Numerical simulation of yield stress fluid flows with X-MESH

DEGROOFF Vincent

May 23, 2023

Contents

Introduction

An introduction

Chapter 1

Governing equations

1.1 General notions of continuum mechanics

Let $\Omega \in \mathbb{R}^3$ denote the domain of interest and $\Gamma = \partial\Omega$ its boundary. Let $\mathbf{u}$ be the fluid velocity, and $\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}$ be the stress tensor with its deviatoric contribution $\boldsymbol{\tau}$.

Let us recall the mass and momentum conservation equations, with the pressure p and body forces $\mathbf{f}$, expressed in an Eulerian reference frame:

$$\nabla \cdot \mathbf{u} = 0 \tag{1.1}$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\, \mathbf{u} \right) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{f} \tag{1.2}$$

Throughout this study, we will consider low Reynolds numbers, i.e. flows that are dominated by viscous forces. We will thus neglect the inertia terms on the left of equation (1.3). The resulting equations are the well known Stokes equations, where we look for velocity and pressure fields $(\mathbf{u}, p)$ solution of

$$0 = -\nabla p + \nabla \cdot \boldsymbol{\tau}(\mathbf{u}) + \mathbf{f} \qquad \text{in } \Omega \tag{1.3}$$

$$0 = \nabla \cdot \mathbf{u} \qquad \text{in } \Omega \tag{1.4}$$

$$\mathbf{u} = \mathbf{U} \qquad \text{on } \Gamma_D \tag{1.5}$$

$$\boldsymbol{\sigma} \cdot \hat{\mathbf{n}} = \mathbf{g} \qquad \text{on } \Gamma_N \tag{1.6}$$

where Γ_D and Γ_N refer to the partitions of $\partial\Omega$ where Dirichlet and Neumann boundary conditions are applied. $\mathbf{U}$ and $\mathbf{g}$ are given beforehand. One could also decide to impose only the normal velocity along with the tangential stress, or the tangential velocity along with the normal stress, as will be explained in more detail in section 3.2.

Before looking for a closure equation for the shear stress tensor, let us first recall the velocity gradient decomposition,

$$\nabla \mathbf{u} = \mathbf{D} + \mathbf{W} \tag{1.7}$$

$$\mathbf{D} = \frac{1}{2} \left(\nabla \mathbf{u} + \nabla^T \mathbf{u} \right) = \frac{1}{2} \dot{\boldsymbol{\gamma}} \tag{1.8}$$

$$\mathbf{W} = \frac{1}{2} \left(\nabla \mathbf{u} - \nabla^T \mathbf{u} \right) \tag{1.9}$$

To be exhaustive:

— **D** is the strain-rate tensor, whose diagonal elements indicate the stretching of the fluid along the basis vectors and whose off-diagonal elements indicate the shearing of the fluid from one basis vector to another.

— **W** is the spin tensor that indicates the axis around which the flow is locally rotating, as well as the speed of this rotation.

1.2 Constitutive equations of Bingham fluids

Newtonian fluid behavior is the most simple one can find in fluid mechanics because it assumes that the fluid deformation, or strain rate, is is directly proportional to the shear stress it undergoes. It is very accurate for many liquids and gases such as air, water, gasoline or alcohol, under normal conditions. However, this model cannot characterize a variety of well known substances from blood to toothpaste to ketchup to name a few.

Such fluids are much better described as *generalized Newtonian* [2]. This model extends Newtonian fluids with a nonlinear constitutive law between the stress $\boldsymbol{\tau}$ and strain rate $\dot{\boldsymbol{\gamma}}$:

$$\boldsymbol{\tau} = \mu(\dot{\boldsymbol{\gamma}}, T, \phi) \dot{\boldsymbol{\gamma}} \tag{1.10}$$

where μ, T and ϕ respectively refer to the fluid viscosity, the temperature and the particle concentration (e.g. polymer chains in dilute solutions). Since the viscosity, a scalar, depends on the tensor $\dot{\boldsymbol{\gamma}}$, then it must depend only on the combinations of components γ_{ij} that are not dependent on the coordinate system [5]. These combinations are the tensor invariants, described below and intimately connected to its eigenvalues λ_i, as one can find by diagonalizing the matrix $\dot{\boldsymbol{\gamma}}$.

$$
\begin{aligned}
I_1(\dot{\boldsymbol{\gamma}}) &:= \mathrm{Tr}(\dot{\boldsymbol{\gamma}}) & &= \lambda_1 + \lambda_2 + \lambda_3 \\
I_2(\dot{\boldsymbol{\gamma}}) &:= \frac{1}{2} \left[\left(\mathrm{Tr}(\dot{\boldsymbol{\gamma}}) \right)^2 - \mathrm{Tr}(\dot{\boldsymbol{\gamma}} \cdot \dot{\boldsymbol{\gamma}}) \right] & &= \lambda_1 \lambda_2 + \lambda_2 \lambda_3 + \lambda_3 \lambda_1 \\
I_3(\dot{\boldsymbol{\gamma}}) &:= \det(\dot{\boldsymbol{\gamma}}) & &= \lambda_1 \lambda_2 \lambda_3
\end{aligned}
\tag{1.11}
$$

The first invariant is always zero when we deal with incompressible fluids since $\mathrm{Tr}(\dot{\boldsymbol{\gamma}}) = 2\nabla \cdot \mathbf{v}$. The third invariant is always neglected because it is also zero for shearing flows in 3 dimensions – $\mathbf{v} = u(y)\mathbf{e}_1$ – precisely those for which the theory was developed. The constitutive law was then later used to model more complex flows, but the absence of I_3 has persisted [5].

The simplest example of Generalized Newtonian fluids are the Power-law Fluids, that assume $\mu(\dot{\gamma}) = K\dot{\gamma}^{n-1}$, with the *consistency K* and *index n*. This can already describe

— *Shear thinning* fluids with viscosity inversely proportional to strain rate ($n < 1$). For example, Ketchup or blood flow more easily with higher strain rates.
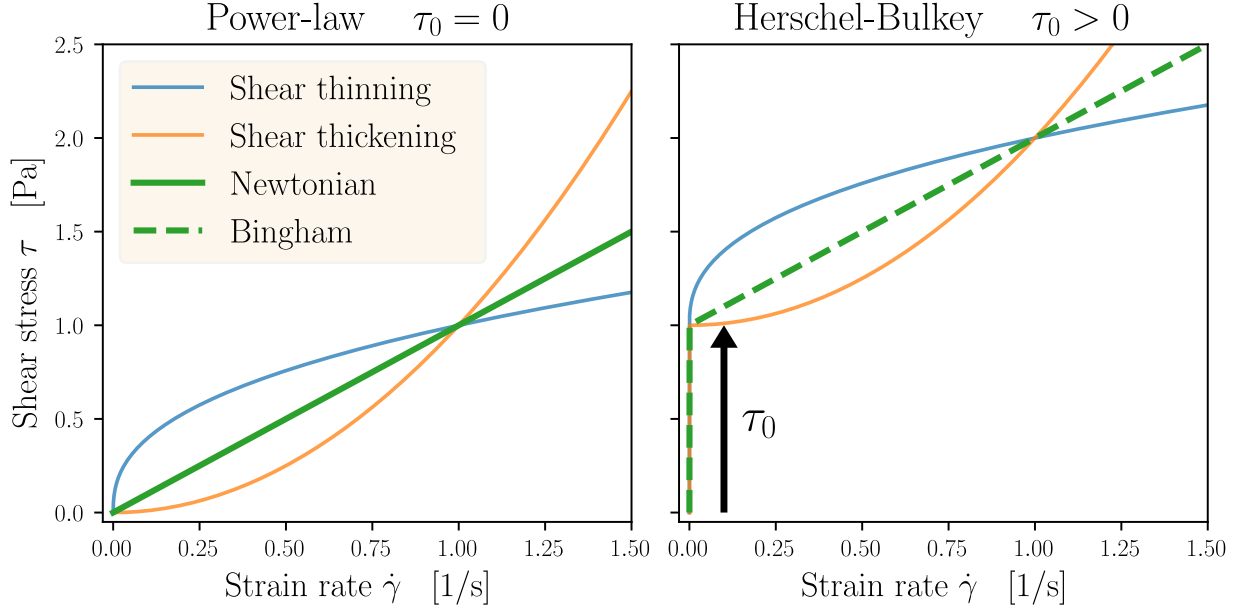
Figure 1.1. Non-Newtonian fluid models.

— *Shear thickening* or *Dilatant* fluids with viscosity proportional to strain rate ($n > 1$). A typical example is the mixture of cornstarch and water, called Oobleck, that appears liquid at rest, but that becomes viscous enough to walk on it if struck hard enough.

— *Newtonian* fluids of course when $n = 1$.

However, this model is not sufficient to describe certain fluids under low shear stress. For example, the toothpaste flows like a liquid out of the tube, but stays in its configuration once on the toothbrush, so we can easily put it in our mouth without dropping any on the floor: at low stress, such a fluid is considered to behave as a solid.

The Herschel-Bulkley Model extends the Power-law model with a yield stress τ_0, that enables us to simulate yield stress fluids like the toothpaste. When the shear stress tensor norm $\tau < \tau_0$, the fluid behaves as a non-deformable solid ($\dot{\boldsymbol{\gamma}} = \mathbf{0}$) and is called *unyielded*. In these regions, the fluid undergoes a rigid-body motion: a uniform translation and/or a uniform rotation, or possibly at rest.

Above that threshold, the fluid becomes *yielded* and follows the Power-law model. The constitutive law of the Herschel-Bulkley Model reads as follows

$$
\begin{aligned}
\dot{\gamma}_{ij} &= 0 && \text{if} \quad \tau \leq \tau_0 \\
\tau_{ij} &= \left(K\dot{\gamma}^{n-1} + \frac{\tau_0}{\dot{\gamma}} \right) \gamma_{ij} && \text{if} \quad \tau \geq \tau_0
\end{aligned}
\tag{1.12}
$$

where $\dot{\gamma}$ and τ are the square root the second invariant tensor of $\dot{\boldsymbol{\gamma}}$ and $\boldsymbol{\tau}$, also equal to their Frobenius norm since they are traceless:

$$
\dot{\gamma} := \sqrt{|I_2(\dot{\boldsymbol{\gamma}})|} = \sqrt{\frac{1}{2}\dot{\boldsymbol{\gamma}} : \dot{\boldsymbol{\gamma}}} = \sqrt{2\mathbf{D} : \mathbf{D}} \qquad \text{and} \qquad \tau := \sqrt{|I_2(\boldsymbol{\tau})|} = \sqrt{\frac{1}{2}\boldsymbol{\tau} : \boldsymbol{\tau}}
\tag{1.13}
$$

Common values of n generally lies between 0.3 and 0.5, especially for Carbopol gels (e.g. hair gel) and emulsions (e.g. mustard, mayonnaise). The Bingham model ($n = 1$) is thus very rarely

Substance	Yield stress τ_0 [Pa]	Consistency K [Pa s^n]	Index n [$-$]
Ketchup [7]	$13 - 32$	2–6	$0.44 - 0.62$
Mayonnaise [13, 12]	$75 - 200$	$10 - 50$	$0.4 - 0.65$
Molten chocolate at $40°C$ [19]	8	2.4	0.88
Hair gel [16]	60	25	0.4
Toothpaste[14]	~ 50 for children ~ 220 for adults	. .	. .
Blood [15]	32.5×10^{-3}	5.62	0.955

Table 1.1. Herschel-Bulkley parameters for common substances.

obtained when it is fitted over more than two orders of magnitude of shear rates. The yield stress value τ_0 generally lies in the range [1, 100 Pa]. [9]. To give a feel of how common substances fit in this model, several sets of parameters are provided in table 1.1.

Those values are not universal. It is in fact hard to find unanimous parameters values of the Herschel-Bulkley model in the literature, especially for food products because their composition may depend on the brand, e.g. the yield stress of mayonnaise is inversely proportional to its fat content [10]. The values may also depend on the test methodology, and on the model for which the experimental data is fitted, which is not always Herschel-Bulkley.

The present study will focus on simple yield stress fluids. By simple yield stress fluids are meant those for which the **shear stress depends only** on the imposed **shear rate**, and not on time (thixotropic/rheopectic) or shear (elastic behavior) [11]. Thixotropic fluids have the property that their viscosity *under constant shear* decreases with time. Though less common, rheopectic are the exact opposite where viscosity increases with time. This history dependence is also called *hysteresis*, and such fluids can thus have an hysteresis loop in their stress-strain diagram.

