Mathematical Modelling of Rheology of Filled Compounds

7 Ottobre 2021

Politecnico di Milano

Scuola di Ingegneria Industriale e dell'Informazione

Mathematical Engineering - Computational Science and Computational Learning

Candidato : Alessandro Cortesi, Matricola : 920493

Relatore : prof. Nicola Parolini

Correlatore : dr. Giorgio Negrini

Anno Accademico 2020-2021

Ai miei genitori, perché,
in piedi sulle loro spalle,
ho potuto guardare lontano.

Mathematical Modelling of Rheology of Filled Compounds

Contents

List of Figures

Listings

Abstract

Filled polymers represent an open wide field of research of great interest in many industrial sectors and applications. In this thesis, filled polymers are described as viscous fluids, whose viscosity depends, other than on the shear rate, both on temperature and filler concentration. In order to describe "completely" viscosity of filled compounds as a function of temperature and filler concentration, some traditional viscosity models have been corrected, employing suitable relative viscosity expressions taken from literature, together with the temperature-dependence Arrhenius model. The implementation of several viscosity models for filled polymers in the *open source* software OpenFOAM (based on Finite Volume Method), is provided and employed in numerical simulations of simple experimental flows. Finally, a detailed comparison between numerical results and experimental data is presented.

Sommario

I polimeri caricati rappresentano un vasto ambito di ricerca di grande interesse per molti settori e applicazioni industriali. In questa tesi, i polimeri caricati sono descritti come fluidi viscosi, la cui viscosità dipende, oltre che dallo shear rate, sia dalla temperatura che dalla concentrazione del filler. Per descrivere "completamente" la viscosità dei polimeri caricati in funzione della temperatura e della concentrazione di filler, alcuni modelli tradizionali di viscosità sono stati corretti attraverso opportune espressioni di viscosità relativa, tratte dalla letteratura, insieme con l'equazione di Arrhenius per la dipendenza dalla temperatura. Si presenta l'implementazione di alcuni modelli di viscosità per polimeri caricati nel software *open source* OpenFOAM (basato sul Metodo ai Volumi Finiti), successivamente utilizzata nelle simulazioni numeriche di semplici flussi sperimentali. Infine, viene presentato un confronto dettagliato tra risultati numerici e dati sperimentali.

Chapter 1

Introduction

In the last century the use of polymeric materials has gained a wider and wider importance in many industrial applications. Frequently, various types of particulate materials (“fillers”) are dispersed in polymers with the aim of improving some desirable physical properties, other than reducing production costs [1]. The term “filler” refers to those inert materials dispersed as solid discrete particles in polymer matrix without significantly affecting the molecular structure of the polymer. Over the years many solid materials have been tested as fillers and dispersed in polymeric compounds, such as carbonates, metal powders, silicates, oxides and carbons [1]. As a matter of fact, the choice of a specific filler to be added to a specific polymer does depend on the physical properties required for the resulting compound: rigid fillers, such as glass fibers, are employed to enhance stiffness of the resulting compound [3], carbon black is widely used in tyre industry in order to increase durability and wear resistance [1], while metal powders or carbons are dispersed in polymers in order to improve the electromagnetic or thermal properties such as thermal conductivity [2]. Moreover many fillers may be employed as flame retardants and combustibility reducers [4, 5]. Actually, fillers dispersion may enhance some desirable polymer properties and alter or spoil some others, that is why the coupling filler-polymer should be deeply analyzed and tested in order to achieve a proper balance between the improvement of some properties and the deterioration of some others. Rheological properties of polymers are likely to change as a consequence of filler addition. In the present study, the influence of filler dispersion on viscosity is analyzed and modelled through some suitable rheological models.

Chapter 2 faces the problem of modelling the viscosity of polymers where solid particulates have been dispersed, with respect to the viscosity of the pure polymer. Some canonical viscosity models are corrected and enabled to describe the viscosity of the polymer-filler compound as a function of the filler loading.

Since polymers where fillers are dispersed are modelled as incompressible viscous fluids, chapter 3 collects and explains the continuum mechanics equations governing the flow of an incompressible viscous fluid.

Chapter 4 deals with rheometry, especially with experimental measurements of viscosity in shear conditions: the flow of an incompressible viscous fluid through a capillary rheometer is presented and analyzed together with Mooney, Rabinowitsch and Bagley corrections to be applied to experimental data.

Chapter 5 contains an introduction to finite volume method for the approximation of partial differential equations problems, with a focus on the solution of the Navier-Stokes system coupled with energy balance equation (chapter 3).The implementation, in the *open source* software OpenFOAM, of some suitable rheological models to describe polymers where fillers have been dispersed (chapter 2), is presented and described.

Chapter 6 collects the analysis of some polymer-filler compound experimental data, which have been employed to choose and characterise some suitable viscosity models in order to describe the rheology of the polymer-filler compound.

Finally, chapter 7 displays the results of numerical simulations run with rheological models introduced in chapter 6, compared with the experimental data.

Chapter 2

Filled Compounds

Polymers are high molecular weight substances resulting from the union (polymerization) of many similar molecules (monomers). Polymer molecules are called macromolecules since they can be thought as flexible chains of monomers, whose length is relatively very larger than the dimensions of the monomers that compose them. As explained in [1], polymers may be classified on the basis of the structure of their macromolecules (linear polymeric chain or branched polymeric chain) rather than on the basis of their degree of crystallinity (the ability of macromelecules to assume a space-ordered arrangement under some suitable pressure-temperature conditions). The dispersion of solid particulates into polymer matrices is widely diffused in industrial contexts in order to reduce costs and to improve some desirable thermal, mechanical, electrical and magnetic properties [1]. The addition of solid material powder into polymeric matrix usually implies a trade-off between the enhancement of some thermal and mechanical properties of the resulting compound (polymer-solid particulate) and the changes in the rheology of the resulting compound (viscosity increase, appearance of yield stresses, different behaviour under shear or extensional flows conditions, changes in storage and loss moduli etc.), the increased difficulty in the melt process, the higher effort needed to eliminate inhomogeneities in solid particulate concentration. The solid particulate dispersed in polymer matrix will be hereby denoted as filler. The polymer without the addition of the filler will be appointed as pure or unfilled polymer while the compound obtained by the dispersion of the filler into the unfilled polymer will be denoted as filled compound. Fillers may be classified by the shape of their particles: spherical fillers are more likely to provide isotropic changes in properties of the filled compound than ellipsoidal fillers or randomly shaped fillers [1]. Moreover the addition of tridimensional particles (e.g. spherical or ellipsoidal fillers) may result in different changes in filled compound properties than bidimensional particles (flakes, platelets) or monodimensional particles (fibers, whiskers).

Also the size distribution of the filler particles plays an important role in determining the properties of filled compounds [1]. As a matter of fact, the addition of solid particles of the same size (monodispersed fillers) or the dispersion of solid particles whose size has a wide range of varibility (polydispersed fillers), have different effects on filled compound properties, for what concerns particle-particle interaction, agglomeration, cluster formation and difficulties in ensuring homogeneous mixing. The investigation of how the thermo-mechanical properties of unfilled polymers are affected by the addition of solid filler particles has gained importance in the last decades but still remains an open research field, due to the large number of variables to take into consideration: the chemical composition of polymer and its molecular structure, the filler particle shape and size distribution and, obviously, the filler concentration. Hystorically, the influence of filler concentration on rheology of filled compounds has been widely investigated, rather than the effects of fillers shape and size distribution. Numerous studies have been focused on the relation between the rheology of the filled compound and the rheology of the unfilled compound with respect to the concentration of the filler. The addition of filler solid particles in polymer matrix leads to an increase in internal friction and energy dissipation during motion, along with the increase in particle-particle interaction and agglomeration of filler solid particles (mostly at high levels of concentration of filler). Hereby, the main focus is to investigate and represent, through some suitable mathematical models, the dependence of the viscosity of the filled compound on the concentration of the filler.

2.1 Relative Viscosity

The addition of a solid compound in a polymeric matrix produces a solid-liquid suspension with different rheological properties with respect to those of the unfilled polymeric matrix, such as viscosity, first normal stress difference, elongational behaviour, onset of yield stresses (viscoplasticity), storage and loss moduli in oscillatory regimes. Hereby, filled and unfilled compounds will be described as viscous fluids and the focus will be on the dependence of viscosity on the filler concentration.

The constitutive relation commonly used to describe polymers as viscous fluids is the generalized Newtonian fluid model:

$$\boldsymbol{T} = -P\boldsymbol{I} + 2\eta(\dot{\gamma}, T)\boldsymbol{D}, \tag{2.1}$$

where $\boldsymbol{T}$ is the Cauchy stress tensor, T is temperature, P denotes pressure while $\boldsymbol{D}$ is the symmetric component of velocity gradient:

$$\boldsymbol{D} = \frac{1}{2}(\nabla\vec{u} + \nabla\vec{u}^T). \tag{2.2}$$

Differently from Newtonian fluids, the viscosity η is assumed to be a function of the shear rate $\dot{\gamma}$ defined as

$$\dot{\gamma} = \sqrt{2\boldsymbol{D} : \boldsymbol{D}} = \sqrt{2D_{ij}D_{ij}}. \tag{2.3}$$

In the following pages, it is assumed that both polymer matrices and filled compounds can be described by equation (2.1) as generalized Newtonian fluids, with $\eta_{pure}(\dot{\gamma}, T)$ and $\eta_{filled}(\dot{\gamma}, T)$ denoting respectively the viscosity of the unfilled polymer matrix and the viscosity of the filled compound.

In past years many mathematical models have been proposed in order to describe the relation between the viscosity of the filled compound ($\eta_{filled}(\dot{\gamma}, T)$) and the viscosity of the unfilled polymer ($\eta_{pure}(\dot{\gamma}, T)$) depending on the concentration of the filler, in particular on its volume fraction.

Definition 2.1. *(Filler Volume Fraction)*
Let V be the volume of a pure polymer matrix. Let $V_{dispersed}$ be the volume of filler dispersed into polymeric matrix. The volume fraction ϕ of the filler is defined as

$$\phi = \frac{V_{dispersed}}{V + V_{dispersed}}. \tag{2.4}$$

In order to relate the viscosity of the filled compound η_{filled} to the viscosity of the unfilled polymer η_{pure}, a definition of relative viscosity it's needed:

$$\eta_r = \frac{\eta_{filled}(\dot{\gamma}, T)}{\eta_{pure}(\dot{\gamma}, T)}. \tag{2.5}$$

Equation (2.5) defines relative viscosity as the ratio between the viscosity of the filled compound and the viscosity of the unfilled polymer, both evaluated at the same shear rate and temperature.

Equation (2.5) turns out to be quite a naive and unfortunate definition of relative viscosity since ratio (2.5) depends on the shear rate $\dot{\gamma}$. For many filled polymers, relative viscosity η_r, computed employing experimental data in (2.5), decreases rapidly in the low shear rates region and mildly in the intermediate shear rate region (for almost all filler volume fractions). In some cases, mostly for low filled compounds, ratio (2.5) reaches an asymptotic value at high shear rates.

In order to provide a definition of relative viscosity η_r which is independent of the shear rate, in [6, 7] it is suggested to substitute equation (2.5) with

$$\eta_r = \lim_{\dot{\gamma} \to \infty} \frac{\eta_{filled}(\dot{\gamma}, T)}{\eta_{pure}(\dot{\gamma}, T)}. \tag{2.6}$$

Definition (2.6) of relative viscosity is quite difficult to use in practice since, especially for highly viscous polymers, it may happen that the ratio

$$\frac{\eta_{filled}(\dot{\gamma}, T)}{\eta_{pure}(\dot{\gamma}, T)}$$

does not achieve an asymptotic value for $\dot{\gamma} \to \infty$. Moreover, especially for highly filled polymers, it is quite difficult to reach high shear rates regimes, since the addition of filler solid particles increases internal friction.

In [8] Kataoka, Kitano et al. proposed a new definition of relative viscosity. Instead of considering the limit of the ratio between the viscosity of filled and unfilled polymer at the same shear rate (and temperature) as in equation (2.5), it's convenient to consider the ratio between the viscosity of filled and unfilled polymer at the same shear stress $\tau = \eta(\dot{\gamma}, T)\dot{\gamma}$ (and temperature):

Definition 2.2. *(Relative Viscosity)*
The relative viscosity η_r of a filled compound is defined as the ratio between the viscosity of the filled compound and the viscosity of the unfilled polymer matrix, at the same shear stress and temperature.

$$\eta_r = \frac{\eta_{filled}(\dot{\gamma}_1, T)|_{\eta_{filled}(\dot{\gamma}_1,T)\dot{\gamma}_1=\tau}}{\eta_{pure}(\dot{\gamma}_2, T)|_{\eta_{pure}(\dot{\gamma}_2,T)\dot{\gamma}_2=\tau}}. \tag{2.7}$$

The ratio (2.7) is independent of τ, at least for sufficiently large shear stresses (i.e. $\tau >> \tau*$).

During the last seventy years a considerable effort has been made in order to model the dependence of relative viscosity on filler volume fraction and several mathematical expressions have been proposed for $\eta_r(\phi)$. As a preliminary consideration, it may be expected that the viscosity of a filled compound is higher than the viscosity of the pure polymer matrix, since the dispersion of solid particulates increases internal friction and, consequently, energy dissipation. Analogously, the higher is the filler volume fraction, the higher is the concentration of the filler, the higher is the relative viscosity of the filler compound. Due to this argument, it should be reasonably expected that

$$\frac{\partial \eta_r(\phi)}{\partial \phi} \geq 0. \tag{2.8}$$

On the other hand, the following condition has to be fulfilled:

$$\eta_r(0) = 1. \tag{2.9}$$

since, if the filler volume fraction is null, the viscosity of the compound is the viscosity of the pure polymer.

One of the first models designed to model the behaviour of the relative viscosity of a filled compound with respect to the filler volume fraction was proposed by Einstein [9]:

$$\eta_r(\phi) = 1 + \frac{5}{2}\phi. \tag{2.10}$$

This relation holds for approximately spherical solid particles of the same magnitude (monodispersed). Moreover equation (2.10) is only valid for low levels of concentration of the filler (dilute solutions, $\phi \simeq 0$). It's worth to notice that, the Einstein model predicts

$$\frac{\partial \eta_r(\phi)}{\partial \phi} = \frac{5}{2} > 0, \tag{2.11}$$

as expected in equation (2.8).

One possible strategy to construct a relative viscosity model is the "effective medium" approach [10].

Assume that it is required to compute the relative viscosity of a filled compound where filler has been dispersed with volume fraction ϕ. Imagine, applying the "effective medium" strategy, to add iteratively some small quantities of filler into the pure polymeric matrix, up to filler volume fraction ϕ. At each iteration some filler is added to a compound which is not pure anymore, but has already a filler volume fraction. Assume that at some iteration the filled compound has volume V and has a filler volume fraction $\phi_1 \simeq \phi$, $\phi_1 < \phi$. To reach filler volume fraction ϕ the addition of a small volume of filler $\Delta V_d << V$ is needed. It holds that

$$\phi = \frac{\phi_1 V + \Delta V_d}{V + \Delta V_d} = \frac{\phi_1 V}{V + \Delta V_d} + \frac{\Delta V_d}{V + \Delta V_d} \simeq \frac{\phi_1 V}{V} + \frac{\Delta V_d}{V + \Delta V_d}. \tag{2.12}$$

Defining

$$\phi_2 = \frac{\Delta V_d}{V + \Delta V_d} << 1.$$

it holds that

$$\phi = \phi_1 + \phi_2. \tag{2.13}$$

Considering the compound with filler volume fraction ϕ_1 as a matrix where filler with volume fraction ϕ_2 has been dispersed, it is possible to state that

$$\eta_r(\phi_2) = \frac{\eta_{filled}(\phi, \dot\gamma_0, T)|_{\eta_{filled}(\phi,\dot\gamma_0,T)\dot\gamma_0=\tau}}{\eta_{filled}(\phi_1, \dot\gamma_1, T)|_{\eta_{filled}(\phi_1,\dot\gamma_1,T)\dot\gamma_1=\tau}} =$$

$$= \frac{\eta_r(\phi)\eta_{pure}(\dot\gamma_2, T)|_{\eta_{pure}(\dot\gamma_2,T)\dot\gamma_2=\tau}}{\eta_r(\phi_1)\eta_{pure}(\dot\gamma_3, T)|_{\eta_{pure}(\dot\gamma_3,T)\dot\gamma_3=\tau}}. \tag{2.14}$$

Since $\phi_2 << 1$, assuming $\phi \simeq \phi_1$, then $\eta_r(\phi_2) \simeq 1$ and $\eta_r(\phi) \simeq \eta_r(\phi_1)$, then equation (2.14) implies

$$\frac{\eta_{pure}(\dot\gamma_2, T)|_{\eta_{pure}(\dot\gamma_2,T)\dot\gamma_2=\tau}}{\eta_{pure}(\dot\gamma_3, T)|_{\eta_{pure}(\dot\gamma_3,T)\dot\gamma_3=\tau}} \simeq 1. \tag{2.15}$$

Equation (2.14) may be simplified in

$$\eta_r(\phi_2) = \frac{\eta_r(\phi)}{\eta_r(\phi_1)}. \tag{2.16}$$

Rearranging equation (2.16), the following relation is obtained:

$$\eta_r(\phi) = \eta_r(\phi_1 + \phi_2) = \eta_r(\phi_1)\eta_r(\phi_2). \tag{2.17}$$

Solving the functional relation (2.17) leads to the following expression for the relative viscosity:

$$\eta_r(\phi) = e^{\alpha\phi}.$$

In order to obtain $\frac{\partial \eta_r(\phi)}{\partial \phi}|_{\phi=0} = 2.5$ as predicted by Einstein (equation (2.10), which holds for $\phi \simeq 0$), it must be $\alpha = 2.5$. The final relative viscosity model is

$$\eta_r(\phi) = e^{2.5\phi}. \tag{2.18}$$

A more precise model may be obtained writing equation (2.12) in the following way

$$\phi = \frac{\phi_1 V + \Delta V_d}{V + \Delta V_d} = \frac{\phi_1 (V + \Delta V_d)}{V + \Delta V_d} - \frac{\phi_1 \Delta V_d}{V + \Delta V_d} + \frac{\Delta V_d}{V + \Delta V_d}, \tag{2.19}$$

which leads to

$$\phi = \phi_1 + \phi_2 - \phi_1 \phi_2. \tag{2.20}$$

Assuming the compound with filler volume fraction ϕ_1 as a matrix where a filler volume fraction ϕ_2 has been dispersed, then

$$\eta_r(\phi_2) = \frac{\eta_{filled}(\phi, \dot{\gamma}_0, T)|_{\eta_{filled}(\phi, \dot{\gamma}_0, T)\dot{\gamma}_0 = \tau}}{\eta_{filled}(\phi_1, \dot{\gamma}_1, T)|_{\eta_{filled}(\phi_1, \dot{\gamma}_1, T)\dot{\gamma}_1 = \tau}} \simeq \frac{\eta_r(\phi)}{\eta_r(\phi_1)}, \tag{2.21}$$

$$\eta_r(\phi) = \eta_r(\phi_1 + \phi_2 - \phi_1\phi_2) = \eta_r(\phi_1)\eta_r(\phi_2). \tag{2.22}$$

Defining $f(1-\phi) = \eta_r(\phi)$, it holds that

$$f(1 - \phi_1 - \phi_2 + \phi_1\phi_2) = f(1-\phi_1)f(1-\phi_2),$$

$$f((1-\phi_1)(1-\phi_2)) = f(1-\phi_1)f(1-\phi_2). \tag{2.23}$$

Letting $\xi = (1-\phi)$, the previous equation becomes

$$f(\xi_1\xi_2) = f(\xi_1)f(\xi_2), \tag{2.24}$$

whose solution is

$$f(\xi) = \xi^\alpha.$$

Eventually, the solution of equation (2.22) is

$$\eta_r(\phi) = (1-\phi)^\alpha.$$

Once again, imposing that $\frac{\partial \eta_r(\phi)}{\partial \phi}|_{\phi=0} = 2.5$, the Roscoe-Brinkman relative viscosity model [16, 17] is obtained:

$$\eta_r(\phi) = (1-\phi)^{-2.5}. \tag{2.25}$$

Remark 2.1. *Mathematical equations for expression of relative viscosity presented so far (equations (2.10)(2.18)(2.25)) do not take into consideration the maximum packing volume fraction ϕ_M.*

Definition 2.3. *The maximum packing filler volume fraction ϕ_M is defined as the filler volume fraction corresponding to the maximum packing arrangement of filler particles, while still retaining a continuous material.*

The value of ϕ_M does depend on the filler particles shape and size distribution. For instance, for a narrow size distribution (not exactly monodispersed but with a narrow range of variability in size) of glass spherical particles, ϕ_M has been found to be 0.62 ([19]). The maximum packing volume fraction may be increased considering fillers with a wider range of variability in size distribution (polydispersed bimodal or multimodal size distribution).

In [11] Krieger and Dougherty suggested the following functional relation:

$$\eta_r(\phi_1 + \phi_2) = \eta_r(\phi_1)\eta_r\Big(\frac{\phi_2}{1 - \frac{\phi_1}{\phi_M}}\Big). \tag{2.26}$$

Equation (2.26) may be interpreted in the following way. Assume that some small quantities ΔV_d of filler are iteratively added into a pure polymeric matrix. At each iteration some filler is added into a compound that already contains some filler. At the first iteration the increase in viscosity due to the addition of ΔV_d is assumed to be smaller than the increase in viscosity at last iterations due to the addition of the same volume ΔV_d. This consideration may suggest that the "effective medium" strategy should be corrected taking into consideration, at each iteration, the "closeness" of the current filler volume fraction to ϕ_M.

Assume that, at some iteration, the filled compound has volume fraction $\phi_1 \simeq \phi$, $\phi_1 < \phi$ and that the addition of a small volume of filler ΔV_d is required to reach volume fraction ϕ. In order to take into account the "closeness" of ϕ_1 to the maximum packing volume fraction ϕ_M, equation (2.16) becomes

$$\eta_r\Big(\frac{\phi_2}{1 - \frac{\phi_1}{\phi_M}}\Big) = \frac{\eta_r(\phi)}{\eta_r(\phi_1)},$$

then

$$\eta_r(\phi) = \eta_r\Big(\frac{\phi_2}{1 - \frac{\phi_1}{\phi_M}}\Big)\eta_r(\phi_1) \tag{2.27}$$

where

$$\phi_2 = \frac{\Delta V_d}{V + \Delta V_d} << 1 \qquad \text{and} \qquad \phi = \phi_1 + \phi_2. \tag{2.28}$$

The solution of equation (2.26) is

$$\eta_r(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{\alpha}.$$

Forcing $\frac{\partial \eta_r(\phi)}{\partial \phi}|_{\phi=0}$ =2.5, coherently with Einstein equation, the Krieger-Dougherty realtive viscosity model is obtained:

$$\eta_r = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2.5\phi_M}. \tag{2.29}$$

Instead of equation (2.26), in [12] Mooney suggested to correct equation (2.17) in the following way:

$$\eta_r(\phi) = \eta_r(\phi_1 + \phi_2) = \eta_r\Big(\frac{\phi_1}{1 - \frac{\phi_2}{\phi_M}}\Big)\eta_r\Big(\frac{\phi_2}{1 - \frac{\phi_1}{\phi_M}}\Big). \tag{2.30}$$

The solution of the equation (2.30) is given by

$$\eta_r(\phi) = exp\Big(\frac{\alpha\phi}{1-\frac{\phi}{\phi_M}}\Big).$$

In order to satisfy condition $\frac{\partial\eta_r(\phi)}{\partial\phi}|_{\phi=0} = 2.5$, α must be set equal to 2.5 and the previous equation becomes

$$\eta_r(\phi) = exp\Big(\frac{2.5\phi}{1-\frac{\phi}{\phi_M}}\Big), \tag{2.31}$$

which is the Mooney relative viscosity model.

Remark 2.2. *Expanding equation (2.18) (2.25) (2.29)(2.31) in Taylor series up to the first order, around $\phi = 0$, it is possible to get Einstein expression (2.10): for dilute filled compounds (i.e. $\phi \simeq 0$) all the previous relative viscosity expressions reduce to the Einstein model.*

Several empirical or semi-empirical models have been proposed to describe relative viscosity as a function of the filler volume fraction, in a wider range of solid particles concentrations: some of them are listed below.

In [13] Eilers introduced the following expression for relative viscosity as function of filler volume fraction:

$$\eta_r(\phi) = \Big(1 + \frac{\frac{5}{4}\phi}{1-\frac{\phi}{\phi_M}}\Big)^2. \tag{2.32}$$

Remark 2.3. *It is worth to notice that $\frac{\partial\eta_r(\phi)}{\partial\phi}|_{\phi=0} = 2.5$: for dilute suspensions (i.e. for $\phi \simeq 0$) the Eilers model reduces to the Einstein equation.*

Maron and Pierce, in [14], designed the following expression to describe the dependence of relative viscosity on concentration of dispersed solid particles:

$$\eta_r(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}. \tag{2.33}$$

Remark 2.4. *The previous equation solves the Krieger and Dougherty equation (2.26)*

$$\eta_r(\phi) = \eta_r(\phi_1 + \phi_2) = \eta_r(\phi_1)\eta_r\Big(\frac{\phi_2}{1-\frac{\phi_1}{\phi_M}}\Big),$$

in such a way that $\frac{\partial\eta_r(\phi)}{\partial\phi}|_{\phi=0} = \frac{2}{\phi_M}$.

An expression similar to equation (2.32) has been suggested by Chong et al. in [15]:

$$\eta_r(\phi) = \Big[1 + \frac{3}{4}\Big(\frac{\frac{\phi}{\phi_M}}{1-\frac{\phi}{\phi_M}}\Big)\Big]^2. \tag{2.34}$$

In [10] Pal made an attempt to take into consideration interactions, collisions and aggregation of suspended particles into the expression of relative viscosity. In particular, Pal ([10]) considered the dispersion of solid spherical particles in non Newtonian fluids. When a filled compound, obtained from the dispersion of solid particulates into polymeric matrix, flows, clusters of solid particles are continuously created and destroyed. The aggregation of solid particles affects the viscosity of the filled compound (i.e. the “resistance” of the filled compound to undergo high velocity gradients).

Consider the shear flow of a filled compound, containing spherical solid particles. When two particles are separated by a sufficiently large distance, they can rotate and translate at different velocities, independently one from the other. When the relative distance between two particles is small, they cannot rotate independently and translate at different velocities anymore (hence a higher ”resistance” to large velocity gradients is expected). Moreover, in [10] it is assumed that, when particles aggregate, a portion of fluid polymer phase close to the cluster is immobilized.

Moving from this consideration, the filler volume fraction ϕ is actually smaller than the effective filler volume fraction ϕ_{eff}

$$\phi_{eff} = \frac{V_{dispersed} + V_{immobilized}}{(V - V_{immobilized}) + V_{dispersed} + V_{immobilized}},$$

$$\phi_{eff} = \frac{V_{dispersed} + V_{immobilized}}{V + V_{dispersed}}, \tag{2.35}$$

where V is the volume of polymeric matrix, $V_{dispersed}$ is the volume of the filler and $V_{immobilized}$ is the volume of polymeric matrix immobilized in the neighbourhood of clusters of solid particles.

Pal expressed the effective filler volume fraction as

$$\phi_{eff} = k(\phi)\phi, \tag{2.36}$$

with $k(\phi)$ being and increasing function of ϕ, since it’s assumed that the volume of clusters increases with the concentration of the filler. Consequently, the volume of immobilized fluid polymer phase frozen around particles clusters increases with filler concentration.

The simplest expression for $k(\phi)$ is

$$k(\phi) = a + b\phi. \tag{2.37}$$

In order to determine constants a and b we consider the following conditions:

$$\begin{cases} k(0) = 1 \\ \phi = \phi_M \Rightarrow \phi_{eff} = k(\phi_M)\phi_M = 1. \end{cases} \tag{2.38}$$

Conditions expressed in (2.38) assume that for pure polymer $\phi = \phi_{eff}$, while for polymer filled up to the maximum packing volume fraction, the whole volume of fluid polymer phase is immobilized. Coefficients a and b turn out to be equal to

$$\begin{cases} a = 1 \\ b = \frac{1-\phi_M}{{\phi_M}^2}. \end{cases}$$

Finally the complete expression for ϕ_{eff} is

$$\phi_{eff} = \Big[1 + \Big(\frac{1-\phi_M}{{\phi_M}^2}\Big)\phi\Big]\phi. \tag{2.39}$$

The "effective medium" strategy is now applied, considering, instead of filler volume fraction, the effective filler volume fraction: equation (2.22) becomes

$$\eta_r(\phi_{eff}) = \eta_r(\phi_{eff,1} + \phi_{eff,2} - \phi_{eff,1}\phi_{eff,2}) = \eta_r(\phi_{eff,1})\eta_r(\phi_{eff,2}). \tag{2.40}$$

The solution of equation (2.40), imposing $\frac{\partial \eta_r(\phi_{eff}}{\partial \phi_{eff}}|_{\phi_{eff}=0} = 2.5$, is

$$\eta_r(\phi_{eff}) = (1 - \phi_{eff})^{-2.5}. \tag{2.41}$$

Combining equations (2.41) and (2.39), Pal relative viscosity model is obtained:

$$\eta_r(\phi) = \Big(1 - \Big(1 + \Big(\frac{1-\phi_M}{\phi_M^2}\Big)\phi\Big)\phi\Big)^{-2.5}. \tag{2.42}$$

Through some straightforward calculations it can be proven that all equations (2.10)(2.18)(2.25)(2.29)(2.31-2.34)(2.42) satisfy conditions (2.8) and (2.9). Moreover, all the models described by equations (2.29)(2.31-2.34)(2.42) verify the following condition

$$\lim_{\phi \to \phi_M} \eta_r(\phi) = \infty. \tag{2.43}$$

If filler volume fraction ϕ reached the value ϕ_M, solid particles would be "packed" preventing the filled compound from flowing and leading to an infinite viscosity.

Figure (2.1) displays the values of relative viscosity $\eta_r(\phi)$ according to equations (2.33)(2.34)(2.31)(2.42)(2.10). The Maron-Pierce and Pal relative viscosities are almost superimposed. The Chong relative viscosity is quite close to the Maron-Pierce and Pal expressions. The Mooney relative viscosity stays below the Maron-Pierce, Chong and Pal models for low filler volume fraction values and above the Maron-Pierce, Chong and Pal expressions for high values of filler concentration. The Einstein relative viscosity stays below all the other models. Finally, for $\phi \to 0$, all the relative viscosity expressions flatten on the Einstein model.

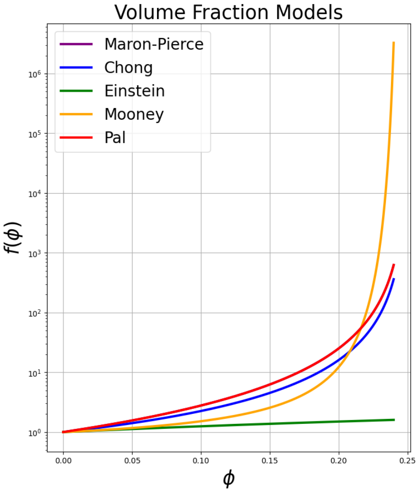


Figure 2.1: Relative Viscosity $\eta_r(\phi)$ for expressions (2.33)(2.34)(2.31) (2.42)(2.10).

2.2 Rheology of Filled Compounds

In order to perform numerical simulations an analytical expression for the viscosity of the filled compound $\eta_{filled}(\dot{\gamma}, T)$ is required. This can be done combining some expressions for the relative viscosity $\eta_r(\phi)$ with the equation for the viscosity of the pure polymeric matrix $\eta_{pure}(\dot{\gamma}, T)$.

One of the most popular models used to describe the viscosity of an incompressible non-Newtonian fluid (independently of temperature) is the Bird-Carreau expression:

$$\eta(\dot{\gamma}) = \eta_0 \Big[1 + \Big(\lambda\dot{\gamma}\Big)^2\Big]^{(n-1)/2}, \tag{2.44}$$

where $\dot{\gamma}$ is the shear rate while η_0 e λ represent, respectively, the zero shear-rate viscosity and the relaxation time of the matrix polymer.

If $\lambda\dot{\gamma} << 1$, then $\eta \simeq \eta_0$: at low shear rates, Bird-Carreau fluids exhibit a Newtonian behaviour.

On the contrary, if $\lambda\dot{\gamma} >> 1$, the following holds

$$\eta \simeq \eta_0 \Big(\lambda\dot{\gamma}\Big)^{(n-1)}. \tag{2.45}$$

Equation (2.45) states that, in high shear rates regime, the viscosity of Bird-Carreau fluids depends on the shear rate through a power law relation.

Assume that the viscosity of the matrix polymer is described by the Bird-Carreau expression (2.44), corrected by a factor $H(T)$ in order to take into consideration the dependence of viscosity on temperature:

$$\eta_{pure}(\dot{\gamma}, T) = H(T)\eta_0 \Big[1 + \Big(\lambda\dot{\gamma}\Big)^2\Big]^{(n-1)/2}. \tag{2.46}$$

To model the viscosity of the filled polymer, where the concentration of the suspended particulates is represented by the filler volume fraction ϕ, Poslinski ([19]) has modified Bird-Carreau equation (2.44) in the following way:

$$\eta(\dot{\gamma}, \phi) = \eta_0 f(\phi)\Big[1 + \Big(f(\phi)\lambda\dot{\gamma}\Big)^2\Big]^{(n-1)/2}, \tag{2.47}$$

with

$$f(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}. \tag{2.48}$$

Since this research is going to take into account the influence of temperature on viscosity of the filled polymer, also equation (2.47) should be corrected as

$$\eta_{filled}(\dot{\gamma}, \phi, T) = H(T)\eta_0 f(\phi)\Big[1 + \Big(f(\phi)\lambda\dot{\gamma}\Big)^2\Big]^{(n-1)/2}, \tag{2.49}$$

Making use of equation (2.49), the relative viscosity of the filled compound with filler volume fraction ϕ is

$$\eta_r(\phi) = f(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}. \tag{2.50}$$

It is worth to notice that the expression of relative viscosity perfectly matches Maron-Pierce equation (2.33).

