

Numerical simulation of yield stress fluid flows

Interface tracking instead of mesh refinement

Vincent Degrooff

Promotor: Jean-François Remacle

Master's thesis



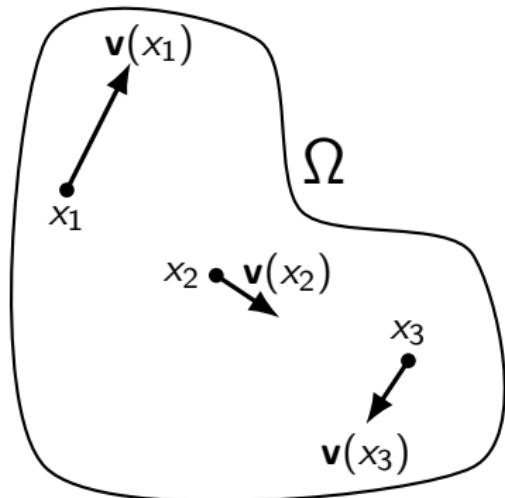
Table of Contents

- 1 The context: what is a yield stress fluid ?
- 2 The motivation: improve flow simulations with interface tracking
- 3 The method: interface tracking algorithm
- 4 Numerical results
- 5 Future work

Table of Contents

- 1 The context: what is a yield stress fluid ?
- 2 The motivation: improve flow simulations with interface tracking
- 3 The method: interface tracking algorithm
- 4 Numerical results
- 5 Future work

- ▶ Domain Ω
- ▶ Position vector $\mathbf{x} \in \Omega$
- ▶ Velocity field $\mathbf{v}(\mathbf{x})$
- ▶ Fluid undergoes stress
$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}$$
- ▶ Need to relate $\boldsymbol{\tau}$ to \mathbf{v}



Velocity modes

$$\mathbf{v}(\mathbf{x}_0 + \boldsymbol{\delta}) = \mathbf{v}(\mathbf{x}_0) + (\nabla \mathbf{v}) \cdot \boldsymbol{\delta} \\ + \text{H.O.T.}$$

$$\nabla \mathbf{v} = \mathbf{W} + \mathbf{D}_s + \mathbf{D}_d$$

$$\tau = \mu 2\mathbf{D}_d \quad \text{Newton}$$

Velocity modes

$$\mathbf{v}(\mathbf{x}_0 + \boldsymbol{\delta}) = \mathbf{v}(\mathbf{x}_0) + (\nabla \mathbf{v}) \cdot \boldsymbol{\delta} \\ + \text{H.O.T.}$$

$$\nabla \mathbf{v} = \mathbf{W} + \mathbf{D}_s + \mathbf{D}_d$$

$$\tau = \mu 2\mathbf{D}_d \quad \text{Newton}$$

Velocity modes

$$\mathbf{v}(\mathbf{x}_0 + \boldsymbol{\delta}) = \mathbf{v}(\mathbf{x}_0) + (\nabla \mathbf{v}) \cdot \boldsymbol{\delta} \\ + \text{H.O.T.}$$

$$\nabla \mathbf{v} = \mathbf{W} + \mathbf{D}_s + \mathbf{D}_d$$

$$\tau = \mu 2\mathbf{D}_d \quad \text{Newton}$$

Velocity modes

$$\mathbf{v}(\mathbf{x}_0 + \boldsymbol{\delta}) = \mathbf{v}(\mathbf{x}_0) + (\nabla \mathbf{v}) \cdot \boldsymbol{\delta} \\ + \text{H.O.T.}$$

$$\nabla \mathbf{v} = \mathbf{W} + \mathbf{D}_s + \mathbf{D}_d$$

$$\tau = \mu 2\mathbf{D}_d \quad \text{Newton}$$

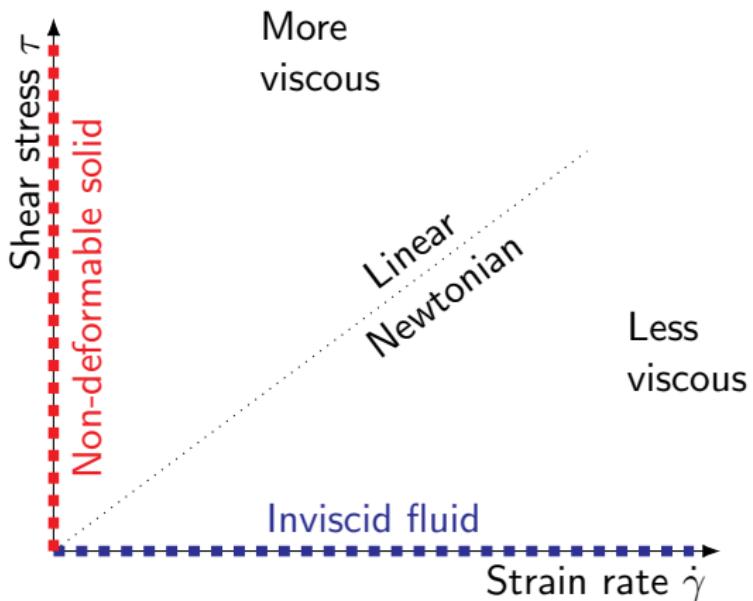
Velocity modes

$$\mathbf{v}(\mathbf{x}_0 + \boldsymbol{\delta}) = \mathbf{v}(\mathbf{x}_0) + (\nabla \mathbf{v}) \cdot \boldsymbol{\delta} \\ + \text{H.O.T.}$$

$$\nabla \mathbf{v} = \mathbf{W} + \mathbf{D}_s + \mathbf{D}_d$$

$$\tau = \mu 2\mathbf{D}_d \quad \text{Newton}$$

Flow curve



Non-Newtonian fluids

- Generalized Newtonian model

$$\boldsymbol{\tau} = \mu(\dot{\gamma}, T) \dot{\gamma} \quad \text{where}$$

$$\begin{cases} \dot{\gamma} = 2\mathbf{D} = \nabla \mathbf{v} + \nabla \mathbf{v}^T \\ \dot{\gamma} = \|\dot{\gamma}\|_F = \sqrt{\text{Tr}(\dot{\gamma} \cdot \dot{\gamma})} \end{cases}$$

Non-Newtonian fluids

- Generalized Newtonian model

$$\boldsymbol{\tau} = \mu(\dot{\gamma}, T) \dot{\gamma} \quad \text{where} \quad \begin{cases} \dot{\gamma} = 2\mathbf{D} = \nabla \mathbf{v} + \nabla \mathbf{v}^T \\ \dot{\gamma} = \|\dot{\gamma}\|_F = \sqrt{\text{Tr}(\dot{\gamma} \cdot \dot{\gamma})} \end{cases}$$

- Power-law model

$$\boldsymbol{\tau} = (K \dot{\gamma}^n) \dot{\gamma}$$

Non-Newtonian fluids

- Generalized Newtonian model

$$\tau = \mu(\dot{\gamma}, T) \dot{\gamma} \quad \text{where} \quad \begin{cases} \dot{\gamma} = 2\mathbf{D} = \nabla \mathbf{v} + \nabla \mathbf{v}^T \\ \dot{\gamma} = \|\dot{\gamma}\|_F = \sqrt{\text{Tr}(\dot{\gamma} \cdot \dot{\gamma})} \end{cases}$$

- Power-law model

$$\tau = (K \dot{\gamma}^n) \dot{\gamma}$$

- Herschel-Bulkley model, for yield stress fluids (*fluides à seuil* in French):

$$\dot{\gamma} = 0 \quad \text{if } \tau < \tau_0$$

$$\tau = \left(K \dot{\gamma}^{n-1} + \frac{\tau_0}{\dot{\gamma}} \right) \dot{\gamma} \quad \text{if } \tau \geq \tau_0$$