Furthermore, only Bingham fluids ($n = 1$), will be considered hereafter. The equations of this model are indeed simpler to solve, but still provide solutions with the desired features that make them interesting to analyze with the X-MESH method. Let us then reformulate model (1.12) for $n = 1$, with $\dot{\gamma}$ defined in (1.13):

$$
\begin{aligned}
\|\boldsymbol{\tau}\| < \tau_0 & \quad \text{if} \quad \dot{\boldsymbol{\gamma}} = \mathbf{0} \\
\boldsymbol{\tau} = \left(K + \frac{\tau_0}{\dot{\gamma}} \right) \dot{\boldsymbol{\gamma}} & \quad \text{if} \quad \dot{\boldsymbol{\gamma}} \neq \mathbf{0}
\end{aligned}
\tag{1.14}
$$

1.3 Weak formulation

In order to solve equations (1.3) and (1.4) with the finite element method one can express them under its weak form. This is done by multiplying the equations of the strong form by velocity and pressure test functions $\mathbf{v}$ and q. The shear stress tensor $\boldsymbol{\tau}$ is eventually replaced with its constitutive law equation (1.14).

$$
\begin{aligned}
0 &= -\nabla p + \nabla \cdot \boldsymbol{\tau}(\mathbf{u}) + \mathbf{f} \\
&= \int_\Omega -\mathbf{v} \cdot \nabla p \, \mathrm{d}x + \int_\Omega \mathbf{v} \cdot (\nabla \cdot \boldsymbol{\tau}) \, \mathrm{d}x + \int_\Omega \mathbf{f} \cdot \mathbf{v} \, \mathrm{d}x \\
&= \int_\Omega p \nabla \cdot \mathbf{v} - \int_\Omega \nabla \cdot (p\mathbf{v}) + \int_\Omega \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{v}) - \int_\Omega \boldsymbol{\tau}(\mathbf{u}) : \nabla^\top \mathbf{v} + \int_\Omega \mathbf{f} \cdot \mathbf{v} \\
&= \int_\Omega p \nabla \cdot \mathbf{v} - \int_{\partial\Omega} p\mathbf{v} \cdot \hat{\mathbf{n}} + \int_{\partial\Omega} (\boldsymbol{\tau} \cdot \mathbf{v}) \cdot \hat{\mathbf{n}} - \int_\Omega \boldsymbol{\tau}(\mathbf{u}) : (\mathbf{D}(\mathbf{v}) + \boldsymbol{\Omega}(\mathbf{v}))^\top + \int_\Omega \mathbf{f} \cdot \mathbf{v} \\
&= \int_\Omega p \nabla \cdot \mathbf{v} \, \mathrm{d}x - \int_\Omega \boldsymbol{\tau}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) \, \mathrm{d}x + \int_\Omega \mathbf{f} \cdot \mathbf{v} \, \mathrm{d}x + \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{v} \, \mathrm{d}s \\
&= \int_\Omega p \nabla \cdot \mathbf{v} \, \mathrm{d}x - \int_\Omega \left(2K + \frac{\tau_0}{\|\mathbf{D}(\mathbf{u})\|} \right) \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) \, \mathrm{d}x + \int_\Omega \mathbf{f} \cdot \mathbf{v} \, \mathrm{d}x + \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{v} \, \mathrm{d}s
\end{aligned}
\tag{1.15}
$$

that must hold $\forall \, \mathbf{v} \in H^1(\Omega)^d$ such that $\mathbf{v} = \mathbf{0}$ on Γ_D.

In equation (1.15), we initially multiplied Equation 1.3 by a test function $\mathbf{v}$ that is zero on Γ_D, then and integrated it over Ω. Afterwards, we integrated by parts, used the Divergence theorem, and the property that the double contraction of symmetric and antisymmetric tensors cancels out.

On the other hand, Equation 1.4 gets multiplied by a pressure test function q.

$$
\int_\Omega q \nabla \cdot \mathbf{u} = 0 \quad \forall q \in L^2(\Omega)
\tag{1.16}
$$

1.4   Energy functional

The velocity field $\mathbf{u}$ solution of this weak formulation can also be found through an energy functional $\mathcal{J}(\mathbf{v})$ [17, 6]. As viscous forces dominate the physics of the problem, we can assume that viscous dissipation is the only source of internal energy. As in linear elasticity for solids, the infinitesimal work of internal stresses in the deformation is

$$
\begin{aligned}
W_{\text{internal}}(D_{ij}) &= \int_0^{D_{ij}} \sigma_{ij} \, \mathrm{d}D'_{ij} \\
&= \int -p\delta_{ij} \, \mathrm{d}D'_{ij} + \int 2K D'_{ij} \, \mathrm{d}D'_{ij} + 2 \int \frac{\tau_0 D'_{ij}}{2\sqrt{\frac{1}{2} D'_{kl} D'_{kl}}} \, \mathrm{d}D'_{ij} \\
&= -p D_{ii} + K D_{ij} D_{ij} + 2\tau_0 \sqrt{\frac{1}{2} D_{ij} D_{ij}} \\
&= -p\nabla \cdot \mathbf{u} + 2K \|\mathbf{D}\|^2 + 2\tau_0 \|\mathbf{D}\| \\
W_{\text{internal}}(\mathbf{u}) &= -p\nabla \cdot \mathbf{u} + \frac{K}{2} \|\dot{\boldsymbol{\gamma}}(\mathbf{u})\|^2 + \tau_0 \|\dot{\boldsymbol{\gamma}}(\mathbf{u})\|
\end{aligned}
\tag{1.17}
$$

Notice that we were able to get rid of the twofold definition of the constitutive law of Bingham fluids. In fact, the stresses under the yield point τ_0 do not generate any work due to the absence of deformation.

The total energy of the system J is the work of internal forces diminished by the work of external forces $\int_\Omega \mathbf{f} \cdot \mathbf{u} + \int_{\Gamma_N} (\boldsymbol{\sigma} \cdot \hat{\mathbf{n}}) \cdot \mathbf{v}$. We can now reformulate equations (1.3) to (1.6) as the solution of the optimization problem (1.18).

$$
\mathbf{u} = \underset{\mathbf{v} \in \mathcal{V}}{\arg\min}\, \mathcal{J}(\mathbf{v})
$$

$$
\mathcal{J}(\mathbf{v}) = \frac{K}{2} \int_\Omega \|\dot{\boldsymbol{\gamma}}(\mathbf{v})\|^2 \, \mathrm{d}x + \tau_0 \int_\Omega \|\dot{\boldsymbol{\gamma}}(\mathbf{v})\| \, \mathrm{d}x - \int_\Omega \mathbf{f} \cdot \mathbf{v} \, \mathrm{d}x - \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{v} \, \mathrm{d}s \tag{1.18}
$$

$$
\mathcal{V} = \{\mathbf{v} \mid \nabla \cdot \mathbf{v} = 0 \text{ in } \Omega,\ \mathbf{v} = \mathbf{u}_D \text{ on } \Gamma_D\}
$$

In equation (1.17), we observe that the first term is the incompressibility constraint multiplied by its Lagrange multiplier: the pressure p, with a minus sign. It is therefore equivalent to to set it as a constraint of problem (1.18). Furthermore, in this study, only homogeneous Neumann boundary conditions will be considered. The boundary integral will therefore be neglected from now on.

We can show that the weak formulation and the energy minimization problem are equivalent. This can be done with a Gâteau derivative of the energy function around its minimum $(\mathbf{u}, p)$.

$$
\mathcal{J}(\mathbf{u} + \epsilon\mathbf{v}, p + \epsilon q) = \int_\Omega \frac{K}{2} \|2\mathbf{D}(\mathbf{u} + \epsilon\mathbf{v})\|^2 + \tau_0 \|2\mathbf{D}(\mathbf{u} + \epsilon\mathbf{v})\| \tag{1.19}
$$

$$
- (p + \epsilon q)\nabla \cdot (\mathbf{u} + \epsilon\mathbf{v}) - \mathbf{f} \cdot (\mathbf{u} + \epsilon\mathbf{v}) \, \mathrm{d}\mathbf{x} \tag{1.20}
$$

$$
\|2\mathbf{D}(\mathbf{u} + \epsilon\mathbf{v})\|^2 = \frac{1}{2} 2\mathbf{D}(\mathbf{u} + \epsilon\mathbf{v}) : 2\mathbf{D}(\mathbf{u} + \epsilon\mathbf{v})
$$

$$
= 2\mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{u}) + 4\epsilon\mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) + 2\epsilon^2\mathbf{D}(\mathbf{v}) : \mathbf{D}(\mathbf{v}) \tag{1.21}
$$

$$
\left.\frac{\mathrm{d}\mathcal{J}(\mathbf{u} + \epsilon\mathbf{v}, p)}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = \int_\Omega \frac{K}{2} 4\mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) + \tau_0 \frac{4\mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v})}{2\sqrt{2\mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{u})}} - p\nabla \cdot (\mathbf{v}) - \mathbf{f} \cdot (\mathbf{v}) \, \mathrm{d}\mathbf{x}
$$

$$
= \int_\Omega \left(2K + \frac{\tau_0}{\|\mathbf{D}(\mathbf{u})\|}\right) \mathbf{D}(\mathbf{u}) : \mathbf{D}(\mathbf{v}) - p\nabla \cdot \mathbf{v} - \mathbf{f} \cdot \mathbf{v} \tag{1.22}
$$

$$
\left.\frac{\mathrm{d}\mathcal{J}(\mathbf{u}, p + \epsilon q)}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = -\int_\Omega q\nabla \cdot \mathbf{u} \tag{1.23}
$$

It is clear that $\mathbf{u}$ is a minimum of J when (1.22) and (1.23) cancel out for every perturbation fields $\mathbf{v}$ and q. Those are precisely the expressions of the weak formulation we derived in (1.15) and (1.16).

Chapter 2

1D solution using conic optimization

2.1 Introduction

To start, we will study the well know Poiseuille flow: the fluid is pushed forward by a constant pressure gradient in a channel between two infinitely long plates. This flow is of course a bit boring since it is stationary and fully developed. But it has the merit of being simple to introduce the optimization method in a one-dimensional framework.

We consider $\Omega = [-h/2, h/2]$ that represents a transversal 1D slice of the channel whose width is h. We also have that the velocity field reduces to $\mathbf{u} = u(y)\,\hat{\mathbf{e}}_\mathbf{x}$, since the flow is x independent and incompressible. The pressure gradient $-\Delta p/\Delta x$ is imposed and considered as a body force f. The fluid obviously does not slip at the wall. Furthermore, the symmetry of the problem implies that the shear stress profile satisfies $\tau_{xy}(-y) = -\tau_{xy}(y)$.

2.2 Analytic solution

We can now solve for $u(y)$. With our hypothesis, the conservation of momentum (1.3) has boiled down to a scalar equation:

$$0 = f + \frac{\partial \tau_{xy}}{\partial y} \qquad \Longrightarrow \qquad \tau_{xy} = -fy \qquad \text{since } \tau_{xy}(0) = 0 \tag{2.1}$$

The expression of τ_{xy} can be obtained from the general expression of the stress tensor in (1.14):

$$\boldsymbol{\tau} = \begin{pmatrix} 0 & K\partial_y u \\ K\partial_y u & 0 \end{pmatrix} + \tau_0 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \operatorname{sign}(\partial_y u) \tag{2.2}$$

$$\Longrightarrow \tau_{xy} = \begin{cases} K\partial_y u - \tau_0 & y > y_0 \\ K\partial_y u + \tau_0 & y < -y_0 \end{cases} \tag{2.3}$$

The threshold shear-stress is reached at $y = \pm y_0$ when

$$\tau_{xy}(\pm y_0) = \mp f\, y_0 = \mp \tau_0 \iff y_0 = \frac{\tau_0}{f} \tag{2.4}$$
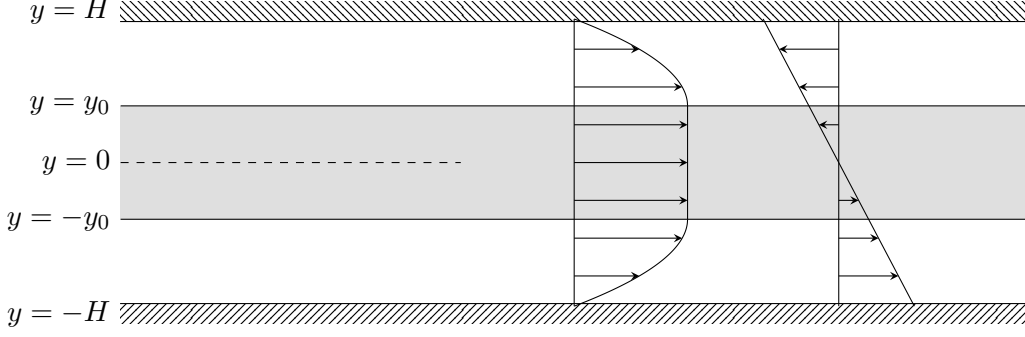
9

Figure 2.1. 1D problem channel situation, with the velocity and shear-stress profiles.