Let $\phi_1 \in [0, \phi_M)$ and let $\dot{\gamma_0}$, $\dot{\gamma_1}$ such that

$$\begin{cases} \tau_0 = \eta_{pure}(\dot{\gamma}_0, T)\dot{\gamma}_0 \\ \tau_1 = \eta_{filled}(\dot{\gamma}_1, \phi_1, T)\dot{\gamma}_1 \\ \tau_0 = \tau_1. \end{cases} \tag{2.51}$$

Then, according to definition (2.2), it holds that

$$\eta_r(\phi_1) = \frac{\eta_{filled}(\dot{\gamma}_1, \phi_1, T)}{\eta_{pure}(\dot{\gamma}_0, T)} = \frac{H(T)\eta_0 f(\phi_1)\Big[1 + \Big(f(\phi_1)\lambda\dot{\gamma}_1\Big)^2\Big]^{(n-1)/2}}{H(T)\eta_0\Big[1 + \Big(\lambda\dot{\gamma}_0\Big)^2\Big]^{(n-1)/2}}. \tag{2.52}$$

Assuming that $\lambda\dot{\gamma}_0 << 1$ and $f(\phi_1)\lambda\dot{\gamma}_1 << 1$, it holds that

$$\eta_r(\phi_1) = \frac{\eta_{filled}(\dot{\gamma}_1, \phi_1, T)}{\eta_{pure}(\dot{\gamma}_0, T)} \simeq f(\phi_1). \tag{2.53}$$

On the other hand, if $\lambda\dot{\gamma}_0 >> 1$ and $f(\phi_1)\lambda\dot{\gamma}_1 >> 1$, it holds that

$$\eta_r(\phi_1) \simeq \frac{f(\phi_1)\Big(f(\phi_1)\lambda\dot{\gamma}_1\Big)^{(n-1)}}{\Big(\lambda\dot{\gamma}_0\Big)^{(n-1)}} = \frac{(f(\phi_1))^n\Big(\dot{\gamma}_1\Big)^{(n-1)}}{\Big(\dot{\gamma}_0\Big)^{(n-1)}}. \tag{2.54}$$

Since $\tau_0 = \tau_1$, the following relation holds true

$$\eta_{pure}(\dot{\gamma}_0, T)\dot{\gamma}_0 = \eta_{filled}(\dot{\gamma}_1, \phi_1, T)\dot{\gamma}_1 \Rightarrow \eta_r(\phi_1) = \frac{\eta_{filled}(\dot{\gamma}_1, \phi_1, T)}{\eta_{pure}(\dot{\gamma}_0, T)} = \frac{\dot{\gamma}_0}{\dot{\gamma}_1}, \tag{2.55}$$

Combining equations (2.54) and (2.55), it holds that

$$\eta_r(\phi_1) = \frac{(f(\phi_1))^n}{\eta_r(\phi_1)^{n-1}} \Rightarrow \eta_r(\phi_1) = f(\phi_1). \tag{2.56}$$

Another popular expression to describe the viscosity of an incompressible non-Newtonian fluid is the following:

$$\eta(\dot{\gamma}) = \eta_0 \Big(\lambda\dot{\gamma}\Big)^{(n-1)}, \tag{2.57}$$

where $\dot{\gamma}$ is the shear rate while η_0 e λ represent, respectively, the zero shear rate viscosity and the relaxation time of the fluid. As before, in order not to neglect dependence of viscosity on temperature, equation (2.57) should be corrected as

$$\eta(\dot{\gamma}, T) = H(T)\eta_0 \Big(\lambda\dot{\gamma}\Big)^{(n-1)}. \tag{2.58}$$

This expression, called power law model, represents a simplification with respect to the Bird-Carreau model, since at high shear rates the Bird-Carreau equation (2.46) reduces to equation (2.58).

Assuming that the polymer matrix viscosity $\eta_{pure}(\dot{\gamma}, T)$ has a power law behaviour (2.58), drawing inspiration from [19], the viscosity of the filled compound, as a function of the filler volume fraction ϕ, may be written in the following way:

$$\eta_{filled}(\dot{\gamma}, \phi, T) = H(T)\eta_0 f(\phi)\Big(f(\phi)\dot{\gamma}\lambda\Big)^{n-1}, \tag{2.59}$$

where $f(\phi)$ should be chosen as in [19]:

$$f(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}. \tag{2.60}$$

It can be proven that, assuming equation (2.59), the relative viscosity of the filled compound matches Maron-Pierce expression (2.33):

$$\eta_r(\phi) = f(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}. \tag{2.61}$$

Let $\phi_1 \in [0, \phi_M)$ and let $\dot{\gamma_0}$, $\dot{\gamma_1}$ such that

$$\begin{cases} \tau_0 = \eta_{pure}(\dot{\gamma}_0, T)\dot{\gamma}_0 \\ \tau_1 = \eta_{filled}(\dot{\gamma}_1, \phi_1, T)\dot{\gamma}_1 \\ \tau_0 = \tau_1. \end{cases} \tag{2.62}$$

The relative viscosity may be computed as

$$\eta_r(\phi_1) = \frac{\eta_{filled}(\dot{\gamma}_1, \phi_1, T)}{\eta_{pure}(\dot{\gamma}_0, T)} = \frac{H(T)\eta_0 f(\phi_1)\Big(f(\phi_1)\lambda\dot{\gamma}_1\Big)^{(n-1)}}{H(T)\eta_0\Big(\lambda\dot{\gamma}_0\Big)^{(n-1)}}, \tag{2.63}$$

$$\eta_r(\phi_1) = \frac{f(\phi_1)\Big(f(\phi_1)\lambda\dot{\gamma}_1\Big)^{(n-1)}}{\Big(\lambda\dot{\gamma}_0\Big)^{(n-1)}} = \frac{(f(\phi_1))^n\Big(\dot{\gamma}_1\Big)^{(n-1)}}{\Big(\dot{\gamma}_0\Big)^{(n-1)}}. \tag{2.64}$$

Taking into account that $\tau_0 = \tau_1$, the following relation holds true:

$$\eta_{pure}(\dot{\gamma}_0, T)\dot{\gamma}_0 = \eta_{filled}(\dot{\gamma}_1, \phi_1, T)\dot{\gamma}_1 \Rightarrow \eta_r(\phi_1) = \frac{\eta_{filled}(\dot{\gamma}_1, \phi_1, T)}{\eta_{pure}(\dot{\gamma}_0, T)} = \frac{\dot{\gamma}_0}{\dot{\gamma}_1}. \tag{2.65}$$

The substitution of the previous equation into equation (2.64) leads to

$$\eta_r(\phi_1) = \frac{(f(\phi_1))^n}{\eta_r(\phi_1)^{n-1}} \Rightarrow \eta_r(\phi_1) = f(\phi_1). \tag{2.66}$$

The Cross model is also widely used to describe the viscosity of an incompressible non Newtonian fluid:

$$\eta(\dot{\gamma}) = \frac{\eta_0}{1 + (\lambda\dot{\gamma})^m}. \tag{2.67}$$

As for Bird Carreau model (2.44) it should be noticed that at low shear rates ($\lambda\dot{\gamma} << 1$), the behaviour of a Cross fluid is Newtonian, since equation (2.67) reduces to

$$\eta(\dot{\gamma}) \simeq \eta_0. \tag{2.68}$$

On the contrary, at high shear rates ($\lambda\dot{\gamma} >> 1$) the Cross fluid exhibits a power law behaviour:

$$\eta(\dot{\gamma}) = \eta_0(\lambda\dot{\gamma})^{-m}. \tag{2.69}$$

Assuming that the viscosity of the polymer matrix exhibits a (temperature dependent) Cross behaviour

$$\eta(\dot{\gamma}, T) = H(T)\frac{\eta_0}{1 + \Big(\lambda\dot{\gamma}\Big)^m} \tag{2.70}$$

and taking inspiration from [19], the following model can be constructed:

$$\eta_{filled}(\dot{\gamma}, \phi, T) = \frac{H(T)\eta_0 f(\phi)}{1 + \Big(\lambda f(\phi)\dot{\gamma}\Big)^m}, \tag{2.71}$$

with

$$f(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}. \tag{2.72}$$

As for equations (2.49) (2.59), the relative viscosity computed exploiting equation (2.71) matches the Maron Pierce expression (2.33).

Let $\phi_1 \in [0, \phi_M)$ and let $\dot{\gamma_0}$, $\dot{\gamma_1}$ such that

$$\begin{cases} \tau_0 = \eta_{pure}(\dot{\gamma}_0, T)\dot{\gamma}_0 \\ \tau_1 = \eta_{filled}(\dot{\gamma}_1, \phi_1, T)\dot{\gamma}_1 \\ \tau_0 = \tau_1. \end{cases} \tag{2.73}$$

The relative viscosity may be computed as

$$\eta_r(\phi_1) = \frac{\eta_{filled}(\dot{\gamma}_1, \phi_1, T)}{\eta_{pure}(\dot{\gamma}_0, T)} = \frac{H(T)\eta_0 f(\phi_1)}{1 + \Big(f(\phi_1)\lambda\dot{\gamma}_1\Big)^m} \frac{1 + \Big(\lambda\dot{\gamma}_0\Big)^m}{H(T)\eta_0}, \tag{2.74}$$

$$\eta_r(\phi_1) = \frac{f(\phi_1)\Big(1 + \Big(\lambda\dot{\gamma}_0\Big)^m\Big)}{1 + \Big(f(\phi_1)\lambda\dot{\gamma}_1\Big)^m}, \tag{2.75}$$

If it is assumed that $\lambda\dot{\gamma}_0 << 1$ and $f(\phi_1)\lambda\dot{\gamma}_1 << 1$, then equation (2.75) reduces to

$$\eta_r(\phi_1) \simeq f(\phi_1). \tag{2.76}$$

On the contrary, if $\lambda\dot{\gamma}_0 >> 1$ and $f(\phi_1)\lambda\dot{\gamma} >> 1$, then it holds

$$\eta_r(\phi_1) \simeq \frac{f(\phi_1)\Big(\lambda\dot{\gamma}_0\Big)^m}{\Big(f(\phi_1)\lambda\dot{\gamma}_1\Big)^m} = \frac{\Big(f(\phi_1)\Big)^{1-m}\dot{\gamma}_0^m}{\dot{\gamma}_1^m}. \tag{2.77}$$

Taking into account that $\tau_0 = \tau_1$, the following relation is obtained:

$$\eta_{pure}(\dot{\gamma}_0, T)\dot{\gamma}_0 = \eta_{filled}(\dot{\gamma}_1, \phi_1, T)\dot{\gamma}_1 \Rightarrow \eta_r(\phi_1) = \frac{\eta_{filled}(\dot{\gamma}_1, \phi_1, T)}{\eta_{pure}(\dot{\gamma}_0, T)} = \frac{\dot{\gamma}_0}{\dot{\gamma}_1}. \tag{2.78}$$

Substituting the previous equation into equation (2.77) allows to get

$$\eta_r(\phi_1) = \Big(f(\phi_1)\Big)^{1-m}\Big(\eta_r(\phi_1)\Big)^m \Rightarrow \eta_r(\phi_1) = f(\phi_1) = \Big(1 - \frac{\phi_1}{\phi_M}\Big)^{-2}. \tag{2.79}$$

In general, it's possible to construct a mathematical expression to describe the viscosity of a filled compound, combining the expression of the viscosity of the pure matrix polymer (Newtonian, Bird-Carreau, power law, Cross etc.)

with a factor $f(\phi)$:

$$\begin{cases} \eta_{pure}(\dot{\gamma},T) = H(T)\eta_0\Big[1+\Big(\lambda\dot{\gamma}\Big)^2\Big]^{(n-1)/2} \Rightarrow \\ \Rightarrow \eta_{filled}(\dot{\gamma},\phi,T) = H(T)\eta_0 f(\phi)\Big[1+\Big(f(\phi)\lambda\dot{\gamma}\Big)^2\Big]^{(n-1)/2} \\ \\ \eta_{pure}(\dot{\gamma},T) = H(T)\eta_0\Big(\lambda\dot{\gamma}\Big)^{(n-1)} \Rightarrow \eta_{filled}(\dot{\gamma},\phi,T) = H(T)\eta_0 f(\phi)\Big(f(\phi)\dot{\gamma}\lambda\Big)^{n-1} \\ \\ \eta_{pure}(\dot{\gamma},T) = H(T)\dfrac{\eta_0}{1+\Big(\lambda\dot{\gamma}\Big)^m} \Rightarrow \eta_{filled}(\dot{\gamma},\phi,T) = H(T)\dfrac{\eta_0 f(\phi)}{1+\Big(\lambda f(\phi)\dot{\gamma}\Big)^m}, \end{cases} \tag{2.80}$$

choosing for $f(\phi)$ one of the filler volume fraction models among (2.10) (2.18) (2.25) (2.29) (2.31 - 2.34) (2.42)

In this way, as showed above, it is obtained, coherently, that

$$\eta_r(\phi) = f(\phi).$$

Remark 2.5. *All the viscosity models presented in (2.80) could be generally written as*

$$\eta_{filled}(\dot{\gamma},\phi,T) = H(T)V(\dot{\gamma},\phi) = H(T)V(\dot{\gamma},f(\phi)). \tag{2.81}$$

Chapter 3

Governing Equations

This chapter presents an overview on the mathematical equations governing the motion of an incompressible generalized Newtonian fluid, taken from continuum mechanics: the mass conservation principle, the conservation of linear momentum equation, the incompressibility constraint and the energy balance equation.

Let Ω_0 be the reference configuration for a Cauchy continuum body and let Ω_t be its configuration at time $t \in [0; +\infty)$. Define the trajectory of the Cauchy continuum body as

$$\mathcal{T} = \{(\boldsymbol{x}, t) \in \mathbb{R}^3 \times [0; +\infty) : \boldsymbol{x} \in \Omega_t\} = \bigcup_{t \in [0;+\infty)} \Big(\Omega_t \times \{t\} \Big). \tag{3.1}$$

The next fundamental theorem will be widely used to compute time derivatives of integral quantities.

Theorem 3.1. *Reynolds Transport Theorem (RTT)*

Let ϕ be a sufficiently regular scalar field defined on $\mathcal{T}$, let $\vec{\phi}$ be a sufficiently regular vectorial field defined on $\mathcal{T}$, let $\mathbf{\Phi}$ be a sufficiently regular tensorial field defined on $\mathcal{T}$.
Denoting with $\mathcal{P}_t \subset \Omega_t$ an arbitrary portion of the continuum body at time t and letting $\vec{u}$ be the eulerian velocity field defined on $\mathcal{T}$,then

$$\frac{d}{dt} \int_{\mathcal{P}_t} \phi dV = \int_{\mathcal{P}_t} \frac{\partial \phi}{\partial t} dV + \int_{\partial \mathcal{P}_t} \phi (\vec{u} \cdot \vec{n}) dS, \tag{3.2}$$

$$\frac{d}{dt} \int_{\mathcal{P}_t} \vec{\phi} dV = \int_{\mathcal{P}_t} \frac{\partial \vec{\phi}}{\partial t} dV + \int_{\partial \mathcal{P}_t} \vec{\phi} (\vec{u} \cdot \vec{n}) dS, \tag{3.3}$$

$$\frac{d}{dt} \int_{\mathcal{P}_t} \mathbf{\Phi} dV = \int_{\mathcal{P}_t} \frac{\partial \mathbf{\Phi}}{\partial t} dV + \int_{\partial \mathcal{P}_t} \mathbf{\Phi} (\vec{u} \cdot \vec{n}) dS. \tag{3.4}$$

Proof. Let $\mathcal{P}_0$ be the portion of the continuum body in the reference configuration Ω_0 corresponding to $\mathcal{P}_t$. The starting point is proving *RTT* for scalar fields.

Performing a change of variables ($\mathcal{L}_t : \Omega_0 \rightarrow \Omega_t$), the first integral in equation (3.2) may be transformed into an integral on the reference configuration:

$$\frac{d}{dt}\int_{\mathcal{P}_t} \phi dV = \frac{d}{dt}\int_{\mathcal{P}_0} \phi_m J dV_m, \tag{3.5}$$

where ϕ_m is the field ϕ mapped back onto the reference configuration, while J is the jacobian of the change of coordinates from $\mathcal{P}_0$ to $\mathcal{P}_t$. Since the domain of integration is no longer depending on t, under sufficient regularity conditions (which are assumed to be satisfied), it is possible to exchange integral and derivative:

$$\frac{d}{dt}\int_{\mathcal{P}_0} \phi_m J dV_m = \int_{\mathcal{P}_0} \frac{d}{dt}(\phi_m J) dV_m = \int_{\mathcal{P}_0} \frac{d\phi_m}{dt} J dV_m + \int_{\mathcal{P}_0} \phi_m \frac{dJ}{dt} dV_m, \tag{3.6}$$

Next, the following relation is taken into consideration without proving it:

$$\frac{dJ}{dt} = J(div(\vec{u}))_m. \tag{3.7}$$

Substituting (3.7) into equation (3.5) and performing an inverse change of variables ($\mathcal{L}_t^{-1} : \Omega_t \rightarrow \Omega_0$), it holds that

$$\int_{\mathcal{P}_0} \frac{d\phi_m}{dt} J dV_m + \int_{\mathcal{P}_0} \phi_m J(div(\vec{u}))_m dV_m = \int_{\mathcal{P}_t} \frac{d\phi}{dt} dV + \int_{\mathcal{P}_t} \phi div(\vec{u}) dV. \tag{3.8}$$

Taking into account that

$$\frac{d\phi}{dt} = \frac{\partial \phi}{\partial t} + \vec{u} \cdot \nabla \phi, \tag{3.9}$$

$$div(\vec{u}\phi) = \phi div(\vec{u}) + \vec{u} \cdot \nabla \phi, \tag{3.10}$$

the following equation is obtained:

$$\int_{\mathcal{P}_t} \frac{d\phi}{dt} dV + \int_{\mathcal{P}_t} \phi div(\vec{u}) dV = \int_{\mathcal{P}_t} \frac{\partial \phi}{\partial t} dV + \int_{\mathcal{P}_t} div(\phi \vec{u}) dV. \tag{3.11}$$

The application of the divergence theorem to the last integral in equation (3.11) leads to

$$\frac{d}{dt}\int_{\mathcal{P}_t} \phi dV = \int_{\mathcal{P}_t} \frac{\partial \phi}{\partial t} dV + \int_{\partial \mathcal{P}_t} \phi (\vec{u} \cdot \vec{n}) dS. \tag{3.12}$$

The next step is proving *RTT* for vectorial fields (equation (3.3)). Let $\vec{a}$ be an arbitrary fixed vector. Define

$$\phi = \vec{a} \cdot \vec{\phi}. \tag{3.13}$$

Since *RTT* for scalar fields has already been proven, it holds that

$$\frac{d}{dt}\int_{\mathcal{P}_t} \vec{a} \cdot \vec{\phi} dV = \int_{\mathcal{P}_t} \frac{\partial(\vec{a} \cdot \vec{\phi})}{\partial t} dV + \int_{\partial \mathcal{P}_t} (\vec{a} \cdot \vec{\phi})(\vec{u} \cdot \vec{n}) dS. \tag{3.14}$$

Exploting the fact that $\vec{a}$ is constant, the previous equation reduces to

$$\vec{a} \cdot \frac{d}{dt} \int_{\mathcal{P}_t} \vec{\phi} dV = \vec{a} \cdot \int_{\mathcal{P}_t} \frac{\partial \vec{\phi}}{\partial t} dV + \vec{a} \cdot \int_{\partial \mathcal{P}_t} \vec{\phi}(\vec{u} \cdot \vec{n}) dS. \tag{3.15}$$

Since $\vec{a}$ it's arbitrary, it holds that

$$\frac{d}{dt} \int_{\mathcal{P}_t} \vec{\phi} dV = \int_{\mathcal{P}_t} \frac{\partial \vec{\phi}}{\partial t} dV + \int_{\partial \mathcal{P}_t} \vec{\phi}(\vec{u} \cdot \vec{n}) dS. \tag{3.16}$$

The proof of *RTT* for tensorial fields is quite similar. Let $\mathbf{\Phi}$ be a tensorial field and let $\vec{w}$ be an arbitrary constant vector and let $\vec{\phi}$ be the following vector field:

$$\vec{\phi} = \mathbf{\Phi} \vec{w}. \tag{3.17}$$

Taking advantage of *RTT* for vectorial fields, it is possible to state that

$$\frac{d}{dt} \int_{\mathcal{P}_t} \mathbf{\Phi} \vec{w} dV = \int_{\mathcal{P}_t} \frac{\partial (\mathbf{\Phi} \vec{w})}{\partial t} dV + \int_{\partial \mathcal{P}_t} (\mathbf{\Phi} \vec{w})(\vec{u} \cdot \vec{n}) dS. \tag{3.18}$$

Since vector $\vec{w}$ is constant, then

$$\Big(\frac{d}{dt} \int_{\mathcal{P}_t} \mathbf{\Phi} dV\Big) \vec{w} = \Big(\int_{\mathcal{P}_t} \frac{\partial \mathbf{\Phi}}{\partial t} dV\Big) \vec{w} + \Big(\int_{\partial \mathcal{P}_t} \mathbf{\Phi}(\vec{u} \cdot \vec{n}) dS\Big) \vec{w}. \tag{3.19}$$

The fact that $\vec{w}$ it's arbitrary ensures that

$$\frac{d}{dt} \int_{\mathcal{P}_t} \mathbf{\Phi} dV = \int_{\mathcal{P}_t} \frac{\partial \mathbf{\Phi}}{\partial t} dV + \int_{\partial \mathcal{P}_t} \mathbf{\Phi}(\vec{u} \cdot \vec{n}) dS. \tag{3.20}$$

□

In next paragraphs *RTT* will be massively used in order to deduce differential equations which govern the motion of a continuum body from a thermomechanical prospective.

3.1 Mass Conservation

The mass conservation principle states that the mass of an arbitrary portion of the continuum body does not change during its motion. Let $\mathcal{P}_t$ be an arbitrary portion of the continuum body in its configuration Ω_t. The mass conservation principle states that

$$\frac{d}{dt} \int_{\mathcal{P}_t} \rho dV = 0, \tag{3.21}$$

where ρ represents the density of the continuum body. Making use of *RTT* and divergence theorem, it's possible to rewrite equation (3.21) as

$$\int_{\mathcal{P}_t} \frac{\partial \rho}{\partial t} dV + \int_{\partial \mathcal{P}_t} \rho(\vec{u} \cdot \vec{n}) dS = \int_{\mathcal{P}_t} \frac{\partial \rho}{\partial t} dV + \int_{\mathcal{P}_t} div(\rho \vec{u}) dV = 0. \tag{3.22}$$

Since equation (3.22) must be satisfied independently of $\mathcal{P}_t$ it must be

$$\frac{\partial \rho}{\partial t} + div(\rho \vec{u}) = 0. \tag{3.23}$$

Taking into consideration that

$$\frac{d\rho}{dt} = \frac{\partial \rho}{\partial t} + \vec{u} \cdot \nabla \rho, \tag{3.24}$$

equation (3.23) may be rewritten as

$$\frac{d\rho}{dt} + \rho div(\vec{u}) = 0. \tag{3.25}$$

3.2 Linear Momentum Conservation

The linear momentum conservation equation corresponds to the second Newton's Law. Let $\mathcal{P}_t$ be an arbitrary portion of the continuum body in its configuration Ω_t. The time derivative of the linear momentum of $\mathcal{P}_t$ equals the sum of surface forces and body forces applied to $\mathcal{P}_t$:

$$\frac{d}{dt}\int_{\mathcal{P}_t} \rho \vec{u} dV = \int_{\partial \mathcal{P}_t} \vec{s} dS + \int_{\mathcal{P}_t} \rho \vec{f} dV. \tag{3.26}$$

The second integral in the right hand side of (3.26) corresponds to the resultant of body forces ($\rho \vec{f}$ is a force per unit volume distributed in $\mathcal{P}_t$). The first integral in the right hand side of (3.26) represents the contribution of surface forces ($\vec{s}$ is a force per unit surface distributed on the boundary of $\mathcal{P}_t$).

Exploiting the Cauchy theorem, $\vec{s}$ may be written as $\vec{s} = \boldsymbol{T}\vec{n}$, where $\boldsymbol{T}$ is the Cauchy stress tensor field and $\vec{n}$ the unit normal vector of $\mathcal{P}_t$, pointing outward:

$$\frac{d}{dt}\int_{\mathcal{P}_t} \rho \vec{u} dV = \int_{\partial \mathcal{P}_t} \boldsymbol{T}\vec{n} dS + \int_{\mathcal{P}_t} \rho \vec{f} dV. \tag{3.27}$$

Making use of *RTT* and divergence theorem, equation (3.27) may be rewritten as

$$\int_{\mathcal{P}_t} \frac{\partial(\rho \vec{u})}{\partial t} dV + \int_{\partial \mathcal{P}_t} (\rho \vec{u})(\vec{u} \cdot \vec{n}) dS = \int_{\mathcal{P}_t} div(\boldsymbol{T}) dV + \int_{\mathcal{P}_t} \rho \vec{f} dV. \tag{3.28}$$

Next, it should be noticed that

$$\rho \vec{u}(\vec{u} \cdot \vec{n}) = \rho(\vec{u} \otimes \vec{u})\vec{n}, \tag{3.29}$$

with $\vec{a} \otimes \vec{b}$ being the tensorial product of vectors $\vec{a}$ and $\vec{b}$.

Equation (3.28) becomes

$$\int_{\mathcal{P}_t} \frac{\partial(\rho \vec{u})}{\partial t} dV + \int_{\partial \mathcal{P}_t} \rho(\vec{u} \otimes \vec{u})\vec{n} dS = \int_{\mathcal{P}_t} div(\boldsymbol{T}) dV + \int_{\mathcal{P}_t} \rho \vec{f} dV. \tag{3.30}$$

Making use of the divergence theorem, it's possible to rewrite the previous equation as

$$\int_{\mathcal{P}_t} \frac{\partial(\rho\vec{u})}{\partial t} dV + \int_{\mathcal{P}_t} div(\rho(\vec{u}\otimes\vec{u}))dV = \int_{\mathcal{P}_t} div(\boldsymbol{T})dV + \int_{\mathcal{P}_t} \rho\vec{f}dV. \tag{3.31}$$

Since the last equation must hold for each portion $\mathcal{P}_t$ of the continuum body, it's possible to state that

$$\frac{\partial(\rho\vec{u})}{\partial t} + div(\rho(\vec{u}\otimes\vec{u})) = div(\boldsymbol{T}) + \rho\vec{f}. \tag{3.32}$$

3.3 Incompressibility

The incompressibility constraint imposes that the volume of each portion of the continuum body does not change during its motion. Letting $\mathcal{P}_t$ be an arbitrary portion of the continuum body in its configuration at time t, the incompressibility constraint may be compactly written as

$$\frac{d}{dt}\int_{\mathcal{P}_t} dV = 0. \tag{3.33}$$

In order to reduce (3.33) to its local form, RTT is needed:

$$\frac{d}{dt}\int_{\mathcal{P}_t} dV = \int_{\mathcal{P}_t} \frac{\partial 1}{\partial t} dV + \int_{\partial\mathcal{P}_t} 1(\vec{u}\cdot\vec{n})dS = \int_{\mathcal{P}_t} div(\vec{u})dV = 0. \tag{3.34}$$

Since equation (3.34) must be true for each portion of the continuum body, the incompressiblity constraint is satisfied if the following equation holds locally:

$$div(\vec{u}) = 0. \tag{3.35}$$

Assuming incompressibility constraint for the continuum body, the local conservation of mass equation (3.23) reduces to

$$\frac{\partial\rho}{\partial t} + div(\rho\vec{u}) = \frac{\partial\rho}{\partial t} + div(\vec{u})\rho + \nabla\rho\cdot\vec{u} = \frac{\partial\rho}{\partial t} + \nabla\rho\cdot\vec{u} = \frac{d\rho}{dt} = 0. \tag{3.36}$$

Equation (3.36) states that the density at each point of the continuum body is constant in time. If the density of the continuum body in the reference configuration Ω_0 is uniform (i.e. if $\rho\in\mathbb{R}^+$), then at each time t, each point of Ω_t has density $\rho\in\mathbb{R}^+$.

Under the assumption of incompressibility, the equation for conservation of linear momentum (3.37) may be further manipulated:

$$\frac{\partial(\rho\vec{u})}{\partial t} + (\vec{u}\cdot\nabla)(\rho\vec{u}) + \rho\vec{u}\,div(\vec{u}) = \frac{\partial(\rho\vec{u})}{\partial t} + (\vec{u}\cdot\nabla)(\rho\vec{u}) = div(\boldsymbol{T}) + \rho\vec{f}. \tag{3.37}$$

If $\rho\in\mathbb{R}^+$, then the previous equation reduces to

$$\rho\frac{\partial\vec{u}}{\partial t} + \rho(\vec{u}\cdot\nabla)\vec{u} = div(\boldsymbol{T}) + \rho\vec{f}. \tag{3.38}$$

3.4 Energy Balance Equation

The first law of thermodynamics states that the rate of change of energy of a system equals the sum of incoming heat per unit time and rate of work done by external forces on the system. More rigorously, the first law of thermodynamics may be formulated in this way:

$$\Pi_{ext} + Q = \frac{dU}{dt} + \frac{dK}{dt}, \tag{3.39}$$

where K and U are, respectively, the kinetic and internal energy of an arbitrary portion of continuum body $\mathcal{P}_t$, while Q and Π_{ext} are the incoming heat per unit time in $\mathcal{P}_t$ and the power (rate of work) done by external forces on $\mathcal{P}_t$. In order to obtain the energy balance equation, definitions for K, Π_{ext}, U and Q are needed.
The kinetic energy K of $\mathcal{P}_t$ is

$$K = \frac{1}{2}\int_{\mathcal{P}_t} \rho\vec{u}\cdot\vec{u}dV. \tag{3.40}$$

In order to manipulate equation (3.39) the following theorem will be used.

Theorem 3.2. *(Kinetic Energy Theorem)*
Let $\mathcal{P}_t$ be an arbitrary portion of a continuum body in its configuration Ω_t at time t. The rate of change of kinetic energy of $\mathcal{P}_t$ equals the sum of the power of external forces Π_{ext} and the power of internal forces Π_{int}.

$$\frac{dK}{dt} = \Pi_{ext} + \Pi_{int}, \tag{3.41}$$

with

$$\Pi_{ext} = \int_{\partial\mathcal{P}_t} (\boldsymbol{T}\vec{n})\cdot\vec{u}dS + \int_{\mathcal{P}_t} \vec{u}\cdot\rho\vec{f}dV,$$

$$\Pi_{int} = -\int_{\mathcal{P}_t} \boldsymbol{T} : \boldsymbol{D}dV.$$

Proof. Let $\mathcal{P}_0$ be the portion of continuum body, in the reference configuration Ω_0, corresponding to $\mathcal{P}_t$. Let K be the kinetic energy of $\mathcal{P}_t$. Performing a change of coordinates $(\mathcal{L}_t : \Omega_0 \to \Omega_t)$, the left hand side of equation (3.41) may be rewritten as an integral on the reference configuration:

$$\frac{dK}{dt} = \frac{d}{dt}\frac{1}{2}\int_{\mathcal{P}_t} \rho\vec{u}\cdot\vec{u} = \frac{d}{dt}\frac{1}{2}\int_{\mathcal{P}_0} \rho_m\vec{u}_m\cdot\vec{u}_m JdV, \tag{3.42}$$

where ρ_m and $\vec{u}_m$ are, respectively, the density and velocity fields mapped back onto $\mathcal{P}_0$ while J is the jacobian of the change of coordinates from $\mathcal{P}_0$ to $\mathcal{P}_t$. Since the integration domain does not depend on t anymore, it's possible to rewrite the previous equation as

$$\frac{dK}{dt} = \frac{1}{2}\int_{\mathcal{P}_0} \frac{d}{dt}\big(\rho_m\vec{u}_m\cdot\vec{u}_m J\big)dV =$$

$$= \frac{1}{2}\int_{\mathcal{P}_0} \frac{d\rho_m}{dt}\vec{u}_m\cdot\vec{u}_m J dV + \int_{\mathcal{P}_0} \rho_m\vec{u}_m\cdot\frac{d\vec{u}_m}{dt} J dV + \frac{1}{2}\int_{\mathcal{P}_0} \rho_m\vec{u}_m\cdot\vec{u}_m\frac{dJ}{dt}dV. \quad (3.43)$$

Taking advantage of equation (3.7), it follows that

$$\frac{dK}{dt} = \frac{1}{2}\int_{\mathcal{P}_0} \frac{d\rho_m}{dt}\vec{u}_m\cdot\vec{u}_m J dV + \int_{\mathcal{P}_0} \rho_m\vec{u}_m\cdot\frac{d\vec{u}_m}{dt} J dV + \frac{1}{2}\int_{\mathcal{P}_0} \rho_m\vec{u}_m\cdot\vec{u}_m J div(\vec{u})_m dV. \quad (3.44)$$

Performing an inverse change of variables from $\mathcal{P}_0$ to $\mathcal{P}_t$ ($\mathcal{L}_t^{-1} : \Omega_t \to \Omega_0$) it is possible to obtain

$$\frac{dK}{dt} = \frac{1}{2}\int_{\mathcal{P}_t} \frac{d\rho}{dt}\vec{u}\cdot\vec{u} dV + \int_{\mathcal{P}_t} \rho\vec{u}\cdot\frac{d\vec{u}}{dt} dV + \frac{1}{2}\int_{\mathcal{P}_0} \rho\vec{u}\cdot\vec{u}\, div(\vec{u}) dV. \quad (3.45)$$

Exploiting the fact that

$$\frac{d\vec{u}}{dt} = \frac{\partial\vec{u}}{\partial t} + (\nabla\vec{u})\vec{u} = \frac{\partial\vec{u}}{\partial t} + (\vec{u}\cdot\nabla)\vec{u}, \quad (3.46)$$

the time derivative of kinetic energy turns out to be

$$\frac{dK}{dt} = \frac{1}{2}\int_{\mathcal{P}_t} \big(\frac{d\rho}{dt} + \rho div(\vec{u})\big)(\vec{u}\cdot\vec{u}) dV + \int_{\mathcal{P}_t} \rho\vec{u}\cdot\big(\frac{\partial\vec{u}}{\partial t} + (\vec{u}\cdot\nabla)\vec{u}\big) dV. \quad (3.47)$$

Taking into consideration the conservation of mass principle (3.25),the previous equation reduces to

$$\frac{dK}{dt} = \int_{\mathcal{P}_t} \rho\vec{u}\cdot\big(\frac{\partial\vec{u}}{\partial t} + (\vec{u}\cdot\nabla)\vec{u}\big) dV. \quad (3.48)$$

Substitution of the linear momentum equation (3.37) into (3.48) leads to

$$\frac{dK}{dt} = \int_{\mathcal{P}_t} \big(div(\boldsymbol{T}) + \rho\vec{f}\big)\cdot\vec{u} dV. \quad (3.49)$$

After noticing that

$$div(\boldsymbol{T}^T\vec{u}) = div(\boldsymbol{T})\vec{u} + \boldsymbol{T} : \nabla\vec{u}, \quad (3.50)$$

the following equation is obtained:

$$\frac{dK}{dt} = \int_{\mathcal{P}_t} div(\boldsymbol{T}^T\vec{u}) dV + \int_{\mathcal{P}_t} \rho\vec{f}\cdot\vec{u} dV - \int_{\mathcal{P}_t} \boldsymbol{T} : \nabla\vec{u} dV. \quad (3.51)$$

Applying, once again, the divergence theorem and taking into consideration the symmetry of the Cauchy stress tensor, the proof is concluded:

$$\frac{dK}{dt} = \int_{\partial\mathcal{P}_t} (\boldsymbol{T}\vec{n})\cdot\vec{u} dS + \int_{\mathcal{P}_t} \vec{u}\cdot\rho\vec{f} dV - \int_{\mathcal{P}_t} \boldsymbol{T} : \boldsymbol{D} dV. \quad (3.52)$$

□

Next, a definition of internal energy is required. The internal energy U of an arbitrary portion $\mathcal{P}_t$ of the continuum body may be defined as

$$U = \int_{\mathcal{P}_t} \rho e dV, \tag{3.53}$$

where e represents the density of internal energy per unit mass.

The heat incoming into $\mathcal{P}_t$ per unit time can be expressed through the following equation

$$Q = \int_{\mathcal{P}_t} \rho r dV - \int_{\partial\mathcal{P}_t} h dS, \tag{3.54}$$

where h stays for the outgoing heat per unit surface through the boundary of $\mathcal{P}_t$, per unit time. The first integral in equation (3.54) represents the incoming heat due to eventual radiation experienced by $\mathcal{P}_t$, being r a density of radiation heat per unit mass, per unit time.