Non-Newtonian fluids

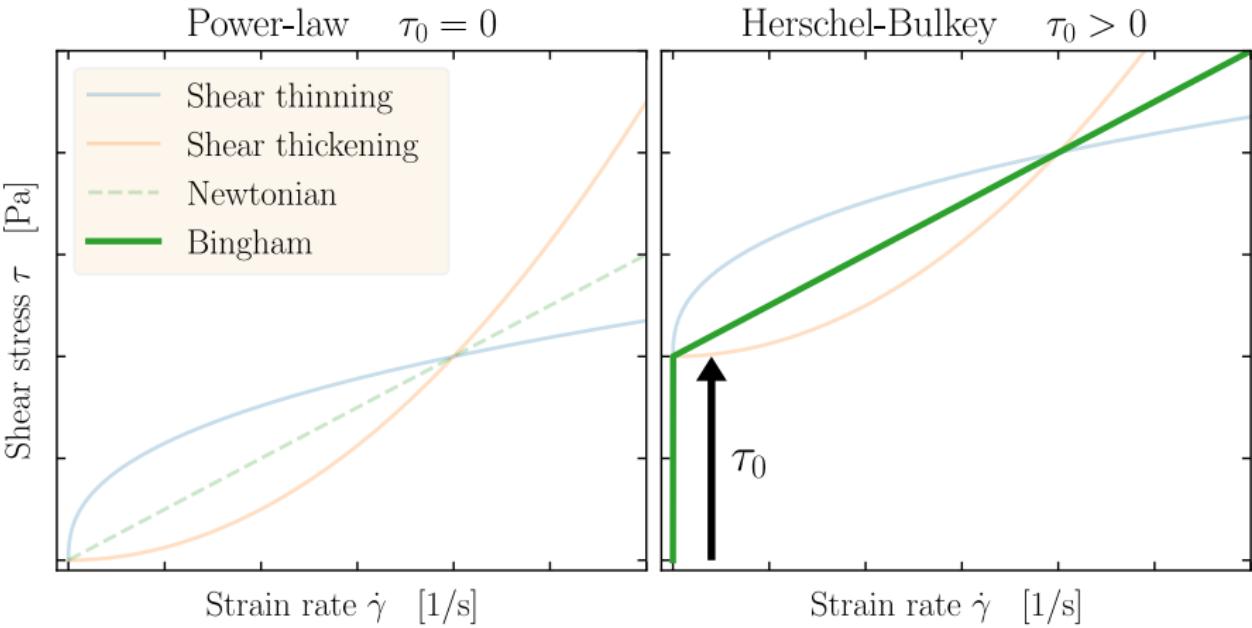
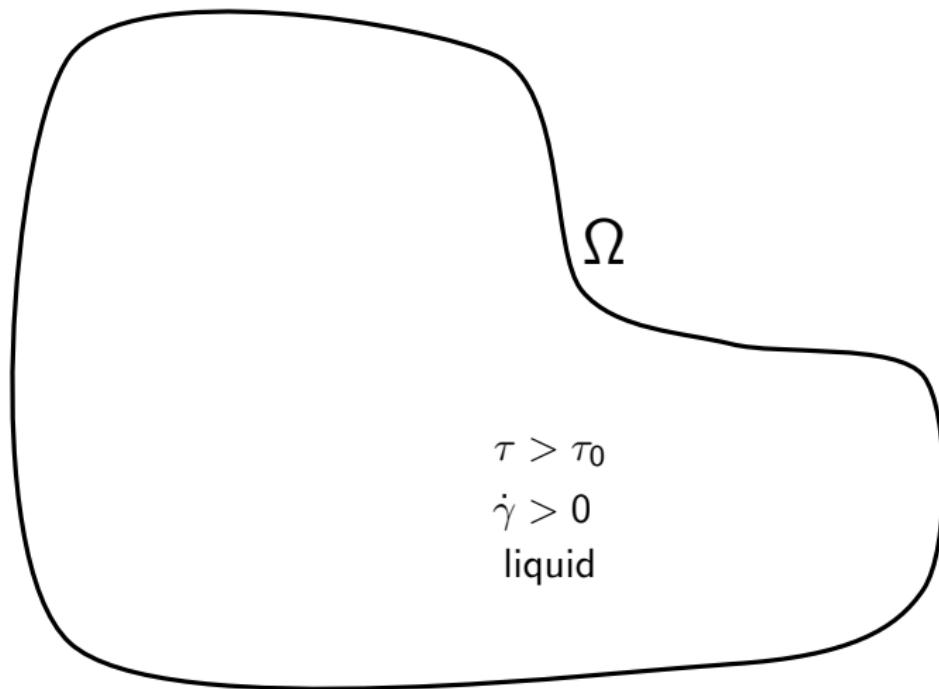


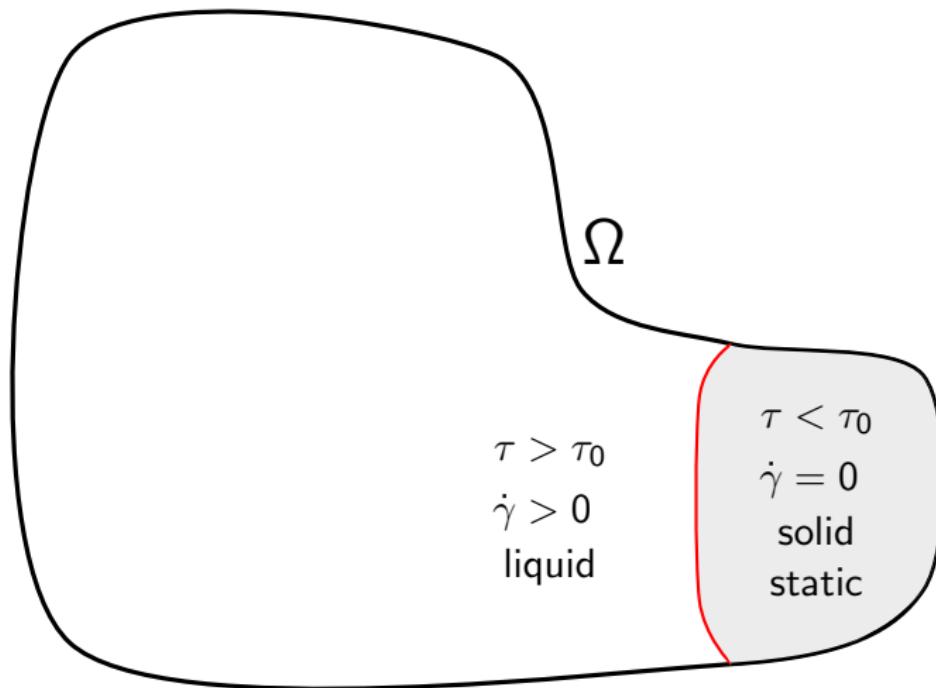
Table of Contents

- 1 The context: what is a yield stress fluid ?
- 2 The motivation: improve flow simulations with interface tracking
- 3 The method: interface tracking algorithm
- 4 Numerical results
- 5 Future work