In view of (2.4), the fluid can be totally in the unyielded state when $\frac{\tau_0}{f} \geq \frac{h}{2}$. In that case, its velocity is zero everywhere to satisfy boundary conditions: the fluid is said to be in the *arrested state* [17].

In the lower zone $[-\frac{h}{2}, -y_0]$, we find the velocity profile $u_3(y)$ by integration equations (2.1) and (2.3):

$$\int_{-\frac{h}{2}}^{y} K \frac{\partial u}{\partial y} + \tau_0 = \int_{-\frac{h}{2}}^{y} -fy \tag{2.5}$$

$$u_3(y) = -\frac{\tau_0}{K}\left(\frac{h}{2}+y\right) + \frac{f}{2K}\left(\left(\frac{h}{2}\right)^2 - y^2\right) \tag{2.6}$$

The continuity of the velocity field allows us to find the velocity of the plug:

$$u_2 = u_3(-y_0) = \frac{f}{8K}(h-2y_0)^2 \tag{2.7}$$

The procedure is very similar for the upper zone $[y_0, h/2]$.

Let us define the non-dimensional coordinate η, the reference velocity U_∞ as the maximum velocity of a classical Poiseuille flow and the Bingham number Bn that relates the yield and viscous stresses.

$$\eta = \frac{2y}{h} \qquad U_\infty = \frac{fh^2}{8K} \qquad Bn = \frac{\tau_0 h}{KU_\infty} = \frac{8\tau_0}{fh} \qquad \eta_0 = \frac{2y_0}{h} = \frac{Bn}{4} \tag{2.8}$$

The velocity profile can then be expressed as follows, in the presence of a yielded region, i.e. when $\eta_0 < 1 \iff Bn < 4$:

$$\frac{u(\eta)}{U_\infty} = -\frac{Bn}{2}(1-\eta) + (1-\eta^2) \qquad\qquad \eta_0 < \eta \leq 1 \tag{2.9}$$

$$\frac{u(\eta)}{U_\infty} = \left(1 - \frac{Bn}{4}\right)^2 \qquad\qquad -\eta_0 \leq \eta \leq \eta_0 \tag{2.10}$$

$$\frac{u(\eta)}{U_\infty} = -\frac{Bn}{2}(1+\eta) + (1-\eta^2) \qquad\qquad -1 \leq \eta < -\eta_0 \tag{2.11}$$

2.3 Finite element formulation

Let us first define the mesh $\Omega = [-h/2, h/2]$ that is made of n elements $\Omega_i = [y_{i-1}, y_i]$, $i = 1, \ldots, N$, and $N+1$ nodal values (y_i). The velocity profile $u(y)$ is approximated by sum of shape functions ϕ_i weighted by their associated nodal value U_i. These shape functions have a compact support around their associated node, and satisfy $\phi_j(x_k) = \delta_{jk}$.

$$u^h(y) = \sum_{j=0}^{N} U_j \phi_j(y) \tag{2.12}$$

We will consider both linear and quadratic shape functions. They are expressed on a reference element $\hat{\Omega} = [-1, 1]$ and numbered with local node index $j = 1, \ldots, n$ in Equation 2.13 and Figure 2.2.

$$
\begin{aligned}
\phi_1(\eta) &= \frac{1 - \eta}{2} & \phi_1(\eta) &= \frac{1}{2}\eta(1 - \eta) \\
\phi_2(\eta) &= \frac{1 + \eta}{2} & \phi_2(\eta) &= \frac{1}{2}\eta(1 + \eta) \\
& & \phi_2(\eta) &= 1 - \eta^2
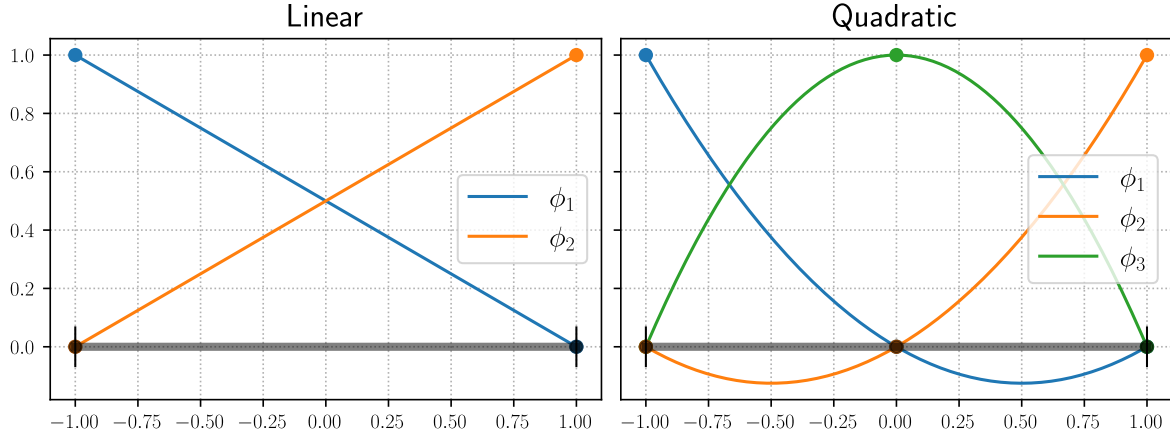\end{aligned}
\tag{2.13}
$$

Figure 2.2. 1-dimensional shape functions ϕ on the reference element.

The strain-rate tensor with a unidirectional flow $u(y)$ only contains the off-diagonal component,

$$D_{12}^h(y) = \frac{1}{2}\frac{\partial u^h}{\partial y} \quad \text{where} \quad \left(\frac{\partial u^h}{\partial y}\right)_{\Omega_i} = \sum_{j=1}^{n} U_j \frac{\mathrm{d}\phi_j}{\mathrm{d}y} \tag{2.14}$$

The minimization function J of Equation 1.18 becomes:

$$
\begin{aligned}
\mathcal{J}(u^h) &= \int_\Omega \left[\frac{K}{2}\left(\frac{\partial u^h}{\partial y}\right)^2 + \tau_0\left|\frac{\partial u^h}{\partial y}\right| - fu^h\right] \mathrm{d}y \\
&= \sum_{i=1}^{N} \int_{\Omega_i} \left[\frac{K}{2}\left(\frac{\partial u^h}{\partial y}\right)_{\Omega_i}^2 + \tau_0\left|\frac{\partial u^h}{\partial y}\right|_{\Omega_i} - fu^h|_{\Omega_i}\right] \mathrm{d}y \\
&= \sum_{i=1}^{N} \sum_{g=1}^{ng} \omega_g \left[\frac{K}{2}\left(\frac{\partial u^h}{\partial y}\right)_{y=y_g}^2 + \tau_0\left|\frac{\partial u^h}{\partial y}\right|_{y=y_g} - fu^h(y_g)\right] \frac{\Delta y_i}{2}
\end{aligned}
\tag{2.15}
$$

where we used Gauss-Legendre quadrature of appropriate order with weights ω_g and coordinates y_g in the element Ω_i, mapped from coordinates η_g in the reference element $[-1, 1]$.

In short, one should solve problem (2.16) hereunder

$$
\begin{aligned}
&\text{minimize} && \mathcal{J}(u^h) \\
&\text{s.t.} && u^h(y = -h) = u^h(y = h) = 0
\end{aligned}
\tag{2.16}
$$

However, this problem cannot be solved straightforwardly with classical optimization techniques such as gradient descent or Newton's method as it contains a non-differentiable term $|\frac{\partial u^h}{\partial y}|$. Next section will introduce a suitable method, known as conic optimization.

One could argue that absolute values can be transformed in multiple linear terms in this case. However, this is no longer true as we go to higher dimensions since the tensor norm will become the square root of multiple squared terms.

2.4 Conic optimization in a nutshell

This theory extends linear programming which is restricted to linear cost and linear constraints, by allowing specific nonlinear inequality constraints. For example, $\sqrt{x^2 + y^2} \leq z$ is expressed as $(x, y, z) \succeq_{L^3} 0$ or $(x, y, z) \in L^3$, where the Lorentz cone L^3 is precisely the set of points (x, y, z) satisfying the nonlinear constraint.

A cone is a subset of a vector space that is closed under linear combinations with positive coefficients. Additional properties can upgrade a cone to a *proper* cone, the only ones we care about in conic optimization. These proper cones $K \subseteq \mathbb{R}^n$ satisfy the following properties:

1. $a \succeq_K 0 \implies \lambda a \succeq_K 0 \quad \forall \lambda \in \mathbb{R}^+$ (cone)

2. $a \succeq_K 0$ and $b \succeq_K 0 \implies a + b \succeq_K 0$ (closed under addition)

3. $x \succeq_K 0$ and $x \preceq_K 0 \implies x = 0$ (pointed)

4. $\text{int}(K) \neq \emptyset$ (solid)

5. if $\{x_i\}_{i \to \infty}$ with $x_i \succeq_K 0 \quad \forall i$, then $\lim_{i \to \infty} x_i = \bar{x} \implies \bar{x} \succeq_K 0$ (closed)

Cones with such properties will ensure us a global convergence of the Newton's iterations through functions known as *self-concordant barriers*. This is of course very appreciated since Newton's algorithm only converges locally in a general framework. A function $g : X \to \mathbb{R}$, where $X \subseteq \mathbb{R}^n$ is called self-concordant iff

- $g \in \mathcal{C}^3$, and

- g is convex, and

- $\nabla^3 g(x)[h, h, h] \leq 2\big(\nabla^2 g(x)[h, h]\big)^{3/2} \quad \forall x \in X \quad \forall h \in \mathbb{R}^n$

where $\nabla^3 g(x)[h, h, h] = \sum_{i,j,k} \frac{\partial^3 g}{\partial x_i \partial x_j \partial x_k}(x) \, h_i h_j h_k$. With univariate functions $g(x) : \mathbb{R} \to \mathbb{R}$, this last property gets simplified to $|g'''(x)| \leq 2\big(g''(x)\big)^{3/2}$. Multivariate functions can then be verified to be self-concordant using the univariate test version on $G_{x,h}(t) : \mathbb{R} \to \mathbb{R} : t \mapsto g(x + th) \quad \forall x \in X \quad \forall h \in \mathbb{R}^n$.

Last but not least, self-concordance (s.c.) is preserved through

- sums: let f and g be s.c. functions, then $h = f + g$ is also a s.c. function.

Name	Definition and barrier
Half-space	$\mathbb{R}_+$
	$g(x) = -\log(x)$
Lorentz/quadratic cone	$L^{n+1} = \{(x,t) \in \mathbb{R}^n \times \mathbb{R} \mid \|x\|_2 \leq t\}$
	$g(x,t) = -\log(t^2 - \|x\|^2)$
Rotated Lorentz cone	$L_R^{n+2} = \{(x,s,t) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \|x\|_2^2 \leq 2st\}$
	$g(x,s,t) = -\log(2st - \|x\|_2^2)$
Exponential cone	$E = \text{closure}\{(x,y,z) \in \mathbb{R}^3 \mid z \geq y\exp(e/y), y > 0\}$
	$g(x,y,z) = -\log(z - y\exp(e/y)) - \log(y) - \log(z)$
Power cone	$P_\alpha = \{(x,y,z) \in \mathbb{R}^3 \mid x^\alpha y^{1-\alpha} \geq |z|, x > 0, y > 0, 0 < \alpha < 1\}$
	$g(x,y,z) = -\log(x^{2\alpha}y^{2-2\alpha} - z^2) - \log(x) - \log(y)$

Table 2.1. Most frequent proper cones used in conic programming.

– linear change of variables: let $x \mapsto f(x)$ be a s.c. functions, then $y \mapsto f(Ay + b)$ is also a s.c. function.

The most common proper cones, with their associated s.c. barrier. are listed in Table 2.1. It may not seem very useful to be limited to such a short list of cones, but a whole zoo of nonlinear constraints can be reformulated to fit in any of these five cones.

Once all the nonlinear constraints have been translated into conic constraints with s.c. barriers $g_i(x)$, the solution is found using an *Interior-point method* that minimizes

$$f_\mu(x) = \frac{c^\top x}{\mu} + \sum_i g_i(x) \tag{2.17}$$

where c is the original linear cost and $\mu > 0$ is progressively brought to zero. For each μ, there is an unique solution x_μ^*. The set of solutions x_μ^* is called the *central path*. One can eventually retrieve the solution of the original problem since $x_\mu^* \to x^*$ as $\mu \to 0$.

Let us take a basic example in 2 dimensions.

$$\min_{x,y} \quad x$$
$$\text{s.t.} \quad 2x + y \geq 3 \quad \text{and} \quad 2x - 4y \leq 1 \quad \text{and} \quad x + 5y \leq 10$$
$$\implies \quad f_\mu(x,y) = \frac{x}{\mu} - \log(2x + y - 3) - \log(1 - 2x + 4y) - \log(10 - x - 5y)$$

In practice, no one computes the x_μ^* on the *central path* because these points are only used as starting point of the next iteration with a lower μ. Instead, we use an iterative algorithm where we alternate between a Newton step and a decrease of μ until we reach the required precision. Newton steps keep the current solution close enough to the central path, while the decrease of μ brings the objective function f_μ to the linear function $c^\top x$.