Taking advantage of the Cauchy theorem, h may be written as $h = \vec{q} \cdot \vec{n}$ and

$$Q = \int_{\mathcal{P}_t} \rho r dV - \int_{\partial\mathcal{P}_t} \vec{q} \cdot \vec{n} dS. \tag{3.55}$$

Remark 3.1. *The surface integral in equation (3.55) describes the heat incoming through the boundary of $\mathcal{P}_t$ (with vector field $\vec{q}$ being the heat flux per unit surface, per unit time).*

Inserting definitions for K, Π_{ext}, U and Q into equation (3.39), it holds that

$$\int_{\partial\mathcal{P}_t} (\boldsymbol{T}\vec{n}) \cdot \vec{u} dS + \int_{\mathcal{P}_t} \vec{u} \cdot \rho \vec{f} dV + \int_{\mathcal{P}_t} \rho r dV - \int_{\partial\mathcal{P}_t} \vec{q} \cdot \vec{n} dS =$$

$$= \frac{d}{dt} \int_{\mathcal{P}_t} \rho e dV + \frac{d}{dt}\frac{1}{2} \int_{\mathcal{P}_t} \rho \vec{u} \cdot \vec{u} dV. \tag{3.56}$$

Exploiting Kinetic Energy Theorem (3.2), the previous equation simplifies in

$$\int_{\mathcal{P}_t} \rho r dV - \int_{\partial\mathcal{P}_t} \vec{q} \cdot \vec{n} dS = \frac{d}{dt} \int_{\mathcal{P}_t} \rho e dV - \int_{\mathcal{P}_t} \boldsymbol{T} : \boldsymbol{D} dV. \tag{3.57}$$

Let $\mathcal{P}_0$ be the portion of continuum body in the reference configuration corresponding to $\mathcal{P}_t$. Performing a change of variables ($\mathcal{L}_t : \Omega_0 \to \Omega_t$), it holds that

$$\frac{d}{dt} \int_{\mathcal{P}_t} \rho e dV = \frac{d}{dt} \int_{\mathcal{P}_0} \rho_m e_m J dV_m =$$

$$= \int_{\mathcal{P}_0} \frac{d\rho_m}{dt} e_m J dV_m + \int_{\mathcal{P}_0} \rho_m \frac{de_m}{dt} J dV_m + \int_{\mathcal{P}_0} \rho_m e_m \frac{dJ}{dt} dV_m. \tag{3.58}$$

Exploiting, once again, equation (3.7) then,

$$\frac{d}{dt} \int_{\mathcal{P}_t} \rho e dV_m = \int_{\mathcal{P}_0} \frac{d\rho_m}{dt} e_m J dV_m + \int_{\mathcal{P}_0} \rho_m \frac{de_m}{dt} J dV_m + \int_{\mathcal{P}_0} \rho_m e_m J div(\vec{u})_m dV_m. \tag{3.59}$$

Performing an inverse change of variables ($\mathcal{L}_t^{-1} : \Omega_t \to \Omega_t$), the following equation is obtained:

$$\frac{d}{dt}\int_{\mathcal{P}_t} \rho e dV = \int_{\mathcal{P}_t} \frac{d\rho}{dt} e dV + \int_{\mathcal{P}_t} \rho \frac{de}{dt} dV + \int_{\mathcal{P}_t} \rho e div(\vec{u}) dV, \tag{3.60}$$

$$\frac{d}{dt}\int_{\mathcal{P}_t} \rho e dV = \int_{\mathcal{P}_t} \big(\frac{d\rho}{dt} + \rho div(\vec{u})\big) e dV + \int_{\mathcal{P}_t} \rho \frac{de}{dt} dV. \tag{3.61}$$

Due to conservation of mass (3.25) the first integral on the right hand side of the previous equation is null.

Substituting back equation (3.61) into (3.57), the next relation holds true:

$$\int_{\mathcal{P}_t} \rho r dV - \int_{\partial\mathcal{P}_t} \vec{q}\cdot\vec{n} dS = \int_{\mathcal{P}_t} \rho \frac{de}{dt} dV - \int_{\mathcal{P}_t} \boldsymbol{T} : \boldsymbol{D} dV. \tag{3.62}$$

Taking into consideration that

$$\frac{de}{dt} = \frac{\partial e}{\partial t} + \vec{u}\cdot\nabla e, \tag{3.63}$$

it holds that

$$\int_{\mathcal{P}_t} \rho r dV - \int_{\partial\mathcal{P}_t} \vec{q}\cdot\vec{n} dS = \int_{\mathcal{P}_t} \rho \frac{\partial e}{\partial t} + \int_{\mathcal{P}_t} \rho\vec{u}\cdot\nabla e dV - \int_{\mathcal{P}_t} \boldsymbol{T} : \boldsymbol{D} dV. \tag{3.64}$$

An expression for $\vec{q}$ and e needs to be specified:

$$\vec{q} = -k\nabla T, \tag{3.65}$$

$$e = C_p T, \tag{3.66}$$

where T represents temperature, while k and C_p are respectively the heat conductivity and the specific heat capacity of the continuum body.

Remark 3.2. *Equation (3.65) is the Fourier Law which states that the heat flux per unit surface, per unit time is proportional to the opposite of temperature gradient.*

The substitution of equations (3.65-3.66) into (3.64) leads to

$$\int_{\mathcal{P}_t} \rho \frac{\partial (C_p T)}{\partial t} dV + \int_{\mathcal{P}_t} \rho\vec{u}\cdot\nabla(C_p T) dV = \int_{\mathcal{P}_t} \rho r dV + \int_{\partial\mathcal{P}_t} (k\nabla T)\cdot\vec{n} dS + \int_{\mathcal{P}_t} \boldsymbol{T} : \boldsymbol{D} dV, \tag{3.67}$$

which, applying divergence theorem, becomes

$$\int_{\mathcal{P}_t} \rho \frac{\partial (C_p T)}{\partial t} dV + \int_{\mathcal{P}_t} \rho\vec{u}\cdot\nabla(C_p T) dV = \int_{\mathcal{P}_t} \rho r dV + \int_{\mathcal{P}_t} div(k\nabla T) dV + \int_{\mathcal{P}_t} \boldsymbol{T} : \boldsymbol{D} dV. \tag{3.68}$$

Since equation (3.68) must hold for each portion $\mathcal{P}_t$ of continuum body, it must be

$$\rho \frac{\partial (C_p T)}{\partial t} + \rho\vec{u}\cdot\nabla(C_p T) = \rho r + div(k\nabla T) + \boldsymbol{T} : \boldsymbol{D}. \tag{3.69}$$

3.5 Governing Equations

Equation (3.69) together with (3.37) and (3.35) describe completely the motion of an incompressible continuum body from a thermo-mechanical point of view:

$$\begin{cases} \frac{\partial(\rho\vec{u})}{\partial t} + (\vec{u}\cdot\nabla)(\rho\vec{u}) = div(\boldsymbol{T}) + \rho\vec{f} \\ div(\vec{u}) = 0 \\ \rho\frac{\partial(C_pT)}{\partial t} + \rho\vec{u}\cdot\nabla(C_pT) = \rho r + div(k\nabla T) + \boldsymbol{T}:\boldsymbol{D}. \end{cases} \tag{3.70}$$

Under the hypothesis $\rho \in \mathbb{R}^+$, it holds that

$$\begin{cases} \rho\frac{\partial\vec{u}}{\partial t} + \rho(\vec{u}\cdot\nabla)\vec{u} = div(\boldsymbol{T}) + \rho\vec{f} \\ div(\vec{u}) = 0 \\ \rho\frac{\partial(C_pT)}{\partial t} + \rho\vec{u}\cdot\nabla(C_pT) = \rho r + div(k\nabla T) + \boldsymbol{T}:\boldsymbol{D}. \end{cases} \tag{3.71}$$

The previous system of equations holds true for any incompressible continuum body. As specified in chapter 2, both polymer matrices and filled compounds are going to be modelled as incompressbile generalized Newtonian fluids, characterized by the constitutive relation

$$\boldsymbol{T} = -P\boldsymbol{I} + 2\eta(\dot{\gamma},T)\boldsymbol{D}. \tag{3.72}$$

System (3.71) may be further manipulated since

$$div(\boldsymbol{T}) = div(-P\boldsymbol{I}) + div(2\eta(\dot{\gamma},T)\boldsymbol{D}) = -\nabla P + div(\eta(\dot{\gamma},T)(\nabla\vec{u} + \nabla\vec{u}^T)), \tag{3.73}$$

$$\boldsymbol{T}:\boldsymbol{D} = \boldsymbol{T}:\nabla\vec{u} = -P\boldsymbol{I}:\nabla\vec{u} + 2\eta(\dot{\gamma},T)\boldsymbol{D}:\nabla\vec{u} =$$

$$= -Pdiv(\vec{u}) + 2\eta(\dot{\gamma},T)\boldsymbol{D}:\nabla\vec{u}. \tag{3.74}$$

Due to incompressibility ($div(\vec{u}) = 0$), it holds that

$$\boldsymbol{T}:\boldsymbol{D} = 2\eta(\dot{\gamma},T)\boldsymbol{D}:\nabla\vec{u} = \eta(\dot{\gamma},T)(\nabla\vec{u} + \nabla\vec{u}^T):\nabla\vec{u}. \tag{3.75}$$

System (3.71), specified for an incompressible generalized Newtonian fluid, is

$$\begin{cases} \rho\frac{\partial\vec{u}}{\partial t} + \rho(\vec{u}\cdot\nabla)\vec{u} = -\nabla P + div(\eta(\dot{\gamma},T)(\nabla\vec{u} + \nabla\vec{u}^T)) + \rho\vec{f} \\ div(\vec{u}) = 0 \\ \rho\frac{\partial(C_pT)}{\partial t} + \rho\vec{u}\cdot\nabla(C_pT) = \rho r + div(k\nabla T) + \eta(\dot{\gamma},T)(\nabla\vec{u} + \nabla\vec{u}^T):\nabla\vec{u}. \end{cases} \tag{3.76}$$

Remark 3.3. *The last equation in system (4.1) equals the time derivative of the internal energy to the sum of radiation heat (ρr), the incoming heat flux ($k\nabla T$) and the kinetic energy that goes dissipated into heat due to viscosity ($\eta(\dot{\gamma},T)(\nabla\vec{u} + \nabla\vec{u}^T):\nabla\vec{u}$).*

Remark 3.4. *In computational fluid dynamics, an eulerian approach is frequently adopted: a fixed domain* Ω *is considered, filled by a fluid at each time instant* $t \geq 0$*:*

$$\Omega \subset \Omega_t \qquad \forall t \in [0, +\infty). \tag{3.77}$$

Since system (4.1) must hold for any $\boldsymbol{x} \in \Omega_t$*, for any* $t \in [0, +\infty)$*, it must be satisfied for any* $(\boldsymbol{x}, t) \in \Omega \times [0, +\infty)$*.*

Chapter 4

Rheometry

In order to analyze viscosity of a generalized Newtonian fluid, rheometers are employed. A rheometer is an apparatus where a generalized Newtonian fluid is poured and may undergo extensional flows (extensional rheometers) rather than steady or oscillatory shear flows (shear rheometers). Couette rheometer, plate-plate rheometer, cone-plate rheometer and capillary rheometer are some examples of shear rheometers, frequently used to investigate viscosity of generalized Newtonian fluids over a wide range of shear rates. Shear rheometers mostly work imposing a kinematic quantity (velocity, shear rate, displacement) and measuring a dynamic quantity (pressure drop, shear stress, momentum). Since the shear flow experienced by the fluid in the rheomether (or an approximation of the flow) is known, combining the imposed kinematic quantity and the measured dynamic quantity, it is possible to estimate the viscosity of the generalized Newtonian fluid at a given shear rate. Temperature is usually set in such a way that the flow of the generalized Newtonian fluid in the rheometer can be approximated as isothermal. Hereafter, the capillary rheometer is going to be investigated.

4.1 Capillary Rheometer

Capillary rheometers are widely employed in order to determine the viscosity of generalized Newtonian fluids, especially highly viscous fluids, in a broad range of shear rates. In a capillary rheometer the fluid is collected into a tank or a reservoir, a piston pushes the fluid into a duct (the capillary) with cross sectional area much smaller than the cross sectional area of the tank. The constant velocity of the piston is imposed (kinematic quantity) while the pressure drop between the reservoir and the outlet of the capillary (dynamic quantity) is measured. The cross section of the capillary may be circular (capillary die) or rectangular (slit die). The temperature of the piston is set equal to the temperature of the fluid as well as the temperature of the walls of both the tank and the capillary. Temperature experimental conditions are set in such a way that the flow experienced by the fluid is approximately isothermal; in practice, especially for highly viscous fluid at high shear rates, energy dissipation due to viscous forces leads to an increase in temperature along the capillary. At low shear rates and sufficiently far from the inlet of the capillary, the fluid experiences approximately an isothermal shear flow (Poiseuille flow). In the next two subsections Poiseuille flows of a power law fluid in cylindrical duct (capillary die) and rectangular duct (slit die) are analytically examined.

4.2 Isothermal Poiseuille Flow for Cylindrical Duct

Consider an incompressible power law fluid flowing in a cylindrical duct of radius R. The motion of an incompressible generalized Newtonian fluid is governed by the following equations (chapter 3):

$$\begin{cases} \rho\frac{\partial \vec{u}}{\partial t} + \rho(\vec{u}\cdot\nabla)\vec{u} = -\nabla P + div(\eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T)) + \rho\vec{f} \\ div(\vec{u}) = 0 \\ \rho\frac{\partial(C_pT)}{\partial t} + \rho\vec{u}\cdot\nabla(C_pT) = \rho r + div(k\nabla T) + \eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T):\nabla\vec{u}. \end{cases} \tag{4.1}$$

In this section the energy equation in (4.1) is neglected since the focus is on the isothermal Poiseuille flow. The conservation of momentum equation and the incompressibility constraint impose that

$$\begin{cases} \frac{\partial \vec{u}}{\partial t} + (\vec{u}\cdot\nabla)\vec{u} = -\nabla p + div(2\nu(\dot{\gamma})\boldsymbol{D}) + \vec{f} \\ div(\vec{u}) = 0, \end{cases} \tag{4.2}$$

where p and $\nu(\dot{\gamma})$ denote pressure P and viscosity $\eta(\dot{\gamma})$ divided by the fluid density ρ. For a power law fluid it holds that

$$\nu(\dot{\gamma}) = \frac{\eta_0}{\rho}(\lambda\dot{\gamma})^{n-1} = \nu_0(\dot{\gamma})^{n-1}. \tag{4.3}$$

Let r, θ, z be cylindrical coordinates such that the z axis is aligned with the axis of the duct. Assuming that the velocity field has the form $\vec{u} = u_z(r)\vec{e}_z$ and neglecting body forces, system (4.2) (rewritten in cylindrical coordinates) reduces to

$$\begin{cases} 0 = -\frac{\partial p}{\partial r} \\ 0 = -\frac{1}{r}\frac{\partial p}{\partial \theta} \\ 0 = -\frac{\partial p}{\partial z} + \left[\frac{1}{r}\frac{\partial}{\partial r}(\nu(\dot{\gamma})r\frac{\partial u_z}{\partial r})\right]. \end{cases} \tag{4.4}$$

from which is possible to deduce the following relations

$$\begin{cases} p = p(z) \\ \frac{\partial p}{\partial z} = \left[\frac{1}{r}\frac{\partial}{\partial r}(\nu(\dot{\gamma})r\frac{\partial u_z}{\partial r})\right] \\ \nu(\dot{\gamma}) = \nu_0(\dot{\gamma})^{n-1} = \nu_0\left|\frac{\partial u_z}{\partial r}\right|^{n-1}. \end{cases} \tag{4.5}$$

The first equation in system (4.5) states that pressure is constant over cross flow sections (i.e. $\frac{\partial p}{\partial r} = \frac{\partial p}{\partial \theta}$). Since the right hand side of the second equation of system (4.5) depends only on r, while the left hand side depends only on coordinate z, by dimensional considerations, it must hold that

$$\frac{\partial p}{\partial z} = \Big[\frac{1}{r}\frac{\partial}{\partial r}(\nu(\dot{\gamma})r\frac{\partial u_z}{\partial r})\Big] = A. \tag{4.6}$$

The previous equation implies that

$$\begin{cases} p = Az + B \\ u_z = -\Big(\frac{|A|}{2\nu_0}\Big)^{1/n}\frac{n}{n+1}r^{\frac{n+1}{n}} + C. \end{cases} \tag{4.7}$$

Remark 4.1. *Equation* $\Big[\frac{1}{r}\frac{\partial}{\partial r}(\nu(\dot{\gamma})r\frac{\partial u_z}{\partial r})\Big] = A$ *has been integrated twice. The first integration leads to*

$$\nu(\dot{\gamma})\frac{\partial u_z}{\partial r} = \frac{1}{2}Ar + \frac{B}{r}. \tag{4.8}$$

To enforce boundedness of the solution, B must be null. The previous equation becomes

$$\nu_0\Big|\frac{\partial u_z}{\partial r}\Big|^{n-1}\frac{\partial u_z}{\partial r} = \frac{1}{2}Ar. \tag{4.9}$$

Assuming that the velocity is maximum at the centre of the duct ($r = 0$) and is minimum at the wall (r=R) (i.e. $\frac{\partial u_z}{\partial r} < 0$) and integrating, the second equation in system (4.5) is obtained.

Since pressure is constant on cross flow sections, if P_1 and P_2 are two points on the z axis, letting Δp and L be, respectively, the pressure drop and the distance in between them, it holds

$$A = \frac{\Delta p}{L} < 0. \tag{4.10}$$

Substitution of the previous equality in equations (4.7) leads to

$$\begin{cases} p = \frac{\Delta p}{L}z + B \\ u_z = -\Big(\frac{|\Delta p|}{2\nu_0 L}\Big)^{1/n}\frac{n}{n+1}r^{\frac{n+1}{n}} + C. \end{cases} \tag{4.11}$$

In order to determine the numerical value of constant C, a velocity boundary condition on the walls of the cylindrical duct needs to be specified:

- no slip boundary condition : $u_z(R) = 0$;
- slip boundary condition : $u_z(R) = f(\vec{\tau}_t)$;

where f is a function of the tangential component of the surface forces exerted on the fluid at the walls of the cylindrical duct:

$$\vec{\tau}_t = \boldsymbol{T}\vec{n} - (\boldsymbol{T}\vec{n} \cdot \vec{n})\vec{n}. \tag{4.12}$$

Imposing a non slip boundary condition $u_z(R) = 0$, pressure and velocity fields are described by

$$\begin{cases} p = \frac{\Delta p}{L} z + B \\ u_z = \left(\frac{|\Delta p|}{2\nu_0 L}\right)^{1/n} \frac{n}{n+1}(R^{\frac{n+1}{n}} - r^{\frac{n+1}{n}}). \end{cases} \tag{4.13}$$

Remark 4.2. *Define*

$$U_{avg} = \frac{1}{\pi R^2} \int_0^R \int_0^{2\pi} u_z r d\theta dr = \frac{2}{R^2} \int_0^R \Big(\frac{|\Delta p|}{2\nu_0 L}\Big)^{1/n} \frac{n}{n+1}(R^{\frac{n+1}{n}} - r^{\frac{n+1}{n}}) r dr, \tag{4.14}$$

then

$$U_{avg} = \frac{2}{R^2}\Big(\frac{|\Delta p|}{2\nu_0 L}\Big)^{1/n} \frac{n}{n+1} \int_0^R (R^{\frac{n+1}{n}} - r^{\frac{n+1}{n}})\, r dr, \tag{4.15}$$

$$U_{avg} = \frac{2}{R^2}\Big(\frac{|\Delta p|}{2\nu_0 L}\Big)^{1/n} \frac{n}{n+1} \frac{n+1}{2(3n+1)} = \Big(\frac{|\Delta p|}{2\nu_0 L}\Big)^{1/n} \frac{n}{3n+1} R^{\frac{n+1}{n}} \tag{4.16}$$

and

$$u_z = U_{avg} \frac{3n+1}{n+1}\Big(1 - \Big(\frac{r}{R}\Big)^{\frac{n+1}{n}}\Big). \tag{4.17}$$

Remark 4.3. *If the fluid is Newtonian (n=1), then the velocity field recovers the well known parabolic expression:*

$$u_z = 2U_{avg}\Big(1 - \Big(\frac{r}{R}\Big)^2\Big).$$

Remark 4.4. *If the fluid experiences sufficiently large shear rate in the cylindrical duct and its viscosity obeys a Bird Carreau model or a Cross model, the velocity profile still is described by (4.17), since the Bird Carreau viscosity expression and Cross viscosity expression reduce to the power law expression for high shear rates.*

If slip boundary condition is imposed on the lateral surface of the cylindrical duct, some more calculations are in order. Since on the walls of the duct

$$\boldsymbol{T} = -P\boldsymbol{I} + \Big(\eta(\dot{\gamma})\frac{\partial u_z}{\partial r}\Big|_R\Big)(\vec{e}_r \otimes \vec{e}_z + \vec{e}_z \otimes \vec{e}_r) \qquad \text{and} \qquad \vec{n} = \vec{e}_r, \tag{4.18}$$

then

$$\vec{\tau}_t = \eta(\dot{\gamma})\frac{\partial u_z}{\partial r}\Big|_R \vec{e}_z. \tag{4.19}$$

Assuming $u_z(R) = f(\vec{\tau_t}) = f(\eta(\dot{\gamma})\frac{\partial u_z}{\partial r}\Big|_R \vec{e}_z)$, system (4.11) becomes

$$\begin{cases} p = \frac{\Delta p}{L}z + B \\ u_z = \Big(\frac{|\Delta p|}{2\nu_0 L}\Big)^{1/n}\frac{n}{n+1}(R^{\frac{n+1}{n}} - r^{\frac{n+1}{n}}) + f(-\eta(\dot{\gamma})\Big(\frac{|\Delta p|}{2L\nu_0}\Big)^{1/n} R^{1/n}\vec{e}_z). \end{cases} \tag{4.20}$$

The most frequently used slip boundary condition is the Navier slip condition [23]:

$$f(\vec{\tau_t}) = K_L||\vec{\tau_t}||. \tag{4.21}$$

Sometimes boundary condition (4.21) is substituted with the non linear Navier slip boundary condition [23]:

$$f(\vec{\tau_t}) = K_{NL}||\vec{\tau_t}||^m \tag{4.22}$$

Some other popular slip boundary conditions [24] are the Hatzikiriakos boundary condition (4.23.A) and the asymptotic boundary condition (4.23.B):

$$f(\vec{\tau_t}) = K_{H1}sinh(K_{H2}||\vec{\tau_t}||), \qquad f(\vec{\tau_t}) = K_{A1}log(1 + K_{A2}||\vec{\tau_t}||). \tag{4.23}$$

In this research only the linear Navier slip boundary condition (4.21) is taken into consideration. Substituting equations (4.21) and (4.19) into (4.20), pressure and velocity fields are described by

$$\begin{cases} p = \frac{\Delta p}{L}z + B \\ u_z = \Big(\frac{|\Delta p|}{2\nu_0 L}\Big)^{1/n}\frac{n}{n+1}(R^{\frac{n+1}{n}} - r^{\frac{n+1}{n}}) + \frac{K_L|\Delta P|R}{2L}. \end{cases} \tag{4.24}$$

Remark 4.5. *Defining the average velocity on a cross section of the cylindrical duct as*

$$U_{avg} = \frac{2}{R^2}\int_0^R u_z r dr = \frac{2}{R^2}\int_0^R \Big(\frac{|\Delta p|}{2\nu_0 L}\Big)^{1/n}\frac{n}{n+1}(R^{\frac{n+1}{n}} - r^{\frac{n+1}{n}})rdr + \frac{1}{2}K_L\frac{|\Delta P|R}{L}, \tag{4.25}$$

it holds

$$U_{avg} = \frac{2}{R^2}\Big(\frac{|\Delta p|}{2\nu_0 L}\Big)^{1/n}\frac{n}{n+1}\int_0^R (R^{\frac{n+1}{n}} - r^{\frac{n+1}{n}})\, rdr + \frac{1}{2}K_L\frac{|\Delta P|R}{L}, \tag{4.26}$$

$$U_{avg} = \Big(\frac{|\Delta p|}{2\nu_0 L}\Big)^{1/n}\frac{n}{3n+1}R^{\frac{n+1}{n}} + \frac{1}{2}K_L\frac{|\Delta P|R}{L}. \tag{4.27}$$

4.3 Isothermal Poiseuille Flow for Rectangular Duct

Consider an incompressible power law fluid flowing through a rectangular duct. If the width W of the duct is much larger than the height $2H$ of the duct (i.e. $W >> 2H$), the flow is well approximated by the flow of an incompressible power law fluid between two parallel planes. Consider a system of cartesian coordinates in $\mathbb{R}^3$. Let $\{O, \vec{e}_x, \vec{e}_y, \vec{e}_z\}$ be cartesian reference frame in $\mathbb{R}^3$ and let π_1 and π_2 be two planes such that:

$$\pi_1 = \{(x, y, z) \in \mathbb{R}^3 \;:\; y = -H\},$$
$$\pi_2 = \{(x, y, z) \in \mathbb{R}^3 \;:\; y = +H\}. \tag{4.28}$$

Assume that the flow is aligned with the x direction. The isothermal Poiseuille flow through the rectangular duct ($W >> 2H$) is governed by the following system of equations :

$$\begin{cases} \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} = -\nabla p + div(2\nu(\dot{\gamma})\boldsymbol{D}) + \vec{f} \\ div(\vec{u}) = 0, \end{cases} \tag{4.29}$$

where $p = P/\rho$ and $\nu(\dot{\gamma}) = \eta(\dot{\gamma})/\rho$. In the next it is assumed that the viscosity of the fluid has a power law behaviour with respect to the shear rate:

$$\nu(\dot{\gamma}) = \nu_0(\dot{\gamma})^{n-1}. \tag{4.30}$$

Assuming $\vec{u} = u_x(y)\vec{e}_x$ and neglecting body forces, the previous system of equations becomes

$$\begin{cases} 0 = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial y}(\nu \frac{\partial u_x}{\partial y}) \\ 0 = -\frac{\partial p}{\partial y} \\ 0 = -\frac{\partial p}{\partial z}, \end{cases} \tag{4.31}$$

with

$$\nu = \nu(\dot{\gamma}) = \nu_0 \left| \frac{\partial u_x}{\partial y} \right|^{n-1}. \tag{4.32}$$

From equations (4.31) it is possible to state

$$\begin{cases} p = p(x) \\ \frac{\partial p}{\partial x} = \nu_0 \frac{\partial}{\partial y}(\frac{\partial u_x}{\partial y} \left| \frac{\partial u_x}{\partial y} \right|^{n-1}). \end{cases} \tag{4.33}$$

In order to proceed with the solution of system (4.33), some more assumptions on velocity profile are in order. As in the cylindrical case, velocity should be maximum for $y = 0$ and should be minimum for $y = \pm H$. Due to these considerations, it's reasonable to assume

$$\begin{cases} \frac{\partial u_x}{\partial y} < 0 \;\text{ for }\; 0 < y < H \\ \frac{\partial u_x}{\partial y} > 0 \;\text{ for }\; -H < y < 0. \end{cases} \tag{4.34}$$

Calculations are developed only for $y > 0$. System (4.33) reduces to

$$\begin{cases} p = p(x) \\ \frac{\partial p}{\partial x} = \nu_0(-1)\frac{\partial}{\partial y}\left| \frac{\partial u_x}{\partial y} \right|^{n} \;\; y > 0. \end{cases} \tag{4.35}$$

The first equation in system (4.35) states that pressure is constant over cross flow sections (i.e. pressure does not depend on coordinates y and z). Since the left hand side of the second equation in system (4.35) depends only on x while the right hand side is a function of the only variable y, by dimensional considerations, it must be

$$\frac{\partial p}{\partial x} = \nu_0(-1)\frac{\partial}{\partial y}\left|\frac{\partial u_x}{\partial y}\right|^n = A. \tag{4.36}$$

The previous equation leads to

$$\begin{cases} p = Ax + B \\ (-1)\nu_0\left|\frac{\partial u_x}{\partial y}\right|^n = Ay + C \;\; y > 0. \end{cases} \tag{4.37}$$

In equations (4.37) A represents the pressure drop per unit length. Let P_1 and P_2 be two points on the x axis and let L be the distance between them. If the pressure drop between P_2 and P_1 is denoted by Δp, then A turns out to be

$$A = \frac{\Delta p}{L} < 0. \tag{4.38}$$

Since $\left(\frac{\partial u_x}{\partial y}\Big|_{y=0}\right)^n = 0$, then $C = 0$ and

$$\begin{cases} p = Ax + B \\ \left|\frac{\partial u_x}{\partial y}\right|^n = \frac{|A|}{\nu_0}y \;\; y > 0. \end{cases} \tag{4.39}$$

Solving the second equation in system (4.39), the following equations are obtained:

$$\begin{cases} p = Ax + B \\ u_x = -\left(\frac{|A|}{\nu_0}\right)^{1/n}\frac{n}{n+1}y^{\frac{n+1}{n}} + D \;\; y > 0. \end{cases} \tag{4.40}$$

Imposing a no slip boundary condition on the walls of the rectangular duct (i.e. $u(H) = 0$) and substituting $A = \frac{\Delta p}{L}$, it holds that

$$\begin{cases} p = \frac{\Delta p}{L}x + B \\ u_x = \left(\frac{|\Delta p|}{L\nu_0}\right)^{1/n}\frac{n}{n+1}(H^{\frac{n+1}{n}} - y^{\frac{n+1}{n}}) \;\; y > 0. \end{cases} \tag{4.41}$$

Analogous calculations show that, for $y < 0$,

$$u_x = \left(\frac{|\Delta p|}{L\nu_0}\right)^{1/n}\frac{n}{n+1}(H^{\frac{n+1}{n}} - (-y)^{\frac{n+1}{n}}). \tag{4.42}$$

Finally, velocity and pressure fields can be described by the following equations:

$$\begin{cases} p = \frac{\Delta p}{L}x + B \\ u_x = \left(\frac{|\Delta p|}{L\nu_0}\right)^{1/n}\frac{n}{n+1}(H^{\frac{n+1}{n}} - |y|^{\frac{n+1}{n}}). \end{cases} \tag{4.43}$$

Remark 4.6. *Defining the average velocity on a cross section of the rectangular duct as*

$$U_{avg} = \frac{1}{2H}\int_{-H}^{H} u_x dy = \frac{1}{2H}\int_{-H}^{H}\Big(\frac{|\Delta p|}{L\nu_0}\Big)^{1/n}\frac{n}{n+1}(H^{\frac{n+1}{n}} - |y|^{\frac{n+1}{n}})dy, \quad (4.44)$$

it is possible to state that

$$U_{avg} = \frac{1}{2H}\Big(\frac{|\Delta p|}{L\nu_0}\Big)^{1/n}\frac{n}{n+1}2\int_{0}^{H}(H^{\frac{n+1}{n}} - y^{\frac{n+1}{n}})dy, \quad (4.45)$$

$$U_{avg} = \frac{1}{H}\Big(\frac{|\Delta p|}{L\nu_0}\Big)^{1/n}\frac{n}{n+1}\,\frac{n+1}{2n+1}H^{\frac{2n+1}{n}} = \Big(\frac{|\Delta p|}{L\nu_0}\Big)^{1/n}\frac{n}{2n+1}H^{\frac{n+1}{n}}, \quad (4.46)$$

and

$$u_x = U_{avg}\frac{2n+1}{n+1}\Big(1 - \Big|\frac{y}{H}\Big|^{\frac{n+1}{n}}\Big). \quad (4.47)$$

Remark 4.7. *If the fluid is Newtonian (n=1), then the velocity field recovers the well known parabolic expression:*

$$u_x = \frac{3}{2}U_{avg}\Big(1 - \Big(\frac{y}{H}\Big)^2\Big).$$

Remark 4.8. *If the viscosity of the fluid in the rectangular duct obeys a Bird-Carreau or a Cross model, the velocity profile is still well described by equation (4.17), provided that high shear rates are achieved: at high shear rates, both the Bird-Carreau and Cross viscosity models reduce to the power law viscosity model.*

On the other hand, if a linear Navier slip boundary condition (4.21) is assumed on the walls of the rectangular duct, pressure and velocity are described by the following system of equations

$$\begin{cases} p = \frac{\Delta p}{L}x + B \\ u_x = \Big(\frac{|\Delta p|}{L\nu_0}\Big)^{1/n}\frac{n}{n+1}(H^{\frac{n+1}{n}} - |y|^{\frac{n+1}{n}}) + K_L\frac{|\Delta P|}{L}H. \end{cases} \quad (4.48)$$

Remark 4.9. *Computing the average velocity on a cross section of the rectangular duct as*

$$U_{avg} = \frac{1}{2H}\int_{-H}^{H} u_x dy = \frac{1}{2H}\int_{-H}^{H}\Big(\frac{|\Delta p|}{L\nu_0}\Big)^{1/n}\frac{n}{n+1}(H^{\frac{n+1}{n}} - |y|^{\frac{n+1}{n}})dy + K_L\frac{|\Delta P|}{L}H, \quad (4.49)$$

it is possible to state that

$$U_{avg} = \frac{1}{2H}\Big(\frac{|\Delta p|}{L\nu_0}\Big)^{1/n}\frac{n}{n+1}2\int_{0}^{H}(H^{\frac{n+1}{n}} - y^{\frac{n+1}{n}})dy + K_L\frac{|\Delta P|}{L}H, \quad (4.50)$$

$$U_{avg} = \Big(\frac{|\Delta p|}{L\nu_0}\Big)^{1/n}\frac{n}{2n+1}H^{\frac{n+1}{n}} + K_L\frac{|\Delta P|}{L}H. \quad (4.51)$$

4.4 Rabinowitsch Analysis for Cylindrical Duct

A capillary rheometer with circular cross section capillary (capillary die) measures a pressure drop (dynamic quantity) corresponding to an imposed flowrate (kinematic quantity). Rabinowitsch analysis allows to compute wall shear rates corresponding to the imposed flowrates. In this paragraph Rabinowitsch analysis is going to be described for a capillary rheometer with capillary die length L and circular capillary die cross section with radius R. Let $(O,\vec{e}_r,\vec{e}_\theta,\vec{e}_z)$ be a cylindrical reference frame such that the z axis is aligned with the axis of the capillary die.