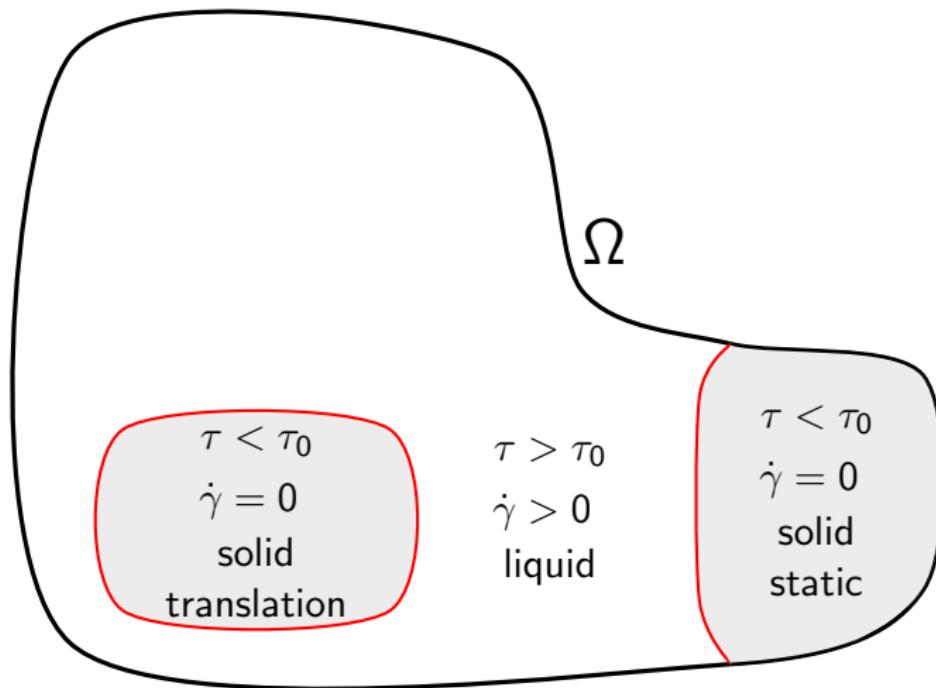
Solid and liquid subdomains



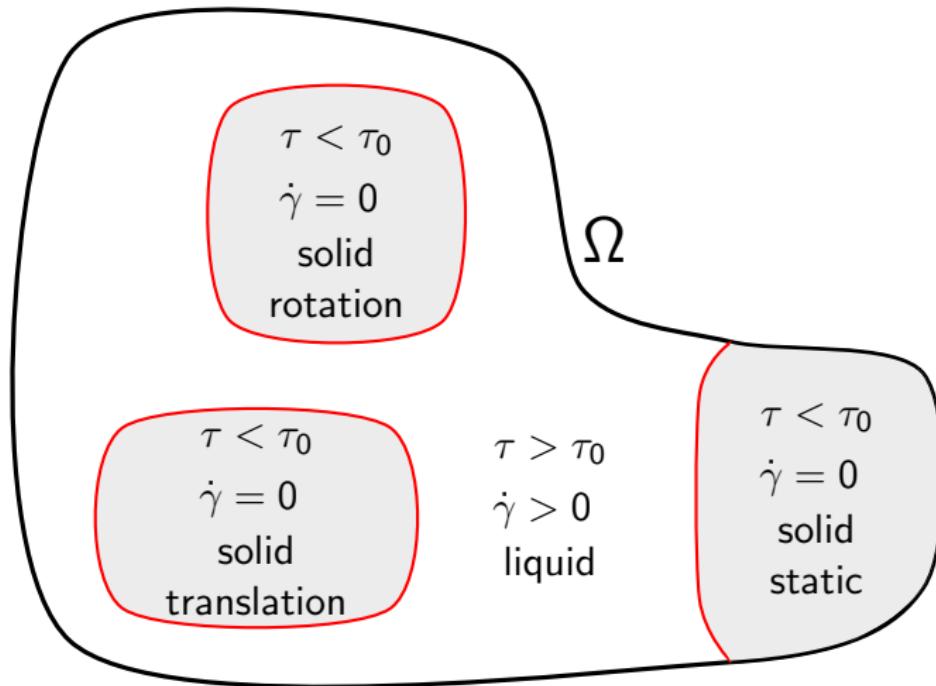
Solid and liquid subdomains



Solid and liquid subdomains



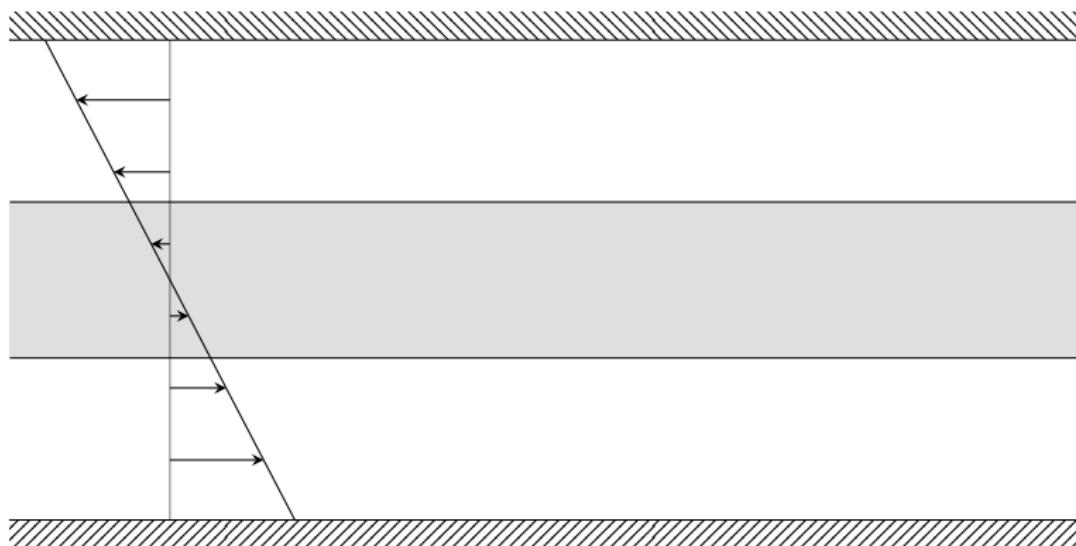
Solid and liquid subdomains



Improve the accuracy near the interfaces

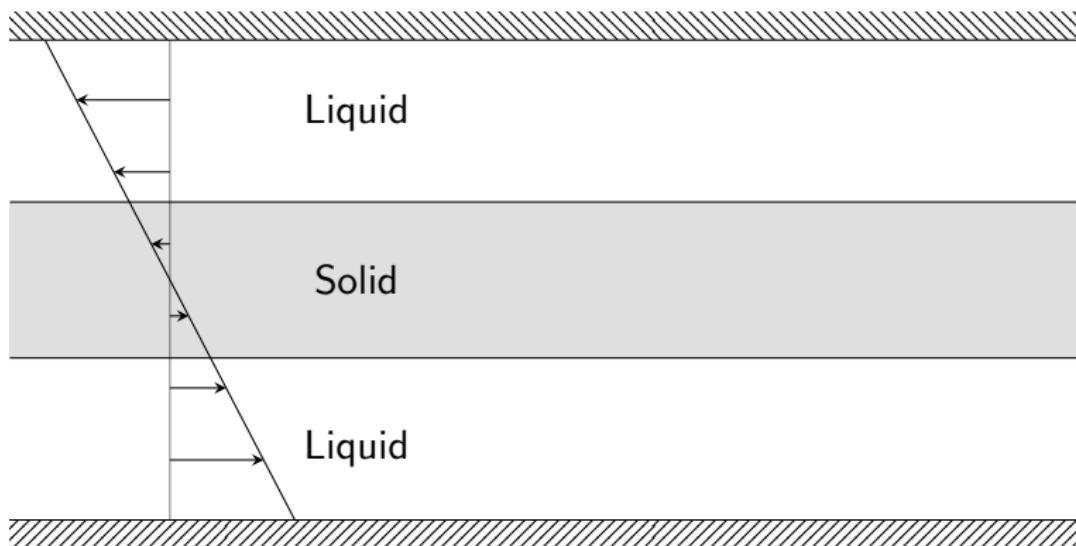
- ▶ Mesh refinement
 - ▶ easier to implement
 - ▶ higher computational cost
- ▶ Interface tracking
 - ▶ harder to implement
 - ▶ lower computational cost

Interface tracking benefits - Poiseuille flow

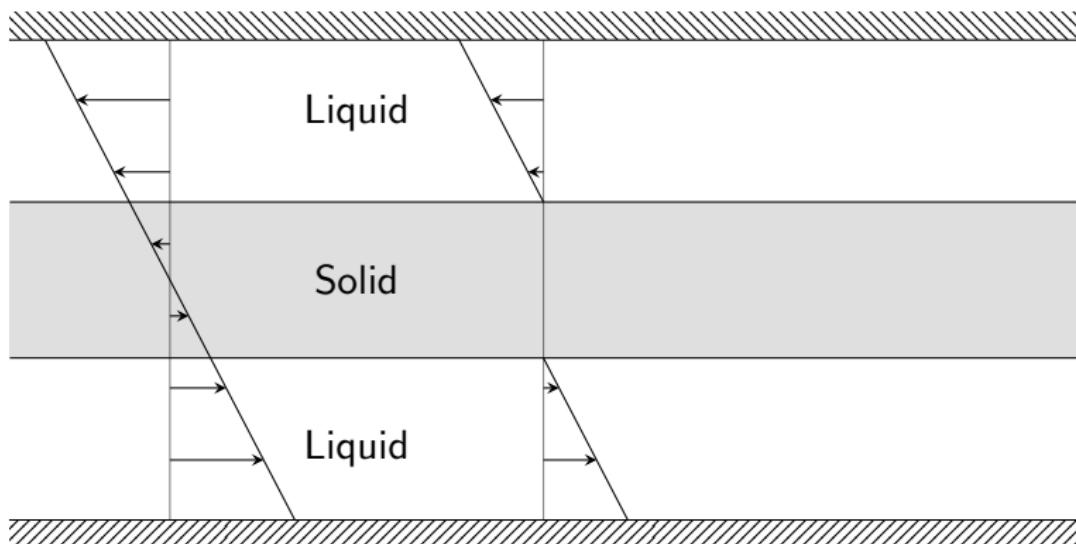


Shear stress

Interface tracking benefits - Poiseuille flow



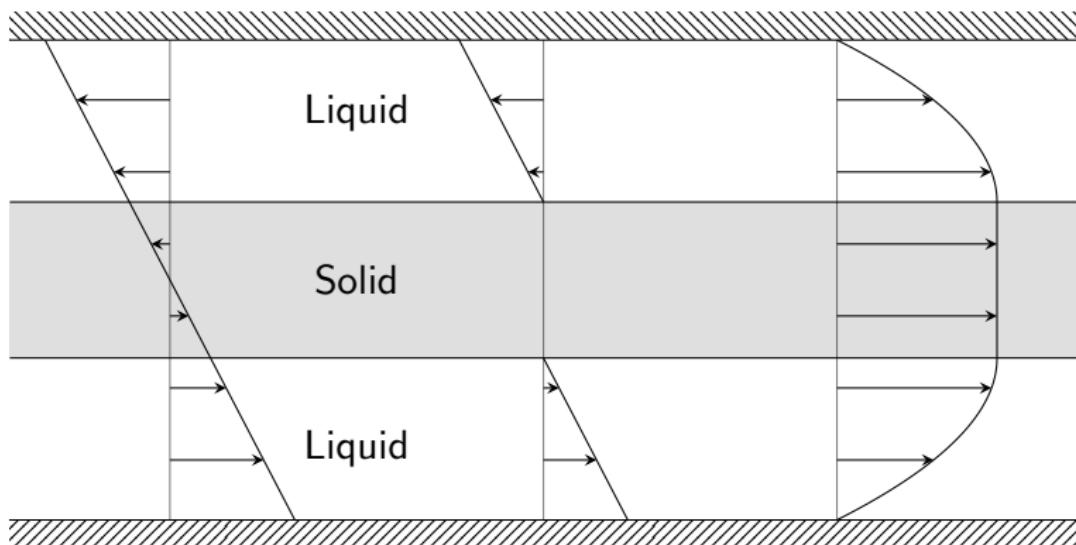
Interface tracking benefits - Poiseuille flow



Shear stress

Strain rate

Interface tracking benefits - Poiseuille flow

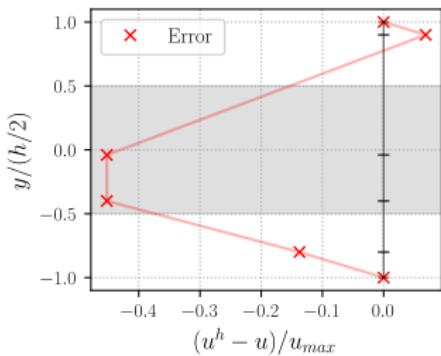
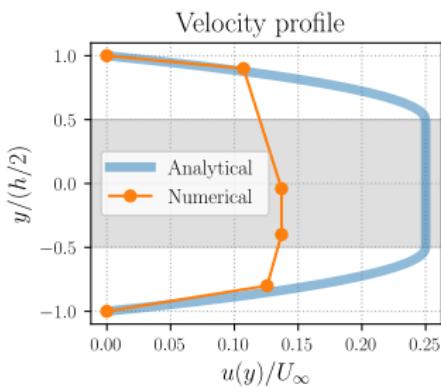