For a precision ϵ s.t. $c^\top x - c^\top x^* < \epsilon$, a solution x is obtained in $\mathcal{O}(\sqrt{\nu}\log\frac{1}{\epsilon})$ iterations with the *short-step algorithm* briefly described here above. $\nu = \sum_i \nu_i$, with the barrier parameter ν_i, in a sense related to the *steepness* of the constraint i. The initial value x must be close enough to the central path. This can be done with damped Newton steps from any admissible $x \in X$ CITE.
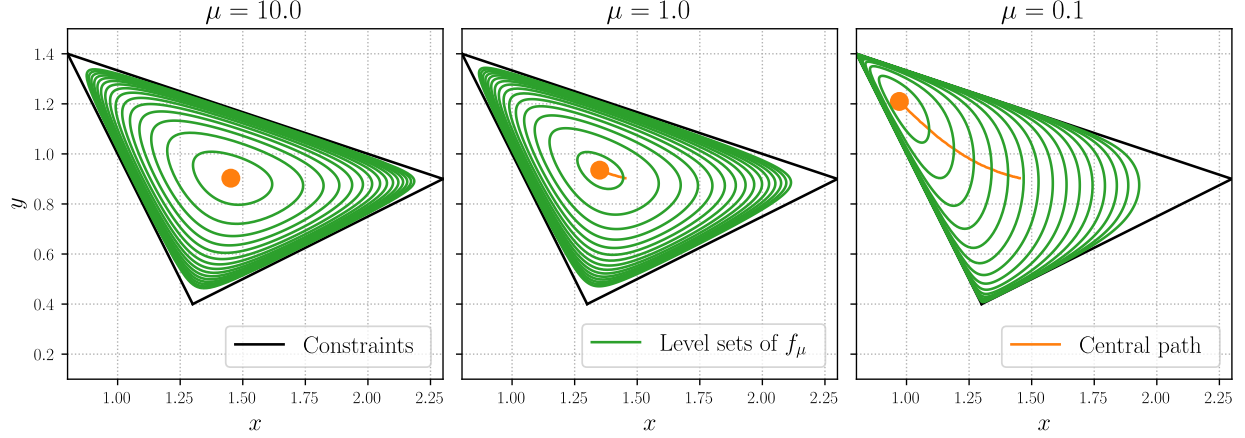
Figure 2.3. Solution of the basic example using the interior point method

2.5 Finite element solution

With our recent knowledge in conic programming, we can reformulate problem (2.16) in terms of second order cones only (SOCP).

$$
\begin{aligned}
\underset{U_j, S_{i,g}, T_{i,g}}{\text{minimize}} \quad & \sum_{i=1}^{N} \sum_{g=1}^{ng} \omega_g \left[\frac{K}{2} S_{i,g} + \tau_0 T_{i,g} - f u^h(y_g) \right] \frac{\Delta y_i}{2} \\
\text{s.t.} \quad & \left(\frac{\partial u^h}{\partial y} \right)^2_{y=y_g} \leq S_{i,g} \quad \forall i \, \forall g \qquad \Longleftrightarrow \qquad \left[S_{i,g}, \frac{1}{2}, \left(\frac{\partial u^h}{\partial y} \right)^2_{y=y_g} \right] \in L^3_R \quad \forall i \, \forall g \\
& \left| \frac{\partial u^h}{\partial y} \right|_{y=y_g} \leq T_{i,g} \quad \forall i \, \forall g \qquad \Longleftrightarrow \qquad \left[T_{i,g}, \left| \frac{\partial u^h}{\partial y} \right|_{y=y_g} \right] \in L^2 \quad \forall i \, \forall g \\
& U_0 = U_N = 0
\end{aligned}
$$

$$\tag{2.18}$$

We minimize this problem over the nodal velocities U, and the newly added S and T variables. Even if the inequalities may confuse at first glance, they are valid from a modelling standpoint. They are always verified as equalities at the optimum. We can show it by contradiction. Let us assume that $\left(\frac{\partial u^h}{\partial y} \right)^2 < S_{i,g}$ for a specific index i, g at the optimum. Then the cost can be reduced by decreasing $S_{i,g}$ by $\epsilon > 0$, keeping all other variables unchanged: we are not at the optimum.

At this stage, the finite element solution will be relevant only when the interface is represented on the mesh. Therefore, the arbitrary mesh is manually modified such that two nodes are placed at $y = \pm y_0$. The goal of the next section, is to develop an algorithm that moves the nodes without knowing the interface position beforehand.

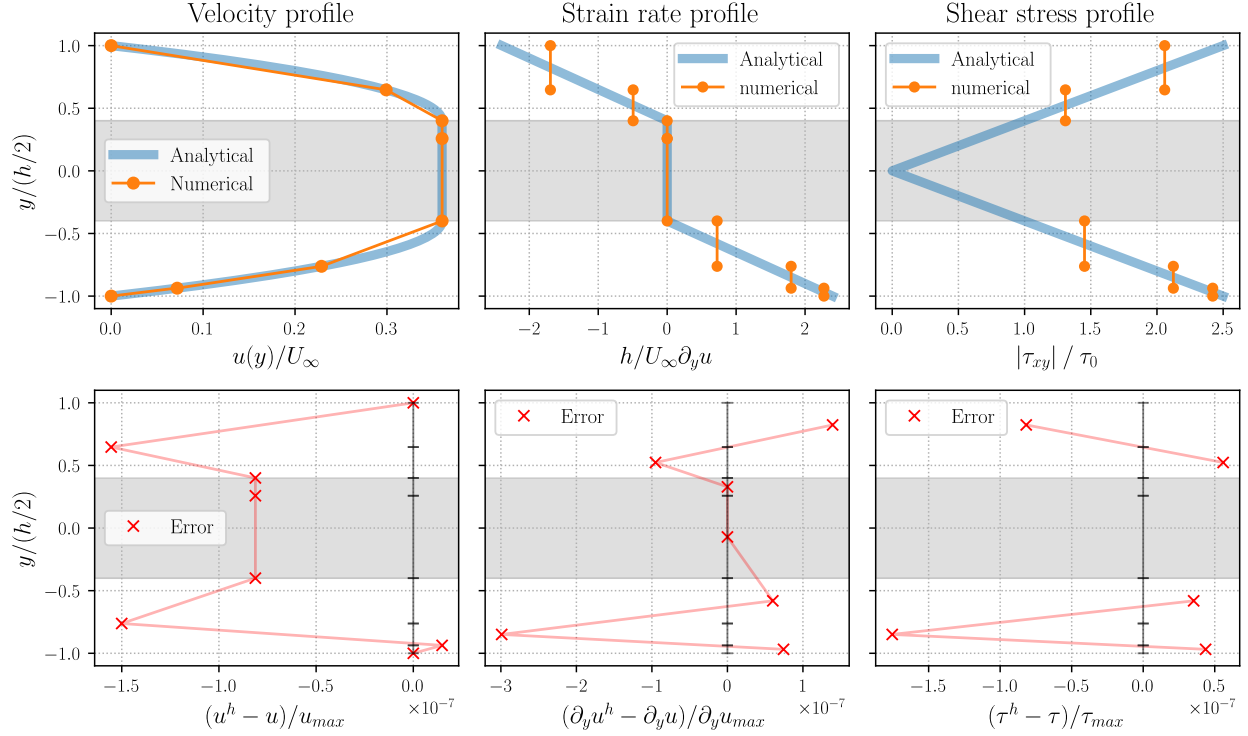The optimization problem was solved with the interior-point solver of the open-source software CVXOPT.

Figure 2.4. FE solution of the Poiseuille flow with $\mathcal{P}_1$ elements. The unyielded zone is shaded in grey. The error is shown at the nodes for $u(y)$ and at the element center for the strain and shear.
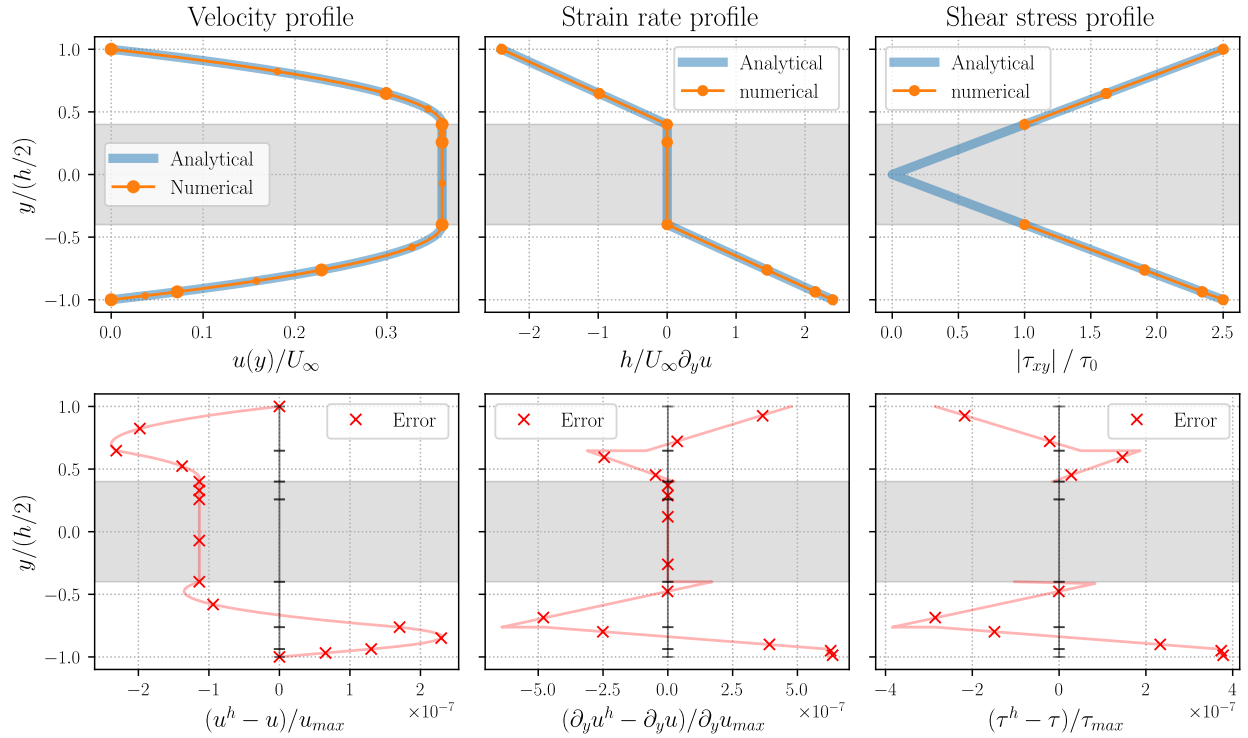

Figure 2.5. FE solution of the Poiseuille flow with $\mathcal{P}_2$ elements. The unyielded zone is shaded in grey. The error is computed on the whole domain Ω.

2.6 Interface tracking

The concept of *front tracking* was initially introduced by James Glimm and co-authors [8] in the context of hydrodynamics. So I let him define it in his own words [18]:

> *Front tracking is the use of surfaces or lower dimensional manifolds as computational degrees of freedom in a numerical algorithm. Its purpose is to improve the resolution of discontinuities or steep gradients in the solution variables or in the laws of physics which describe them.*

In our model, the velocity field and even the strain rate field are continuous. In fact, one can infer that on the liquid side of the interface, where $\tau = \tau_0$, the strain rate must also equal to 0.

$$\tau_0 = \|\boldsymbol{\tau}\| \stackrel{(1.14)}{=} (K + \tau_0/\dot{\gamma})\|\dot{\boldsymbol{\gamma}}\| = K\dot{\gamma} + \tau_0 \qquad \Longrightarrow \qquad \dot{\gamma} = 0 \qquad\qquad (2.19)$$

The discontinuity is to be found in the derivative of the deformation, e.g. the concavity of the velocity profile in the 1-dimensional case. In fact, $\partial_{yy}u = 0$ in the solid plug as $\partial_y u = 0$ over a whole nonempty interval, while $\partial_{yy}u > 0$ in the liquid region. This is what we observed in figures 2.4 and 2.5.

2.6.1 Reconstructed strain

We will therefore make this continuity of the strain rate field our primary objective. Sadly, $\mathcal{P}_1$-elements for the velocity only provide piecewise constant strains, which are of course discontinuous. The idea is thus to reconstruct a continuous strain field as a linear approximation of the finite element solution at the Gauss points. This approximation is one-sided at each interface as we should only take the information from the yielded regions.