Let CV be a cylindrical control volume inside the capillary die with radius $r < R$ and length L, with $CS = \partial CV$ (figure (4.1)). The conservation of momentum equation states that

$$\int_{CV} \rho\vec{f}dV + \int_{CV} div(\boldsymbol{T})dV = \int_{CV} \rho\frac{\partial\vec{u}}{\partial t}dV + \int_{CV} \rho(\vec{u}\cdot\nabla)\vec{u}dV, \tag{4.52}$$

$$\int_{CV} \rho\vec{f}dV + \int_{CV} div(-P\boldsymbol{I} + 2\eta(\dot{\gamma})\boldsymbol{D})dV = \int_{CV} \rho\frac{\partial\vec{u}}{\partial t}dV + \int_{CV} \rho(\vec{u}\cdot\nabla)\vec{u}dV. \tag{4.53}$$

In the capillary die, the velocity field satisfies

$$\begin{cases} \vec{u} = u_z(r)\vec{e}_z \\ \frac{\partial u_z}{\partial r} < 0. \end{cases} \tag{4.54}$$

As a straightforward consequence, the shear rate turns out to be

$$\dot{\gamma} = \left|\frac{\partial u_z}{\partial r}\right| = -\frac{\partial u_z}{\partial r}. \tag{4.55}$$

Neglecting body forces, equation (4.53) becomes

$$\int_{CV} div(-P\boldsymbol{I})dV + \int_{CV} div(2\eta(\dot{\gamma})\boldsymbol{D})dV = 0, \tag{4.56}$$

since the flow is stationary and $(\vec{u}\cdot\nabla)\vec{u} = 0\cdot$

Considering that the velocity field has the form specified by equation (4.54), it is possible to prove that

$$div(\eta(\dot{\gamma})(\nabla\vec{u})^T) = 0. \tag{4.57}$$

Finally, applying divergence theorem, the conservation of momentum equation reduces to

$$\int_{CS} P\vec{n}dS = \int_{CS} \eta(\dot{\gamma})\nabla\vec{u}\ \vec{n}dS. \tag{4.58}$$

The projection of the previous equation along the z direction leads to

$$\int_{Sb1} PdS - \int_{Sb0} PdS = \int_{Slat} \eta(\dot{\gamma})\frac{\partial u_z}{\partial r}dS, \tag{4.59}$$

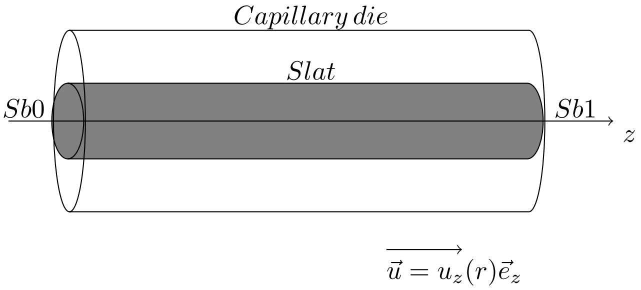


Figure 4.1: Cylindrical control volume (grey) inside capillary die

where $Sb0$ and $Sb1$ are, respectively, the upstream and the downstream faces of CV and $Slat$ is the lateral face of CV (figure (4.1)).

It is worth to notice that $\frac{\partial u_z}{\partial r}$,$\dot{\gamma}$ and $\eta(\dot{\gamma})$ are constant on $Slat$. Assuming that P is constant on faces $Sb0$ and $Sb1$, equation (4.59) can be rewritten as

$$P_1 \pi r^2 - P_0 \pi r^2 = 2\pi r L \tau, \tag{4.60}$$

where τ is $\mu(\dot{\gamma})\frac{\partial u_z}{\partial r}$ evaluated on $Slat$, P_1 and P_0 are pressures on faces $Sb1$ and $Sb0$, respectively.

The previous equation can be further manipulated, obtaining

$$\tau = \frac{P_1 - P_0}{2L} r = \frac{\Delta P}{2L} r = -\frac{|\Delta P|}{2L} r. \tag{4.61}$$

Writing the previous equation for $r = R$, it is possible to compute the wall shear stress $\tau_w = \tau|_{r=R}$:

$$\tau_w = \frac{P_1 - P_0}{2L} R = \frac{\Delta P}{2L} R. \tag{4.62}$$

Let Q be the flux through the cross section of the capillary die:

$$Q = \int_0^R \int_0^{2\pi} u_z r d\theta dr = 2\pi \int_0^R u_z r dr. \tag{4.63}$$

Integrating by parts and assuming $u_z(R) = 0$, it holds that

$$Q = 2\pi \Big[\frac{1}{2} r^2 u_z\Big]_0^R - 2\pi \int_0^R \frac{1}{2} r^2 \frac{\partial u_z}{\partial r} dr = -\pi \int_0^R r^2 \frac{\partial u_z}{\partial r} dr = \pi \int_0^R r^2 \dot{\gamma} dr. \tag{4.64}$$

Equation (4.61) is exploited in order to perform the following change of variables:

$$\begin{cases} r = -2\frac{L\tau}{|\Delta P|} \\ dr = -2\frac{L}{|\Delta P|} d\tau \\ r = R \Leftrightarrow \tau = \tau_w \\ r = 0 \Leftrightarrow \tau = 0 \end{cases} \tag{4.65}$$

Equation (4.64) becomes

$$Q = \pi \int_0^{\tau_w} \left(-2\frac{L}{|\Delta P|}\right)^3 \tau^2 \dot{\gamma} d\tau. \tag{4.66}$$

Substituting equation (4.62) into equation (4.66) the flowrate is

$$Q = \frac{\pi R^3}{\tau_w^3} \int_0^{\tau_w} \tau^2 \dot{\gamma} d\tau. \tag{4.67}$$

Define the apparent shear rate as

$$\dot{\gamma}_{app} = \frac{4Q}{\pi R^3}. \tag{4.68}$$

Making use of equation (4.68), it is possible to state the following relation

$$\dot{\gamma}_{app} = 4\frac{Q}{\pi R^3} = 4\frac{1}{\tau_w^3} \int_0^{\tau_w} \tau^2 \dot{\gamma} d\tau. \tag{4.69}$$

Rearranging the previous equation as

$$\tau_w^3 \dot{\gamma}_{app} = 4 \int_0^{\tau_w} \tau^2 \dot{\gamma} d\tau \tag{4.70}$$

and performing partial derivative $\frac{\partial}{\partial \tau_w}$, equation (4.71) is obtained:

$$\frac{\partial \dot{\gamma}_{app}}{\partial \tau_w} \tau_w^3 + 3\dot{\gamma}_{app} \tau_w^2 = 4\tau_w^2 \dot{\gamma}_w, \tag{4.71}$$

where $\dot{\gamma}_w$ is the shear rate at the wall.

After some calculations, the following relation is obtained:

$$\frac{1}{4}\frac{\partial \dot{\gamma}_{app}}{\partial \tau_w} \tau_w + \frac{3}{4}\dot{\gamma}_{app} = \dot{\gamma}_w. \tag{4.72}$$

Since $\tau_w = \frac{\partial \tau_w}{\partial log(\tau_w)}$ and $\dot{\gamma}_{app} = \frac{\partial \dot{\gamma}_{app}}{\partial log(\dot{\gamma}_{app})}$, the previous equation can be rewritten in the form

$$\frac{1}{4}\frac{\partial log(\dot{\gamma}_{app})}{\partial log(\tau_w)} \dot{\gamma}_{app} + \frac{3}{4}\dot{\gamma}_{app} = \dot{\gamma}_w. \tag{4.73}$$

Defining $\frac{1}{n} = \frac{\partial log(\dot{\gamma}_{app})}{\partial log(\tau_w)}$, then

$$\frac{1}{4n}\dot{\gamma}_{app} + \frac{3}{4}\dot{\gamma}_{app} = \dot{\gamma}_w \Rightarrow \dot{\gamma_w} = \frac{3n+1}{4n}\dot{\gamma}_{app}. \tag{4.74}$$

Remark 4.10. *If the decimal logarithm is used in equation (4.73), still (4.74) holds*

$$\frac{1}{4}\frac{\partial Log(\dot{\gamma}_{app})}{\partial Log(\tau_w)} \dot{\gamma}_{app} + \frac{3}{4}\dot{\gamma}_{app} = \dot{\gamma}_w, \tag{4.75}$$

since $\tau_w = \frac{1}{log(10)} \frac{\partial \tau_w}{\partial Log(\tau_w)}$ *and* $\dot{\gamma}_{app} = \frac{1}{log(10)} \frac{\partial \dot{\gamma}_{app}}{\partial Log(\dot{\gamma}_{app})}$.

Defining $\frac{1}{n} = \frac{\partial Log(\dot{\gamma}_{app})}{\partial Log(\tau_w)} = \frac{\partial log(\dot{\gamma}_{app})}{\partial log(\tau_w)}$, *it still holds that*

$$\dot{\gamma_w} = \frac{3n+1}{4n} \dot{\gamma}_{app}. \tag{4.76}$$

Once computed $\dot{\gamma}_w$ through equation (4.74), the viscosity of the generalized Newtonian fluid under examination is estimated as

$$\eta \simeq \frac{\tau_w}{\dot{\gamma}_w}. \tag{4.77}$$

4.5 Mooney Analysis for Cylindrical Duct

If the no slip boundary condition on the walls of the capillary die fails, it is not possible to assume $u_z(R) = 0$ when integrating by parts equation (4.63). In order to perform Rabinowitsch analysis, the apparent shear rates $\dot{\gamma}_{app}$ (kinematic quantity imposed on the rheometer) must undergo Mooney correction procedure (4.90).

Let $u_z(R) = v_{slip}$, due to the symmetry of the cylindrical domain v_{slip} (which is unknown) is assumed to be independent of θ and z (i.e. $v_{slip} \in \mathbb{R}^+$).

Integration by parts of equation (4.63) leads to

$$Q = 2\pi \Big[\frac{1}{2} r^2 u_z\Big]_0^R - 2\pi \int_0^R \frac{1}{2} r^2 \frac{\partial u_z}{\partial r} dr = \pi R^2 v_{slip} + \pi \int_0^R r^2 \dot{\gamma} dr. \tag{4.78}$$

Next, equation (4.61) is exploited to perform the following change of variables:

$$\begin{cases} r = -2\frac{L\tau}{|\Delta P|} \\ dr = -2\frac{L}{|\Delta P|} d\tau \\ r = R \Leftrightarrow \tau = \tau_w \\ r = 0 \Leftrightarrow \tau = 0. \end{cases} \tag{4.79}$$

Equation (4.78) becomes

$$Q = \pi R^2 v_{slip} + \frac{\pi R^3}{\tau_w^3} \int_0^{\tau_w} \tau^2 \dot{\gamma} d\tau. \tag{4.80}$$

Equation (4.80) may be rewritten in one of the following equivalent ways:

$$\frac{4Q}{\pi R^3} = \frac{4 v_{slip}}{R} + \frac{4}{\tau_w^3} \int_0^{\tau_w} \tau^2 \dot{\gamma} d\tau, \tag{4.81}$$

$$\frac{4Q}{\pi D^3} = \frac{v_{slip}}{D} + \frac{1}{2\tau_w^3} \int_0^{\tau_w} \tau^2 \dot{\gamma} d\tau. \tag{4.82}$$

Analogously equation (4.78) may be rewritten as

$$\frac{4Q}{\pi R^3} = \frac{4v_{slip}}{R} + \frac{4}{R^3}\int_0^R r^2\dot{\gamma}dr, \tag{4.83}$$

$$\frac{4Q}{\pi D^3} = \frac{v_{slip}}{D} + \frac{4}{D^3}\int_0^R r^2\dot{\gamma}dr. \tag{4.84}$$

In order to determine the value of v_{slip} at least two rheometers with different capillary die lengths L_1, L_2 and radii R_1, R_2 need to be employed.

Imagine to fix incoming flowrates Q_1 and Q_2 in such a way that

$$\tau_{w1} = \tau_{w2} \Rightarrow \frac{\Delta P_1}{2L_1}R_1 = \frac{\Delta P_2}{2L_2}R_2 = \tau_w. \tag{4.85}$$

Then,exploiting equation (4.83), it is possible to state that

$$\frac{4Q_1}{\pi R_1^3} = \frac{4v_{slip}}{R_1} + \frac{4}{R_1^3}\int_0^{R_1} r_1^2\left|\frac{\partial u_{z1}}{\partial r_1}\right| dr_1, \tag{4.86}$$

$$\frac{4Q_2}{\pi R_2^3} = \frac{4v_{slip}}{R_2} + \frac{4}{R_2^3}\int_0^{R_2} r_2^2\left|\frac{\partial u_{z2}}{\partial r_2}\right| dr_2, \tag{4.87}$$

with the assumption that v_{slip} depends only on $\tau_w = \tau_{w1} = \tau_{w2}$.

Since it holds that

$$\frac{\Delta P_1}{2L_1}R_1 = \frac{\Delta P_2}{2L_2}R_2, \tag{4.88}$$

it can be proven that

$$\frac{4}{R_1^3}\int_0^{R_1} r_1^2\left|\frac{\partial u_{z1}}{\partial r_1}\right| dr_1 = \frac{4}{R_2^3}\int_0^{R_2} r_2^2\left|\frac{\partial u_{z2}}{\partial r_2}\right| dr_2. \tag{4.89}$$

Taking into consideration equation (4.89), v_{slip} turns out to be equal to the slope of the straight line between points $(\frac{4}{R_1}, \frac{4Q_1}{\pi R_1^3}) = (\frac{4}{R_1}, \dot{\gamma}_{app1})$ and $(\frac{4}{R_2}, \frac{4Q_2}{\pi R_2^3}) = (\frac{4}{R_2}, \dot{\gamma}_{app2})$ in $\mathbb{R}^2$. Performing a linear regression between these two points it's possible to determine the slippage velocity v_{slip}. Finally the correct apparent shear rate may be computed as

$$\dot{\gamma}_{app,corr} = \frac{4Q_{corr}}{\pi R^3} = \frac{4(Q - \pi R^2 v_{slip})}{\pi R^3}. \tag{4.90}$$

Remark 4.11. *Supposing that the flow under examination is a power law fluid it's easy to prove equation (4.89):*

$$\begin{cases} \frac{4}{R_1^3}\int_0^{R_1} r_1^2\left|\frac{\partial u_{z1}}{\partial r_1}\right| dr_1 \\ \frac{4}{R_2^3}\int_0^{R_2} r_2^2\left|\frac{\partial u_{z2}}{\partial r_2}\right| dr_2. \end{cases} \tag{4.91}$$

Performing the following change of variables

$$\begin{cases} r = \frac{r_1}{R_1} = \frac{r_2}{R_2} \\ dr = \frac{dr_1}{R_1} = \frac{dr_2}{R_2}, \end{cases} \tag{4.92}$$

it holds that

$$\begin{cases} \frac{4}{R_1^3}\int_0^{R_1} r_1^2 \left|\frac{\partial u_{z1}}{\partial r_1}\right| dr_1 = 4\int_0^1 r^2 \left|\frac{\partial u_{z1}}{\partial r_1}\right| dr \\ \frac{4}{R_2^3}\int_0^{R_2} r_2^2 \left|\frac{\partial u_{z2}}{\partial r_2}\right| dr_2 = 4\int_0^1 r^2 \left|\frac{\partial u_{z2}}{\partial r_2}\right| dr. \end{cases} \tag{4.93}$$

Taking into consideration that v_{slip} depends only on $\tau_w = \tau_{w1} = \tau_{w2}$, for a power law fluid (4.3), it holds that

$$\begin{cases} u_{z1} = \left(\frac{|\Delta p_1|}{2\nu_0 L_1}\right)^{1/n} \frac{n}{n+1}(R_1^{\frac{n+1}{n}} - r_1^{\frac{n+1}{n}}) + v_{slip} \\ u_{z2} = \left(\frac{|\Delta p_2|}{2\nu_0 L_2}\right)^{1/n} \frac{n}{n+1}(R_2^{\frac{n+1}{n}} - r_2^{\frac{n+1}{n}}) + v_{slip}. \end{cases} \tag{4.94}$$

Then

$$\begin{cases} \frac{\partial u_{z1}}{\partial r_1} = -\left(\frac{|\Delta p_1|}{2\nu_0 L_1}\right)^{1/n} r_1^{\frac{1}{n}} \\ \frac{\partial u_{z2}}{\partial r_2} = -\left(\frac{|\Delta p_2|}{2\nu_0 L_2}\right)^{1/n} r_2^{\frac{1}{n}} \end{cases} \tag{4.95}$$

and, finally,

$$\begin{cases} \frac{\partial u_{z1}}{\partial r_1} = -\left(\frac{1}{\nu_0}\right)^{\frac{1}{n}} \left(\frac{|\Delta p_1|}{2L_1}R_1\right)^{1/n} r^{\frac{1}{n}} = -\left(\frac{1}{\nu_0\rho}\right)^{\frac{1}{n}} \tau_w^{1/n} r^{\frac{1}{n}} \\ \frac{\partial u_{z2}}{\partial r_2} = -\left(\frac{1}{\nu_0}\right)^{\frac{1}{n}} \left(\frac{|\Delta p_2|}{2L_2}R_2\right)^{1/n} r^{\frac{1}{n}} = -\left(\frac{1}{\nu_0\rho}\right)^{\frac{1}{n}} \tau_w^{1/n} r^{\frac{1}{n}}. \end{cases} \tag{4.96}$$

Inserting equation (4.96) into equation (4.93), the result is obtained:

$$\frac{4}{R_1^3}\int_0^{R_1} r_1^2 \left|\frac{\partial u_{z1}}{\partial r_1}\right| dr_1 = \frac{4}{R_2^3}\int_0^{R_2} r_2^2 \left|\frac{\partial u_{z2}}{\partial r_2}\right| dr_2. \tag{4.97}$$

Remark 4.12. *Condition (4.129) is quite difficult to impose. The usual practice prescribes to employ at least three capillary rheometers with different radii (R_1, R_2, R_3) to measure pressure drops on a wide range of apparent shear rates. Once fixed a value of τ_w, the apparent shear rates ($\dot\gamma_{app,1}$, $\dot\gamma_{app,2}$, $\dot\gamma_{app,3}$) corresponding to τ_w can be computed through linear interpolation, for each capillary rheometer. Performing a linear regression of $\dot\gamma_{app,1}$, $\dot\gamma_{app,2}$, $\dot\gamma_{app,3}$ against $\frac{4}{R_1}, \frac{4}{R_2}, \frac{4}{R_3}$, the slip velocity is computed as the slope of the regression line.*

Remark 4.13. *Once corrected the apparent shear rate (4.90), Rabinowitsch analysis can be performed substituting $\dot\gamma_{app}$ with $\dot\gamma_{app,corr}$ in equation (4.71).*

4.6 Rabinowitsch Analysis for Rectangular Duct

The goal of this paragraph is to present Rabinowitsch analysis for a rheometer with rectangular capillary cross section (slit die), with length L, height $2H$ and width W ($W >> 2H$). Let $(O,\vec{e}_x,\ \vec{e}_y,\vec{e}_z)$ be a cartesian reference frame such that the x axis is aligned with the longitudinal axis of the slit die.

Letting CV be a rectangular control volume, inside the slit die, with height $2y$ ($y < H$) and width W, with $CS = \partial CV$. The momentum equation may be written as

$$\int_{CV} \rho\vec{f}dV + \int_{CV} div(\boldsymbol{T})dV = \int_{CV} \rho\frac{\partial\vec{u}}{\partial t}dV + \int_{CV} \rho(\vec{u}\cdot\nabla)\vec{u}dV, \tag{4.98}$$

$$\int_{CV} \rho\vec{f}dV + \int_{CV} div(-P\boldsymbol{I} + 2\eta(\dot{\gamma})\boldsymbol{D})dV = \int_{CV} \rho\frac{\partial\vec{u}}{\partial t}dV + \int_{CV} \rho(\vec{u}\cdot\nabla)\vec{u}dV. \tag{4.99}$$

In the slit die, the velocity field $\vec{u}$ is assumed to satisfy the following conditions

$$\begin{cases} \vec{u} = u_x(y)\vec{e}_x \\ \dot{\gamma} = \left|\frac{\partial u_x}{\partial y}\right| = \begin{cases} -\frac{\partial u_x}{\partial y} \ \ if \ \ y > 0 \\ \frac{\partial u_x}{\partial y} \ \ if \ \ y < 0. \end{cases} \end{cases} \tag{4.100}$$

Neglecting boy forces, equation (4.99) becomes

$$\int_{CV} div(-P\boldsymbol{I})dV + \int_{CV} div(2\eta(\dot{\gamma})\boldsymbol{D})dV = 0, \tag{4.101}$$

since the flow is stationary and $(\vec{u}\cdot\nabla)\vec{u} = 0$.

Exploiting the fact that the velocity field is in the form of equation (4.100), it is possible to prove that

$$div(\eta(\dot{\gamma})(\nabla\vec{u})^T) = 0. \tag{4.102}$$

Finally, conservation of momentum equation reduces to

$$\int_{CS} P\vec{n}dS = \int_{CS} \eta(\dot{\gamma})\nabla\vec{u}\ \vec{n}dS. \tag{4.103}$$

Projecting the previous equation along the x direction, it holds

$$\int_{Sb1} PdS - \int_{Sb0} PdS = \int_{Sup} \eta(\dot{\gamma})\frac{\partial u_x}{\partial y}dS - \int_{Sdown} \eta(\dot{\gamma})\frac{\partial u_x}{\partial y}dS, \tag{4.104}$$

where $Sb0$ and $Sb1$ are, respectively, the upstream and the downstream faces of CV, Sup and $Sdown$ are the two lateral faces of CV orthogonal to $\vec{e}_y$. The contribution of surface forces on the lateral faces of CV orthogonal to $\vec{e}_z$ ($Sright$ and $Sleft$) is neglected since $W >> 2H \geq 2y$ (figure (4.2)).

It might be noticed that $\frac{\partial u_x}{\partial y}$,$\dot{\gamma}$ and $\eta(\dot{\gamma})$ are constant on $Sup \cup Sdown$. Assuming that P is constant on faces $Sb0$ and $Sb1$, equation (4.104) reduces to

$$P_1 2yW - P_0 2yW = 2WL\tau \tag{4.105}$$

where τ is $\left(-\eta(\dot{\gamma})\left|\frac{\partial u_z}{\partial r}\right|\right)$ evaluated on $Sup \cup Sdown$, P_1 and P_0 are pressures on faces $Sb1$ and $Sb0$, respectively.

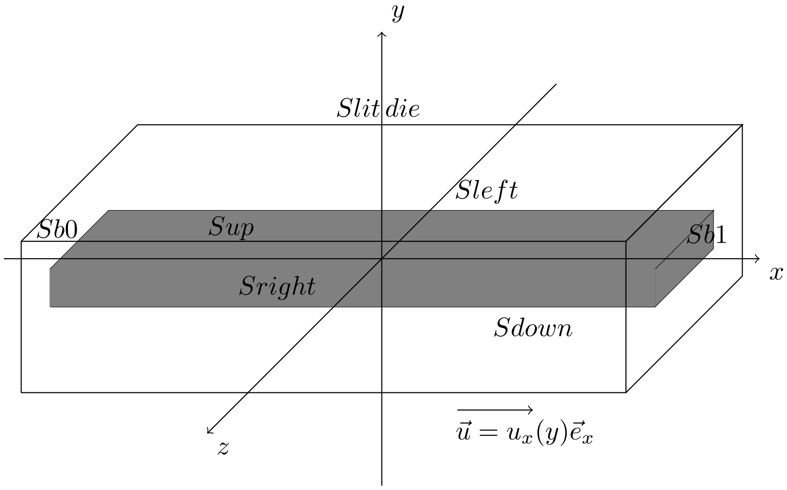


Figure 4.2: Rectangular control volume (grey) inside slit die

The previous equation may be arranged int the following form:

$$\tau = \frac{P_1 - P_0}{L} y = \frac{\Delta P}{L} y = -\frac{|\Delta P|}{L} y \tag{4.106}$$

Writing the previous equation for $y = H$, it holds that

$$\tau_w = \frac{P_1 - P_0}{L} H = \frac{\Delta P}{L} H, \tag{4.107}$$

where $\tau_w = \tau|_{|y|=H}$ is the wall shear stress.

Denote as Q the flux through the cross section of the rectangular duct:

$$Q = \int_{-H}^{H} \int_{0}^{W} u_x dz dy = W \int_{-H}^{H} u_x dy. \tag{4.108}$$

After performing an integration by parts, assuming $u_x(\pm H) = 0$, it holds that

$$Q = W \Big[y u_x \Big]_{-H}^{H} - W \int_{-H}^{H} y \frac{\partial u_x}{\partial y} dy = -W \int_{-H}^{H} y \frac{\partial u_x}{\partial y} dy, \tag{4.109}$$

$$Q = W \int_{-H}^{H} |y| \Big| \frac{\partial u_x}{\partial y} \Big| dy = W \int_{-H}^{H} |y| \dot{\gamma} dy = 2W \int_{0}^{H} y \dot{\gamma} dy. \tag{4.110}$$

Exploiting equation (4.107), it is possible to perform the following change of variables:

$$\begin{cases} y = -\frac{L\tau}{|\Delta P|} \\ dy = -\frac{L}{|\Delta P|} d\tau \\ y = H \Leftrightarrow \tau = \tau_w \\ y = 0 \Leftrightarrow \tau = 0. \end{cases} \tag{4.111}$$

Equation (4.110) becomes

$$Q = 2W \int_0^{\tau_w} \Big(-\frac{L}{|\Delta P|}\Big)^2 \tau \dot{\gamma} d\tau. \tag{4.112}$$

Combination of equation (4.107) and equation (4.112) results in

$$Q = \frac{2WH^2}{\tau_w^2} \int_0^{\tau_w} \tau \dot{\gamma} d\tau. \tag{4.113}$$

Define the apparent shear rate as

$$\dot{\gamma}_{app} = \frac{3Q}{2WH^2}. \tag{4.114}$$

Substituting equation (4.114) into equation (4.113), the following relation holds:

$$\dot{\gamma}_{app} = \frac{3Q}{2WH^2} = 3\frac{1}{\tau_w^2} \int_0^{\tau_w} \tau \dot{\gamma} d\tau. \tag{4.115}$$

The previous equation may be rewritten as

$$\tau_w^2 \dot{\gamma}_{app} = 3 \int_0^{\tau_w} \tau \dot{\gamma} d\tau \tag{4.116}$$

and, performing partial derivative $\frac{\partial}{\partial \tau_w}$,

$$\frac{\partial \dot{\gamma}_{app}}{\partial \tau_w} \tau_w^2 + 2\dot{\gamma}_{app} \tau_w = 3\tau_w \dot{\gamma}_w, \tag{4.117}$$

where $\dot{\gamma}_w$ is the shear rate at the wall.

After some manipulations the following relation is obtained:

$$\frac{1}{3}\frac{\partial \dot{\gamma}_{app}}{\partial \tau_w} \tau_w + \frac{2}{3}\dot{\gamma}_{app} = \dot{\gamma}_w. \tag{4.118}$$

Taking into account that $\tau_w = \frac{\partial \tau_w}{\partial log(\tau_w)}$ and $\dot{\gamma}_{app} = \frac{\partial \dot{\gamma}_{app}}{\partial log(\dot{\gamma}_{app})}$, the previous equation may be transformed as

$$\frac{1}{3}\frac{\partial log(\dot{\gamma}_{app})}{\partial log(\tau_w)} \dot{\gamma}_{app} + \frac{2}{3}\dot{\gamma}_{app} = \dot{\gamma}_w. \tag{4.119}$$

Defining $\frac{1}{n} = \frac{\partial log(\dot{\gamma}_{app})}{\partial log(\tau_w)}$, then

$$\frac{1}{3n}\dot{\gamma}_{app} + \frac{2}{3}\dot{\gamma}_{app} = \dot{\gamma}_w \Rightarrow \dot{\gamma_w} = \frac{2n+1}{3n}\dot{\gamma}_{app}. \tag{4.120}$$

Remark 4.14. *If the decimal logarithm is used in equation (4.119) still holds that*

$$\frac{1}{3}\frac{\partial Log(\dot{\gamma}_{app})}{\partial Log(\tau_w)} \dot{\gamma}_{app} + \frac{2}{3}\dot{\gamma}_{app} = \dot{\gamma}_w, \tag{4.121}$$

since $\tau_w = \frac{1}{log(10)} \frac{\partial \tau_w}{\partial Log(\tau_w)}$ *and* $\dot{\gamma}_{app} = \frac{1}{log(10)} \frac{\partial \dot{\gamma}_{app}}{\partial Log(\dot{\gamma}_{app})}$.

Defining $\frac{1}{n} = \frac{\partial Log(\dot{\gamma}_{app})}{\partial Log(\tau_w)} = \frac{\partial log(\dot{\gamma}_{app})}{\partial log(\tau_w)}$, *it still holds that*

$$\dot{\gamma_w} = \frac{2n+1}{3n} \dot{\gamma}_{app}. \tag{4.122}$$

After computing $\dot{\gamma}_w$ as suggested by equation (4.120), it is possible to estimate the viscosity of the generalized Newtonian fluid under examination as

$$\eta \simeq \frac{\tau_w}{\dot{\gamma}_w}. \tag{4.123}$$

4.7 Mooney Analysis for Rectangular Duct

If it is not possible to assume $u_x(\pm H) = 0$ when integrating by parts equation (4.108), in order to perform Rabinowitsch analysis, the apparent shear rates $\dot{\gamma}_{app}$ (kinematic quantity imposed on the rheometer) must undergo Mooney correction procedure (4.134).

Assume that $u_x(\pm H) = v_{slip}$. v_{slip} (which is unknown) is assumed to be independent of x and z (i.e. $v_{slip} \in \mathbb{R}^+$).

Integration by parts of equation (4.108) leads to

$$Q = W\Big[yu_x\Big]_{-H}^{H} - W\int_{-H}^{H} y\frac{\partial u_x}{\partial y}dy = 2WHv_{slip} + 2W\int_0^H y\dot{\gamma}dy. \tag{4.124}$$

Next, equation (4.61) is taken into consideration to perform the following change of variables:

$$\begin{cases} y = -\frac{L\tau}{|\Delta P|} \\ dy = -\frac{L}{|\Delta P|}d\tau \\ y = H \Leftrightarrow \tau = \tau_w \\ y = 0 \Leftrightarrow \tau = 0. \end{cases} \tag{4.125}$$

Equation (4.124) becomes

$$Q = 2WHv_{slip} + \frac{2WH^2}{\tau_w^2}\int_0^{\tau_w} \tau^2\dot{\gamma}d\tau. \tag{4.126}$$

Equation (4.126) may be rewritten as

$$\frac{3Q}{2WH^2} = \frac{3v_{slip}}{H} + \frac{3}{\tau_w^2}\int_0^{\tau_w} \tau\dot{\gamma}d\tau. \tag{4.127}$$

Analogously equation (4.124) may be rewritten as

$$\frac{3Q}{2WH^2} = \frac{3v_{slip}}{H} + \frac{3}{H^2}\int_0^{H} y\dot{\gamma}dr. \tag{4.128}$$

In order to determine the value of v_{slip} at least two rheometers with different slit die lengths L_1, L_2, heights $2H_1$, $2H_2$ and widths W_1, W_2 need to be employed ($W_1 >> 2H_1$, $W_2 >> 2H_2$).

Imagine to fix incoming flowrates Q_1 and Q_2 in such a way that

$$\tau_{w1} = \tau_{w2} \Rightarrow \frac{\Delta P_1}{L_1} H_1 = \frac{\Delta P_2}{L_2} H_2 = \tau_w. \tag{4.129}$$

Then, exploiting equation (4.128), it is possible to state that

$$\frac{3Q_1}{2W_1H_1} = \frac{3v_{slip}}{H_1} + \frac{3}{H_1^2}\int_0^{H_1} y_1^2 \left|\frac{\partial u_{x1}}{\partial y_1}\right| dy_1, \tag{4.130}$$

$$\frac{3Q_2}{2W_2H_2} = \frac{3v_{slip}}{H_2} + \frac{3}{H_2^2}\int_0^{H_2} y_2^2 \left|\frac{\partial u_{x2}}{\partial y_2}\right| dy_2, \tag{4.131}$$

with the assumption that v_{slip} depends only on $\tau_w = \tau_{w1} = \tau_{w2}$.

Since it holds that

$$\frac{\Delta P_1}{L_1} H_1 = \frac{\Delta P_2}{L_2} H_2, \tag{4.132}$$

it can be proven that

$$\frac{3}{H_1^2}\int_0^{H_1} y_1^2 \left|\frac{\partial u_{x1}}{\partial y_1}\right| dy_1 = \frac{3}{H_2^2}\int_0^{H_2} y_2^2 \left|\frac{\partial u_{x2}}{\partial y_2}\right| dy_2. \tag{4.133}$$

Taking into consideration equation (4.133), v_{slip} turns out to be equal to the slope of the straight line between points $(\frac{3}{H_1}, \frac{3Q_1}{2W_1H_1}) = (\frac{3}{H_1}, \dot\gamma_{app1})$ and $(\frac{3}{H_2}, \frac{3Q_2}{2W_2H_2}) = (\frac{3}{H_2}, \dot\gamma_{app2})$ in $\mathbb{R}^2$. Performing a linear regression between these two points, it is possible to determine the slippage velocity v_{slip}. Finally the correct apparent shear rate can be computed as

$$\dot\gamma_{app,corr} = \frac{3Q_{corr}}{2WH^2} = \frac{3(Q - 2WHv_{slip})}{2WH^2}. \tag{4.134}$$

Remark 4.15. *Supposing that the flow under examination is a power law fluid (4.32), it is easy to prove equation (4.133).*

$$\begin{cases} \frac{3}{H_1^2}\int_0^{H_1} y_1 \left|\frac{\partial u_{x1}}{\partial y_1}\right| dy_1 \\ \frac{3}{H_2^2}\int_0^{H_2} y_2 \left|\frac{\partial u_{x2}}{\partial y_2}\right| dy_2. \end{cases} \tag{4.135}$$

Performing the following change of variables

$$\begin{cases} y = \frac{y_1}{H_1} = \frac{y_2}{H_2} \\ dy = \frac{dy_1}{H_1} = \frac{dy_2}{H_2}, \end{cases} \tag{4.136}$$

it holds that

$$\begin{cases} \frac{3}{H_1^2}\int_0^{H_1} y_1 \left|\frac{\partial u_{x1}}{\partial y_1}\right| dy_1 = 3\int_0^1 y \left|\frac{\partial u_{x1}}{\partial y_1}\right| dy \\ \frac{3}{H_2^2}\int_0^{H_2} y_2 \left|\frac{\partial u_{x2}}{\partial y_2}\right| dy_2 = 3\int_0^1 y \left|\frac{\partial u_{x2}}{\partial y_2}\right| dy \end{cases} \tag{4.137}$$

Since it has been assumed that v_{slip} depends only on $\tau_w = \tau_{w1} = \tau_{w2}$, for a power law fluid, it follows that

$$\begin{cases} u_{x1} = \left(\frac{|\Delta p_1|}{\nu_0 L_1}\right)^{1/n} \frac{n}{n+1}(H_1^{\frac{n+1}{n}} - |y_1|^{\frac{n+1}{n}}) + v_{slip} \\ u_{x2} = \left(\frac{|\Delta p_2|}{\nu_0 L_2}\right)^{1/n} \frac{n}{n+1}(H_2^{\frac{n+1}{n}} - |y_2|^{\frac{n+1}{n}}) + v_{slip}. \end{cases} \tag{4.138}$$

Then,

$$\begin{cases} \left|\frac{\partial u_{x1}}{\partial y_1}\right| = \left(\frac{|\Delta p_1|}{\nu_0 L_1}\right)^{1/n} |y_1|^{\frac{1}{n}} \\ \left|\frac{\partial u_{x2}}{\partial y_2}\right| = \left(\frac{|\Delta p_2|}{\nu_0 L_2}\right)^{1/n} |y_2|^{\frac{1}{n}} \end{cases} \tag{4.139}$$

and, finally,

$$\begin{cases} \left|\frac{\partial u_{x1}}{\partial y_1}\right| = \left(\frac{1}{\nu_0}\right)^{\frac{1}{n}} \left(\frac{|\Delta p_1|}{L_1} H_1\right)^{1/n} |y|^{\frac{1}{n}} = \left(\frac{1}{\nu_0 \rho}\right)^{\frac{1}{n}} \tau_w^{1/n} |y|^{\frac{1}{n}} \\ \left|\frac{\partial u_{x2}}{\partial y_2}\right| = \left(\frac{1}{\nu_0}\right)^{\frac{1}{n}} \left(\frac{|\Delta p_2|}{L_2} H_2\right)^{1/n} |y|^{\frac{1}{n}} = \left(\frac{1}{\nu_0 \rho}\right)^{\frac{1}{n}} \tau_w^{1/n} |y|^{\frac{1}{n}}. \end{cases} \tag{4.140}$$

Inserting equation (4.140) into equation (4.137), the result is obtained:

$$\frac{3}{H_1^2} \int_0^{H_1} y_1^2 \left|\frac{\partial u_{x1}}{\partial y_1}\right| dy_1 = \frac{3}{H_2^2} \int_0^{H_2} y_2^2 \left|\frac{\partial u_{x2}}{\partial y_2}\right| dy_2. \tag{4.141}$$

Remark 4.16. *In practice, condition (4.129) is quite difficult to impose. The usual procedure to preform Mooney correction prescribes to employ at least three capillary rheometers with different slit die heights ($2H_1$, $2H_2$, $2H_3$) to measure pressure drops on a wide range of apparent shear rates. Once fixed a value of τ_w, the apparent shear rates ($\dot\gamma_{app,1}$, $\dot\gamma_{app,2}$, $\dot\gamma_{app,3}$) corresponding to τ_w can be computed through linear interpolation, for each capillary rheometer. Performing a linear regression of $\dot\gamma_{app,1}$, $\dot\gamma_{app,2}$, $\dot\gamma_{app,3}$ against $\frac{3}{H_1}, \frac{3}{H_2}, \frac{3}{H_3}$, the slip velocity is computed as the slope of the regression line.*

Remark 4.17. *Once corrected the apparent shear rate (4.134), Rabinowitsch analysis can be performed substituting $\dot\gamma_{app}$ with $\dot\gamma_{app,corr}$ in equation (4.117).*

4.8 Bagley Correction

Capillary rheometers impose the velocity of the piston which push the fluid into the capillary and measure the difference of pressure (ΔP_{tot}) between the tank and the outlet of the capillary. The following expression for ΔP_{tot} is assumed :

$$\Delta P_{tot} = \Delta P_c + \Delta P_d, \tag{4.142}$$

where ΔP_c is the pressure drop concentrated at the inlet of the capillary (independent of the length of the capillary), while ΔP_d represents the pressure drop distributed along the capillary (directly proportional to the length of the capillary). In order to estimate correctly the wall shear stress through formulas (4.62) and (4.107), the pressure drop distributed along the capillary must be known (i.e. $\Delta P = \Delta P_d$). The Bagley correction prescribes to measure, at each fixed piston velocity, the pressure loss P_{tot} with capillaries of different length. Performing a linear regression of the measured pressure drops at fixed piston velocity and different capillary lengths, it is possible to estimate the term ΔP_c and, then, compute

$$\Delta P = \Delta P_d = \Delta P_{tot} - \Delta P_c. \tag{4.143}$$

For what concerns rheometers with circular die cross section, if the ratio between the length and the diameter of the die is such that

$$\frac{L}{D} > 30, \tag{4.144}$$

the Bagley correction may be not applied to experimental data, since concentrated pressure drop is negligible with respect to distributed pressure drop:

$$\Delta P_{tot} \simeq \Delta P_d. \tag{4.145}$$

On the other hand, if $L/D < 30$, then the concentrated pressure drop is not negligible and Bagley correction becomes necessary.