Shear stress

Strain rate

Velocity

Interface tracking benefits - Poiseuille flow



Interface tracking benefits - Poiseuille flow

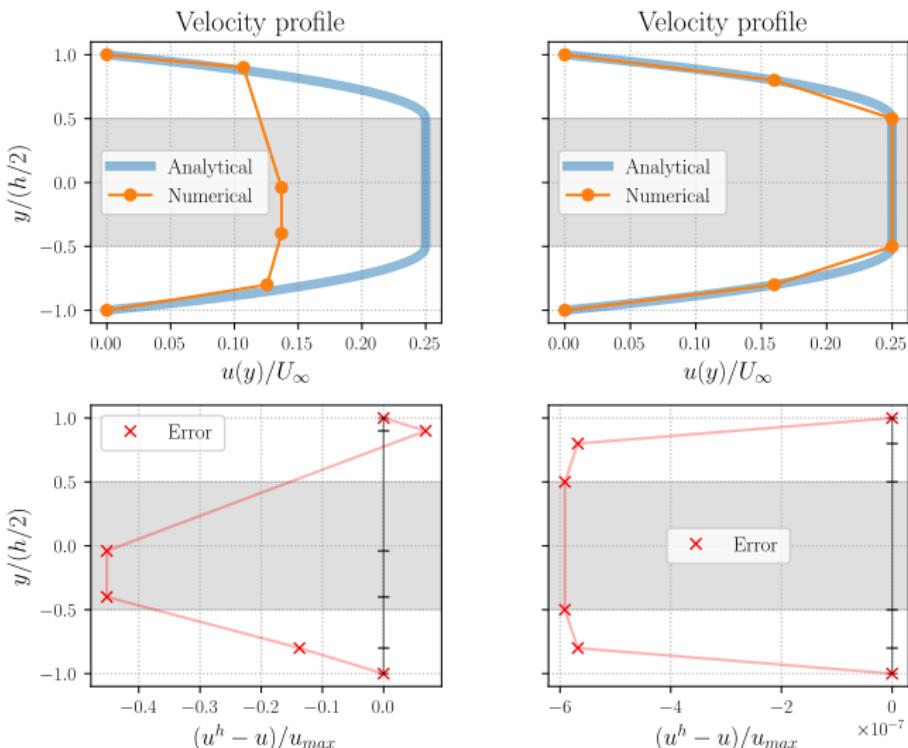


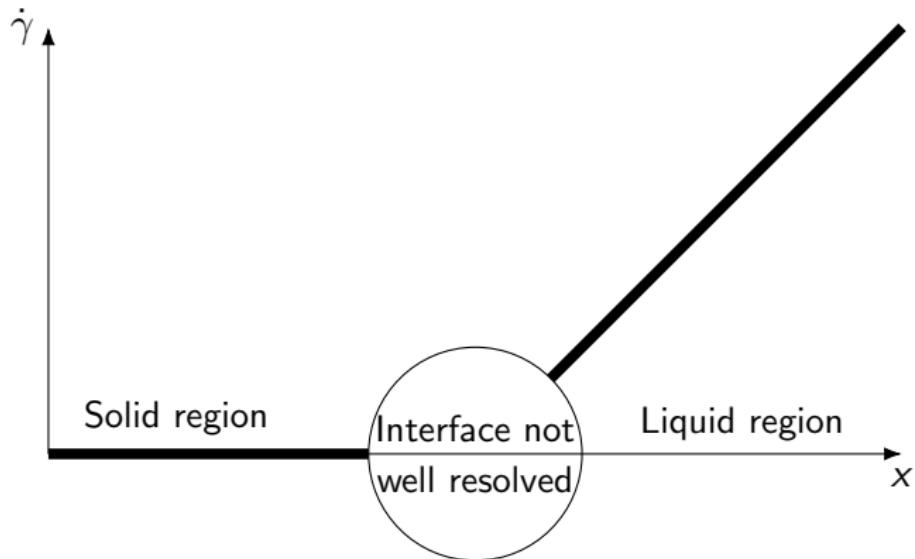
Table of Contents

- 1 The context: what is a yield stress fluid ?
- 2 The motivation: improve flow simulations with interface tracking
- 3 The method: interface tracking algorithm
- 4 Numerical results
- 5 Future work

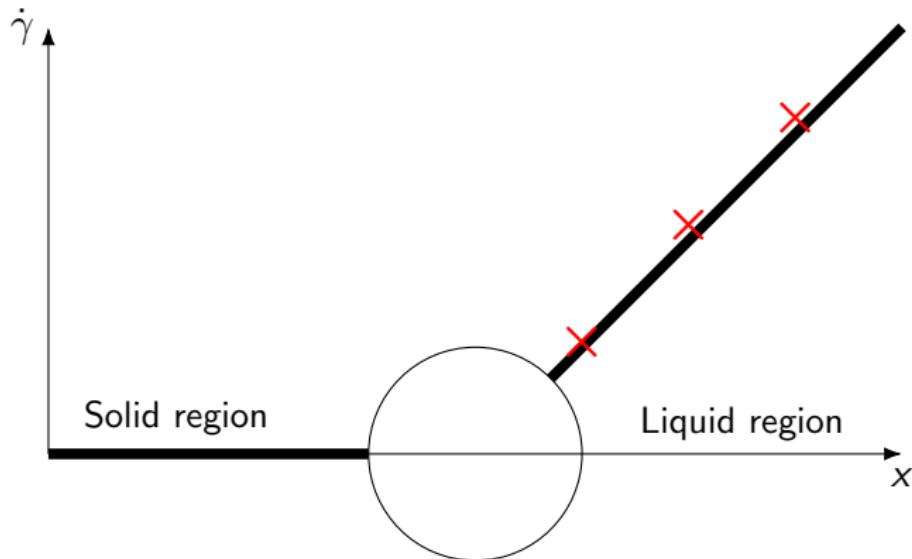
Procedure

- 1 Obtain the flow \mathbf{v} with a finite element method
 - ▶ Solution of an optimization problem, obtained with an interior-point method
- 2 Locate interface(s) based on the strain rate field $\dot{\gamma}(\mathbf{v})$
 - ▶ Not trivial: not a simple root of $\dot{\gamma}^h > 0$, and $\dot{\gamma}^h$ is discontinuous
- 3 Deform the mesh to match the physical interface
 - ▶ With the X-MESH algorithm, where we allow extreme deformations of the mesh
- 4 Repeat until convergence