The choice of gauss points as abscissae of the linear approximation may seem trivial in this 1-dimensional model case, but the topic was seriously investigated for general geometries. Barlow showed that reduced integration points are superconvergent points, i.e. at which the stress is an order of magnitude more accurate than in any other point within the element [4]. Later on, Zienkiewicz and Zhu developed an error estimator by extrapolating values at the integration points [20].

The procedure is described in algorithm 1 and illustrated in figures 2.6 and 2.8. The strain field $\partial_y u$ is shown at each iteration with with its corresponding mesh. The complete overview of the solution is provided for the first and last iterations, for both $\mathcal{P}_1$ and $\mathcal{P}_2$ elements, in figures 2.7 and 2.9.

Algorithm 1: Interface-tracking algorithm in 1 dimension.

---

optimalMesh ← false;
$\mathcal{M} \leftarrow$ initial mesh with nodes y_i;
while *not optimalMesh* **do**
  Minimize the energy functional $\mathcal{J}$ with $\mathcal{M}$ and retrieve the solutions $U_j, T_{i,g}$;
  $\mathcal{I} \leftarrow$ set of current interfaces (node shared between yielded and unyielded elements);
  //Stop if $\mathcal{I} = \emptyset$: the mesh should be finer to have unyielded elements
  optimalMesh ← true;
  forall $k \in \mathcal{I}$ **do**
    $\ell_k \leftarrow$ one-sided linear approximation function of the strain close to y_k;
    if $\ell_k(y_k) \not\approx 0$ **then**
      optimalMesh ← false ;   //since $\partial_y u$ is not continuous at the interface
      $y_k \leftarrow \text{root}(\ell_k(y))$;   //update $\mathcal{M}$ using the linear approximation
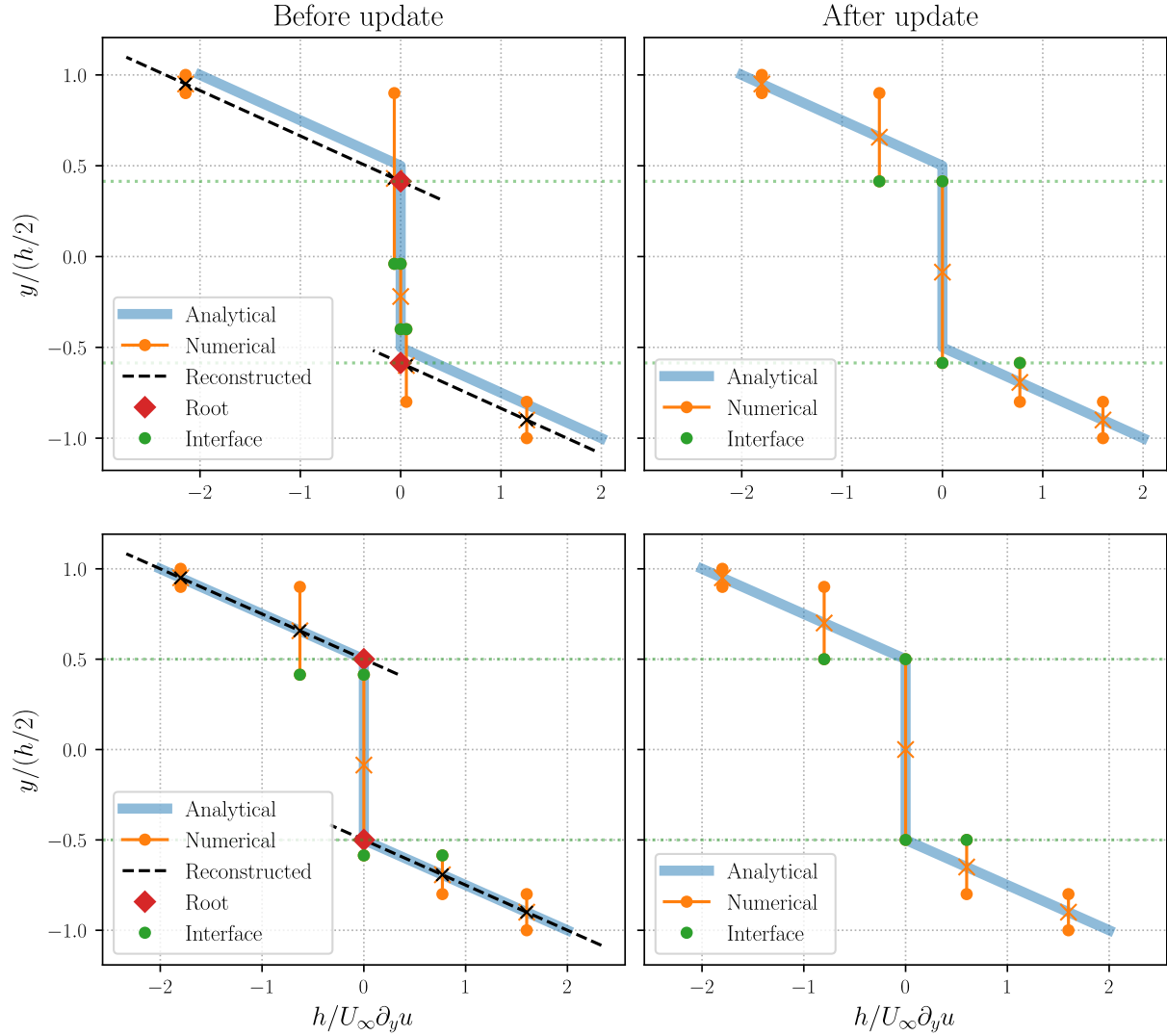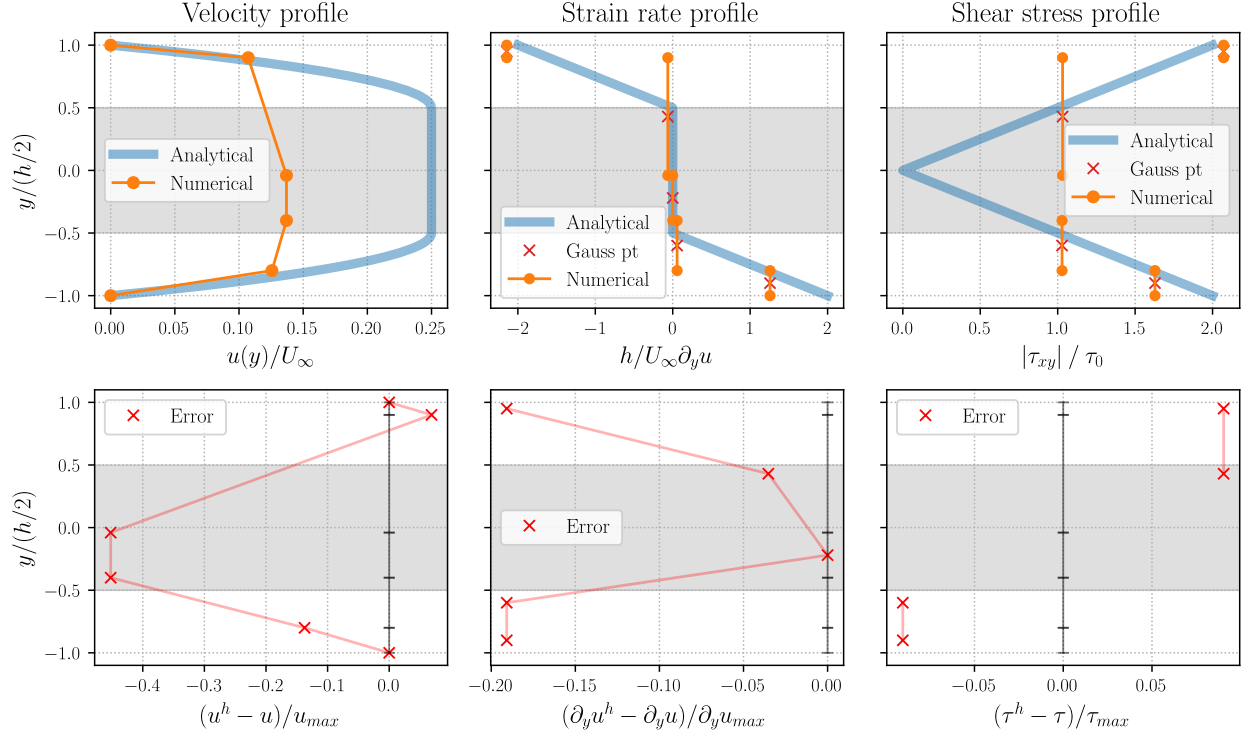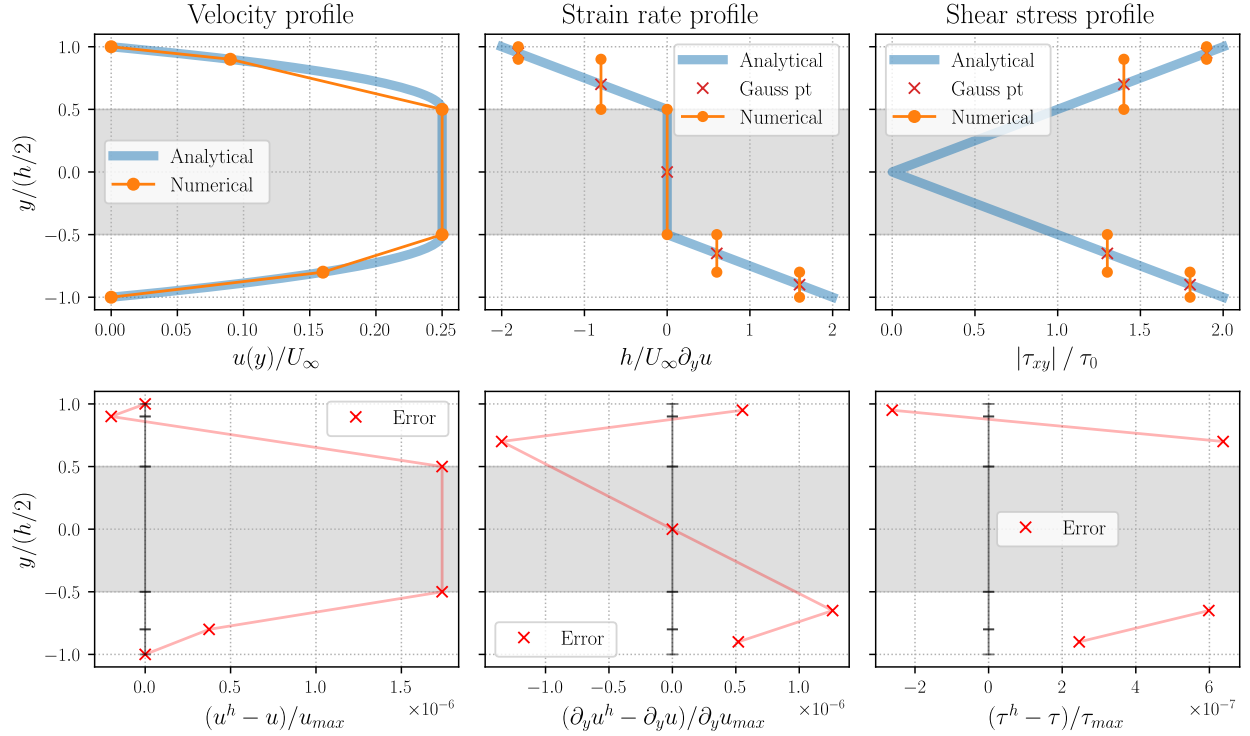**return** U_j

---

Figure 2.6. Strain field and its reconstruction during the interface-tracking algorithm. First to second iteration above, and second to third iteration below. Mesh with 5 $\mathcal{P}_1$-elements.

(a) Initial mesh.

(b) Final mesh.

Figure 2.7. Finite element solution of the Poiseuille flow at the start, and after convergence of the algorithm. $\mathcal{P}_1$ elements.

Although the initial velocity profile was far from correct, two iterations were enough to bring the nodes to the interface, and provide an accurate numerical solution with errors around 10^{-6}.