Chapter 5

Implementation of the Viscosity Models

OpenFOAM (Open Field Operation And Manipulation) is a C++ implemented *open source* software for the solution of fluid dynamics problems. OpenFOAM includes a wide set of ready-to-use solvers, utilities, tools and applications suitable for the modelization and solution of a wide variety of CFD (Computational Fluid Dynamics) problems. Many of the OpenFOAM provided solvers are equipped with appropriate tutorials in order to get the user familiar with those applications. Moreover, OpenFOAM works also as a powerful objected oriented framework where to build customized solvers, application and libraries for user-specific CFD problems. The object-oriented C++ implementation of OpenFOAM ensures extensive modularity and massive code re-use. Within this research a custom library including rheological models described in chapter 2 has been employed.

5.1 Finite Volume Method

The Finite Volume Method (FVM) is a numerical method for the approximation of Partial Differential Equations (PDEs). Through FVM the continuous differential problem is discretized into an algebraic system of equations. Once constructed, the linear system of equations is solved through direct or, more likely, iterative methods. As a first step in discretizing the differential problem, the geometrical domain on which the continuous problem has been formulated, is subdivided into a collection of non overlapping discrete polyhedral elements (meshing). The mesh or computational grid consists of elements called cells or finite volumes (hence Finite Volume Method).

As pointed out in chapter 3, the motion of an incompressible generalized Newtonian fluid is described, from a thermo-mechanical point of view, by the following set of equations:

$$\begin{cases} \rho\frac{\partial\vec{u}}{\partial t} + \rho(\vec{u}\cdot\nabla)\vec{u} = -\nabla P + div(\eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T)) + \rho\vec{f} \\ div(\vec{u}) = 0 \\ \rho\frac{\partial(C_pT)}{\partial t} + \rho\vec{u}\cdot(C_p\nabla T) = \rho r + div(k\nabla T) + \eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T):\nabla\vec{u} \\ +\,\text{boundary conditions} \\ +\,\text{initial conditions} \end{cases} \tag{5.1}$$

or, equivalently,

$$\begin{cases} \rho\frac{\partial\vec{u}}{\partial t} + \rho div(\vec{u}\otimes\vec{u}) = -\nabla P + div(\eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T)) + \rho\vec{f} \\ div(\vec{u}) = 0 \\ \rho\frac{\partial(C_pT)}{\partial t} + \rho div(\vec{u}C_pT) = \rho r + div(k\nabla T) + \eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T):\nabla\vec{u} \\ +\,\text{boundary conditions} \\ +\,\text{initial conditions,} \end{cases} \tag{5.2}$$

where $\eta(\dot{\gamma},T)$ is the viscosity of the generalized Newtonian fluid, ρ is its density (hereby assumed constant: $\rho \in \mathbb{R}^+$), while k and C_p are, respectively, the thermal conductivity and the specific heat capacity of the fluid. Since only stationary flows will be numerically simulated, the focus should be on the following system of equations:

$$\begin{cases} \rho div(\vec{u}\otimes\vec{u}) = -\nabla P + div(\eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T)) + \rho\vec{f} \\ div(\vec{u}) = 0 \\ \rho div(\vec{u}C_pT) = \rho r + div(k\nabla T) + \eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T):\nabla\vec{u} \\ +\,\text{boundary conditions.} \end{cases} \tag{5.3}$$

Let C be a cell of the computational grid, the second step in discretizing the original differential problem is integrating equations in (5.3) on the finite volume C:

$$\begin{cases} \int_C \rho div(\vec{u}\otimes\vec{u})dV = -\int_C \nabla P dV + \int_C div(\eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T))dV + \int_C \rho\vec{f}dV \\ \int_C div(\vec{u})dV = 0 \\ \int_C \rho div(\vec{u}C_pT)dV = \int_C \rho r dV + \int_C div(k\nabla T)dV + \\ \qquad\qquad + \int_C \eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T):\nabla\vec{u}dV. \end{cases} \tag{5.4}$$

Making use of the divergence theorem, the previous system of equations may be rewritten as

$$\begin{cases}\displaystyle\int_{\partial C}\rho(\vec{u}\cdot\vec{n})\vec{u}dS=-\int_{C}\nabla PdV+\int_{\partial C}\eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T)\vec{n}dS+\int_{C}\rho\vec{f}dV\\ \displaystyle\int_{\partial C}\vec{u}\cdot\vec{n}dS=0\\ \displaystyle\int_{\partial C}\rho C_pT(\vec{u}\cdot\vec{n})dS=\int_{C}\rho rdV+\int_{\partial C}k(\nabla T)\cdot\vec{n}dS+\\ \displaystyle\qquad\qquad+\int_{C}\eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T):\nabla\vec{u}dV,\end{cases}\tag{5.5}$$

where $\vec{n}$ is the outward pointing normal vector defined on ∂C.

Integrals over the boundary of C may be decomposed into the sum of the integrals on the faces f that compose ∂C.

$$\begin{cases}\displaystyle\sum_{f\in\partial C}\int_{f}\rho(\vec{u}\cdot\vec{n})\vec{u}dS=-\int_{C}\nabla PdV+\sum_{f\in\partial C}\int_{f}\eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T)\vec{n}dS+\\ \displaystyle\qquad\qquad\qquad\qquad+\int_{C}\rho\vec{f}dV\\ \displaystyle\sum_{f\in\partial C}\int_{f}\vec{u}\cdot\vec{n}dS=0\\ \displaystyle\sum_{f\in\partial C}\int_{f}\rho C_pT(\vec{u}\cdot\vec{n})dS=\int_{C}\rho rdV+\sum_{f\in\partial C}\int_{f}k(\nabla T)\cdot\vec{n}dS+\\ \displaystyle\qquad\qquad+\int_{C}\eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T):\nabla\vec{u}dV,\end{cases}\tag{5.6}$$

Adopting a one-point Gaussian quadrature formula both for volume and face integrals, the previous system of equations may be transformed in

$$\begin{cases}\displaystyle\sum_{f\in\partial C}(\rho\vec{u}\cdot\vec{n})_f\vec{u}_f|f|=-(\nabla P)_C|C|+\sum_{f\in\partial C}(\eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T)\vec{n})_f|f|+\\ \displaystyle\qquad\qquad\qquad\qquad+(\rho\vec{f})_C|C|\\ \displaystyle\sum_{f\in\partial C}(\vec{u}\cdot\vec{n})_f|f|=0\\ \displaystyle\sum_{f\in\partial C}(\rho C_pT\vec{u}\cdot\vec{n})_f|f|=(\rho r)_C|C|+\sum_{f\in\partial C}(k(\nabla T)\cdot\vec{n})_f|f|+\\ \displaystyle\qquad\qquad+(\eta(\dot{\gamma},T)(\nabla\vec{u}+\nabla\vec{u}^T):\nabla\vec{u})_C|C|,\end{cases}\tag{5.7}$$

where Ψ_C and Ψ_f represent, respectively, the value of the field Ψ at the element centroid and at face centroid, while $|C|$ and $|f|$ are, respectively, the volume of the cell C and the area of the face f.

The first two equations in system (5.7) represent the semi-discretized form of the Navier-Stokes system (linear momentum equation + incompressibility constraint), while the third equation in system (5.7) corresponds to the semi-dicretized version of the energy balance equation.

The semi-discretized momentum equation contains both velocity and pressure but also temperature, since the viscosity of the fluid η is assumed to depend on T. Also the semi-discretized energy balance equation links temperature and velocity, due to the presence of the viscous heating term

$$(\eta(\dot{\gamma}, T)(\nabla\vec{u} + \nabla\vec{u}^T) : \nabla\vec{u})_C|C|. \tag{5.8}$$

In order to face the coupling between pressure-velocity (semi-discretized Navier-Stokes system) and temperature (semi-discretized energy balance equation), equations in system (5.7) are handled in an iterative fashion: at each iteration the Navier-Stokes system is solved for $\vec{u}$ and P, then the energy balance equation is solved for T.

The semi-discretized energy balance equation in system (5.7) could be arranged in the following way:

$$\sum_{f\in\partial C} C_{p,f}(\rho\vec{u}\cdot\vec{n})_f T_f|f| + \sum_{f\in\partial C} k_f((\nabla T)\cdot\vec{n})_f|f| =$$

$$= \rho_C r_C|C| + (\eta(\dot{\gamma}, T)(\nabla\vec{u} + \nabla\vec{u}^T) : \nabla\vec{u})_C|C|. \tag{5.9}$$

Assuming that, at some iteration n, the velocity field $\vec{u}^{(n)}$ is known (and hence the shear rate $\dot{\gamma}^{(n)}$), the last term in equation (5.9) will be treated as a source term:

$$\sum_{f\in\partial C} C_{p,f}(\rho\vec{u}^{(n)}\cdot\vec{n})_f T_f|f| + \sum_{f\in\partial C} k_f((\nabla T)\cdot\vec{n})_f|f| =$$

$$= \rho_C r_C|C| + (\eta(\dot{\gamma}^{(n)}, T)(\nabla\vec{u}^{(n)} + (\nabla\vec{u}^{(n)})^T) : \nabla\vec{u}^{(n)})_C|C|. \tag{5.10}$$

In principle, both the specific heat capacity C_p and the thermal conductivity k may depend on temperature, other than viscosity η which does depend on temperature. The linearized temperature equation (5.11) reads

$$\sum_{f\in\partial C} C_{p,f}^{(n-1)}(\rho\vec{u}^{(n)}\cdot\vec{n})_f T_f|f| + \sum_{f\in\partial C} k_f^{(n-1)}((\nabla T)\cdot\vec{n})_f|f| =$$

$$= \rho_C r_C|C| + (\eta(\dot{\gamma}^{(n)}, T^{(n-1)})(\nabla\vec{u}^{(n)} + (\nabla\vec{u}^{(n)})^T) : \nabla\vec{u}^{(n)})_C|C|, \tag{5.11}$$

where $T^{(n-1)}$ denotes temperature at previous iteration, while $C_{p,f}^{(n-1)}$ and $k_f^{(n-1)}$ denote the specific heat capacity and the thermal conductivity (at centroid of face f) computed using $T^{(n-1)}$. Equation (5.11) should be further discretized approximating both $((\nabla T)\cdot\vec{n})_f$ and T_f employing the values of T at cells centroids. With this procedure, (5.11) could be transformed into a linear equation of the following form, being T_C and T_F the (unknown) values of temperature at centroid of cell C and at the centroid of its neighbouring cells:

$$a_C^T T_C + \sum_{F\in NB(C)} a_F^T T_F = b_C, \tag{5.12}$$

where $NB(C)$ denotes the set of elements sharing a face f with element C. Collecting an equation of the form (5.12) for each element C of the computational grid, the following linear system could be assembled:

$$[\boldsymbol{A}][T] = [b], \tag{5.13}$$

where $[T]$ is the vector storing the (unknown) centroid values of temperature, $[b]$ is the vector containing the discretization of the source terms in energy balance equation over each element and $[\boldsymbol{A}]$ is a square matrix collecting diagonal coefficient a_C^T and off-diagonal coefficients a_F^T for each cell of the computational grid.

The solution of the semi-discretized Navier-Stokes system is a little more involved:

$$\begin{cases} \displaystyle\sum_{f\in\partial C} (\rho\vec{u}\cdot\vec{n})_f \vec{u}_f |f| = -(\nabla P)_C |C| + \sum_{f\in\partial C} (\eta(\dot{\gamma}, T)(\nabla\vec{u} + \nabla\vec{u}^T)\vec{n})_f |f| + \\ \hfill +(\rho\vec{f})_C |C| \\ \displaystyle\sum_{f\in\partial C} (\vec{u}\cdot\vec{n})_f |f| = 0. \end{cases} \tag{5.14}$$

Let $\vec{u}^{(n-1)}$ and $T^{(n-1)}$ be the velocity and temperature fields at the previous iteration, then system (5.14) is transformed into

$$\begin{cases} \displaystyle\sum_{f\in\partial C} (\rho\vec{u}\cdot\vec{n})_f \vec{u}_f |f| = -(\nabla P)_C |C| + \sum_{f\in\partial C} (\eta(\dot{\gamma}^{(n-1)}, T^{(n-1)})(\nabla\vec{u} + \nabla\vec{u}^T)\vec{n})_f |f| \\ \hfill +(\rho\vec{f})_C |C| \\ \displaystyle\sum_{f\in\partial C} (\vec{u}\cdot\vec{n})_f |f| = 0. \end{cases} \tag{5.15}$$

The non linearity of the advection term in the semi-discretized momentum equation is canonically overcome by modifying system (5.15) in the following way:

$$\begin{cases} \displaystyle\sum_{f\in\partial C} (\rho\underline{\vec{u}}\cdot\vec{n})_f \vec{u}_f |f| = -(\nabla P)_C |C| + \sum_{f\in\partial C} (\eta(\dot{\gamma}^{(n-1)}, T^{(n-1)})(\nabla\vec{u} + \nabla\vec{u}^T)\vec{n})_f |f| + \\ \hfill +(\rho\vec{f})_C |C| \\ \displaystyle\sum_{f\in\partial C} (\vec{u}\cdot\vec{n})_f |f| = 0, \end{cases} \tag{5.16}$$

with $\underline{\vec{u}}$ being a vector field sufficiently “close” to the unknown velocity field $\vec{u}$: if an iterative procedure is employed for the solution of system (5.16), $\underline{\vec{u}}$ could be the velocity field at the previous iteration, which should be sufficiently “close” to the velocity field to be computed at the current iteration.

Usually, the viscous stress term in the momentum equation is split into a diffusion term (the one containing the velocity gradient) and an explicit term (the one containing the transposed velocity gradient):

$$\begin{cases} \displaystyle\sum_{f\in\partial C}(\rho\underline{\vec{u}}\cdot\vec{n})_f\vec{u}_f|f| - \sum_{f\in\partial C}(\eta(\dot{\gamma}^{(n-1)},T^{(n-1)})(\nabla\vec{u})\vec{n})_f|f| = -(\nabla P)_C|C| + \\ \qquad\qquad\qquad\qquad + \displaystyle\sum_{f\in\partial C}(\eta(\dot{\gamma}^{(n-1)},T^{(n-1)})(\nabla(\underline{\vec{u}})^T)\vec{n})_f|f| + (\rho\vec{f})_C|C| \\ \displaystyle\sum_{f\in\partial C}(\vec{u}\cdot\vec{n})_f|f| = 0. \end{cases} \tag{5.17}$$

The main difficulty in solving system (5.17) is the presence of the term $(\nabla P)_C|C|$ in the momentum equation and the absence of an equation to be solved for pressure. This issue is overcome by combining equations of system (5.15) in order to obtain a discrete equation to be solved for velocity and a discrete equation to be solved for pressure. Finally the fully discretized velocity equation and the fully discretized pressure equation are solved sequentially and iteratively.

Remark 5.1. *The first equation in system (5.17) is fully discretized (treating the pressure gradient term as a source term) in a similar way to the energy balance equation (5.9)-(5.12):*

$$a_C\vec{u}_C + \sum_{F\in NB(C)} a_F\vec{u}_F = -(\nabla P)_C|C| + \vec{b}_C, \tag{5.18}$$

where $NB(C)$ is the set of elements sharing a face with element C and $\vec{b}_C$ collects all the source contributions in momentum equation (5.17), with the exception of the pressure gradient term. Defining the following vector operator

$$\mathcal{H}_C(\vec{u}) = \vec{b}_C - \sum_{F\in NB(C)} a_F\vec{u}_F, \tag{5.19}$$

the previous equation can be transformed in

$$\vec{u}_C = -(\nabla P)_C\frac{|C|}{a_C} + \frac{1}{a_C}\mathcal{H}_C(\vec{u}). \tag{5.20}$$

In order to get the values $\vec{u}_f$ of velocity at the faces of the cell C, vectors

$$\vec{u}_C = -(\nabla P)_C\frac{|C|}{a_C}+\frac{1}{a_C}\mathcal{H}_C(\vec{u})\,,\ \vec{u}_F = -(\nabla P)_F\frac{|F|}{a_F}+\frac{1}{a_F}\mathcal{H}_F(\vec{u})\ \text{with}\ F\in NB(C)$$

should be interpolated. The substitution of interpolated $\vec{u}_f$ into the second equation of system (5.17) gives an equation for pressure.

Many algorithms have been designed ([20, 21]) in order to preform this task: one of the most famous is the SIMPLE method (Semi Implicit Method for Pressure Linked Equations) [25, 26, 27], which gave raise to a huge family of algorithms [28] (SIMPLEC, SIMPLEM, SIMPLEST, SIMPLEX etc).

In conclusion, at each iteration n, the momentum and pressure discretized equations are solved through an iterative algorithm, such as, for example, the SIMPLE algorithm, then the discretized energy balance equation is solved for temperature.

5.2 Rheology of Filled Compounds in OpenFOAM

In OpenFOAM, in order to solve the Navier-Stokes system coupled with energy balance equation, as explained in the previous section, a temperature and filler volume fraction dependent viscosity model has to be introduced, together with density, thermal conductivity and specific heat capacity expressions.

5.2.1 Thermal Properties

The specific heat capacity of a substance or compound represents the amount of energy, per unit mass, required to increase its temperature by $1\,K$. As a first approximation the specific heat capacity could be thought as a constant, assuming that the specific amount of energy requested to increase temperature by $1\,K$ is the same whichever is the temperature of the compound. As a matter of fact, the specific heat capacity is dependent on temperature implying that the energy per unit mass needed to increase temperature by $1\,K$ does depend on temperature. Let T_0 be a reference temperature and let $C_p(T)$ be the specific heat capacity of a compound. A simple model for $C_p(T)$ in the neighbourhood of T_0 is

$$C_p(T) = C_{p0} + C_{p1}(T - T_0). \tag{5.21}$$

If the slope C_{p1} of expression (5.21) is positive, the higher is the temperature of the compound the higher is the amount of energy, per unit mass, required to increase the temperature of the compound by $1\,K$. If, on the other hand, the coefficient C_{p1} is negative, the specific amount of energy requested to increase temperature by $1\,K$ increases when temperature decreases. Experimental data suggest that the specific heat capacity of a filled compound depends, other than on the temperature of the compound, on the filler volume fraction ϕ (i.e. on the concentration of the filler). In this research it is assumed that the specific heat capacity depends on temperature and filler volume fraction through the following relation:

$$C_p(T, \phi) = C_{p0} + C_{p1}(T - T_0) + C_{p2}\phi. \tag{5.22}$$

The positiveness of coefficient C_{p2} implies that the addition of filler increases the specific amount of energy required to increase the temperature of the compound by $1\,K$. On the other hand, if C_{p2} is negative the dispersion of filler in the compound reduces the specific amount of energy needed to increase the temperature of the compound by $1\,K$.

The thermal conductivity k of a substance or compound corresponds to the proportional coefficient between the instantaneous heat flux per unit surface and the opposite of the temperature gradient. Let $P_1 \in \Omega_t$ and $P_2 \in \Omega_t$ be two material points of a compound in the actual configuration Ω_t at time t, close one to the other. Let T_1 and T_2 be the temperatures of the two points. The higher is the thermal conductivity of the compound, the higher is the instant heat flux between P_1 and P_2 (flowing from the hotter point to the colder point). Thermal conductivity is always positive since heat flows from higher temperature points to lower temperature points.

The thermal diffusivity δ of a substance or compound is defined as the ratio between the thermal conductivity of the material and the product of its density with its specific heat capacity:

$$\delta = \frac{k}{\rho C_p}. \tag{5.23}$$

Manipulating equation (5.23), thermal conductivity may be computed as

$$k = \delta \rho C_p. \tag{5.24}$$

Concerning filled compounds, thermal conductivity is assumed to be a function of temperature and filler volume fraction:

$$k(T, \phi) = \delta(T, \phi)\rho(\phi)C_p(T, \phi). \tag{5.25}$$

In equation (5.25), the density of the filled compound ρ is a function of the filler volume fraction, indeed

$$\rho(\phi) = (1 - \phi)\rho_{unfilled} + \phi\rho_{filler}, \tag{5.26}$$

with $\rho_{unfilled}$ being the density of the unfilled compound and ρ_{filler} being the density of the filler. As pointed out in equation (5.22), C_p is a function of both temperature and filler volume fraction and, in principle, also thermal diffusivity δ should be a function of temperature and filler concentration. In this research, for semplicity, the dependence of thermal diffusivity on temperature is dropped, leading to the following expression for thermal conductivity:

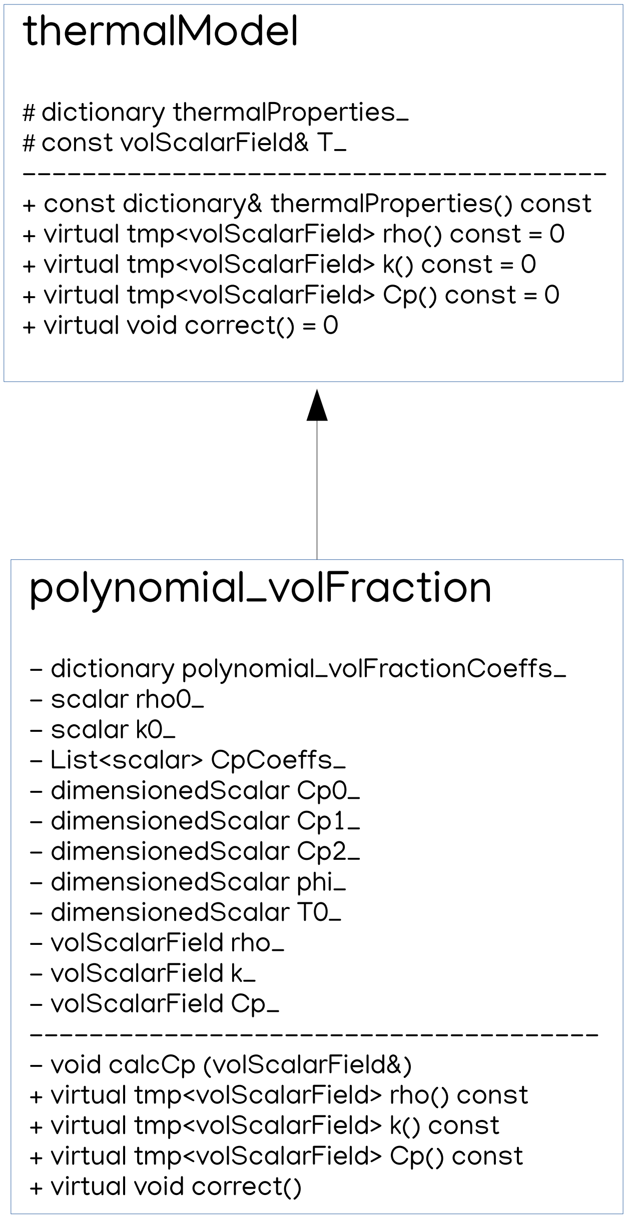
$$k(T, \phi) = \delta(\phi)\rho(\phi)C_p(T, \phi). \tag{5.27}$$

Assuming that the flow experienced by a generalized Newtonian fluid is approximately isothermal, at a nominal temperature $T_{nominal}$, equation (5.27) may be further simplified in

$$k(\phi) = \delta(\phi)\rho(\phi)C_p(T_{nominal}, \phi). \tag{5.28}$$

5.2.2 Thermal Properties Implementation

In OpenFOAM, the information about density, thermal conductivity and specific heat capacity is stored in a class called `thermalModel`. Once that the filler volume fraction has been fixed, both density and thermal conductivity are constants, while specific heat capacity still depends on temperature. Since, beside equation (5.22), other models can be chosen for specific heat capacity, the class `thermalModel` has been made an abstract class: each specific heat capacity expression will be modelled through a derived class of the abstract class `thermalModel`. The next page presents a scheme of the implementation of the `thermalModel` class. The focus should be on the protected data member `thermalProperties_` which stores the thermal properties supplied by the user. Also a constant reference `T_` to the temperature field is stored. Some pure virtual functions are defined (`thermalModel.rho()`,`thermalModel.k()`, `thermalModel.Cp()`): they should be in charge of returning the density scalar field, the thermal conductivity scalar field and the specific heat capacity scalar field. The pure virtual function `thermalModel.correct()`, in particular, should compute the specific heat capacity on the computational domain, according to temperature field.



```
thermalModel

# dictionary thermalProperties_
# const volScalarField& T_
--------------------------------------
+ const dictionary& thermalProperties() const
+ virtual tmp<volScalarField> rho() const = 0
+ virtual tmp<volScalarField> k() const = 0
+ virtual tmp<volScalarField> Cp() const = 0
+ virtual void correct() = 0
```

```
polynomial_volFraction

- dictionary polynomial_volFractionCoeffs_
- scalar rho0_
- scalar k0_
- List<scalar> CpCoeffs_
- dimensionedScalar Cp0_
- dimensionedScalar Cp1_
- dimensionedScalar Cp2_
- dimensionedScalar phi_
- dimensionedScalar T0_
- volScalarField rho_
- volScalarField k_
- volScalarField Cp_
-------------------------------------
- void calcCp (volScalarField&)
+ virtual tmp<volScalarField> rho() const
+ virtual tmp<volScalarField> k() const
+ virtual tmp<volScalarField> Cp() const
+ virtual void correct()
```

The class polynomial_volFraction is a derived class of the abstract base class thermalModel. The private data members of polynomial_volFraction class store :

- a dictionary polynomial_volFractionCoeffs_ which stores coefficients supplied by the user for the current thermal model;
- a scalar rho0_ needed to initialise the (constant) density scalar field ρ;
- a scalar k0_ required for the initialisation of the (constant) thermal conductivity scalar field;
- a list of scalars storing the values of the coefficients of equation (5.22);
- scalars Cp0_,Cp1_,Cp2_ representing the coefficients of equation (5.22);
- the filler volume fraction ϕ, named phi_;
- a scalar T0_ corresponding to the reference temperature T_0 in equation (5.22);
- three scalar fields rho_,k_,Cp_ representing density, thermal conductivity and specific heat capacity fields defined on the computational domain.

Overriding functions polynomial_volFraction.rho(), polynomial_volFraction.k(), polynomial_volFraction.Cp() return respectively the density field ρ, the thermal conductivity field k and the specific heat capacity field C_p. The polynomial_volFraction.correct() public method is in charge of computing the specific heat capacity according to the temperature field, through the auxiliary member function polynomial_volFraction.calcCp(), which implements equation (5.22).

Fields rho_ and k_ will be constant during the lifetime of each variable of class polynomial_volFraction, since no class method changes them, while the specific heat capacity field Cp_ will be corrected according to the temperature field.

5.2.3 Viscosity Model Implementation

As pointed out in chapter 2, filled polymer compounds will be hereby described by the generalized Newtonian fluid model: the kinematic viscosity of the filled compound is then modelled as (see equations (2.80))

$$\nu(\dot{\gamma}, \phi, T) = \frac{\eta(\dot{\gamma}, \phi, T)}{\rho(\phi)} = H(T)\frac{V(\dot{\gamma}, \phi)}{\rho(\phi)} = H(T)\frac{V(\dot{\gamma}, f(\phi))}{\rho(\phi)}, \tag{5.29}$$

with $f(\phi)$ being the correction factor taking into consideration the concentration of filler, chosen among a relative viscosity expression proposed in literature (see chapter 2). The factor $H(T)$ represents the dependence of viscosity of the filled compound on temperature. For the temperature correction factor $H(T)$ an Arrhenius expression has been chosen:

$$H(T) = exp\Big(\alpha\Big(\frac{1}{T} - \frac{1}{T_\alpha}\Big)\Big), \tag{5.30}$$

where α is a positive coefficient and T_α is a reference (absolute) temperature. If the absolute temperature T equals the reference temperature T_α, the temperature correction factor $H(T)$ is 1, which makes it irrelevant in expression (5.29). If the absolute temperature T is higher than T_α, $H(T)$ is smaller than unity. On the contrary, if the absolute temperature T is smaller than T_α the temperature correction factor is bigger than unity: as should be expected, the increase in temperature reduces viscosity.

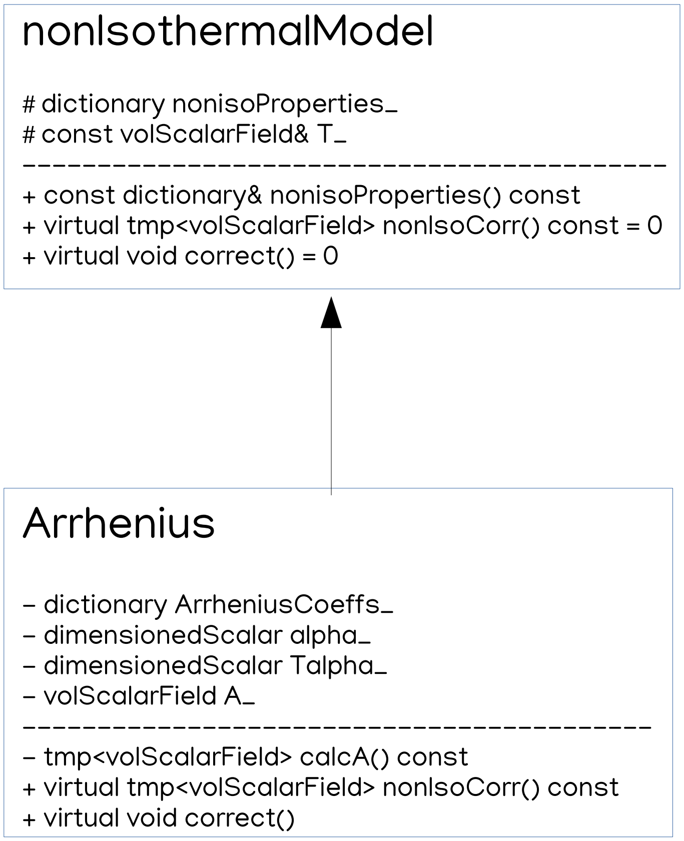
In OpenFOAM the dependence of viscosity on temperature has been implemented through the class `nonIsothermalModel`. Since, other than Arrhenius model (5.30), many expressions can be chosen for $H(T)$, the class `nonIsothermalModel` has been designed to be an abstract class.

The class `nonIsothermalModel` stores a dictionary named `nonisoProperties_`, collecting the information provided by the user on the temperature correction factor $H(T)$, together with a constant reference to the temperature field `T_`. Pure virtual member functions `nonIsothermalModel.nonIsoCorr()` and `nonIsothermalModel.correct()` are defined: the former should return the temperature correction factor $H(T)$ as a scalar field defined on the computational domain while the latter should compute $H(T)$ according to the temperature field.

The Arrhenius expression (5.30) for temperature correction factor is implemented in class `Arrhenius`. The class `Arrhenius` inherits from the class `nonIsothermalModel` and declares as private data members :

- a dictionary named `ArrheniusCoeffs_`, collecting information about coefficients of the Arrhenius expression (5.30);
- a scalar `alpha_` representing the coefficient α in equation (5.30);
- a scalar `Talpha_` corresponding to the reference temperature T_α in equation (5.30);
- a scalar field `A_` defined on the computational domain which represents the temperature correction factor $H(T)$.

The member function `Arrhenius.nonIsoCorr()` overrides `nonIsothermalModel.nonIsoCorr()`, returning the temperature correction factor field `A_`. The overriding function `Arrhenius.correct()` computes the scalar field `A_` through the auxiliary private method `Arrhenius.calcA()`, which implements formula (5.30).


nonIsothermalModel
# dictionary nonisoProperties_
# const volScalarField& T_

+ const dictionary& nonisoProperties() const
+ virtual tmp<volScalarField> nonIsoCorr() const = 0
+ virtual void correct() = 0
Arrhenius
– dictionary ArrheniusCoeffs_
– dimensionedScalar alpha_
– dimensionedScalar Talpha_
– volScalarField A_

– tmp<volScalarField> calcA() const
+ virtual tmp<volScalarField> nonIsoCorr() const
+ virtual void correct()

The volume fraction correction factor $f(\phi)$ in equation (5.29) is implemented through the abstract class `volumeFractionModel`. The implementation of this class is quite similar to the implementation of the `nonIsothermalModel` class.