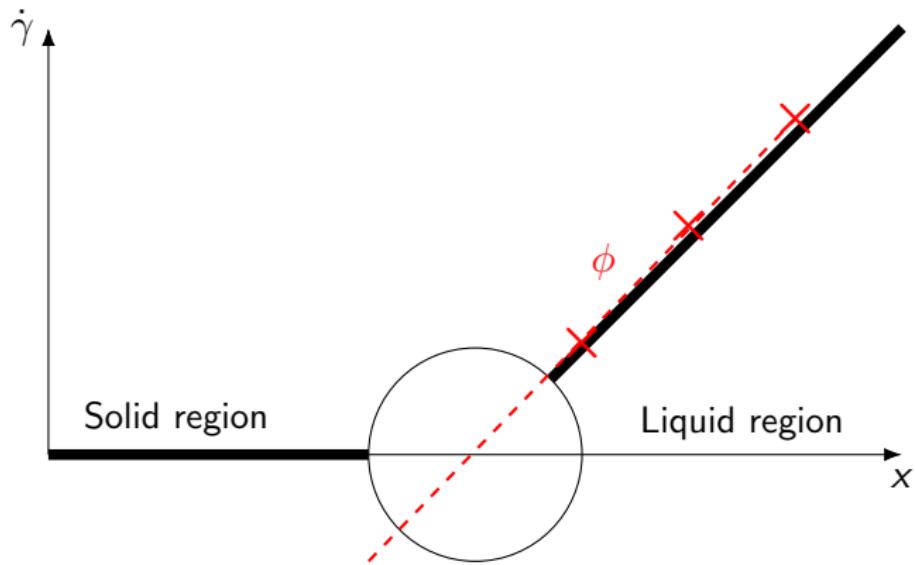
Locating the interface



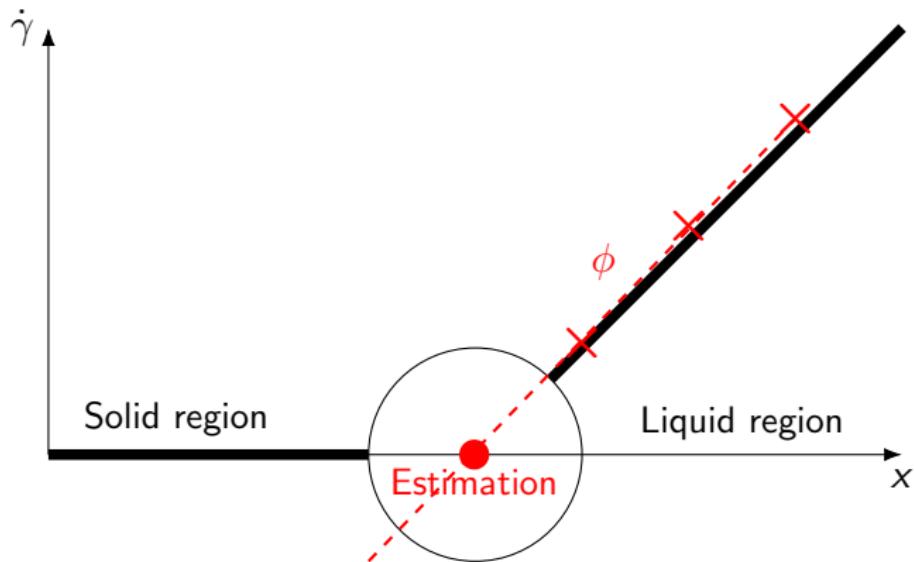
Locating the interface



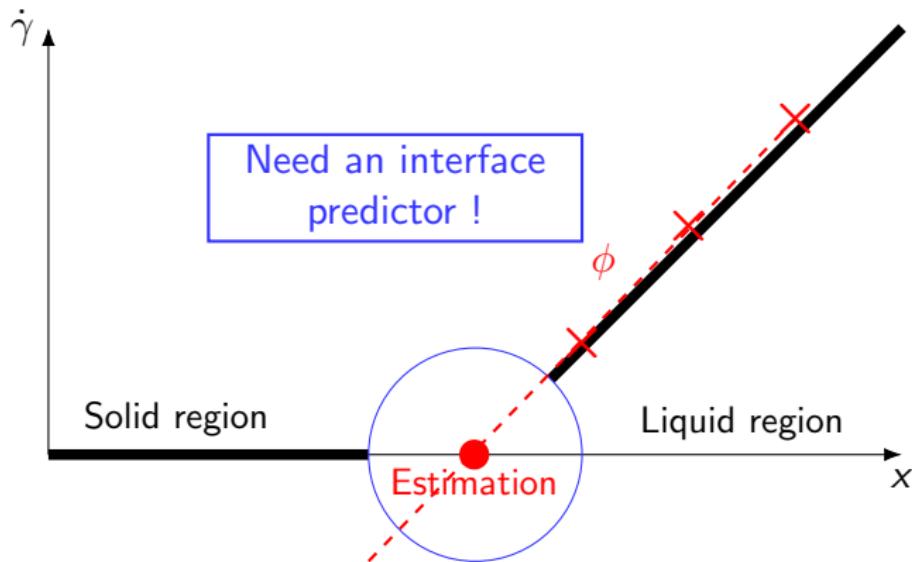
Locating the interface



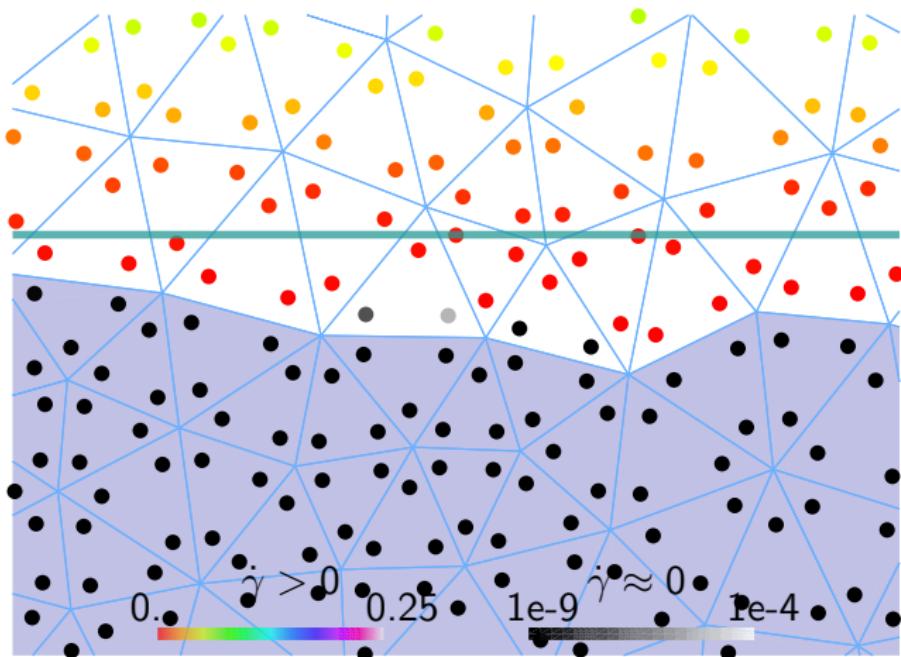
Locating the interface



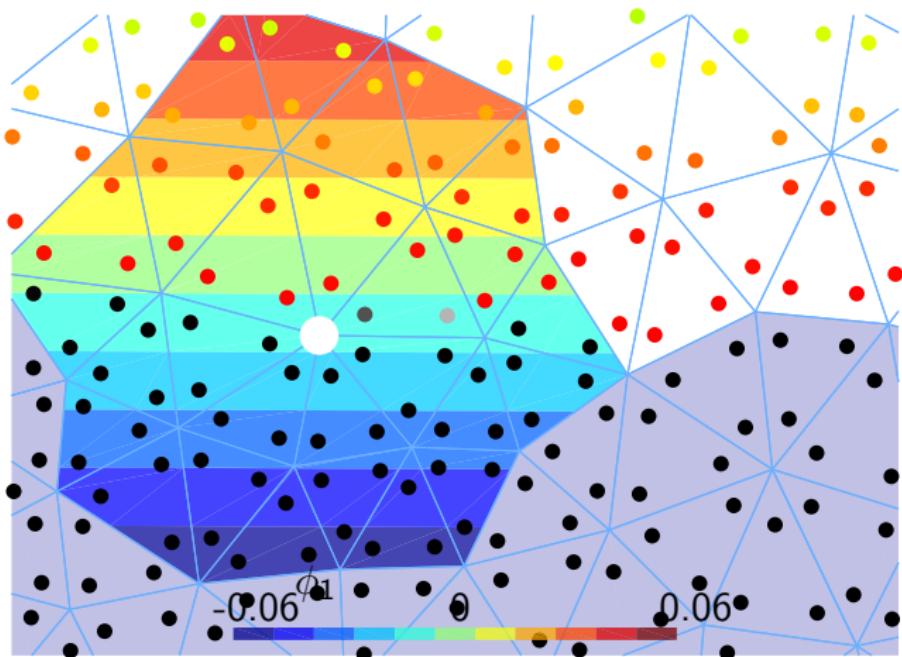
Locating the interface



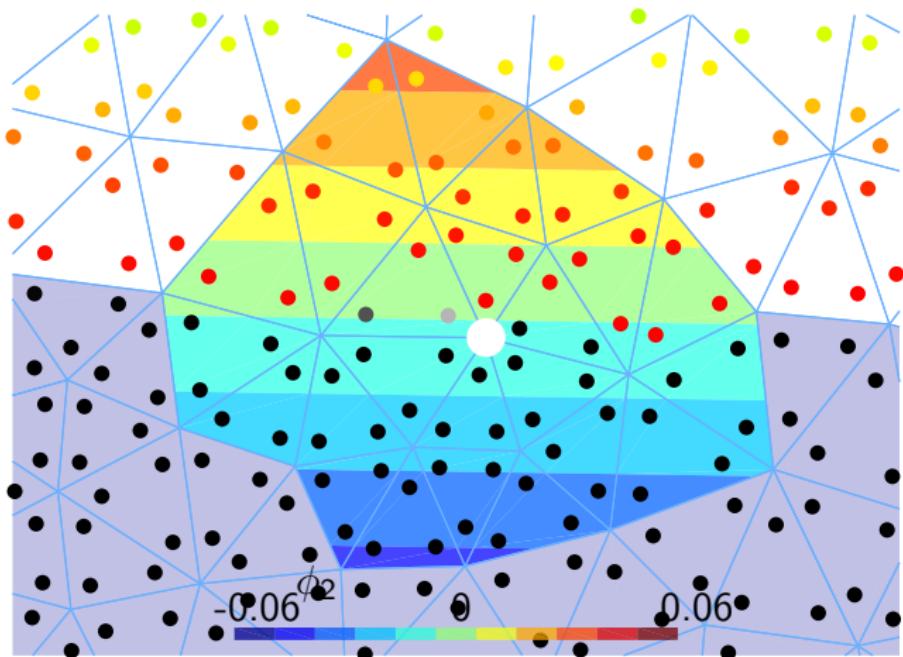
Locating the interface - Predictor



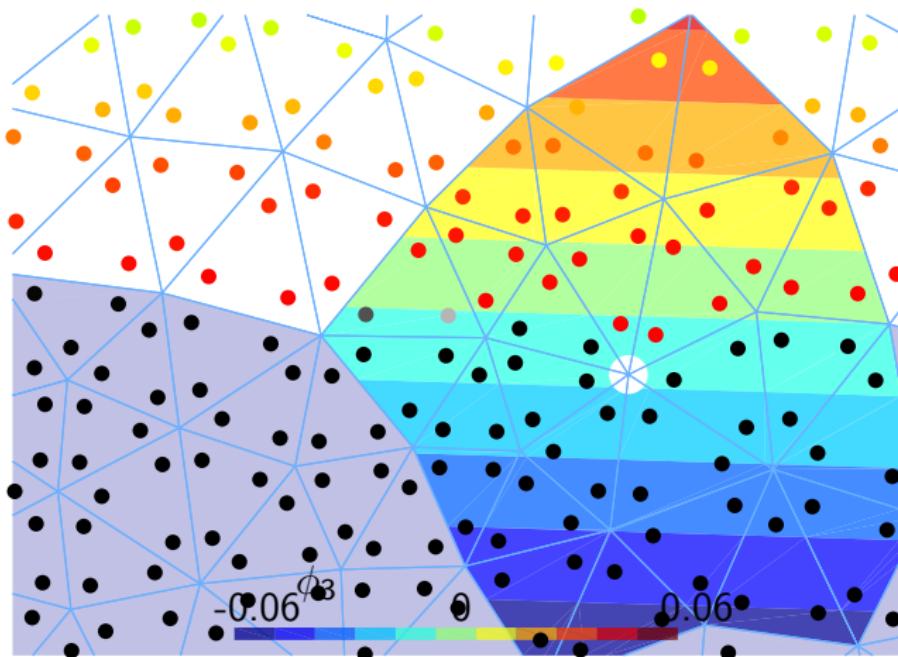
Locating the interface - Compute linear approximation