Surprisingly the convergence is faster for $\mathcal{P}_1$-elements (piecewise constant strain rate) than for $\mathcal{P}_2$-elements (piecewise linear strain rate). This can be easily explained by a flaw in algorithm 1: the reconstruction is made over all the Gauss points of elements near the current interface. However, one of these gauss point can be inside the unyielded region, as can be seen in figure 2.8. In that case, the reconstruction is deteriorated because it uses information from the wrong side, leading to a slower convergence.
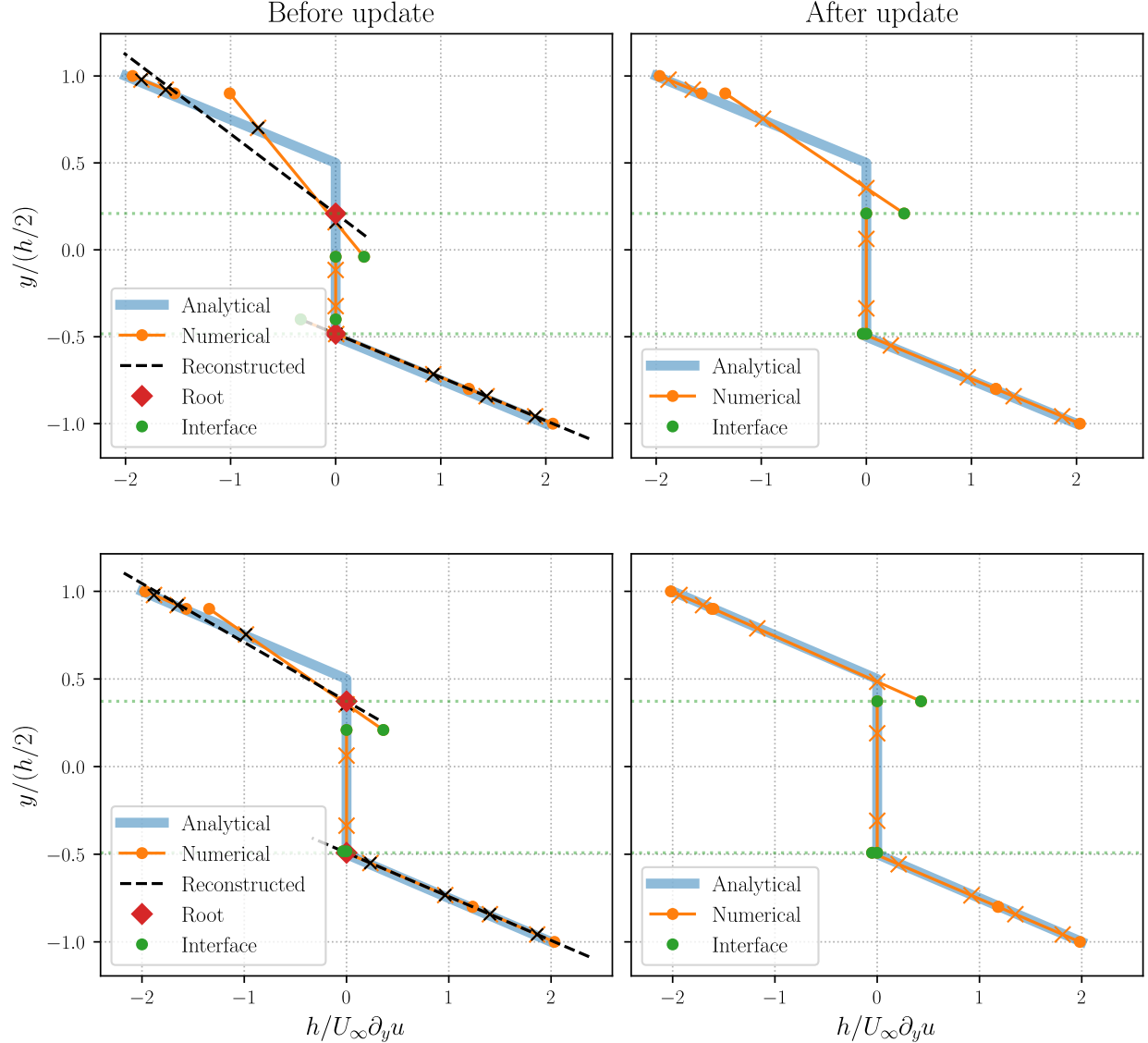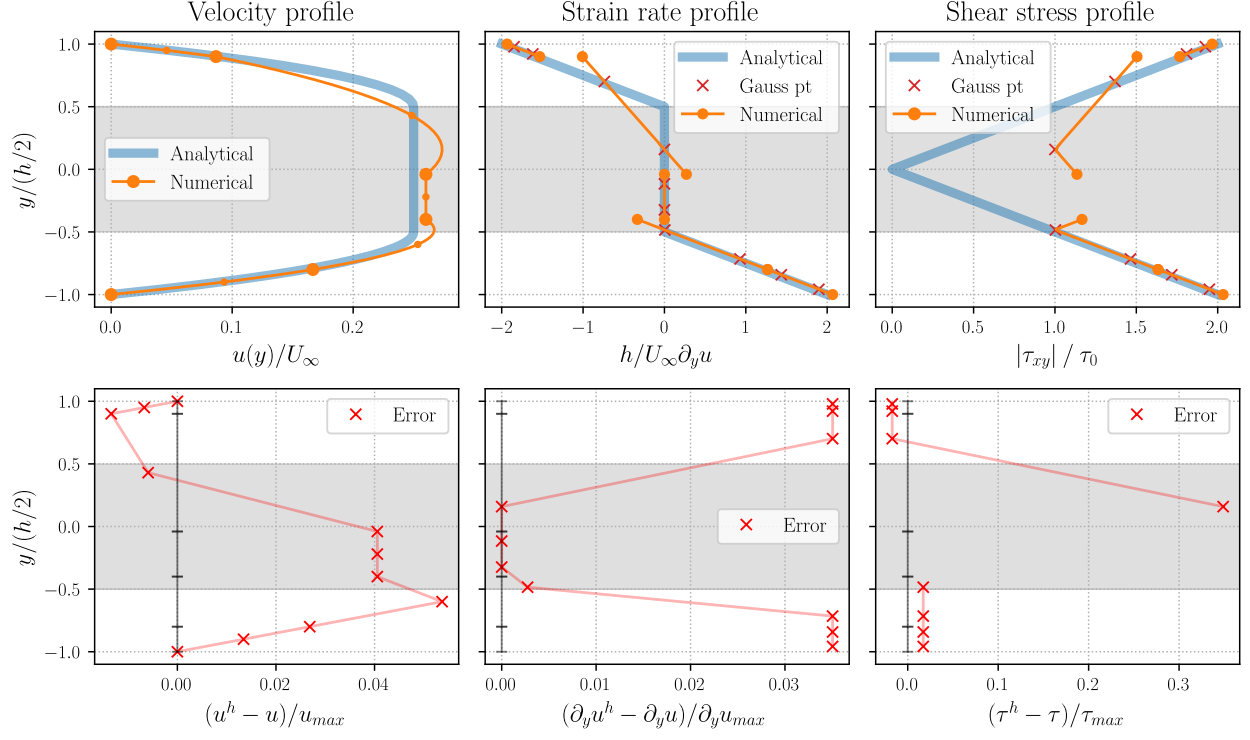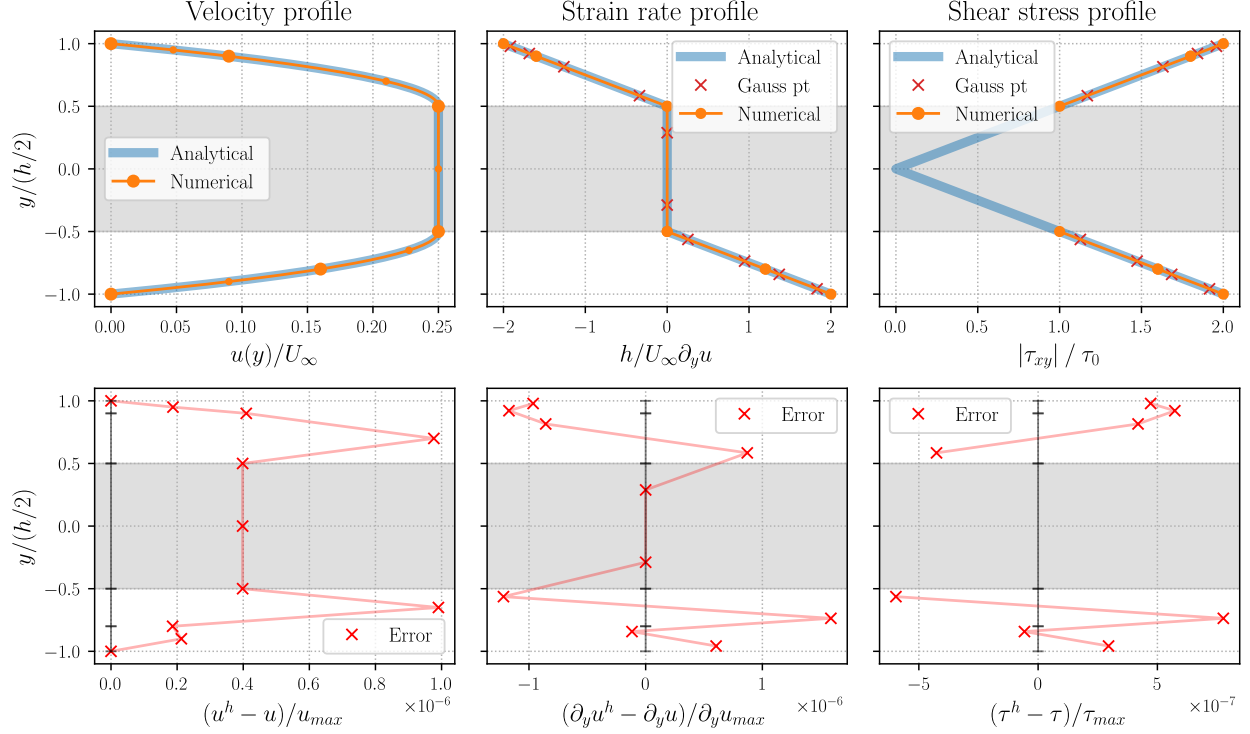

Figure 2.8. Strain field and its reconstruction during the interface-tracking algorithm. First to second iteration above, and second to third iteration below. $\mathcal{P}_1$ elements.

(a) Initial mesh.

(b) Final mesh.

Figure 2.9. Finite element solution of the Poiseuille flow at the start, and after convergence of the algorithm. $\mathcal{P}_1$ elements.

2.6.2 Energy-based tracking

An alternative approach was considered for the interface tracking. It consists in minimizing the functional not only over the nodal values, but also over the position of the nodes. Two difficulties then arose. The first one is that the optimization problem becomes non-convex, because of the body force term $fu^h(y_g)$.

$$\underset{U_j, y_i}{\text{minimize}} \sum_i \sum_g -f \sum_{j=1}^n U_j \phi_j(y_g) \frac{\Delta y_i}{2} \tag{2.20}$$

where the product $U_j(y_i - y_{i-1})$ is obviously not convex in the variables U_j, y_i, y_{i-1} since U_j can be positive or negative. This issue could be overcome by first solving the optimization problem with the velocities as variables, and then fix the velocities while minimizing over the nodal positions. This problem is in fact convex because the three terms of (2.18) are convex in y_i:

$$\left(\frac{\partial u^h}{\partial y} \right)^2 \Delta y_i = \left(\sum_j U_j \frac{\partial \phi}{\partial \eta} \frac{2}{\Delta y_i} \right)^2 \Delta y_i \qquad \text{convex when } \Delta y_i > 0 \tag{2.21}$$

$$\left| \frac{\partial u^h}{\partial y} \right| \Delta y_i = \left| \sum_j U_j \frac{\partial \phi}{\partial \eta} \frac{\partial \eta}{\partial y} \right| \Delta y_i = 2 \left| \sum_j U_j \frac{\partial \phi}{\partial \eta} \right| \qquad \text{convex because constant in } y_i \tag{2.22}$$

$$-f \sum_j U_j \phi_j \Delta y_i \qquad \text{convex because linear in } y_i \tag{2.23}$$

The second difficulty is even greater: the optimum of this problem (fixing the nodal values or not) is not a mesh with nodes placed at the analytical interface: with P^1 elements, the energy functional can be better minimized by *wrongly* placing the nodes. An example is shown in figure 2.10 with a mesh of only four nodes.