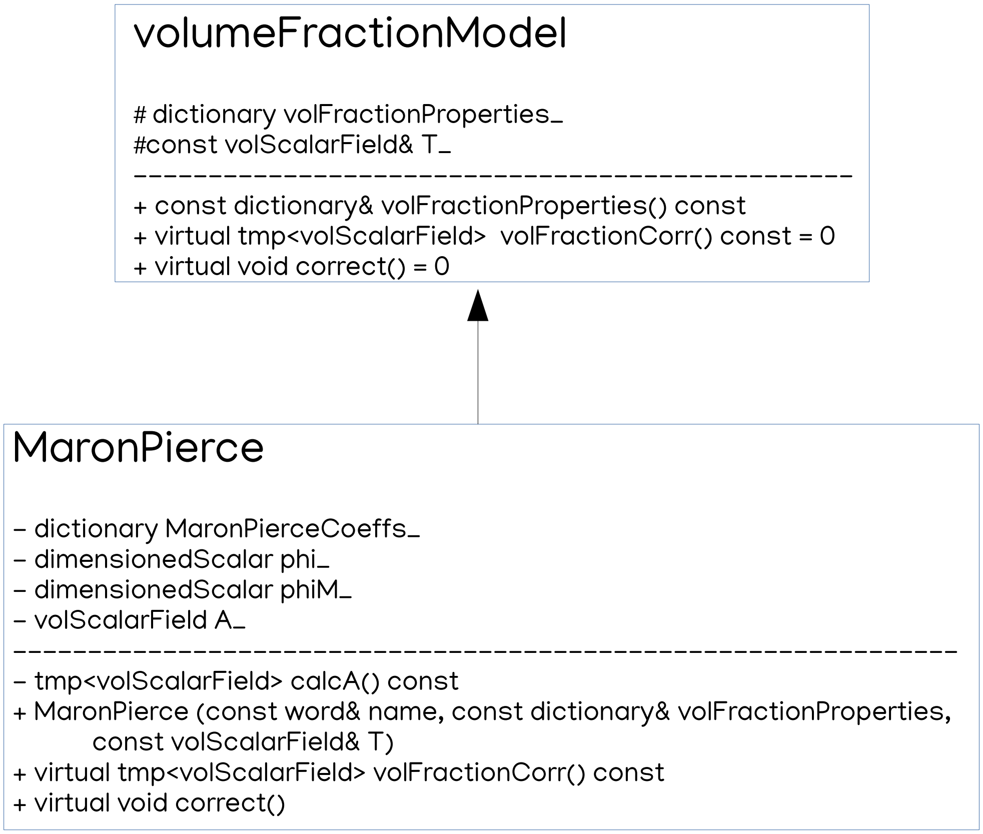
Among the protected data members stored in class `volumeFractionModel` there is a constant reference to the temperature field `T_` together with a dictionary named `volFractionProperties_` which contains all the information supplied by the user about the volume fraction correction $f(\phi)$.

Pure virtual functions `volumeFractionModel.correct()` and `volumeFractionModel.volFractionCorr()`, which should, respectively, compute and return the filler volume fraction correction factor $f(\phi)$, make `volumeFractionModel` an abstract class and must be overridden in any derived class.

Class `MaronPierce` is an example of derived class from base class `volumeFractionModel`. The class `MaronPierce` implements the Maron-Pierce volume fraction correction factor:

$$f(\phi) = \left(1 - \frac{\phi}{\phi_M}\right)^{-2}. \tag{5.31}$$

The class `MaronPierce` stores a dictionary collecting the coefficients of the Maron Pierce model, the scalar `phi_` representing the filler volume fraction ϕ and the scalar `phiM_` corresponding to the maximum packing filler volume fraction ϕ_M. The volume fraction correction field $f(\phi)$ is stored as a (constant) scalar field defined on the computational domain and denoted as `A_`. Pure virtual function `volumeFractionModel.volFractionCorr()` is overridden by `MaronPierce.volFractionCorr()`, which returns the volume fraction correction factor field `A_`. Analogously, the member function `MaronPierce.correct()` overrides the base class function `volumeFractionModel.correct()`, computing the filler volume fraction correction field `A_`. Since, once fixed the filler volume fraction, the correction factor $f(\phi)$ does not change, the class method `MaronPierce.correct()`, actually, has an empty function body.



```
volumeFractionModel

# dictionary volFractionProperties_
#const volScalarField& T_
------------------------------------------------
+ const dictionary& volFractionProperties() const
+ virtual tmp<volScalarField>  volFractionCorr() const = 0
+ virtual void correct() = 0
```

```
MaronPierce

- dictionary MaronPierceCoeffs_
- dimensionedScalar phi_
- dimensionedScalar phiM_
- volScalarField A_
---------------------------------------------------------------
- tmp<volScalarField> calcA() const
+ MaronPierce (const word& name, const dictionary& volFractionProperties,
        const volScalarField& T)
+ virtual tmp<volScalarField> volFractionCorr() const
+ virtual void correct()
```

The constructor of class MaronPierce (see listing 5.1), once invoked the constructor of the base class volumeFractionModel, initialises the dictionary MaronPierceCoeffs_, stores the values of filler volume fraction and maximum packing filler volume fraction in phi_ and phiM_ and initializes the filler volume fraction correction field A_ with expression (5.31). Once initialised, the field A_ will be constant during the lifetime of the any variable of class MaronPierce, since no class method changes it.

```
Foam::volumeFractionModels::MaronPierce::MaronPierce
(
    const word& name,
    const dictionary& volFractionProperties,
    const volScalarField& T
)
:
    volumeFractionModel(name, volFractionProperties, T),
    MaronPierceCoeffs_(
    volFractionProperties.optionalSubDict(typeName + "Coeffs")
    ),
    phi_("phi", dimless, MaronPierceCoeffs_),
    phiM_("phiM", dimless, MaronPierceCoeffs_),
    A_
    (
        IOobject
        (
            name,
            T_.time().timeName(),
            T_.db(),
            IOobject::NO_READ,
            IOobject::NO_WRITE
        ),
        T_.mesh(),
        Foam::pow((1-phi_/phiM_),-2)
    )
{}
```

Listing 5.1: MaronPierce.C - Constructor

Beside the Maron-Pierce expression (5.31), some other models have been implemented in OpenFOAM:

- Chong filler volume fraction model:

$$f(\phi) = \Big[1 + \frac{3}{4}\Big(\frac{\frac{\phi}{\phi_M}}{1 - \frac{\phi}{\phi_M}}\Big)\Big]^2; \tag{5.32}$$

- Mooney filler volume fraction model:

$$f(\phi) = exp\Big(\frac{2.5\phi}{1 - \frac{\phi}{\phi_M}}\Big); \tag{5.33}$$

- Pal filler volume fraction model:

$$f(\phi) = \Big(1 - \Big(1 + \Big(\frac{1-\phi_M}{\phi_M^2}\Big)\phi\Big)\phi\Big)^{-2.5}; \tag{5.34}$$

- Einstein filler volume fraction model:

$$f(\phi) = 1 + \frac{5}{2}\phi. \tag{5.35}$$

Each of the aforementioned filler volume fraction models is implemented as a derived class of the base class `volumeFractionModel`. All the previously enumerated volume fraction models are implemented in the same way of the Maron-Pierce expression (5.31), with the exception of the initialisation of the scalar field `A_`. The constructors of the classes corresponding to equations (6.35-6.38) are reported below.

```
Foam::volumeFractionModels::Chong::Chong
(
    const word& name,
    const dictionary& volFractionProperties,
    const volScalarField& T
)
:
    volumeFractionModel(name, volFractionProperties, T),
    ChongCoeffs_(
    volFractionProperties.optionalSubDict(typeName + "Coeffs")
    ),
    phi_("phi", dimless, ChongCoeffs_),
    phiM_("phiM", dimless, ChongCoeffs_),
    A_
    (
        IOobject
        (
            name,
            T_.time().timeName(),
            T_.db(),
            IOobject::NO_READ,
            IOobject::NO_WRITE
        ),
        T_.mesh(),
        Foam::pow(1+0.75*(phi_/phiM_)/(1-phi_/phiM_),2)
    )
{}
```

Listing 5.2: Chong.C - Constructor

```
Foam::volumeFractionModels::Mooney::Mooney
(
    const word& name,
    const dictionary& volFractionProperties,
    const volScalarField& T
)
:
    volumeFractionModel(name, volFractionProperties, T),
    MooneyCoeffs_(
    volFractionProperties.optionalSubDict(typeName + "Coeffs")
    ),
    phi_("phi", dimless, MooneyCoeffs_),
    phiM_("phiM", dimless, MooneyCoeffs_),
    A_
    (
        IOobject
        (
            name,
            T_.time().timeName(),
            T_.db(),
            IOobject::NO_READ,
            IOobject::NO_WRITE
        ),
        T_.mesh(),
        Foam::exp(2.5*phi_/(1-phi_/phiM_))
    )
{}
```

Listing 5.3: Mooney.C - Constructor

```
Foam::volumeFractionModels::Pal::Pal
(
    const word& name,
    const dictionary& volFractionProperties,
    const volScalarField& T
)
:
    volumeFractionModel(name, volFractionProperties, T),
    PalCoeffs_(
    volFractionProperties.optionalSubDict(typeName + "Coeffs")
    ),
    phi_("phi", dimless, PalCoeffs_),
    phiM_("phiM", dimless, PalCoeffs_),
    A_
    (
        IOobject
        (
            name,
            T_.time().timeName(),
            T_.db(),
            IOobject::NO_READ,
            IOobject::NO_WRITE
        ),
        T_.mesh(),
        Foam::pow(1-
                  (1+
                    ((1-phiM_)/Foam::pow(phiM_,2))
                  *phi_)
                *phi_,
              -2.5)
    )
{}
```

Listing 5.4: Pal.C - Constructor

```
Foam::volumeFractionModels::Einstein::Einstein
(
    const word& name,
    const dictionary& volFractionProperties,
    const volScalarField& T
)
:
    volumeFractionModel(name, volFractionProperties, T),
    EinsteinCoeffs_(
    volFractionProperties.optionalSubDict(typeName + "Coeffs")
    ),
    phi_("phi", dimless, EinsteinCoeffs_),
    A_
    (
        IOobject
        (
            name,
            T_.time().timeName(),
            T_.db(),
            IOobject::NO_READ,
            IOobject::NO_WRITE
        ),
        T_.mesh(),
        1+2.5*phi_
    )
{}
```

Listing 5.5: Einstein.C - Constructor

Once chosen expressions for the temperature correction factor $H(T)$ and the filler volume fraction correction factor $f(\phi)$, the complete viscosity model should be implemented :

$$\nu(\dot{\gamma}, \phi, T) = H(T)\frac{V(\dot{\gamma}, f(\phi))}{\rho(\phi)}. \tag{5.36}$$

Complete viscosity models (taking into consideration both the temperature and filler volume fraction correction factors) are implemented in OpenFOAM through the abstract class `nonIsothermalViscosityModel` where two pointers are stored as protected members: a pointer to a `nonIsothermalModel` object (`nonIsothermalModelPtr_`) which is in charge of including in the viscosity model the temperature correction factor $H(T)$ and a pointer to a `volumeFractionModel` object (`volumeFRactionModelPtr_`)which, on the other hand, takes into account the filler volume fraction correction factor $f(\phi)$. Moreover, the class `nonIsothermalViscosityModel` stores a dictionary (`viscosityProperties_`) collecting information about the viscosity model supplied by the user, a constant reference to the velocity vector field `U_`, a constant reference to the velocity flux ($\Phi = \vec{u} \cdot \vec{n}$) scalar field `phi_`, defined on the faces of the elements of the computational domain, and a constant reference to the temperature scalar field `T_`. Pure virtual functions `nonIsothermalViscosityModel.nu()`, `nonIsothermalViscosityModel.correctIso()` and `nonIsothermalViscosityModel.correct()` make the class `nonIsothermalViscosityModel` an abstract class: classes inheriting from this abstract base class must override these methods.

Among the public function members of class `nonIsothermalViscosityModel`, the method `nonIsothermalViscosityModel.strainRate()` is in charge of computing the strain rate scalar field $\dot{\gamma}$, employing the formula

$$\dot{\gamma} = \sqrt{2}||\boldsymbol{D}||_2 = \sqrt{2}\sqrt{\boldsymbol{D} : \boldsymbol{D}}. \tag{5.37}$$

An example of derived class of the abstract base class `nonIsothermalViscosityModel` is the class `BirdCarreau` which implements the Bird-Carreau viscosity model:

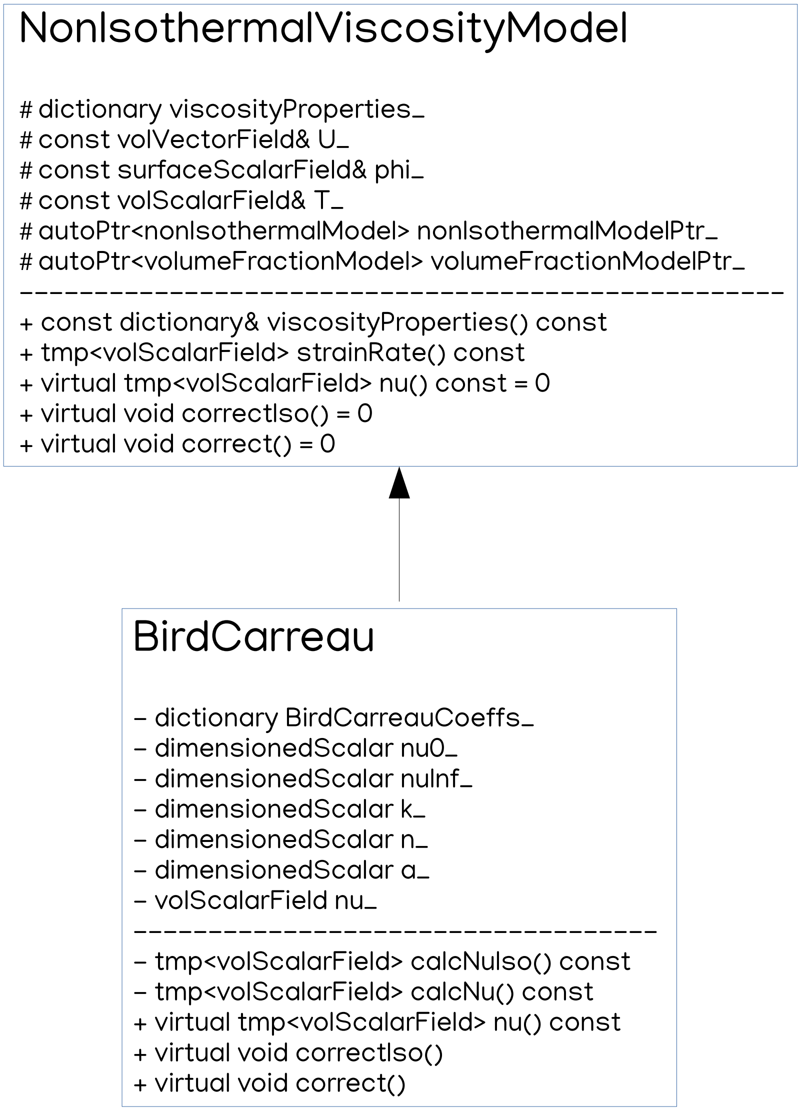
$$\begin{aligned}\nu(\dot{\gamma}, \phi, T) &= H(T)\frac{\eta_0}{\rho(\phi)} f(\phi)\Big[1 + \Big(f(\phi)\lambda\dot{\gamma}\Big)^2\Big]^{(n-1)/2} = \\ &= H(T)\nu_0 f(\phi)\Big[1 + \Big(f(\phi)\lambda\dot{\gamma}\Big)^2\Big]^{(n-1)/2},\end{aligned} \tag{5.38}$$

with $n < 1$, since filled polymer compounds are likely to show shear thinning behaviour (i.e. viscosity decreases when the shear rate increases). Actually, in order to ensure viscosity to be bounded from below and not to give rise to instability issues in numerical simulations, the viscosity expression implemented in class `BirdCarreau` is

$$\nu(\dot{\gamma}, \phi, T) = H(T)f(\phi)\Big(\nu_\infty + (\nu_0 - \nu_\infty)\Big[1 + \Big(f(\phi)\lambda\dot{\gamma}\Big)^a\Big]^{(n-1)/a}\Big), \tag{5.39}$$

with $\nu_\infty > 0$ and $a > 0$. In this way, it holds that

$$\lim_{\dot{\gamma}\to\infty} \nu(\dot{\gamma}, \phi, T) = H(T)f(\phi)\nu_\infty > 0. \tag{5.40}$$



```
NonIsothermalViscosityModel

# dictionary viscosityProperties_
# const volVectorField& U_
# const surfaceScalarField& phi_
# const volScalarField& T_
# autoPtr<nonIsothermalModel> nonIsothermalModelPtr_
# autoPtr<volumeFractionModel> volumeFractionModelPtr_
--------------------------------------------------
+ const dictionary& viscosityProperties() const
+ tmp<volScalarField> strainRate() const
+ virtual tmp<volScalarField> nu() const = 0
+ virtual void correctIso() = 0
+ virtual void correct() = 0
```

```
BirdCarreau

- dictionary BirdCarreauCoeffs_
- dimensionedScalar nu0_
- dimensionedScalar nuInf_
- dimensionedScalar k_
- dimensionedScalar n_
- dimensionedScalar a_
- volScalarField nu_
-----------------------------------
- tmp<volScalarField> calcNuIso() const
- tmp<volScalarField> calcNu() const
+ virtual tmp<volScalarField> nu() const
+ virtual void correctIso()
+ virtual void correct()
```

User supplied information about the Bird-Carreau model is stored in the dictionary BirdCarreauCoeffs_ . Moreover, the coefficients ν_0, ν_∞, λ, n and a of expression (5.39) are stored, respectively, as nu0_, nuInf_, k_, n_ and a_. Among the private data member, the class BirdCarreau collects also the kinematic viscosity scalar field nu_, defined on the computational domain. The overriding method BirdCarreau.correct() updates the temperature correction factor field (through nonIsothermalModelPtr_->correct()) and the filler volume fraction correction factor field (through volumeFractionModelPtr_->correct()) and computes the kinematic viscosity field according to equation (5.39) through the auxiliary private method BirdCarreau.calcNu(). The overriding method BirdCarreau.correctIso() updates the filler volume fraction correction factor field (through volumeFractionModelPtr_->correct()) and computes, through the auxiliary private method BirdCarreau.calcNuIso(), the kinematic viscosity field, neglecting the temperature correction factor $H(T)$ in equation (5.39). The overriding member function BirdCarreau.nu() returns the kinematic viscosity scalar field.

The implementation of the auxiliary functions BirdCarreau.calcNu() and BirdCarreau.calcNuIso() is displayed in the source file BirdCarreau.C (listing 5.6).

```

Foam::tmp<Foam::volScalarField>
Foam::nonIsothermalViscosityModels::BirdCarreau::
calcNuIso() const
{
    return
    volumeFractionModelPtr_->volFractionCorr()*
      (nuInf_
    + (nu0_ - nuInf_)
     *pow(scalar(1) +
     pow(k_* volumeFractionModelPtr_->volFractionCorr()*
     strainRate(), a_),
     (n_ - 1.0)/a_));
}


Foam::tmp<Foam::volScalarField>
Foam::nonIsothermalViscosityModels::BirdCarreau::
calcNu() const
{
    return
        nonIsothermalModelPtr_->nonIsoCorr() *
        volumeFractionModelPtr_->volFractionCorr()*
        (nuInf_
      + (nu0_ - nuInf_)
       *pow(scalar(1) +
       pow(k_*volumeFractionModelPtr_->volFractionCorr()*
       strainRate(), a_),
       (n_ - 1.0)/a_));
}
```

Listing 5.6: BirdCarreau.C (partial)

The implementation of the power law viscosity model

$$\nu(\dot{\gamma},\phi,T) = H(T)\frac{\eta_0}{\rho(\phi)}f(\phi)\Big(f(\phi)\dot{\gamma}\lambda\Big)^{n-1} = H(T)\nu_0 f(\phi)\Big(f(\phi)\dot{\gamma}\lambda\Big)^{n-1}, \quad (5.41)$$

with $n < 1$, is quite similar to the implementation of the BirdCarreau model (5.38). Once again, in order to avoid unboundedness in viscosity and consequent instabilities in the numerical simulations, the implemented expression for viscosity is the following

$$\nu(\dot{\gamma},\phi,T) = max\Big\{\nu_{min}, min\Big\{\nu_{max}, H(T)\nu_0 f(\phi)\Big(max\big\{\epsilon, f(\phi)\dot{\gamma}\lambda\big\}\Big)^{n-1}\Big\}\Big\}, \quad (5.42)$$

with ν_{min} and ν_{max} being, respectively, the lower and the upper bound for kinematic viscosity, while $\epsilon > 0$ is a small quantity preventing the quantity raised to the $(n-1)$ to be infinitesimal. The power law viscosity model is implemented in OpenFOAM through the class `PowerLaw` (derived from the abstract base class `nonIsothermalViscosityModel`), whose main difference with respect to the class `BirdCarreau` is in the private member functions `PowerLaw.calcNu()` and `PowerLaw.calcNuIso()`. The implementation of both these memeber functions is displayed in listing 5.7.

```
Foam::tmp<Foam::volScalarField>
Foam::nonIsothermalViscosityModels::PowerLaw::
calcNuIso() const
{
    return max
    (
        nuMin_,
        min
        (
            nuMax_,
            volumeFractionModelPtr_->volFractionCorr()*
            K_ * pow
            (
                max
                (
                    lambda_ *
                    volumeFractionModelPtr_->volFractionCorr()*
                    strainRate(),
                    dimensionedScalar(dimless, small)
                ),
                n_.value() - scalar(1)
            )
        )
    );
}
```

```


Foam::tmp<Foam::volScalarField>
Foam::nonIsothermalViscosityModels::PowerLaw::calcNu() const
{
    return max
    (
        nuMin_,
        min
        (
            nuMax_,
            nonIsothermalModelPtr_->nonIsoCorr() *
            volumeFractionModelPtr_->volFractionCorr()*
            K_ * pow
            (
                max
                (
                    lambda_ *
                    volumeFractionModelPtr_->volFractionCorr()*
                    strainRate(),
                    dimensionedScalar(dimless, small)
                ),
                n_.value() - scalar(1)
            )
        )
    );
}
```

Listing 5.7: PowerLaw.C (partial)

Analogously, the Cross viscosity model

$$\nu(\dot{\gamma},\phi,T) = \frac{\eta_0}{\rho(\phi)}\frac{H(T)f(\phi)}{1+\Big(\lambda f(\phi)\dot{\gamma}\Big)^m} = \nu_0\frac{H(T)f(\phi)}{1+\Big(\lambda f(\phi)\dot{\gamma}\Big)^m}, \tag{5.43}$$

with $m > 0$, is implemented through the class Cross. Actually, in order to ensure boundedness and guarantee numerical stability, the implemented expression for viscosity is

$$\nu(\dot{\gamma},\phi,T) = H(T)f(\phi)\Big(\nu_\infty + \frac{\nu_0-\nu_\infty}{1+\Big(\lambda f(\phi)\dot{\gamma}\Big)^m}\Big), \tag{5.44}$$

with $\nu_\infty > 0$.

Equation (5.44) is such that

$$\lim_{\dot{\gamma}\to\infty}\nu(\dot{\gamma},\phi,T) = H(T)f(\phi)\nu_\infty > 0. \tag{5.45}$$

The implementation of the member functions Cross.calcNu() and Cross.calcNuIso() is reported in listing 5.8.

```
Foam::tmp<Foam::volScalarField>
Foam::nonIsothermalViscosityModels::Cross::calcNuIso() const
{
    return
    volumeFractionModelPtr_->volFractionCorr()*
      (nuInf_
    + (nu0_ - nuInf_)
     /(scalar(1)+
     pow(t_* volumeFractionModelPtr_->volFractionCorr()*
     strainRate(),m_)
     ));
}


Foam::tmp<Foam::volScalarField>
Foam::nonIsothermalViscosityModels::Cross::calcNu() const
{
    return
        nonIsothermalModelPtr_->nonIsoCorr() *
        volumeFractionModelPtr_->volFractionCorr()*
        (nuInf_
      + (nu0_ - nuInf_)
      / (scalar(1)+
      pow(t_*volumeFractionModelPtr_->volFractionCorr()*
      strainRate(), m_)));
}
```

Listing 5.8: Cross.C (partial)

Finally, the class nonIsothermalSinglePhaseTransportModel has been designed in order to group the information about viscosity of the filled compound together with its thermal properties (density, thermal conductivity and specific heat capacity).

The class nonIsothermalSinglePhaseTransportModel contains two pointers (nonIsothermalViscosityModelPtr_, thermalModelPtr_): a pointer to a nonIsothermalViscosityModel object (collecting information about viscosity model together with temperature correction factor and filler volume fraction correction factor) and a pointer to a thermalModel object (gathering information about density, thermal conductivity and specific heat capacity).

These pointers are exploited in the implementation of function members of class nonIsothermalSinglePhaseTransportModel:

- the method nonIsothermalSinglePhaseTransportModel.nu() calls the method nonIsothermalViscosityModel.nu() (returning the viscosity scalar field) on the nonIsothermalViscosityModel object pointed by nonIsothermalViscosityModelPtr_;

- the methods nonIsothermalSinglePhaseTransportModel.rho(), nonIsothermalSinglePhaseTransportModel.k(), nonIsothermalSinglePhaseTransportModel.Cp() invoke the corresponding functions of the thermalModel object pointed by thermalModelPtr_, returning, respectively, the (constant) density scalar field, the (constant) thermal conductivity scalar field and the specific heat capacity scalar field;

- the member function nonIsothermalSinglePhaseTransportModel.correctIso() does nothing more than calling the homonymous method on the nonIsothermalViscosityModel object pointed by nonIsothermalViscosityModelPtr_;
- the method nonIsothermalSinglePhaseTransportModel.correct() calls the homonymous methods both on the object pointed by nonIsothermalViscosityModelPtr_ and by thermalModelPtr_.

To conclude, a call to the method nonisothermalSinglePhaseTransportModel.correct() will

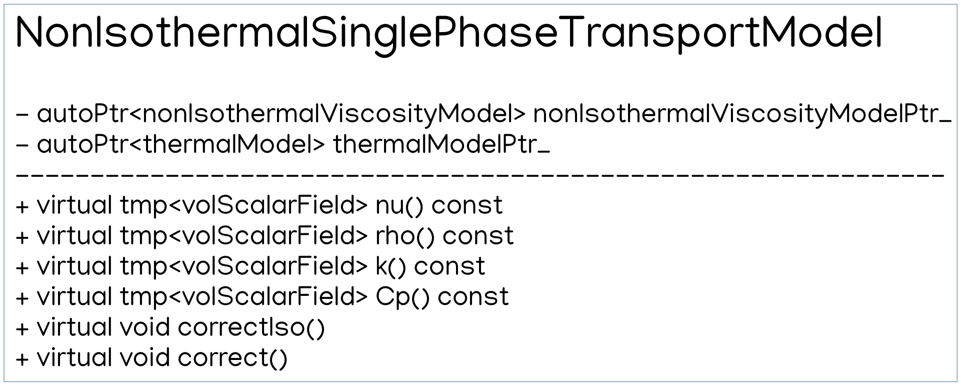
- compute/update the specific heat capacity through thermalModelPtr_ -> correct();
- compute/update the temperature correction factor $H(T)$ through the method nonIsothermalViscosityModelPtr_-> nonIsothermalModelPtr_ -> correct();
- compute/update the filler volume fraction correction factor through the method nonIsothermalViscosityModelPtr_ -> volumeFractionModelPtr_ -> correct().

Actually, the public method nonisothermalSinglePhaseTransportModel.correct() updates also the kinematic viscosity scalar field through nonIsothermalViscosityModelPtr_ -> correct(). On the other hand, calls to methods nonisothermalSinglePhaseTransportModel.rho(), nonisothermalSinglePhaseTransportModel.k(), nonisothermalSinglePhaseTransportModel.Cp() and nonisothermalSinglePhaseTransportModel.nu() return, respectively the density scalar field, the thermal conductivity scalar field, the specific heat capacity scalar field and the viscosity scalar field.

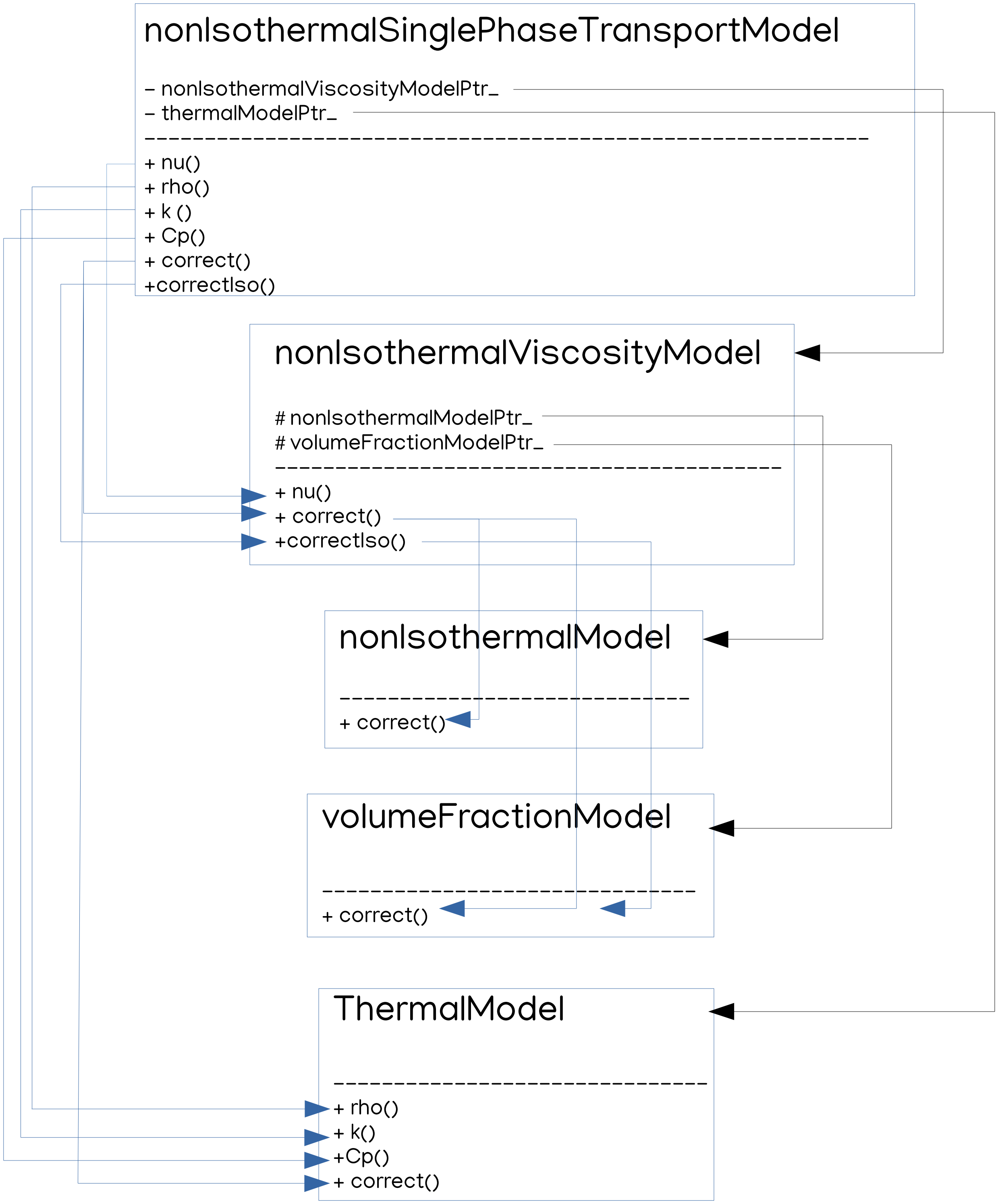
```
#include "nonIsothermalSinglePhaseTransportModel.H"
#include "nonIsothermalViscosityModel.H"
#include "thermalModel.H"


// * * * * * * * * * Member Functions  * * * * * * * * * * //

Foam::tmp<Foam::volScalarField>
Foam::nonIsothermalSinglePhaseTransportModel::nu() const
```

```
{
    return nonIsothermalViscosityModelPtr_->nu();
}


Foam::tmp<Foam::volScalarField>
Foam::nonIsothermalSinglePhaseTransportModel::rho() const
{
    return thermalModelPtr_->rho();
}


Foam::tmp<Foam::volScalarField>
Foam::nonIsothermalSinglePhaseTransportModel::k() const
{
    return thermalModelPtr_->k();
}


Foam::tmp<Foam::volScalarField>
Foam::nonIsothermalSinglePhaseTransportModel::Cp() const
{
    return thermalModelPtr_->Cp();
}


void Foam::nonIsothermalSinglePhaseTransportModel::
correctIso()
{
    nonIsothermalViscosityModelPtr_->correctIso();
}


void Foam::nonIsothermalSinglePhaseTransportModel::correct()
{
    nonIsothermalViscosityModelPtr_->correct();
    thermalModelPtr_->correct();
}


// ******************************************************* //
```

Listing 5.9: nonIsothermalSinglePhaseTransportModel.C (partial)



```
NonIsothermalSinglePhaseTransportModel

– autoPtr<nonIsothermalViscosityModel> nonIsothermalViscosityModelPtr_
– autoPtr<thermalModel> thermalModelPtr_
------------------------------------------------------------
+ virtual tmp<volScalarField> nu() const
+ virtual tmp<volScalarField> rho() const
+ virtual tmp<volScalarField> k() const
+ virtual tmp<volScalarField> Cp() const
+ virtual void correctIso()
+ virtual void correct()
```


nonIsothermalSinglePhaseTransportModel
– nonIsothermalViscosityModelPtr_
– thermalModelPtr_
+ nu()
+ rho()
+ k ()
+ Cp()
+ correct()
+correctIso()
nonIsothermalViscosityModel
# nonIsothermalModelPtr_
# volumeFractionModelPtr_
+ nu()
+ correct()
+correctIso()
nonIsothermalModel
+ correct()
volumeFractionModel
+ correct()
ThermalModel
+ rho()
+ k()
+Cp()
+ correct()

Chapter 6

Data Analysis

In order to test the OpenFOAM implementation of the rheological models for filled compounds presented in the previous chapters, some experimental data are needed. Pirelli Tyre S.p.A. supplied experimental data relative to a polymer compound filled with silica (SiO_2) at different levels of concentration. On one hand, these data have been analyzed in order to estimate the numerical value of the coefficients of some models chosen to describe the rheology of the filled compound. On the other hand, they have been employed to check if numerical simulations, involving the chosen rheological models, are able to predict pressure drops along the capillary die of a rheometer (i.e. to check if the numerically computed pressure drops match the experimental values). Because of confidentiality reasons, graphs in this chapter and in the next chapter are reported without numerical values on axes.

6.1 Viscosity Models

For what concerns the choice of a viscosity model and the estimation of its coefficients, experimental data have been collected employing a capillary rheometer, with a circular cross section capillary (capillary die) and a length over diameter ratio equal to $L/D = 30/2$. During a rheometer experiment, as explained in chapter 4, the filled compound, whose temperature is set equal to a fixed temperature $T_{nominal}$, is collected in a reservoir and pushed by a piston through the capillary die, whose walls are at temperature $T_{nominal}$. Also the temperature of the piston and the temperature of the walls of the reservoir are set equal to $T_{nominal}$.The experimental dataset collects measured pressure drops along the capillary die at different values of apparent shear rate (i.e. at different piston velocities v_{piston}):

$$\dot{\gamma}_{app,min} \leq \dot{\gamma}_{app} = \frac{4Q}{\pi R^3} = \frac{4(v_{piston}\pi R_{tank}^2)}{\pi R^3} \leq \dot{\gamma}_{app,max}, \tag{6.1}$$

where R is the radius of the capillary die and R_{tank} is the radius of the rheometer reservoir.