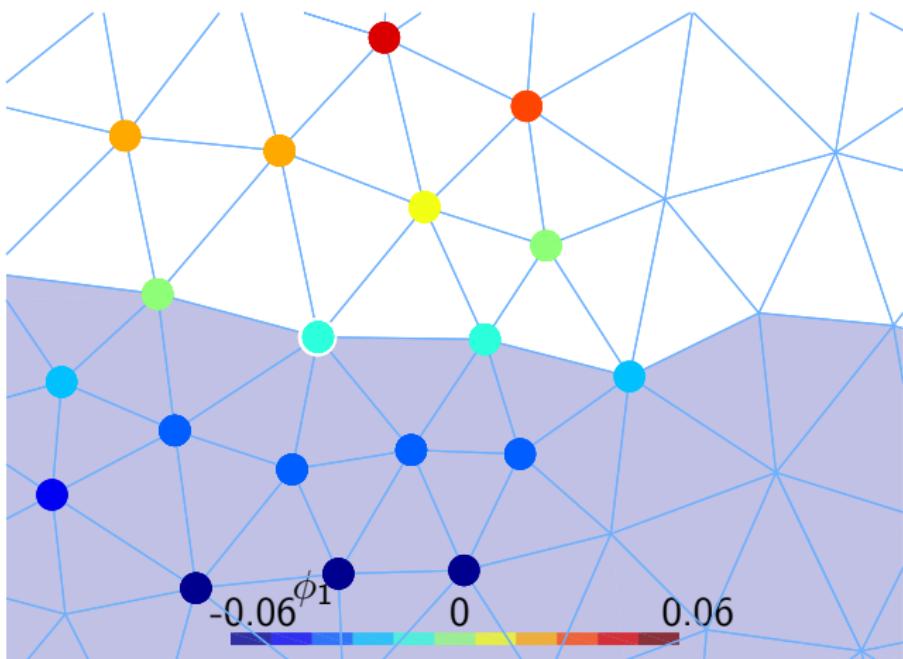
Locating the interface - Compute linear approximation



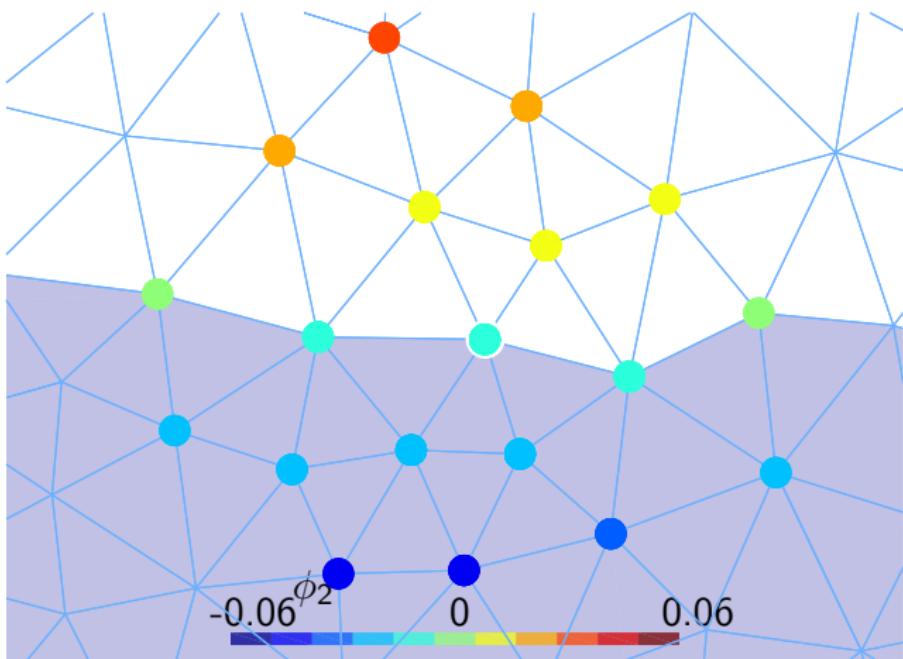
Locating the interface - Compute linear approximation



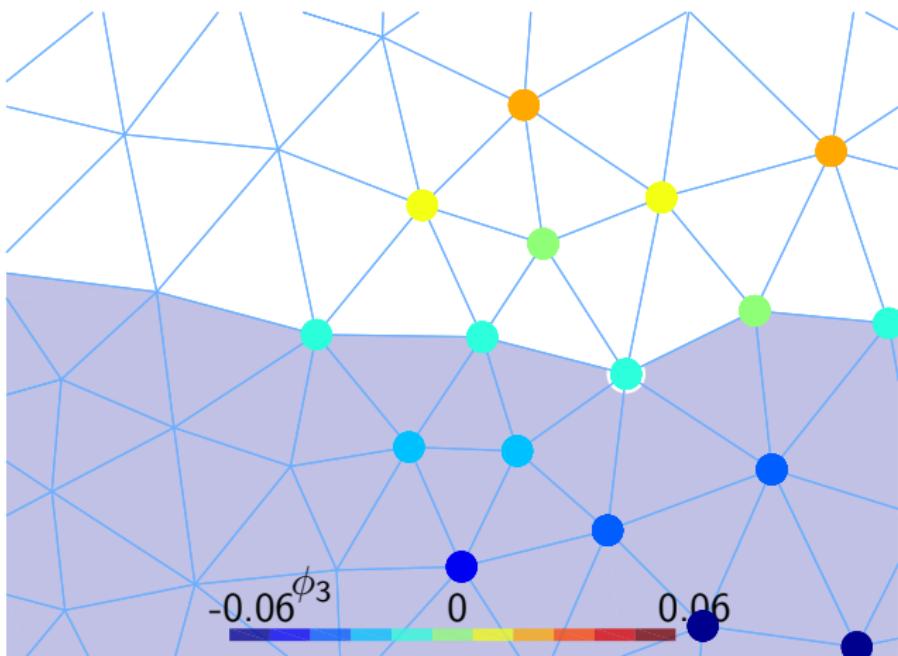
Locating the interface - Evaluate linear approximation



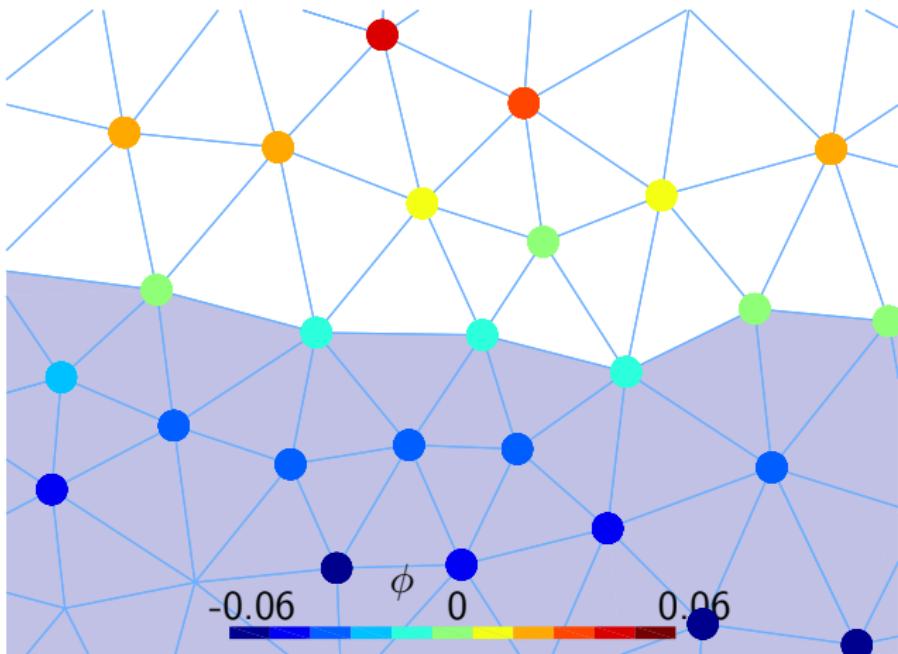
Locating the interface - Evaluate linear approximation