With P^2 elements, this energy-based approach would provide the optimal piecewise quadratic velocity field over all possible meshes. Since, by chance, the analytical solution is quadratic, it would also be the solution of the finite element problem. Hence, the minimization would give the exact velocity field and the correct mesh with nodes at the interface. In that case, it would be impossible to further decrease the cost as the exact solution is optimal over all velocity fields in $H^1(\Omega)$. The approach is however not suitable in general where the exact velocity field is not necessarily part of the finite element space.
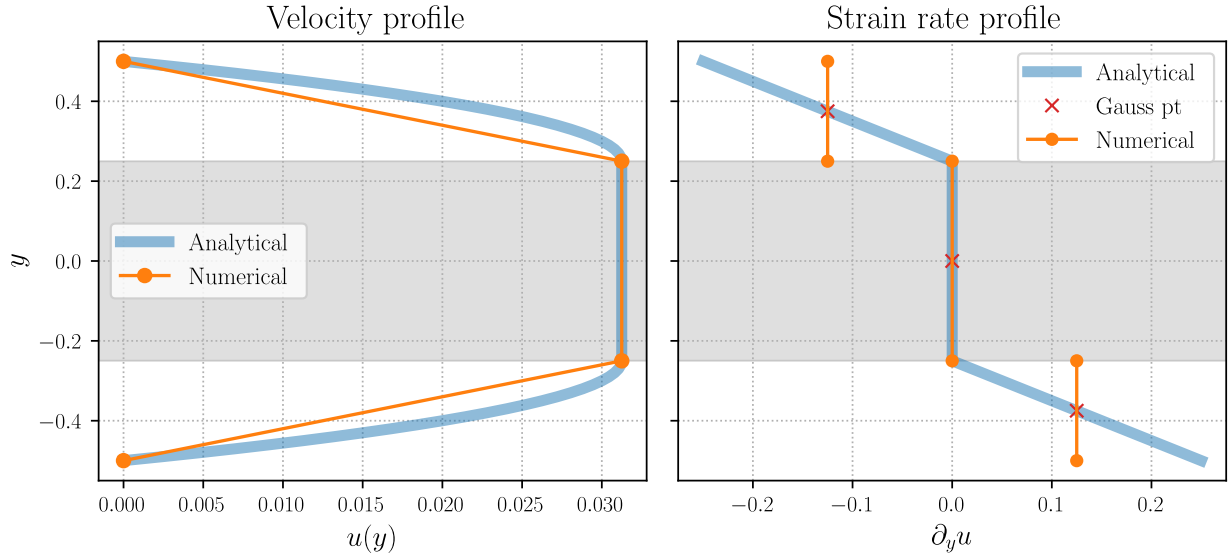
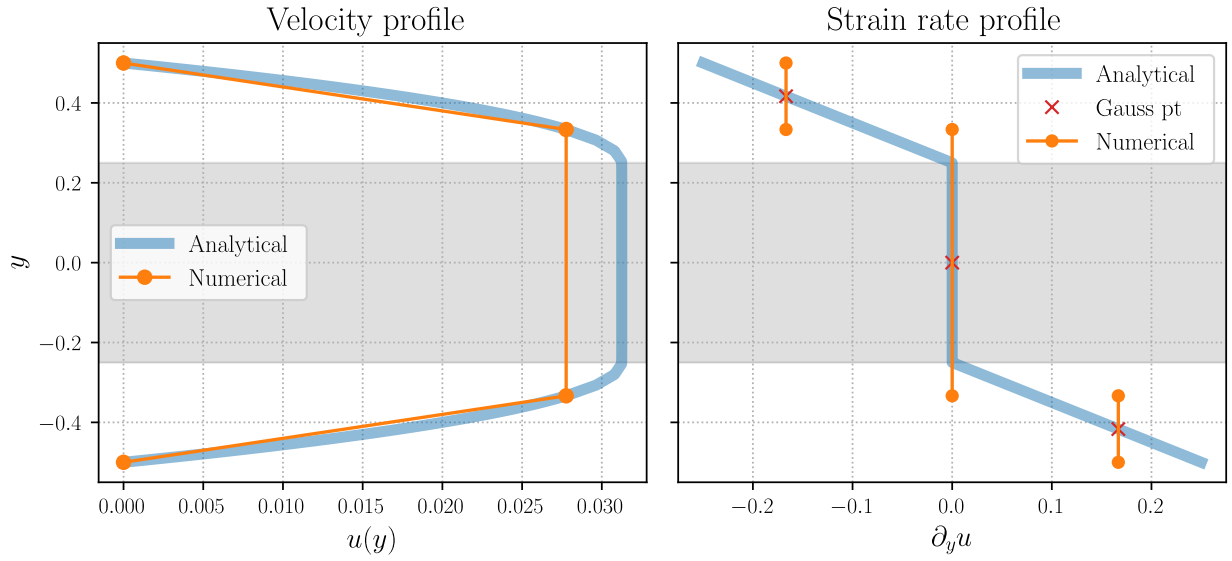**Figure 2.10.** Nodes at the interface: $\mathcal{J}(u^h) = \mathcal{J}(u^*) + 1.302 \times 10^{-3}$.

Figure 2.11. Optimal nodes: $\mathcal{J}(u^h) = \mathcal{J}(u^*) + 5.787 \times 10^{-4}$.

Chapter 3

2D problem

3.1 Finite element formulation

It is well known that velocity-pressure fields discretized with $\mathcal{P}_1$–$\mathcal{P}_0$ and $\mathcal{P}_1$–$\mathcal{P}_1$ elements do not provide stable discretizations of the Stokes equations, or equivalently of the incompressible linear elasticity equations. A first option is to enrich the velocity space with a bubble function: this is the MINI element [1]. A second option is to use Taylor-Hood elements, i.e. $\mathcal{P}_k$–$\mathcal{P}_{k-1}$ for $2 \leq k$. This element was implemented in the code for $k = 2$, along with the MINI element. The simulations presented here were done with Taylor-Hood elements unless otherwise specified.

The shape functions ϕ_j are represented graphically in figure 3.1. Again, the finite element approximation velocity field $\mathbf{v}^h$ is a sum of shape functions attached to each node j, weighted by the nodal values $\mathbf{V}_j$:

$$\mathbf{v}^h(x, y) = \sum_j \mathbf{V}_j \, \phi_j(x, y) \tag{3.1}$$

To express the deformation norm $\|\dot{\boldsymbol{\gamma}}\|$ needed in the functional, we need the velocity gradient:

$$\nabla \mathbf{v} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} \tag{3.2}$$

$$\implies \quad \|\dot{\boldsymbol{\gamma}}(u, v)\|^2_{\mathrm{Cart}} = 2 \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + 2 \left(\frac{\partial v}{\partial y} \right)^2 \tag{3.3}$$

Each of these partial derivatives is also computed as a sum of shape functions derivatives. These spatial derivatives in the physical domain (x, y) are themselves computed from derivatives in the parametric domain (ξ, η) through the Jacobian of the transformation.

$$\frac{\partial u^h_m}{\partial x_n} = \sum_j U_j \frac{\partial \phi_j}{\partial x_n} \qquad m = 1, 2 \quad n = 1, 2 \tag{3.4}$$

$$(\nabla_{\mathbf{x}} \, \phi)_{\mathrm{Cart}} = \nabla_{\boldsymbol{\xi}} \phi \, \cdot \frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{x}} \tag{3.5}$$

$$\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{x}} = \left(\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\boldsymbol{\xi}} \right)^{-1} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}^{-1} \tag{3.6}$$

(a) Lagrange $\mathcal{P}_1$

(b) Bubble enriched Lagrange

(c) Lagrange $\mathcal{P}_2$

Figure 3.1. Shape functions used for the pressure and velocity fields in 2-dimensions.

The functional of the approximate field $\mathbf{v}^h$ is recalled below, along with the *weak* incompressibility constraint, where ψ_l denotes the pressure linear shape function, attached to every primary node l.

$$\mathcal{J}(u^h, v^h) = \int_\Omega \frac{K}{2} \|\dot{\boldsymbol{\gamma}}(u^h, v^h)\|^2 + \tau_0 \|\dot{\boldsymbol{\gamma}}(u^h, v^h)\| - \mathbf{f} \cdot \mathbf{v}^h \, \mathrm{d}x \tag{3.7}$$

$$0 = \int_\Omega \psi_l \, \nabla \cdot \mathbf{v}^h \qquad \forall \quad \psi_l \tag{3.8}$$

It is also possible to impose the incompressibility constraint in a *strong* way, i.e. locally at each integration point[6]. Both options (3.9) and (3.8) were implemented in the code.

$$\frac{\partial u^h}{\partial x} + \frac{\partial v^h}{\partial y} = 0 \qquad \forall i \; \forall g \tag{3.9}$$

24

The minimization problem can for Cartesian coordinate systems is given hereunder:

$$
\begin{aligned}
\underset{\mathbf{V}_j, S_{i,g}, T_{i,g}}{\text{minimize}} \quad & \sum_i \sum_g \omega_g \left[\frac{K}{2} S_{i,g} + \tau_0 T_{i,g} - \mathbf{f} \cdot \mathbf{v}^h |_{\mathbf{x}_g} \right] \det \left(\frac{d\mathbf{x}}{d\boldsymbol{\xi}} \right)_{i,g} - \sum_e \sum_g \tilde{\omega}_g \, \mathbf{g} \cdot \mathbf{v}^h |_{\tilde{\mathbf{x}}_g} \frac{\ell_e}{2} \\
\text{s.t.} \quad & \left(S_{i,g}, \frac{1}{2}, \sqrt{2} \frac{\partial u^h}{\partial x}, \sqrt{2} \frac{\partial v^h}{\partial y}, \frac{\partial u^h}{\partial y} + \frac{\partial v^h}{\partial x} \right) \in L_R^5 \qquad \forall i, g \\
& \left(T_{i,g}, \sqrt{2} \frac{\partial u^h}{\partial x}, \sqrt{2} \frac{\partial v^h}{\partial y}, \frac{\partial u^h}{\partial y} + \frac{\partial v^h}{\partial x} \right) \in L^4 \qquad \forall i, g \\
& 0 = \sum_i \sum_g \omega_g \psi_l |_{\mathbf{x}_g} \left(\frac{\partial u^h}{\partial x} + \frac{\partial v^h}{\partial y} \right) \qquad \forall \, l \\
& \mathbf{V}_j = \mathbf{U} \qquad \forall \, \mathbf{V}_j \in \Gamma_D
\end{aligned}
$$

$$\tag{3.10}$$

where $\tilde{\omega}_g$, $\tilde{\mathbf{x}}_g$ are the integration weights and positions on the edges, while ℓ_e indicates its length.

3.2   Boundary conditions

Boundary conditions should never be overlooked, especially in the case of the Poiseuille flow.

No-slip walls are easily handled, with homogeneous Dirichlet boundary conditions. However, the situation is not as simple at the inflow and outflow. We can list the three most obvious choices:

1. Impose the velocity profile at the inflow (Dirichlet)

2. Impose the pressure at the inflow and at the outflow (Neumann)

3. Impose the pressure gradient over the domain Ω (Body force)

However, care must be taken because we are only providing the normal force on the boundary, due to the pressure gradient, but not the tangential force due to the shear stress that is not known yet. If we prescribe only the normal component of the traction $\mathbf{g} = (\hat{\mathbf{n}} \cdot \sigma \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$, we also need to prescribe the tangential component of the velocity $\mathbf{v} \cdot \hat{\mathbf{n}}^\perp$, as can be derived from the weak formulation equation (1.15) [3].

Each description must then be completed in order to obtain a correct physical flow.

1. At the inflow, impose the velocity profile with $\mathbf{v} \cdot \hat{\mathbf{n}} = v_n$ and $\mathbf{v} \cdot \hat{\mathbf{n}}^\perp = v_t = 0$. At the outflow, set zero tangential velocity $v_t = 0$, along with a normal force $\mathbf{g} = (-p + \hat{\mathbf{n}} \cdot \boldsymbol{\tau} \cdot \hat{\mathbf{n}}) \hat{\mathbf{n}} = -p_{\text{out}} \hat{\mathbf{n}}$. It is not very suitable for Bingham fluids because it requires the knowledge of the velocity profile beforehand.

2. Impose the normal boundary force at the inflow and outflow, made up of the pressure and normal shear stress: $\mathbf{g} \cdot \hat{\mathbf{n}} = -p_{\text{in/out}}$. Also set $v_t = 0$ at the two boundaries.

3. Set the pressure gradient as body force $\mathbf{f} = -\partial_x p$. It still requires to specify the normal shear stress and tangential velocity: $\mathbf{g} = 0 \, \hat{\mathbf{n}}$, and $v_t = 0$.

If one forgets to cancel the tangential velocity at the outflow, the problem becomes equivalent to impose zero tangential shear stress at the outflow, which is of course not the situation of a Poiseuille flow where $\tau_{12} = K \partial_y u \neq 0$.