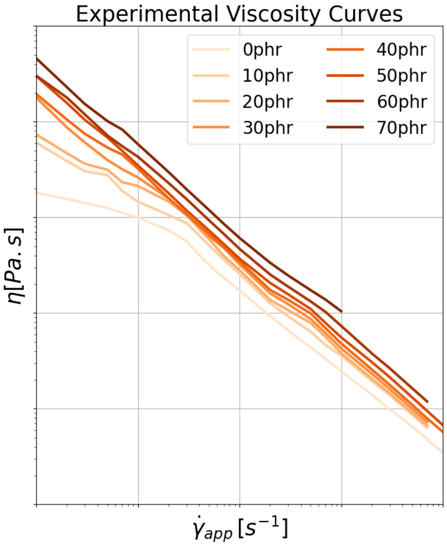


Figure 6.1: Experimental Viscosity Curves for $\phi_{0phr} \leq \phi \leq \phi_{70phr}$.

Different levels of concentration of filler (SiO_2) have been tested: from $0\,phr$ (neat of unfilled polymer) up to $70\,phr$ (highly filled polymer).

Since experimental data have been collected employing only one capillary rheometer neither the Mooney correction nor the Bagley correction have been performed because they require capillary rheometers with different capillary die lengths and radii.

The viscosity of the filled compound at a given apparent shear rate $\dot{\gamma}_{app}$ is estimated as

$$\eta = \frac{\tau_w}{\dot{\gamma}_{app}} = \frac{\Delta P}{2L}\frac{R}{\dot{\gamma}_{app}} \tag{6.2}$$

and no Rabinowitsch correction has been employed, since the ratio between the length and the diameter of the capillary die is $L/D = 30/2 < 30$ and (necessary) Bagley correction has not been applied to experimental data.

Cross Maron-Pierce Viscosity Model

Figure (6.1) displays the experimental viscosity curves at silica concentration ranging from $0\,phr$ to $70\,phr$. It's worth to notice that the filled compound, at each level of filler concentration, exhibits a shear thinning behaviour, since the value of viscosity decreases with the increase in apparent shear rates. Moreover for a fixed value of apparent shear rate, viscosity increases with the increase in concentration of silica. At high levels of concentration in silica (i.e. $\phi = \phi_{50phr}, \phi_{60phr}, \phi_{70phr}$) the experimental viscosity curves look like straight lines, denoting, in log-log scale, a power law behaviour. At low levels of filler concentration, the experimental viscosity exhibit a plateau for low apparent shear rates: for $\phi = \phi_{0phr}$, $\phi = \phi_{10phr}$, $\phi = \phi_{20phr}$, the filled compound shows a zero shear rate viscosity:

$$\lim_{\dot{\gamma}_{app} \to 0} \eta < +\infty. \tag{6.3}$$

Because of this consideration a Cross viscosity model has been chosen:

$$\eta(\dot{\gamma}, \phi, T) = \frac{H(T)\eta_0 f(\phi)}{1 + \Big(\lambda f(\phi)\dot{\gamma}\Big)^m}, \tag{6.4}$$

with an Arrhenius-like expression for $H(T)$:

$$H(T) = exp\Big(\alpha\Big(\frac{1}{T} - \frac{1}{T_\alpha}\Big)\Big), \tag{6.5}$$

where the absolute temperature T_α has been set equal to the temperature $T_{nominal}$ of the rheometer experiments. Moreover, for the filler volume fraction correction factor $f(\phi)$, the Maron Pierce model has been selected:

$$f(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}. \tag{6.6}$$

The expression for viscosity (6.4) becomes

$$\eta(\dot{\gamma}, \phi, T) = H(T)\frac{\eta_0}{\Big(1 - \frac{\phi}{\phi_M}\Big)^2}\frac{1}{1 + \Big(\frac{\lambda}{\left(1 - \frac{\phi}{\phi_M}\right)^2}\dot{\gamma}\Big)^m} \tag{6.7}$$

In order to estimate the numerical value of coefficients η_0 and λ and ϕ_M, the temperature correction factor $H(T)$ has been neglected, since the flow experienced by the filled compound in the rheometer could be approximately considered isothermal and

$$T \simeq T_{nominal} = T_\alpha \;\; \Rightarrow \;\; H(T) \simeq 1. \tag{6.8}$$

With the aim of estimating the values of η_0, λ and ϕ_M, it is assumed that, at each level of filler concentration, the viscosity of the filled compound has a Cross expression whose coefficients depend on filler volume fraction ϕ:

$$\eta(\dot{\gamma}, \phi) = \frac{\mathcal{N}_0(\phi)}{1 + \Big(\Lambda(\phi)\dot{\gamma}\Big)^{m(\phi)}}. \tag{6.9}$$

Values of $\mathcal{N}_0(\phi)$, $\Lambda(\phi)$ and $m(\phi)$ are computed, for each filler volume fraction, fitting the experimental viscosity curves through expression (6.9).

Since equation (6.7) does not allow exponent m to depend on ϕ, the value of m has been computed as

$$m = \bar{m} = \frac{1}{8}\sum_{\phi=\phi_{0phr}}^{\phi_{70phr}} m(\phi). \tag{6.10}$$

Comparing equations (6.7) and (6.9), it must be

$$\mathcal{N}_0(\phi) = \eta_0\Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}, \qquad\qquad \Lambda(\phi) = \lambda\Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}. \tag{6.11}$$

Remark 6.1. *Since, as pointed out in chapter 2, f(0)=1, then parameters η_0 and λ in equation (6.7) are estimated as*

$$\mathcal{N}_0(0) = \eta_0 f(0) = \eta_0, \qquad\qquad \Lambda(0) = \lambda f(0) = \lambda. \tag{6.12}$$

The last task to perform is the estimation of coefficient ϕ_M which represents the maximum packing filler volume fraction. In order to estimate the value of ϕ_M, the ratios $\mathcal{N}_0(\phi)/\eta_0$ are computed. Rearranging equation (6.11A), it holds that

$$\frac{\mathcal{N}_0(\phi)}{\eta_0} = f(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2} \quad\Rightarrow\quad \phi = \phi_M\Big(1 - \Big(\frac{\mathcal{N}_0(\phi)}{\eta_0}\Big)^{-\frac{1}{2}}\Big). \tag{6.13}$$

The maximum packing filler volume fraction is then estimated as the slope of the regression line ϕ *vs* $\Big(1 - \Big(\frac{\mathcal{N}_0(\phi)}{\eta_0}\Big)^{-\frac{1}{2}}\Big)$. Obviously, to ensure that the estimation of the maximum packing filler volume fraction is consistent with the viscosity model (6.7), it must hold $\phi_M > \phi_{70phr}$.

Figure (6.2) reports the experimental viscosity curves compared with theoretical viscosity curves computed employing equation (6.7). For what concerns the unfilled polymer, the experimental viscosity curve is well matched by equation (6.7). At low levels of concentration of silica, the agreement between experimental and theoretical values of viscosity is quite satisfactory, with the exception of low shear rates range. At higher levels of concentration of filler, the discrepancy between experimental and theoretical values of viscosity becomes more evident both at low and high shear rates.

Power Law Maron-Pierce Viscosity Model

The second viscosity model taken into consideration is the power law model:

$$\eta(\dot\gamma, \phi, T) = H(T)\eta_0 f(\phi)\Big(f(\phi)\dot\gamma\lambda\Big)^{n-1}, \tag{6.14}$$

with

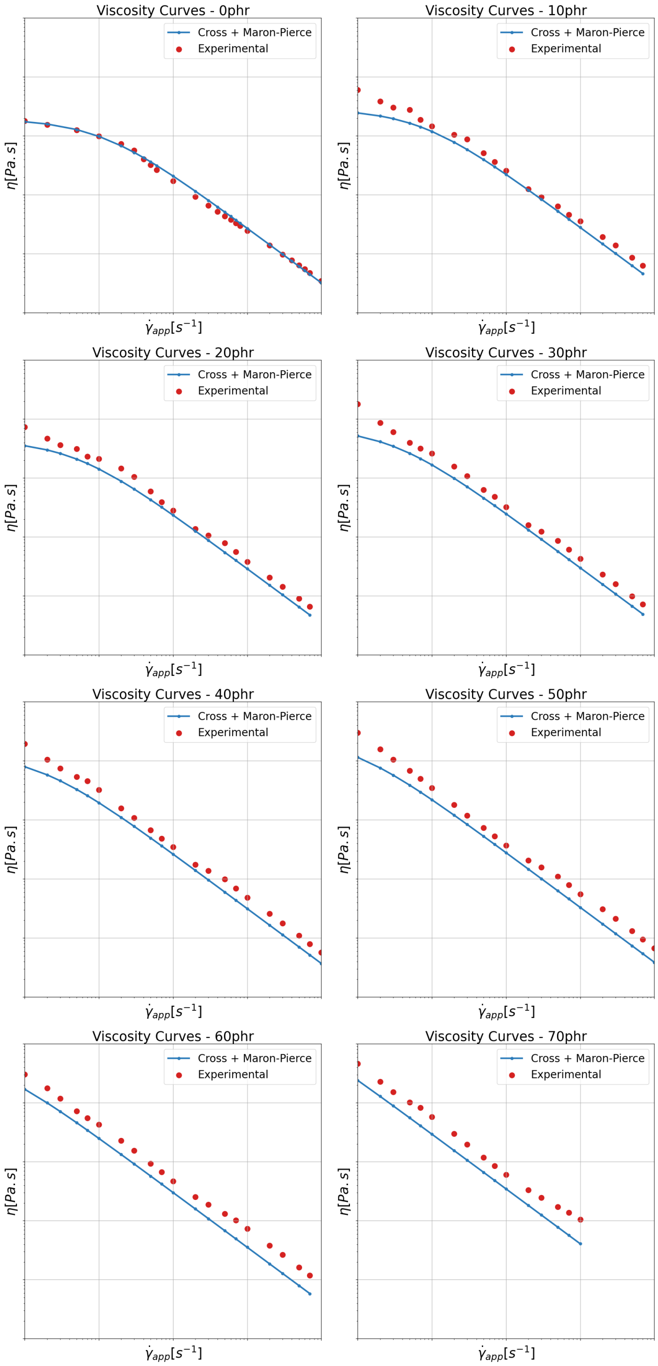
$$H(T) = exp\Big(\alpha\Big(\frac{1}{T} - \frac{1}{T_\alpha}\Big)\Big), \tag{6.15}$$

and

$$f(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}. \tag{6.16}$$

For the sake of simplicity and with no loss of generality, the coefficient λ has been set equal to 1 and the expression for viscosity (6.14) becomes

$$\eta(\dot\gamma, \phi, T) = H(T)\frac{\eta_0}{\Big(1 - \frac{\phi}{\phi_M}\Big)^2}\Big(\frac{\dot\gamma}{\Big(1 - \frac{\phi}{\phi_M}\Big)^2}\Big)^{n-1}. \tag{6.17}$$



Viscosity Curves - 0phr

Viscosity Curves - 10phr

Viscosity Curves - 20phr

Viscosity Curves - 30phr

Viscosity Curves - 40phr

Viscosity Curves - 50phr

Viscosity Curves - 60phr

Viscosity Curves - 70phr

Figure 6.2: Experimental viscosity data (red) compared to theoretical values of viscosity (blue) obtained through Cross + Maron-Pierce viscosity model (6.7).

Since the reference temperature T_α in the Arrhenius expression for $H(T)$ has been chosen to be equal to $T_{nominal}$, $H(T) \simeq 1$ is negligible in equation (6.17), due to the fact that the flow undergone by the filled compound in the rheometer is approximately isothermal:

$$\eta(\dot{\gamma}, \phi) = \frac{\eta_0}{\left(1 - \frac{\phi}{\phi_M}\right)^2} \left(\frac{\dot{\gamma}}{\left(1 - \frac{\phi}{\phi_M}\right)^2} \right)^{n-1}. \tag{6.18}$$

In order to estimate the values of coefficients η_0 and ϕ_M, it is assumed that, at each filler volume fraction, the viscosity of the filled compound has a power law expression with coefficients depending on ϕ:

$$\eta(\dot{\gamma}, \phi) = \mathcal{N}_0(\phi) \left(\dot{\gamma} \right)^{n(\phi)-1}. \tag{6.19}$$

For filler concentrations from $10\,phr$ up to $70\,phr$, the values of $\mathcal{N}_0(\phi)$ and $n(\phi)$ are estimated performing a linear regression $Log(\eta)$ vs $Log(\dot{\gamma})$.

Remark 6.2. *Assuming equation (6.19),*

$$\eta = \eta(\dot{\gamma}, \phi) = \mathcal{N}_0(\phi) \left(\dot{\gamma} \right)^{n(\phi)-1} \quad \Rightarrow \quad Log(\eta) = Log(\mathcal{N}_0(\phi)) + (n(\phi) - 1) Log(\dot{\gamma}).$$

$\mathcal{N}_0(\phi)$ and $n(\phi)$ may be estimated as, respectively, the quote of the regression line $Log(\eta)$ vs $Log(\dot{\gamma})$ and the slope of the regression line $Log(\eta)$ vs $Log(\dot{\gamma})$, incremented by one.

Remark 6.3. *As displayed in figure (6.1), viscosity curves, at low filler concentration levels (ϕ_{10phr},ϕ_{20phr},ϕ_{30phr}), show a power law behaviour only for high values of apparent shear rate. This is the reason why, for each filler volume fraction $\phi_{10phr} \leq \phi \leq \phi_{70phr}$, only values of viscosity corresponding to $\dot{\gamma}_{app}$ such that*

$$\dot{\gamma}_{app,min} < \dot{\gamma}^*_{app,1} \leq \dot{\gamma}_{app} \leq \dot{\gamma}^*_{app,2} < \dot{\gamma}_{app,max} \tag{6.20}$$

have been employed in performing linear regression.

Remark 6.4. *The viscosity curve corresponding to $0\,phr$, clearly presents a zero shear viscosity: only at high apparent shear rates the power law behaviour is matched. The value of $\mathcal{N}_0(0)$ has been computed as the quote of the regression line $Log(\eta)$ vs $Log(\dot{\gamma})$, including values of viscosity corresponding to $\dot{\gamma}_{app}$ such that*

$$\dot{\gamma}_{app,min} << \dot{\gamma}_{app} \simeq \dot{\gamma}_{app,max}. \tag{6.21}$$

Since expression (6.18) does not consider a dependence of exponent n on filler volume fraction ϕ, n is set equal to

$$n = \bar{n} = \frac{1}{7} \sum_{\phi=\phi_{10phr}}^{\phi_{70phr}} n(\phi). \tag{6.22}$$

The viscosity of the filled compound is then assumed to be

$$\eta(\dot{\gamma},\phi) = \mathfrak{N}_0(\phi)\Big(\dot{\gamma}\Big)^{n-1} \tag{6.23}$$

For $\phi_{10phr} \le \phi \le \phi_{70phr}$, assuming $\eta = \eta(\dot{\gamma},\phi) = \mathfrak{N}_0(\phi)\Big(\dot{\gamma}\Big)^{n-1}$, $\mathfrak{N}_0$ has been computed as

$$Log(\mathfrak{N}_0(\phi)) = \overline{(Log(\eta) - (n-1)Log(\dot{\gamma}_{app}))} =$$

$$= \frac{1}{N_{\dot{\gamma}_{app}}} \sum_{i=1}^{N_{\dot{\gamma}_{app}}} (Log(\eta_i) - (n-1)Log(\dot{\gamma}_{app,i})),$$

$$\mathfrak{N}_0(\phi) = 10^{\overline{(Log(\eta)-(n-1)Log(\dot{\gamma}_{app}))}}, \tag{6.24}$$

while for $\phi_{0phr} = 0$, $\mathfrak{N}_0(0) = \mathcal{N}_0(0)$.

Remark 6.5. *Equation (6.24) represents the least squares estimate of* $\mathfrak{N}_0$*. Assume to fix* ϕ*, let* η_i *be the experimental values of viscosity and let* $\hat{\eta}_i = \mathfrak{N}_0(\phi)(\dot{\gamma}_{app,i})^{n-1}$*. Define*

$$Err = \sum_{i=1}^{N_{\dot{\gamma}_{app}}} \Big(Log(\eta_i) - Log(\hat{\eta}_i)\Big)^2 =$$

$$= \sum_{i=1}^{N_{\dot{\gamma}_{app}}} \Big(Log(\eta_i) - (n-1)Log(\gamma_{app,i}) - Log(\mathfrak{N}_0(\phi))\Big)^2. \tag{6.25}$$

Then

$$\frac{\partial Err}{\partial Log(\mathfrak{N}_0(\phi))} = -2\sum_{i=1}^{N_{\dot{\gamma}_{app}}} \Big(Log(\eta_i) - (n-1)Log(\gamma_{app,i})\Big) + 2N_{\dot{\gamma}_{app}} Log(\mathfrak{N}_0(\phi)), \tag{6.26}$$

$$\frac{\partial^2 Err}{\partial Log(\mathfrak{N}_0(\phi))^2} = 2N_{\dot{\gamma}_{app}} > 0. \tag{6.27}$$

Then, the quantity Err *is minimised for*

$$Log(\mathfrak{N}_0(\phi)) = \frac{1}{N_{\dot{\gamma}_{app}}} \sum_{i=1}^{N_{\dot{\gamma}_{app}}} (Log(\eta_i) - (n-1)Log(\dot{\gamma}_{app,i})). \tag{6.28}$$

Comparing equations (6.18), and (6.23) it must hold that

$$\mathfrak{N}_0(\phi) = \eta_0\Big(1 - \frac{\phi}{\phi_M}\Big)^{-2n}. \tag{6.29}$$

Remark 6.6. *Since* $f(0) = 1$*, the coefficient* η_0 *could be estimated as*

$$\mathfrak{N}_0(0) = \eta_0(f(0))^n = \eta_0. \tag{6.30}$$

Finally, with the aim of estimating the value of the maximum packing filler volume fraction ϕ_M, ratios $\mathfrak{N}_0(\phi)/\eta_0$ are computed. Indeed equation (6.29) leads to

$$\frac{\mathfrak{N}_0(\phi)}{\eta_0} = (f(\phi))^n = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2n} \quad \Rightarrow \quad \phi = \phi_M\Big(1 - \Big(\frac{\mathfrak{N}_0(\phi)}{\eta_0}\Big)^{-\frac{1}{2n}}\Big). \tag{6.31}$$

The maximum packing filler volume fraction ϕ_M can be estimated as the slope of the regression line $\phi\ vs\ \Big(1 - \Big(\frac{\mathfrak{N}_0(\phi)}{\eta_0}\Big)^{-\frac{1}{2n}}\Big)$. Unsurprisingly, the values of maximum packing filler volume fraction estimated with equations (6.13) and (6.31) are such that

$$\phi_{M,Cross} \simeq \phi_{M,Power\,Law} \tag{6.32}$$

Figure (6.3) displays the experimental viscosity curves compared with the theoretical values of viscosity computed through expression (6.17). At high levels of filler concentration, equation (6.17) matches quite well with experimental viscosity data, both at low and high shear rates. For low levels of concentration of silica, the agreement between theoretical and experimental values of viscosity is quite satisfactory at high shear rates, while at low shear rates experimental viscosity exhibits a plateau, differently from what prescribed by equation (6.17).

In conclusion, in numerical simulations, the employed viscosity models are:

$$\eta(\dot\gamma, \phi, T) = H(T)\frac{\eta_0}{\Big(1 - \frac{\phi}{\phi_M}\Big)^2}\frac{1}{1 + \Bigg(\frac{\lambda}{\Big(1 - \frac{\phi}{\phi_M}\Big)^2}\dot\gamma\Bigg)^m}, \tag{6.33}$$

$$\eta(\dot\gamma, \phi, T) = H(T)\frac{\eta_0}{\Big(1 - \frac{\phi}{\phi_M}\Big)^2}\Bigg(\frac{\dot\gamma}{\Big(1 - \frac{\phi}{\phi_M}\Big)^2}\Bigg)^{n-1}. \tag{6.34}$$

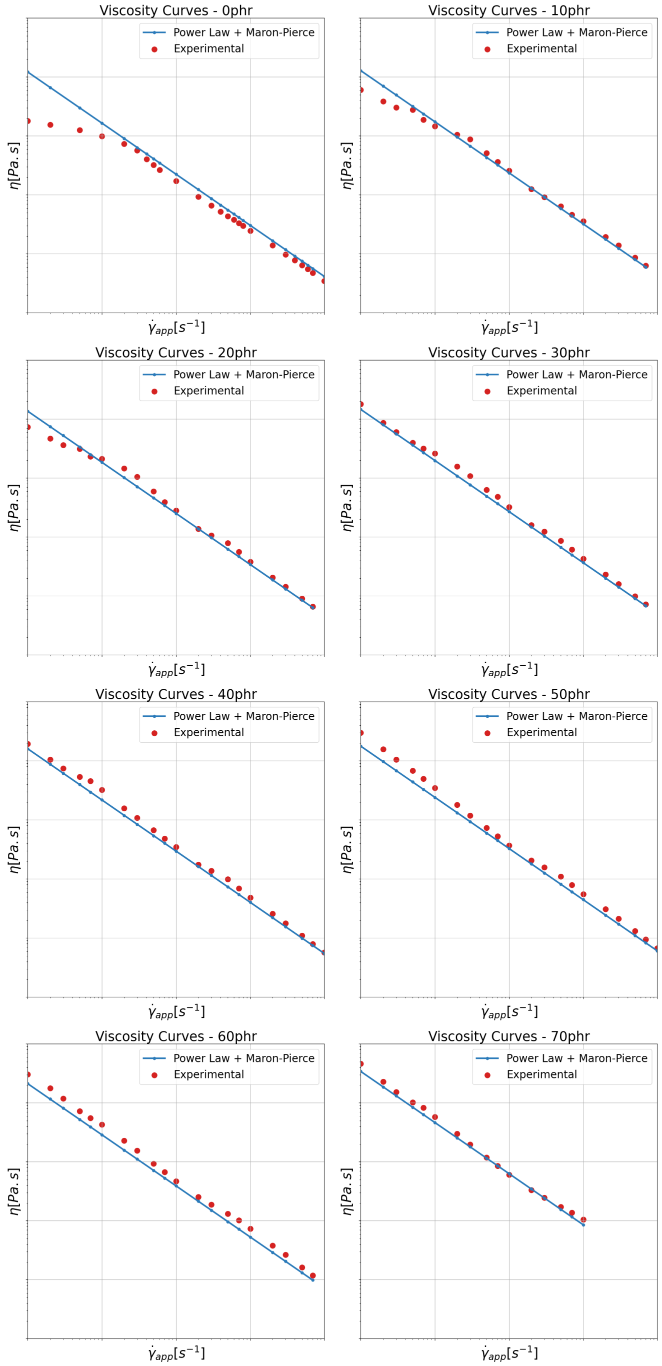
Beside equation (6.6), once known the numerical value of ϕ_M, some other filler volume fraction corrections factors $f(\phi)$ may be considered both in the Cross model (6.4) and the power law model (6.14):

- Chong:
$$f(\phi) = \Big[1 + \frac{3}{4}\Big(\frac{\frac{\phi}{\phi_M}}{1 - \frac{\phi}{\phi_M}}\Big)\Big]^2; \tag{6.35}$$

- Mooney:
$$f(\phi) = exp\Big(\frac{2.5\phi}{1 - \frac{\phi}{\phi_M}}\Big); \tag{6.36}$$

- Pal:
$$f(\phi) = \Big(1 - \Big(1 + \Big(\frac{1 - \phi_M}{\phi_M^2}\Big)\phi\Big)\phi\Big)^{-2.5}; \tag{6.37}$$

- Einstein:
$$f(\phi) = 1 + \frac{5}{2}\phi. \tag{6.38}$$


Viscosity Curves - 0phr

Viscosity Curves - 10phr

Viscosity Curves - 20phr

Viscosity Curves - 30phr

Viscosity Curves - 40phr

Viscosity Curves - 50phr

Viscosity Curves - 60phr

Viscosity Curves - 70phr

Figure 6.3: Experimental viscosity data (red) compared to theoretical values of viscosity (blue) obtained through Power Law + Maron-Pierce viscosity model (6.17).

6.2 Thermal Properties

For what concerns the modelling of thermal properties of the filled compound, the supplied experimental dataset collects the DSC (Differential Scanning Calorimetry) measures of specific heat capacity of the filled compound, at different filler concentrations and temperatures $T_{min} \leq T \leq T_{max}$ (with $T_{min} < T_{nominal} < T_{max}$) . As pointed out in chapter 5, the specific heat capacity has been considered to be dependent both on temperature and filler volume fraction:

$$C_p(T, \phi) = C_{p0} + C_{p1}(T - T_0) + C_{p2}\phi. \tag{6.39}$$

With no loss of generality the temperature T_0 has been set equal to 0, then

$$C_p(T, \phi) = C_{p0} + C_{p1}T + C_{p2}\phi. \tag{6.40}$$

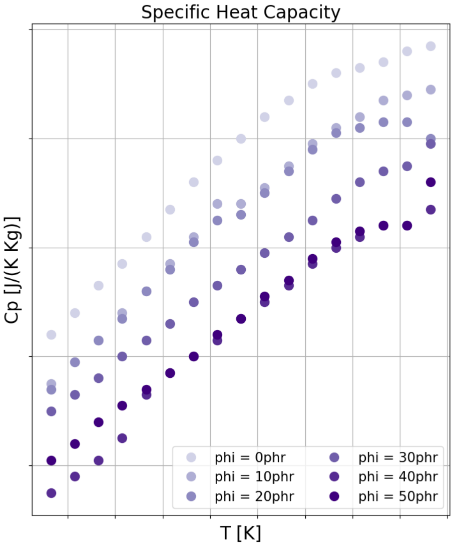


Figure 6.4: DSC experimental values of specific heat capacity.

Figure (6.4) displays the Differential Scanning Calorimetry (DSC) experimental values of specific heat capacity: the specific heat capacity seems to increase with increasing temperature, at each level of filler concentration, while seems to decrease with the increase in filler volume fraction, at each temperature $T_{min} \leq T \leq T_{max}$.

Coefficients in equation (6.40) have been estimated performing a multiple linear regression. Coherently with the previous considerations, coefficient C_{p1} turned out to be positive: the energy per unit mass required to increase the temperature of the filled compound by $1\,K$ increases with temperature. On the contrary, coefficient C_{p2} turned out to be negative implying that the specific energy needed to increase the temperature of the filled compound by $1\,K$ decreases with the increase in filler concentration.

As pointed out in chapter 5, the density of the filled compound at different levels of concentration of silica has been estimated through a convex combination of the densities of the unfilled compound and the density of the filler:

$$\rho(\phi) = (1-\phi)\rho_{unfilled} + \phi\rho_{SiO_2}. \tag{6.41}$$

Finally, the thermal conductivity of the filled compound has been computed through the following formula, presented in chapter 5,

$$k(\phi) = \delta(\phi)\rho(\phi)C_p(T_{nominal}, \phi), \tag{6.42}$$

where the thermal diffusivity values of $\delta(\phi_{0phr})$ and $\delta(\phi_{50phr})$ has been found in literature and then linearly interpolated to compute $\delta(\phi)$, for ϕ ranging from ϕ_{0phr} up to ϕ_{70phr}.

Chapter 7

Numerical Simulations

This chapter collects the results of numerical simulations run in OpenFOAM, considering different rheological models. As discussed in the previous chapter, several models have been designed in order to describe the rheology of the silica filled compound to which the experimental data refer. Those rheological models have been employed in numerical simulations which replicate the flow of the filled compound through a capillary rheometer with a circular cross section capillary (capillary die) and a length over diameter ratio equal to $L/D = 30/2$. The numerically computed pressure drops along the capillary die are then compared with the experimental pressure drops in order to verify the capability of the proposed rheological models to reproduce the behaviour of the filled compound.

7.1 Poiseuille Flow

In the capillary die of a capillary rheometer, the flow experienced by the filled compound is approximately a Poiseuille flow in a cylindrical duct. The first computational domain considered is a cylinder with radius $R = 1\,mm$ and length $L = 30\,mm$. In order to reduce the computational cost of simulations, instead of constructing a cylindrical computational grid, an axisymmetric grid, corresponding to a wedge of the cylindrical duct with an angle $\theta = \frac{2\pi}{90}$ (figure (7.1)), is considered.

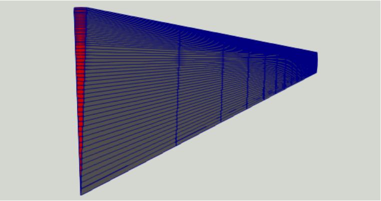


Figure 7.1: Axisymmetric Computational Domain - Poiseuille Flow.

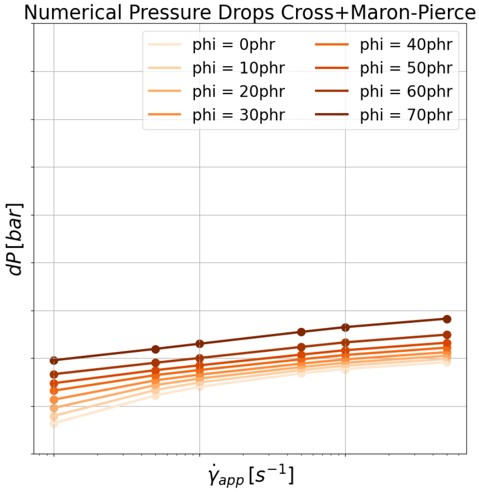


Figure 7.2: Numerically computed pressure drops at different apparent shear rates for different levels of concentration of silica (Cross + Maron-Pierce viscosity model (7.3)).

For what concerns boundary conditions, temperature has been set equal to $T_{nominal}$ both at the inflow boundary and at the wall of the duct, while the condition $\frac{\partial T}{\partial \vec{n}} = 0$ has been imposed at the outflow boundary. Pressure has been set null at the outlet and $\frac{\partial P}{\partial \vec{n}} = 0$ has been prescribed at the inlet and at the walls of the cylindrical duct. Regarding velocity, a no-slip boundary condition has been assumed on the walls of the duct while an incoming mass flowrate has been imposed at the inflow boundary:

$$\int \rho\vec{u}\cdot\vec{n} = Q, \tag{7.1}$$

with

$$\dot{\gamma}_{app,min} < \dot{\gamma}_{app} = \frac{4Q}{\pi R^3} < \dot{\gamma}_{app,max}. \tag{7.2}$$

A null normal velocity derivative ($\frac{\partial \vec{u}}{\partial \vec{n}} = 0$) has been imposed onto the outflow boundary of the computational domain. Numerical pressure drops have been computed by taking the difference between the mean pressure on the inflow boundary and the mean pressure on the outflow boundary.

Figure (7.2) displays numerically computed pressure drops obtained with the Cross + Maron-Pierce viscosity model:

$$\mu(\phi, T, \dot{\gamma}) = H(T)\frac{\eta_0}{\left(1-\frac{\phi}{\phi_M}\right)^2}\frac{1}{1+\left(\frac{t}{\left(1-\frac{\phi}{\phi_M}\right)^2}\dot{\gamma}\right)^m}, \tag{7.3}$$

with

$$H(T) = exp\Big(\alpha\Big(\frac{1}{T} - \frac{1}{T_\alpha}\Big)\Big). \tag{7.4}$$

For each level of filler concentration (ranging from $0phr$ to $70phr$), pressure drops increase with the increase in apparent shear rate: for high apparent shear rates, the fluid is flowing faster into the capillary die, hence pressure has to "push" higher at the inflow boundary. On the other side, at each apparent shear rate, pressure drops increase with the increase in silica concentration:

$$\frac{\partial\eta(\dot\gamma,\phi,T)}{\partial\phi} = \Big\{\frac{\frac{H(T)\eta_0}{\left(1-\frac{\phi}{\phi_M}\right)^2}\left(1-\frac{\phi}{\phi_M}\right)^{2m-1}\frac{2}{\phi_M}}{\left(1-\frac{\phi}{\phi_M}\right)^{2m}+(t\dot\gamma)^m}\Big\}\Big\{\frac{(1-m)(t\dot\gamma)^m+\left(1-\frac{\phi}{\phi_M}\right)^{2m}}{\left(1-\frac{\phi}{\phi_M}\right)^{2m}+(t\dot\gamma)^m}\Big\} > 0. \tag{7.5}$$

Increasing the filler volume fraction ϕ, the viscosity $\eta(\dot\gamma,\phi,T)$ increases (7.5), hence internal friction increases and a higher pressure is required in order to push the fluid through the capillary die.

Figure (7.3) displays a comparison between experimentally measured pressure drops and numerically computed pressure drops at different concentrations of silica. For low levels of concentration of silica ($\phi = \phi_{0phr}$, $\phi = \phi_{10phr}$), the numerically computed pressure drops and the experimental pressure drops are quite close one to the other, especially at low apparent shear rates, while at high apparent shear rates, numerical results and experimental data tend to diverge. On the contrary, for higher levels of concentration of silica, the numerical pressure drops are below the experimental measures, and the disagreement between them increases with the increase in filler concentration. Beside the Cross + Maron-Pierce viscosity model (7.3), many other viscosity expressions may be constructed:

$$\mu(\phi,T,\dot\gamma) = H(T)\eta_0 f(\phi)\frac{1}{1+\Big(tf(\phi)\dot\gamma\Big)^m}, \tag{7.6}$$

choosing for the volume fraction correction $f(\phi)$ one of the models described by equations (6.35-6.38).