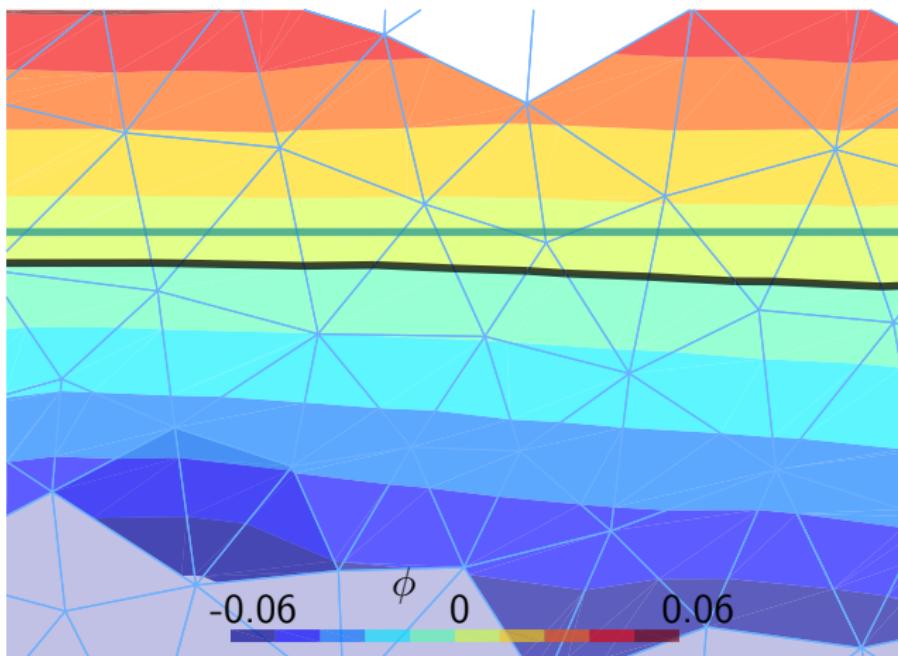
Locating the interface - Evaluate linear approximation



Locating the interface - Average linear approximations



Locating the interface - Corrector



Deform the mesh - Move nodes along edges

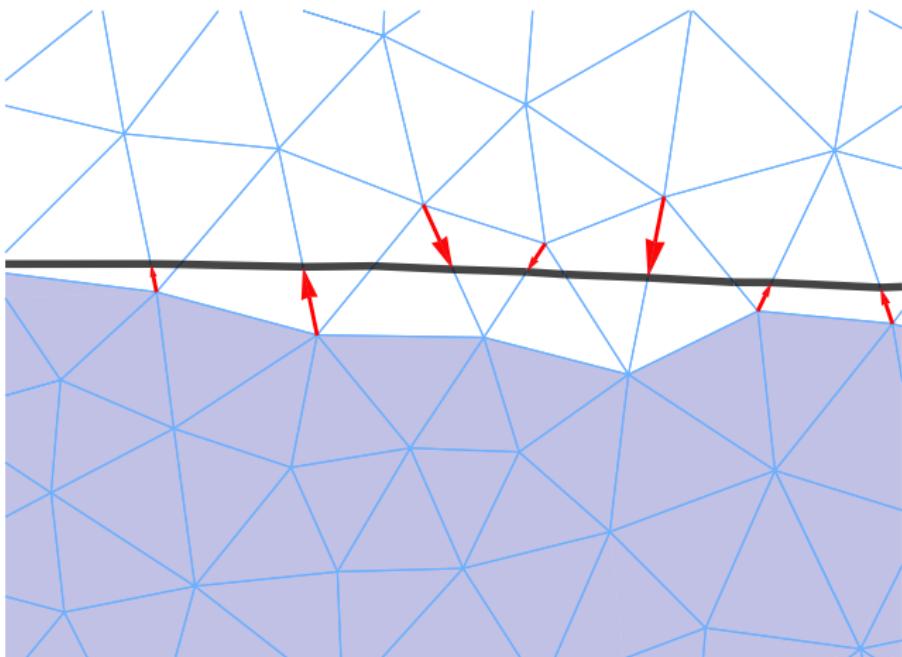
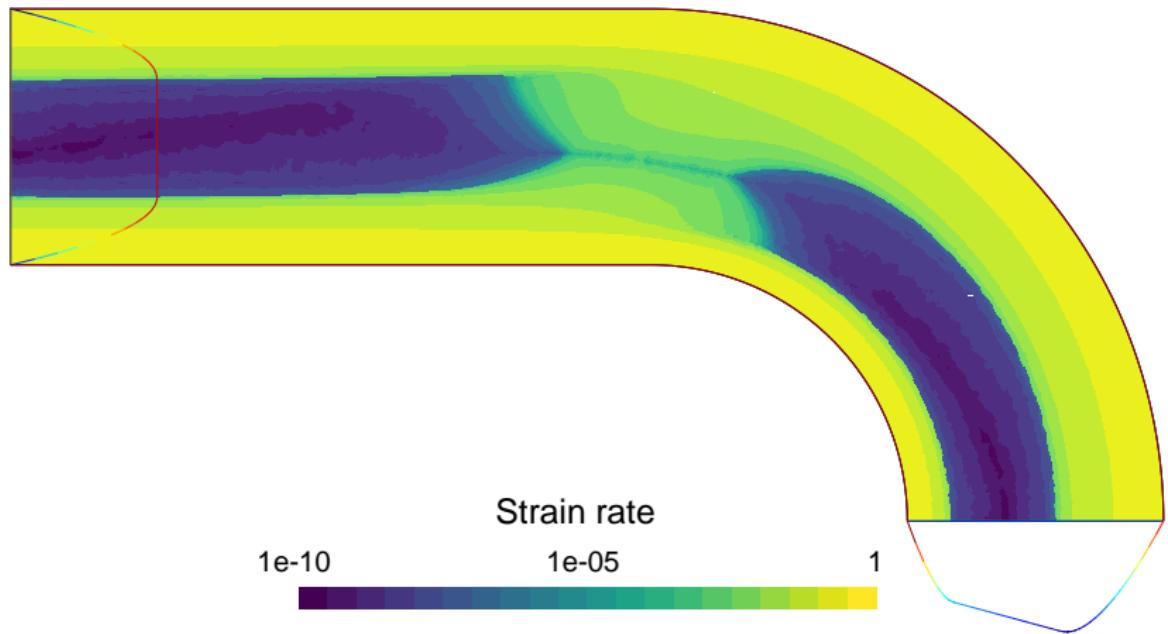


Table of Contents

- 1 The context: what is a yield stress fluid ?
- 2 The motivation: improve flow simulations with interface tracking
- 3 The method: interface tracking algorithm
- 4 Numerical results
- 5 Future work