I ran a first set of simulations with the correct setup, imposing $v_t = 0$, and a second one with the incorrect setup where the condition $v_t = 0$ was intentionally forgotten, both without yield stress ($\tau_0 = 0$). These two situations are compared in figure 3.2. When no condition is applied on the tangential outflow velocity,the finite element solution gives a very high pressure gradient near the outflow corners in order to achieve zero tangential shear force. Also note that the streamlines are curved at the outflow, towards the corners.
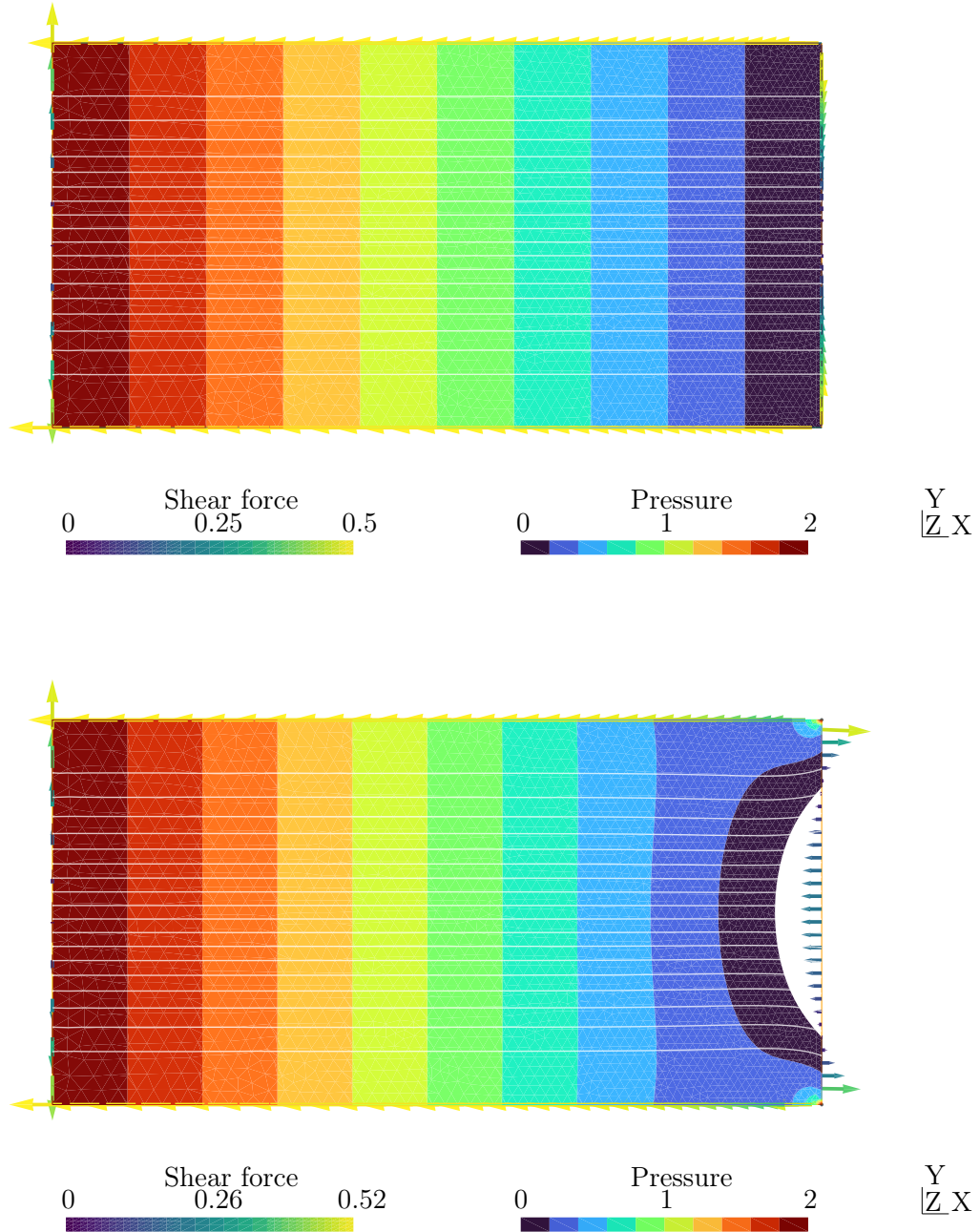

Figure 3.2. Comparison of the pressure field, streamlines and boundary forces of the correct (top) and incorrect (bottom) flows.

One can also wonder if the choice of boundary condition (1-2-3), or the choice of incompressibility condition (weak-strong) influences quality of the solution. In the case of a flow in a channel of 2 meters long and 1 meter wide, the exact energy functional is $-1/12$. The deviation of the numerical functional from the exact functional is given in table 3.1 for different setups.

Incompressibility	Dirichlet/Neumann Body force	Correct setup $v_t = 0$	Incorrect setup v_t free
Weak	Dirichlet (1)	6.095×10^{-10}	-2.077×10^{-3}
	Neumann (2)	2.201×10^{-9}	-2.130×10^{-3}
	Body force (3)	1.858×10^{-11}	-2.130×10^{-3}
Strong	Dirichlet (1)	7.603×10^{-10}	-2.037×10^{-3}
	Neumann (2)	8.898×10^{-10}	-2.088×10^{-3}
	Body Force (3)	5.483×10^{-10}	-2.088×10^{-3}

Table 3.1. Deviation from the analytical functional: $\mathcal{J}(u^h) - \mathcal{J}(u)$.

3.3 Poiseuille flow

3.4 Curved channel flow

3.5 Lid-driven cavity

3.6 Flow around cylinder

Conclusion

A conclusion

Bibliography

[1] D. N. Arnold, F. Brezzi, and M. Fortin, *A stable finite element for the stokes equations*, Calcolo, 21 (1984), pp. 337–344.

[2] N. Balmforth and R. Craster, *Geophysical Aspects of Non-Newtonian Fluid Mechanics*, vol. 582, Springer, 01 2008, pp. 34–51, `https://doi.org/10.1007/3-540-45670-8_2`.

[3] W. Bangerth, *Finite element methods in scientific computing: Boundary conditions*, 2014, `https://www.math.colostate.edu/~bangerth/videos/676/slides.21.5.pdf`.

[4] J. Barlow, *Optimal stress locations in finite element models*, International Journal for Numerical Methods in Engineering, 10 (1976), pp. 243–251, `https://doi.org/10.1002/nme.1620100202`.

[5] R. Bird, R. C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids, Volume 1: Fluid Mechanics*, Dynamics of Polymeric Liquids, Wiley, 1987, p. 170.

[6] J. Bleyer, M. Maillard, P. de Buhan, and P. Coussot, *Efficient numerical computations of yield stress fluid flows using second-order cone programming*, Computer Methods in Applied Mechanics and Engineering, 283 (2015), pp. 599–614, `https://doi.org/10.1016/j.cma.2014.10.008`.

[7] P. Bottiglieri, F. De Sio, G. Fasanaro, G. Mojoli, M. Impembo, and D. Castaldo, *Rheological characterization of ketchup*, Journal of Food Quality, 14 (1991), p. 507, `https://doi.org/10.1111/j.1745-4557.1991.tb00089.x`.

[8] I.-L. Chern, J. Glimm, O. McBryan, B. Plohr, and S. Yaniv, *Front tracking for gas dynamics*, Journal of Computational Physics, 62 (1986), pp. 83–110.

[9] P. Coussot, *Yield stress fluid flows: A review of experimental data*, Journal of Non-Newtonian Fluid Mechanics, 211 (2014), pp. 31–49, `https://doi.org/10.1016/j.jnnfm.2014.05.006`.

[10] F. De Vito, F. Meyer, and S. Fritz, *Investigating the effect of fat content on the yield stress of mayonnaise*, tech. report, Thermo Fisher Scientific, 2022, `https://assets.thermofisher.com/TFS-Assets/MSD/Application-Notes/V271-investigation-effect-fat-content-yield-stress-mayonnaise.pdf`.

[11] I. Frigaard, *Simple yield stress fluids*, Current Opinion in Colloid & Interface Science, 43 (2019), p. 1, `https://doi.org/10.1016/j.cocis.2019.03.002`.

[12] D. Izidoro, M. Sierakowski, N. Waszczynskyj, C. Haminiuk, and A. Scheer, *Sensory evaluation and rheological behavior of commercial mayonnaise*, International Journal of Food Engineering, 3 (2007), p. 10, `https://doi.org/10.2202/1556-3758.1094`.

[13] G. Katsaros, M. Tsoukala, M. Giannoglou, and P. Taoukis, *Effect of storage on the rheological and viscoelastic properties of mayonnaise emulsions of different oil droplet size*, Heliyon, 6 (2020), p. 3, `https://doi.org/10.1016/j.heliyon.2020.e05788`.

[14] C. Küchenmeister and J. P. Plog, *How much squeezing power is required to get the toothpaste out of the tube?*, tech. report, Thermo Fisher Scientific, 2014, `https://assets.thermofisher.com/TFS-Assets/CAD/Application-Notes/V-275-e-How-much-squeezing-power-is-required-to-get-the-tooth-paste-out-of-the-tube.pdf`.

[15] B.-K. Lee, S. Xue, J. Nam, H. Lim, and S. Shin, *Determination of the blood viscosity and yield stress with a pressure-scanning capillary hemorheometer using constitutive models*, Korea-Australia Rheology Journal, 23 (2011), p. 5, `https://doi.org/10.1007/s13367-011-0001-y`.

[16] NETZSCH-Gerätebau, *Determination of yield stress of complex fluids by model fitting using a rotational rheometer*, tech. report, NETZSCH-Gerätebau, 2022, `https://analyzing-testing.netzsch.com/_Resources/Persistent/5/1/6/5/51659abc829511ca44f93d7159ed1c74a3274975/AN145_~1.pdf`.

[17] P. Saramito, *Complex fluids : modeling and algorithms*, Springer, 2016.

[18] D. She, R. Kaufman, H. Lim, J. Melvin, A. Hsu, and J. Glimm, *Chapter 15 - front-tracking methods*, in Handbook of Numerical Methods for Hyperbolic Problems, R. Abgrall and C.-W. Shu, eds., vol. 17 of Handbook of Numerical Analysis, Elsevier, 2016, p. 383, `https://doi.org/10.1016/bs.hna.2016.07.004`.

[19] E. Talansier, A. Bacconnier, F. Caton, C. Chastel, L. Costa, D. Gunes, and D. Roux, *Accurate methodology to determine slip velocity, yield stress and the constitutive relation for molten chocolate*, Journal of Food Engineering, 244 (2018), p. 7, `https://doi.org/10.1016/j.jfoodeng.2018.09.031`.

[20] O. C. Zienkiewicz and J. Z. Zhu, *The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique*, International Journal for Numerical Methods in Engineering, 33 (1992), pp. 1331–1364, `https://doi.org/10.1002/nme.1620330702`.

Appendix A

Curvilinear coordinates

In axisymmetric cylindrical coordinates, the solution only depends on the radial and axial coordinates r, z. We also assume that the velocity field has no tangential component.

$$\begin{bmatrix} x(r, \theta = 0, z) \\ y(r, \theta = 0, z) \\ z(r, \theta = 0, z) \end{bmatrix} = \begin{bmatrix} r \\ 0 \\ z \end{bmatrix} \tag{A.1}$$

$$\mathbf{u}(r, z) = u_r(r, z)\mathbf{e}_r + u_z(r, z)\mathbf{e}_z \tag{A.2}$$

$$(\nabla\, \phi)_{\text{Cyl}} = \nabla_{\boldsymbol{\xi}}\phi \,\cdot \frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{x}} \cdot \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}(r, z)} \tag{A.3}$$

$$\nabla \cdot \mathbf{v} = \frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial u_z}{\partial z} \tag{A.4}$$

$$\|\dot{\boldsymbol{\gamma}}(u, v)\|^2_{\text{Cyl}} = 2\left(\frac{\partial u_r}{\partial r}\right)^2 + 2\left(\frac{u_r}{r}\right)^2 + 2\left(\frac{\partial u_z}{\partial r}\right)^2 + \left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right)^2 \tag{A.5}$$

Polar coordinates provide quite similar equations. Here again, we assume axisymmetry and the absence of azimuthal velocity.

$$\begin{bmatrix} x(r, \theta, \varphi = 0) \\ y(r, \theta, \varphi = 0) \\ z(r, \theta, \varphi = 0) \end{bmatrix} = \begin{bmatrix} r\sin\theta \\ 0 \\ r\cos\theta \end{bmatrix} \tag{A.6}$$

$$\mathbf{u}(r, z) = u_r(r, \theta)\mathbf{e}_r + u_\theta(r, \theta)\mathbf{e}_\theta \tag{A.7}$$

$$(\nabla\, \phi)_{\text{Pol}} = \nabla_{\boldsymbol{\xi}}\phi \,\cdot \frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\mathbf{x}} \cdot \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}(r, \theta)} \tag{A.8}$$

$$\nabla \cdot \mathbf{v} = \frac{1}{r^2}\frac{\partial(r^2 u_r)}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(u_\theta \sin\theta)}{\partial\theta} \tag{A.9}$$

$$\|\dot{\boldsymbol{\gamma}}(u, v)\|^2_{\text{Cyl}} = 2\left(\frac{\partial u_r}{\partial r}\right)^2 + 2\left(\frac{1}{r}\frac{\partial u_\theta}{\partial\theta} + \frac{u_r}{r}\right)^2 + 2\left(\frac{u_r}{r} + \cot\theta\frac{u_\theta}{r}\right)^2$$

$$+ \left(\frac{1}{r}\frac{\partial u_r}{\partial\theta}\frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r}\right)^2 \tag{A.10}$$