Figure (7.4) displays numerical pressure drops computed employing different filler volume fraction corrections. For what concerns the unfilled polymer, the numerically computed pressure drops are independent of the filler volume fraction model, since for all filler volume fraction models $f(\phi_{0phr}) = f(0) = 1$. At low levels of filler concentration, all the filler volume fraction models almost coincide with the Einstein model, then the pressure drops computed employing different volume fraction corrections almost coincide. At higher levels of concentration of filler, differences between numerically computed pressure drops become more evident. Actually the pressure drops predicted by the Maron-Pierce, Pal and Chong models are similar at all levels of filler concentration. For $\phi_{30phr} \leq \phi \leq \phi_{70phr}$, the pressure drops computed employing the Einstein volume fraction model are clearly below the pressure drops obtained through the other filler volume fraction models, since the Einstein filler volume fraction

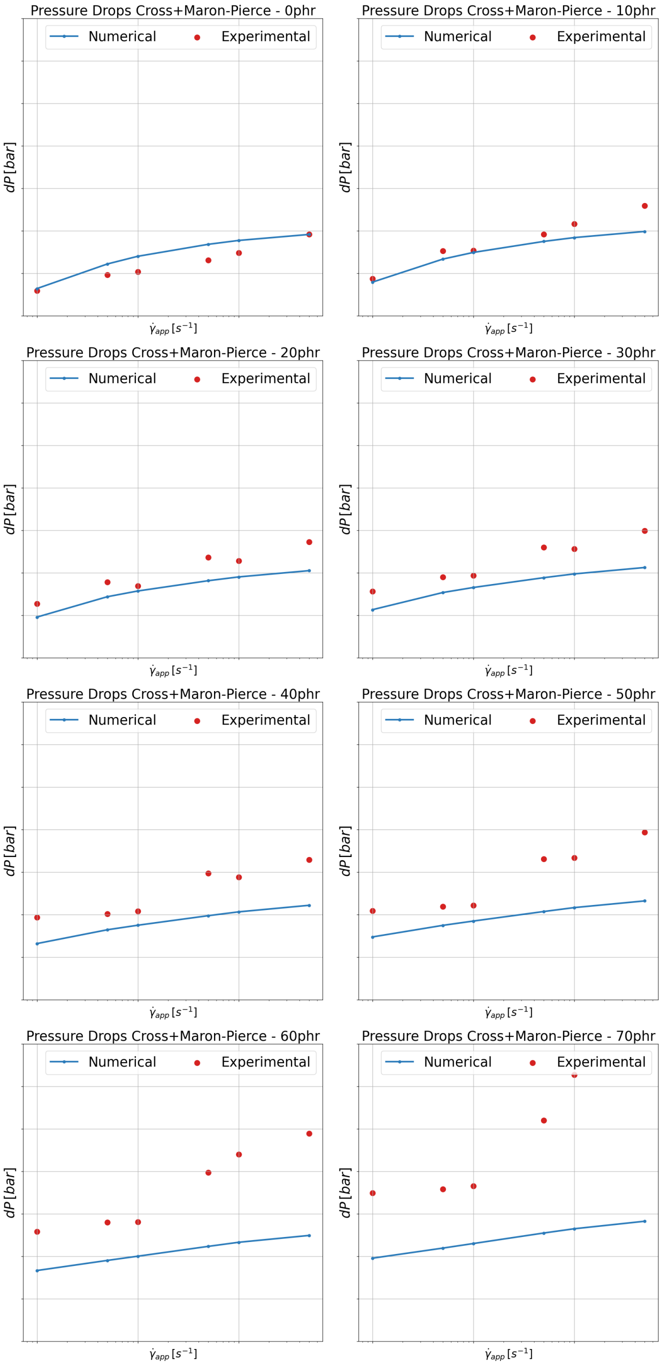


Figure 7.3: Experimental pressure drops compared with numerical pressure drops for different levels of concentration of silica (Cross + Maron-Pierce viscosity model (7.3)).

correction, differently from the others, has been designed to describe the rheology of dilute solutions (i.e.$\phi \simeq 0$). For intermediate filler volume fraction values ($\phi_{30phr} \leq \phi \leq \phi_{50phr}$) ,the pressure drops predicted by the Mooney filler volume fraction model are below the pressure drops obtained with the other models (with the only exception of the Einstein expression (6.38)). For $\phi = \phi_{60phr}$ the pressure drops computed employing different volume fraction models almost coincide (with the exception of the Einstein model (6.38)). For $\phi = \phi_{70phr}$ the Mooney expression for filler volume fraction correction $f(\phi)$ produces some unphysical values for pressure, which are not reported in figure (7.4): at high values of filler volume fraction the expression (6.36) diverges more rapidly with respect to other filler volume fraction models, affecting heavily the quality of the numerical results. Figure (7.4) displays also the experimental pressure drops. The best filler volume fraction model in approximating the experimental pressure drops seems to be the Maron-Pierce expression $f(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}$.

Figure (7.5) displays pressure drops obtained from numerical simulations involving the Power Law+ Maron-Pierce viscosity model:

$$\mu(\phi, T, \dot{\gamma}) = H(T)\frac{\eta_0}{\big(1 - \frac{\phi}{\phi_M}\big)^2}\Big(\frac{\dot{\gamma}}{\big(1 - \frac{\phi}{\phi_M}\big)^2}\Big)^{n-1}, \tag{7.7}$$

with

$$H(T) = exp\Big(\alpha\Big(\frac{1}{T} - \frac{1}{T_\alpha}\Big)\Big). \tag{7.8}$$

As can be seen in figure (7.5), the pressure drop along the capillary die increases with the increase in apparent shear rate (i.e. with the increase in flowrate through the inflow boundary), at each level of concentration of silica. Moreover, at each apparent shear rate the pressure drop increases with the increase in filler concentration since the higher is the filler volume fraction ϕ, the higher is the viscosity according to equation (7.7), since

$$\frac{\partial \eta(\dot{\gamma}, \phi, T)}{\partial \phi} = \frac{H(T)\eta_0\dot{\gamma}^{n-1}(2n)}{\Big(1 - \frac{\phi}{\phi_M}\Big)^{2n+1}}\Big(\frac{1}{\phi_M}\Big) > 0. \tag{7.9}$$

It's worth to notice that the increase in pressure drop between 60 *phr* and 70 *phr* is larger than the increase in pressure drop between 50 *phr* and 60 *phr*. Since viscosity increases more than linearly with respect to filler volume fraction

$$\frac{\partial^2 \mu}{\partial \phi^2} = H(T)\eta_0\dot{\gamma}^{n-1}\frac{2n(2n+1)}{\phi_M^2}\frac{1}{\Big(1 - \frac{\phi}{\phi_M}\Big)^{2n+2}} > 0, \tag{7.10}$$

the increment in viscosity (hence in pressure drops) increases as the filler volume fraction ϕ gets closer to ϕ_M.

Figure (7.6) performs a comparison between experimental pressure drops and numerical pressure drops computed employing the Power Law + Maron-Pierce

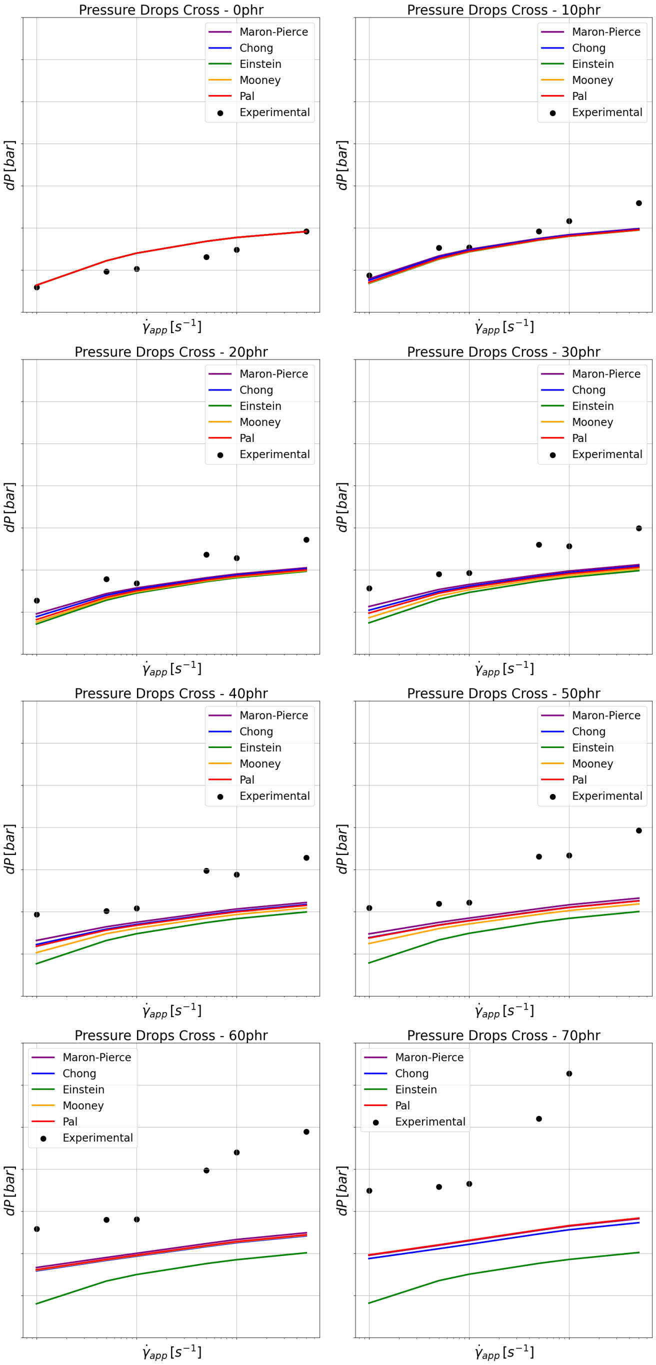


Figure 7.4: Experimental pressure drops compared with numerical pressure drops for different levels of concentration of silica and different filler volume fraction models.

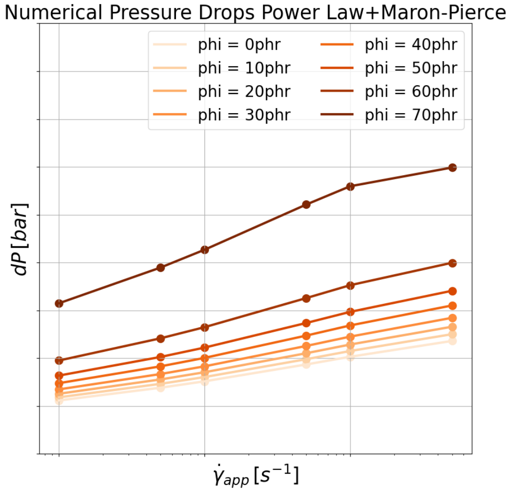


Figure 7.5: Numerically computed pressure drops at different apparent shear rates for different levels of concentration of silica (Power Law + Maron-Pierce viscosity model(7.7)).

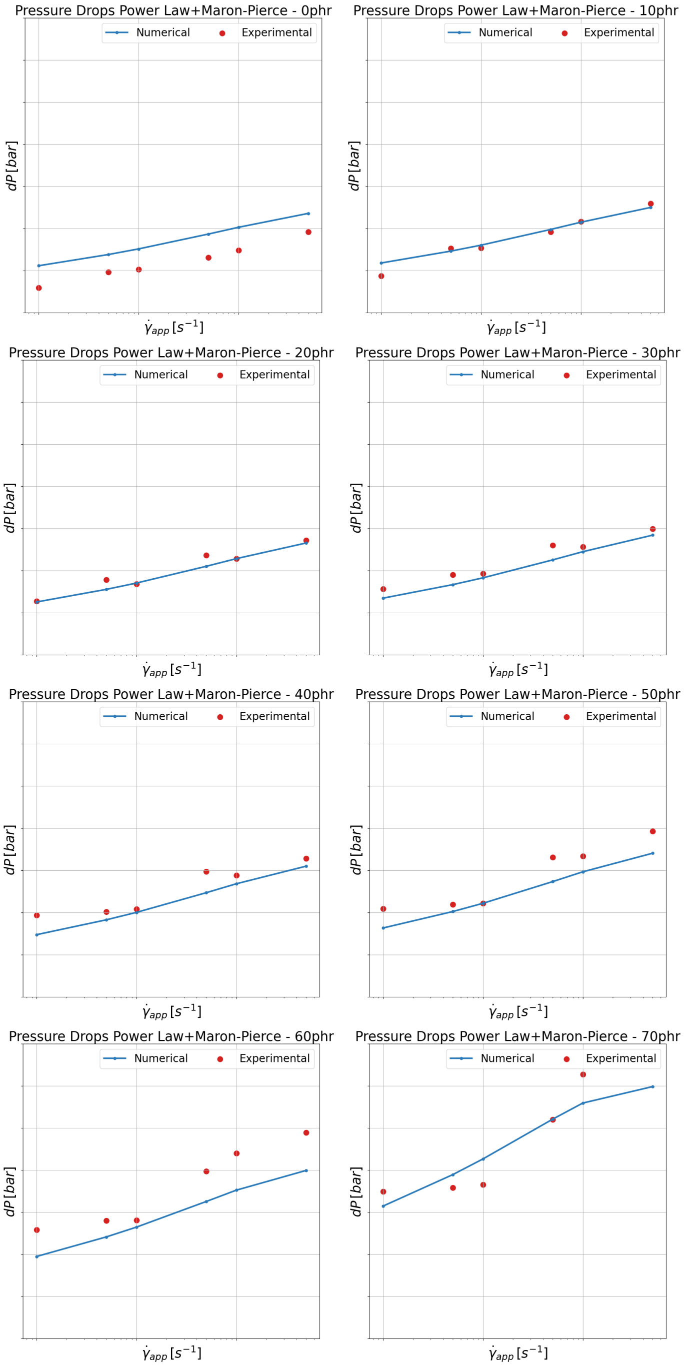
viscosity model (7.7). The agreement between numerical and experimental values is good at all levels of concentration of silica with the exception of 0 phr. For 0 phr, the Power Law + Maron-Pierce viscosity curve differs significantly from the experimental viscosity curve which, instead, presents a zero shear viscosity (see figure (6.3)): it should be expected that the numerical pressure drops do not match the experimental ones. Moreover, the Power Law + Maron-Pierce viscosity model overestimates the value of the viscosity of unfilled compound with respect to the experimental viscosity, as reported in figure (6.3), then the numerical pressure drops are coherently higher than the experimental ones.

As for the Cross viscosity model, also the Power Law viscosity expression may give raise to many different viscosity models :

$$\mu(\phi, T, \dot{\gamma}) = H(T)\eta_0 f(\phi)\big(f(\phi)\dot{\gamma}\big)^{n-1}, \tag{7.11}$$

choosing for $f(\phi)$, instead of the Maron Pierce model $f(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}$, one of the expressions (6.35-6.38).

Figure (7.7) compares experimental pressure drops with numerical pressure drops computed employing different filler volume fraction models (6.35-6.38) and $f(\phi) = \Big(1 - \frac{\phi}{\phi_M}\Big)^{-2}$. For what concerns the unfilled compound, the numerically computed pressure drops are independent of the filler volume fraction correction, since for all models $f(\phi_{0phr}) = f(0) = 1$. At low levels of filler concentration, the pressure drops obtained through different filler volume fraction models are quite similar, since all the volume fraction corrections almost coincide with the Ein-



Pressure Drops Power Law+Maron-Pierce - 0phr

Pressure Drops Power Law+Maron-Pierce - 10phr

Pressure Drops Power Law+Maron-Pierce - 20phr

Pressure Drops Power Law+Maron-Pierce - 30phr

Pressure Drops Power Law+Maron-Pierce - 40phr

Pressure Drops Power Law+Maron-Pierce - 50phr

Pressure Drops Power Law+Maron-Pierce - 60phr

Pressure Drops Power Law+Maron-Pierce - 70phr

Figure 7.6: Experimental pressure drops compared with numerical pressure drops for different levels of concentration of silica (Power Law + Maron-Pierce viscosity model (7.7)).

stein expression (6.38) for $\phi \to 0$. At higher values of filler volume fraction the numerically computed pressure drops differ on the basis of the volume fraction model employed. The Maron-Pierce, Chong and Pal volume fraction corrections predict similar pressure drops at all levels of filler concentration. The numerical pressure drops obtained through the Einstein filler volume fraction model clearly lay below the values obtained through other filler volume fraction corrections, since the Einstein model, differently form the others, is suited for low filled compounds. For what concerns the Mooney filler volume fraction correction, the numerically computed pressure drops are below the pressure drops obtained employing the Maron-Pierce, Chong and Pal volume fraction models for intermediate levels of filler concentration ($\phi_{30phr} \leq \phi \leq \phi_{50phr}$). At $\phi = \phi_{60phr}$ the pressure drops predicted by the Mooney filler volume fraction model almost coincide with the Maron-Pierce pressure drops. For high filler volume fraction values ($\phi = \phi_{70phr}$), the Mooney correction give raise to unphysical pressure drops which are not reported in figure (7.7): the Mooney highly divergent expression for $f(\phi)$ (6.36) heavily influences the reliability of numerical results. Figure (7.7), displays also the experimentally measured pressure drops. A comparison between the experimental pressure drops and the numerical pressure drops obtained through the different volume fraction models, seems to suggest the Maron-Pierce volume fraction model $f(\phi) = \left(1 - \frac{\phi}{\phi_M}\right)^{-2}$ as the best filler volume fraction correction in approximating the experimental data.

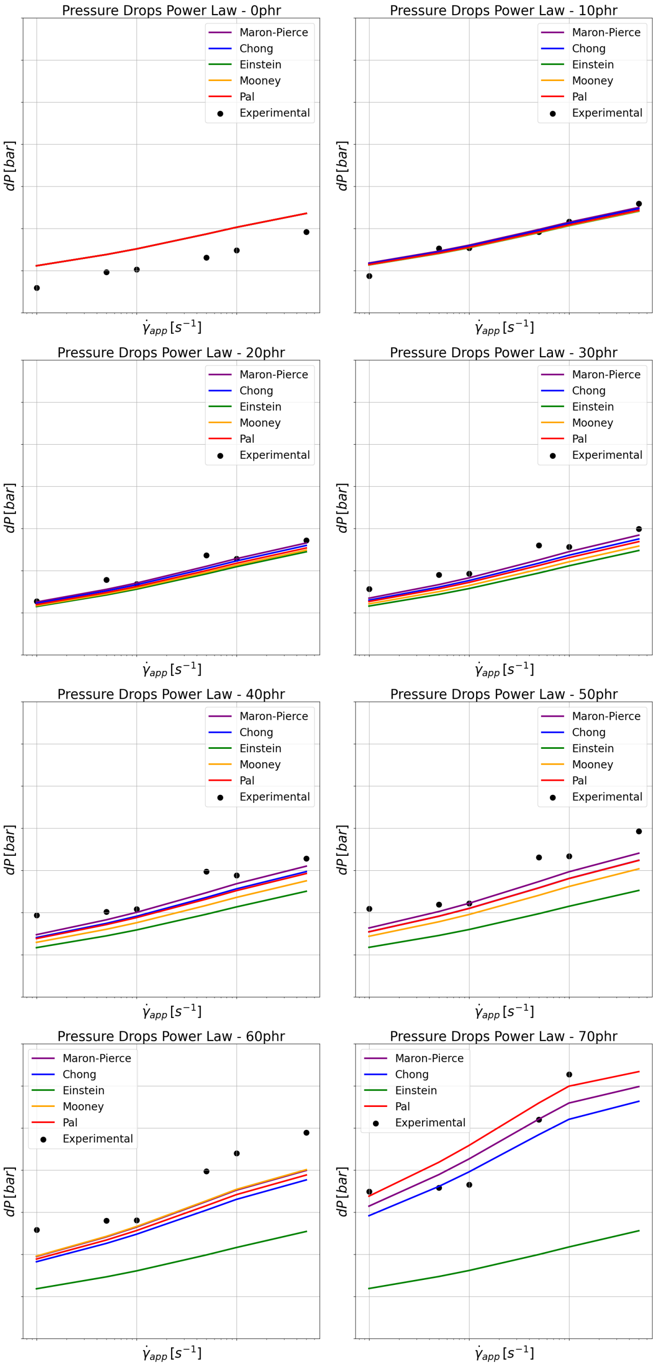


Figure 7.7: Experimental pressure drops compared with numerical pressure drops for different levels of concentration of silica and different filler volume fraction models.

7.2 Contraction Flow

A second slot of numerical simulations have been run on a wedge grid representing not only the capillary die but also the reservoir of the rheometer (figure (7.8)).

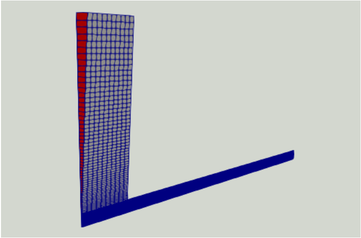
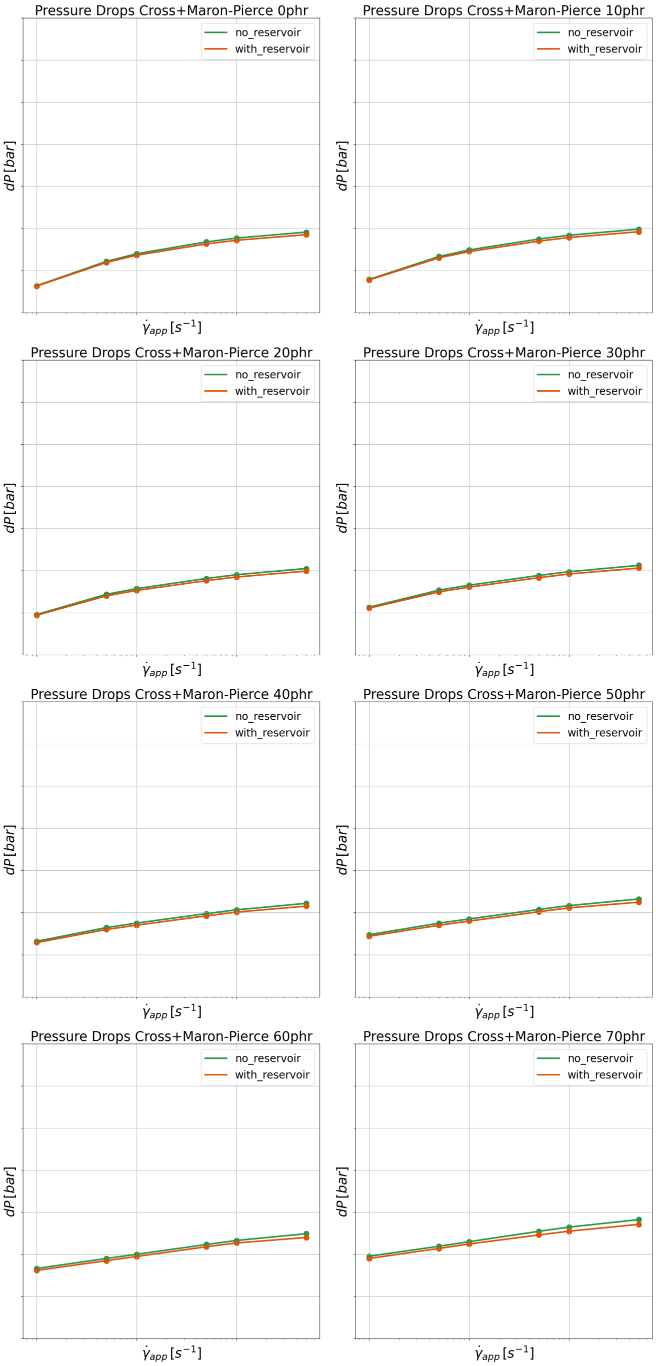


Figure 7.8: Axisymmetric Computational Domain - Contraction Flow.

The capillary die has still dimensions $R = 1\,mm$, $L = 30\,mm$, while the radius of the reservoir (a cylinder aligned with the capillary die) is $R_{tank} = 15\,mm$. Boundary conditions are the same as in the previous computational domain (conditions on the walls of the capillary die have been extended to the walls of the tank). Still the grid is an axisymmetric wedge grid with an angle $\theta = \frac{2\pi}{90}$. Pressure drops have been computed as the difference between the average pressure at the entrance of the the capillary die (not the inflow boundary) and the average pressure on the outflow boundary.

In figure (7.9) are plotted numerical pressure drops computed with the Cross + Maron-Pierce viscosity model (7.3) on the axisymmetric grid including the reservoir of the rheometer, compared with the pressure drops obtained with the grid representing only the capillary die of the rheometer. The difference between the pressure drops obtained with these two grids is minimal: including the reservoir in the computational domain is not really significant but requires a higher computational effort.

Finally in figure (7.10) is shown the difference between numerical pressure drops computed employing the Power Law + Maron-Pierce viscosity model (7.7) on different computational grids: the grid reproducing only the capillary die and the grid including the reservoir of the capillary rheometer. Once again, the discrepancy between the two curves is negligible, with the only exception of high filler volume fractions and high apparent shear rates, then it's not very relevant to include or not the reservoir in the computational domain.


Pressure Drops Cross+Maron-Pierce 0phr

Pressure Drops Cross+Maron-Pierce 10phr

Pressure Drops Cross+Maron-Pierce 20phr

Pressure Drops Cross+Maron-Pierce 30phr

Pressure Drops Cross+Maron-Pierce 40phr

Pressure Drops Cross+Maron-Pierce 50phr

Pressure Drops Cross+Maron-Pierce 60phr

Pressure Drops Cross+Maron-Pierce 70phr

Figure 7.9: Numerically computed pressure drops with Cross + Maron-Pierce viscosity model (7.3) with different computational grids.

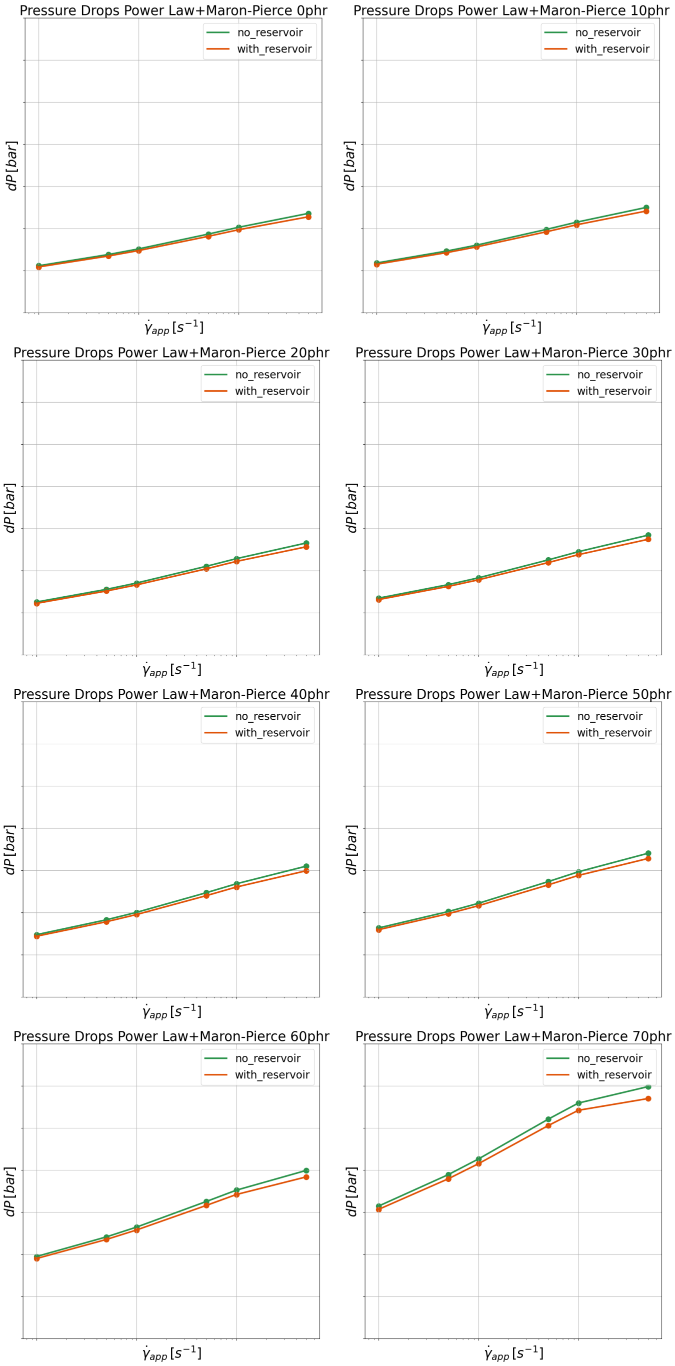


Figure 7.10: Numerically computed pressure drops with Power Law + Maron-Pierce viscosity model (7.7) with different computational grids.

Chapter 8

Conclusions

Numerical simulations have been undertaken for the flow of a filled polymer compound through the die of capillary rheometer. The filled compound has been modeled as an incompressible generalized Newtonian fluid, employing several non isothermal and filler-concentration-dependent viscosity models, which have been constructed combining some canonical viscosity models (together with the well known Arrhenius expression, to include the dependence of viscosity on temperature) with some relative viscosity expressions extracted from both recent and traditional literature on filled compounds. This approach leads to high modularity and flexibility in describing the rheology of a filled compound, assembling a suitable "complete" viscosity model by gathering an expression for viscosity and suitable models in order to represent the dependence of viscosity on temperature and filler concentration (i.e filler volume fraction). In order to approach the investigation of the flow of a filled compound through the die of a capillary rheometer, Poiseuille flows of an incompressible generalized Newtonian fluid in ducts with different cross sections have been analyzed, once governing equations (i.e. the Navier-Stokes system coupled with the energy balance equation) have been introduced together with their theoretical continuum mechanics background. Moreover, some corrections to be applied on rheometers experimental data have been presented and discussed. The aforementioned viscosity models have been implemented in OpenFOAM, an *open source* software for the solution of continuous differential problems through the Finite Volume Method (FVM). Finally, the supply of experimental rheometer data allowed to estimate the numerical value of coefficients of many of the above mentioned viscosity models, which have been lately employed in OpenFOAM to run numerical simulations.

If the unfilled polymer exhibits a Newtonian plateau at low shear rates, Cross or Bird-Carreau viscosity expressions (corrected through some suitable models to make viscosity temperature and filler-volume-fraction dependent) should be preferred, at least at low filler concentration levels. On the other hand, if the unfilled compound presents a power law viscosity behaviour, temperature and filler volume fraction corrections should be combined with a power law viscosity model, in order to describe the rheology of the filled compound. Moreover, even if the unfilled compound presents a Newtonian behaviour at low shear rates, a temperature and filler volume-fraction-dependent power law model, may be employed in order to describe the rheology of the filled compound on a wide range of filler concentration levels : the addition of filler, especially at high levels of concentration, usually erases the low shear rates Newtonian plateau. Among expressions employed to make viscosity filler-concentration-dependent, the one by Maron-Pierce, although quite traditional, is confirmed as a skillful and manageable filler volume fraction model.

As pointed out in chapter 2, the dispersion of fillers into polymer matrices is likely to alter several rheological properties, other than viscosity. In this prospective, an investigation on yield stress appearance (viscoplasticity), shear oscillatory and elongational behaviours, with respect to filler loading level, may be useful to improve the actual knowledge and mathematical modelling tools about filler compounds. Moreover, some more complex and accurate viscosity models than those presented in this research (quite "handy" and intuitive) may be corrected and enabled to describe the viscosity of filled compounds depending on the filler concentration. Actually, also viscoplastic or viscoelastic rheological models may be suited to describe the rheology of unfilled polymers and, consequently, corrected and made suitable to catch the rheological behaviour of filled compounds. As a matter of fact, also the effect of filler particles shape (aspect ratio) and size distribution plays an important role in determining the rheology of filled compounds and could be included in the mathematical description of their rheology. Finally, viscosity models proposed in this research have been employed in numerical simulations of simple experimental flows but could be tested in CFD simulations of more complex flows which may occur in industrial filled polymers processing.

Bibliography

[1] Shenoy A.V., Rheology of Filled Polymer Systems,Springer-Science+Business Media Dordrecht (1999)

[2] Bloor D., 22 - Electrical Conductivity, Comprehensive Polymer Science and Supplements, Pergamon, 687-705 (1989)

[3] Baird D.G., Polymer Processing, Encyclopedia of Physical Science and Technology (Third Edition), Academic Press, 611-643 (2003)

[4] Hornsby P.R., Fire Retardant Fillers for Polymers, International Materials Reviews, 46, 199-210 (2001)

[5] Purser D.A., Chapter 2 - Fire Safety Performance of Flame Retardants Compared with Toxic and Environmental Hazards, Polymer Green Flame Retardants,Elsevier, 45-86 (2014)

[6] Thomas, D. G., Transport characteristics of suspension: VIII. A note on the viscosity of Newtonian suspensions of uniform spherical particles, J. Colloid Sci. 20, 267 (1965)

[7] Cross, M. M.,Viscosity-concentration-shear rate relations for suspensions, Rheol. Acta 14, 402 (1975)

[8] Kataoka T., Kitano T., Sasahara M., Nishijima K., Viscosity of particle filled polymer melts, Rheol. Acta 17, 149-155 (1978)

[9] Einstein A., Investigation on the Theory of the Brownian Movement, Dover, New York (1956)

[10] Pal R., New Generalized Viscosity Model for Non-Colloidal Suspensions and Emulsions, Fluids, 5, 150 (2020)

[11] Krieger, I.M., Dougherty, T.J. Mechanism for non-Newtonian flow in suspensions of rigid particles, Trans. Soc. Rheol., 3, 137–152 (1959)

[12] Mooney, M. The viscosity of a concentrated suspensions of spherical particles. J. Colloid Sci., 6, 162–170 (1951)

[13] Eilers H., Die viskositat von emulsionen hochviskoser stoffe als funktion der konzentration, Kolloid. Zh., 97, 313 (1941)

[14] Maron S.H., Pierce P.E., Application of Ree-Eyring generalized flow theory to suspensions of spherical particles, J. Colloid Sci., 11, 80–95 (1956)

[15] Chong J.S., Christiansen E.B., Baer A.D., Rheology of concentrated suspensions, J. Appl. Polym. Sci., 15, 2007 (1971)

[16] Roscoe, R. The viscosity of suspensions of rigid spheres. Br. J. Appl. Phys., 3, 267–269 (1952)

[17] Brinkman, H.C. The Viscosity of Concentrated Suspensions and Solutions. J. Chem. Phys., 20, 571–581 (1952)

[18] Honek T., Hausnerova B., Saha P., Relative Viscosity Models and Their Application to Capillary Flow Data of Highly Filled Hard-Metal Carbide Powder Compounds, Polymer Composites, 26, 29-30 (2005)

[19] Poslinski A. J., Ryan M. E., Gupta R. K., Seshadri S. G., Frechette F. J., Rheological Behavior of Filled Polymeric Systems I. Yield Stress and Shear-Thinning Effects, Journal of Rheology 32, 703 (1988)

[20] Moukalled F., Mangani L., Darwish M., The Finite Volume Method in Computational Fluid Dynamics,Springer (2016)

[21] Versteeg H. K., Malalasekera W., An Introduction to Computational Fluid Dynamics, Pearson (2007)

[22] Forte S., Preziosi L., Vianello M., Meccanica dei Continui, Springer (2019)

[23] Fernandes C., Ferras L.L., Habla F., Carneiro O. S., Nobrega J.M., Implementation of partial slip boundary conditions in an open-source finite-volume-based computational library, J. Polym. Eng. (2019)

[24] Ferras L.L., Nobrega J.M., Pinho F.T., Implementation of slip boundary conditions in the finite volume method: new techniques (2013)

[25] Patankar S.V., A calculation procedure for two dimensional elliptic situations, Numerical Heat Transfer 4(4), 409–425 (1981)

[26] Patankar S.V., Numerical heat transfer and fluid flow, Hemisphere (1980)

[27] Patankar S.V., Spalding D.B., A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, International Journal of Heat and Mass Transfer 15(10), 1787–1806 (1972)

[28] Darwish M., Asmar D., Moukalled F., A comparative assessment within a multigrid environment of segregated pressure-based algorithms for fluid flow at all speeds, Numerical Heat Transfer, Part B 45(1), 49–74 (2004)

[29] https://www.openfoam.com

Ringraziamenti

Desidero innanzitutto ringraziare il mio relatore, il professor Nicola Parolini, per avermi permesso di partecipare al progetto svolto in collaborazione con Pirelli Tyre S.p.A. su cui si innesta questa tesi, per i suoi preziosi suggerimenti e correzioni, per essere sempre riuscito ad incastrarmi tra i suoi numerosi impegni e, in generale, per aver tenuto l'ago della bussola rivolto nella giusta direzione, rendendo possibile questo lavoro.

Il secondo ringraziamento, non meno sincero, va al dr. Giorgio Negrini, per avermi accolto all'interno del suddetto progetto, per avermi pazientemente introdotto nel mondo di OpenFOAM che inizialmente appare scuro e nebuloso, per la sua generosa disponibilità, per il supporto nell'imparare ad utilizzare LINUX e Python, nonché per i preziosi stimoli e per avermi spesso riportato sulla giusta rotta: a lui faccio i migliori auguri per il futuro.

Ringrazio caldamente il team di Pirelli Tyre S.p.A., per la valida collaborazione, la disponibilità nel fornire dati sperimentali, oltre che per gli utili riferimenti bibliografici.