Curved channel

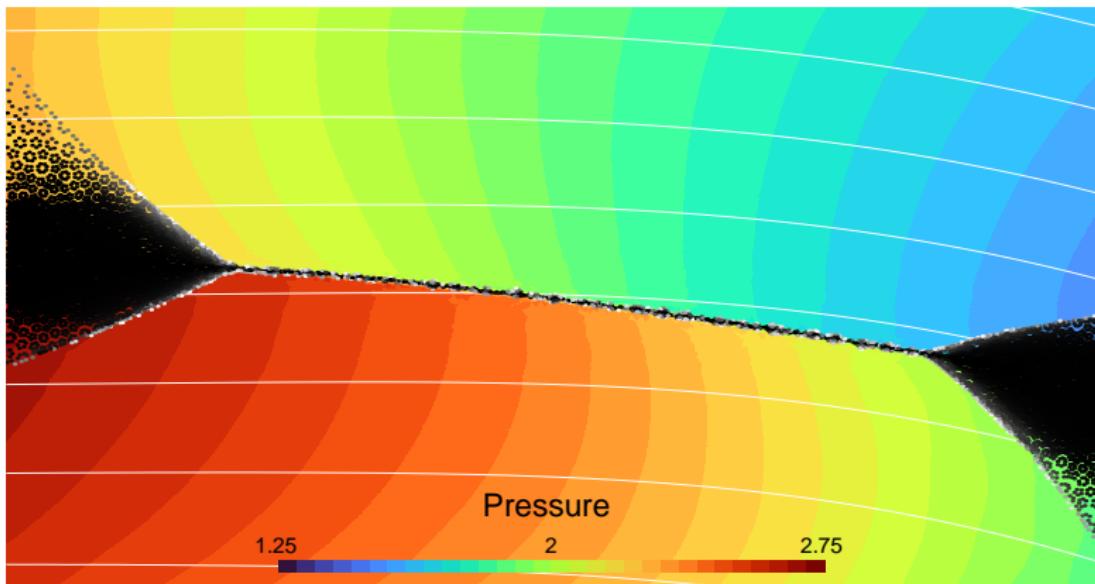


Curved channel - Increasing Bingham

Curved channel - Pressure discontinuity

Jump condition

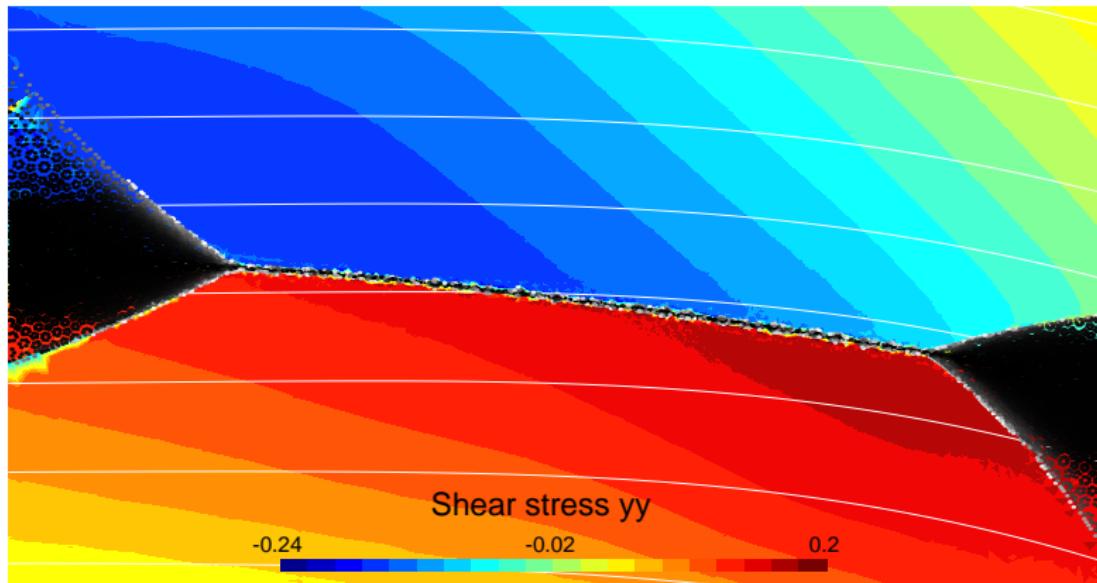
$$[\![\sigma \cdot \hat{n}]\!] = 0 \implies [\![p \hat{n}]\!] = [\![\tau \cdot \hat{n}]\!] = [\![K \dot{\gamma} \cdot \hat{n}]\!] + \tau_0 [\![\dot{\gamma} / \dot{\gamma} \cdot \hat{n}]\!]$$



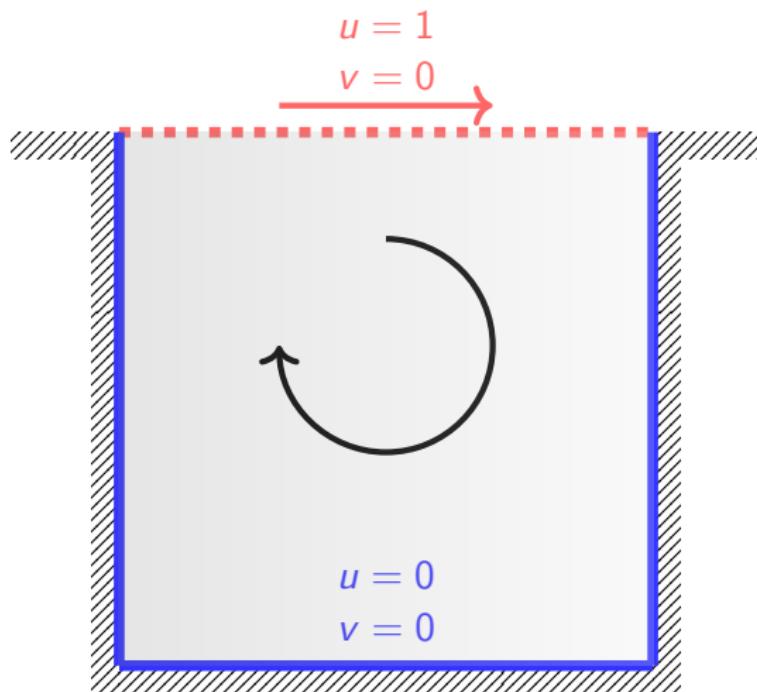
Curved channel - Pressure discontinuity

Jump condition

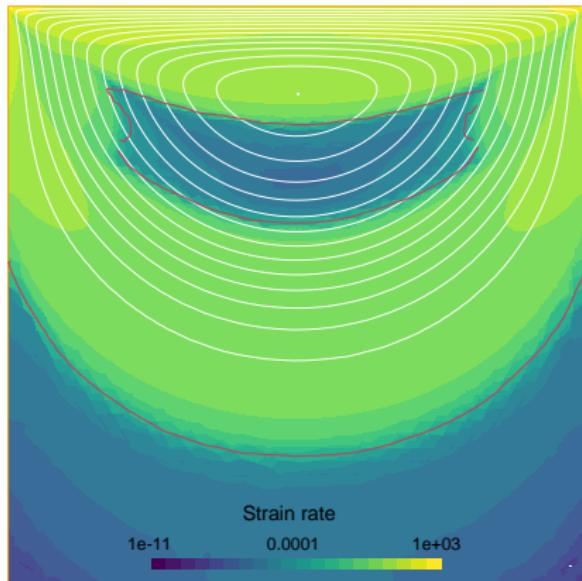
$$[\![\sigma \cdot \hat{n}]\!] = 0 \implies [\![p\hat{n}]\!] = [\![\tau \cdot \hat{n}]\!] = [\![K\dot{\gamma} \cdot \hat{n}]\!] + \tau_0 [\![\dot{\gamma}/\dot{\gamma} \cdot \hat{n}]\!]$$



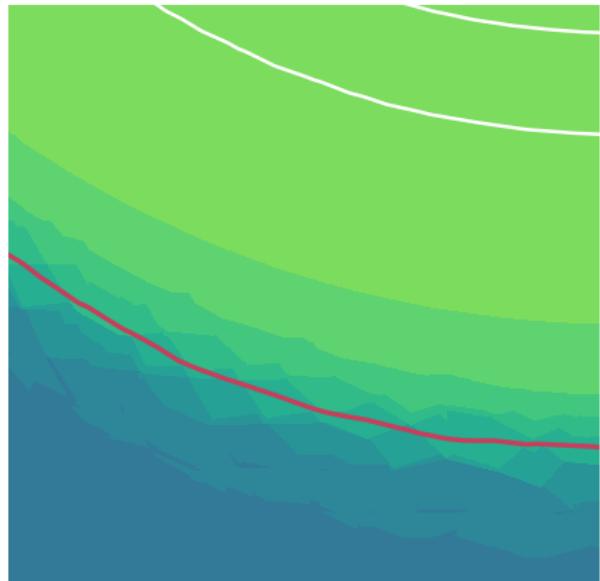
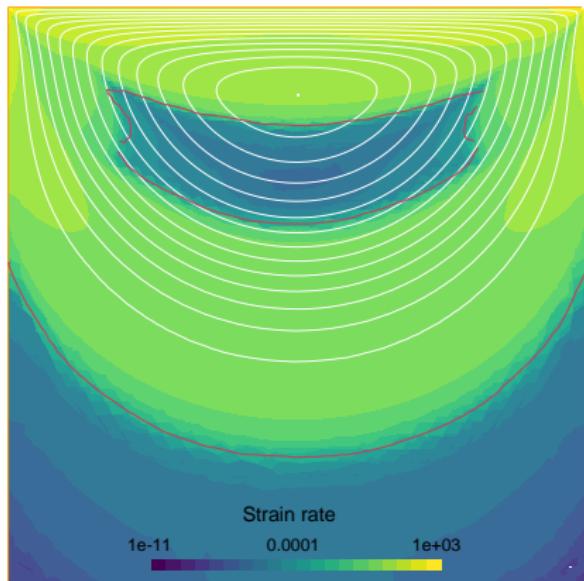
Lid-driven cavity



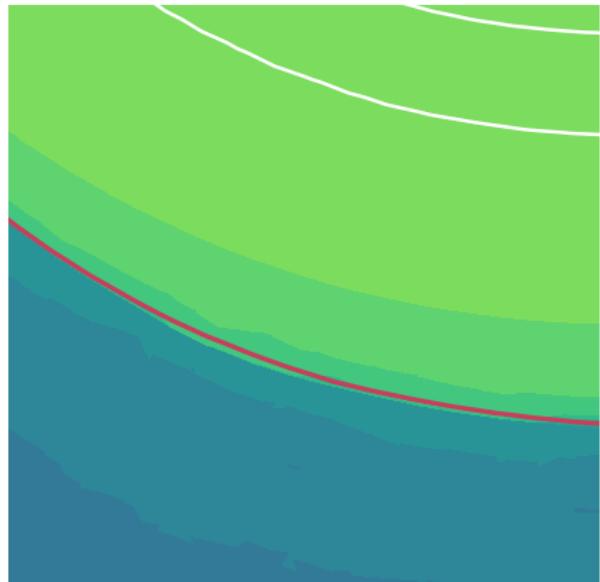
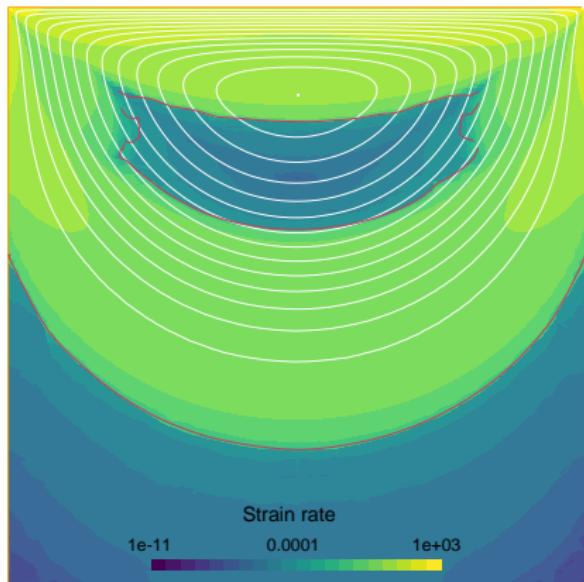
Lid-driven cavity - before interface tracking



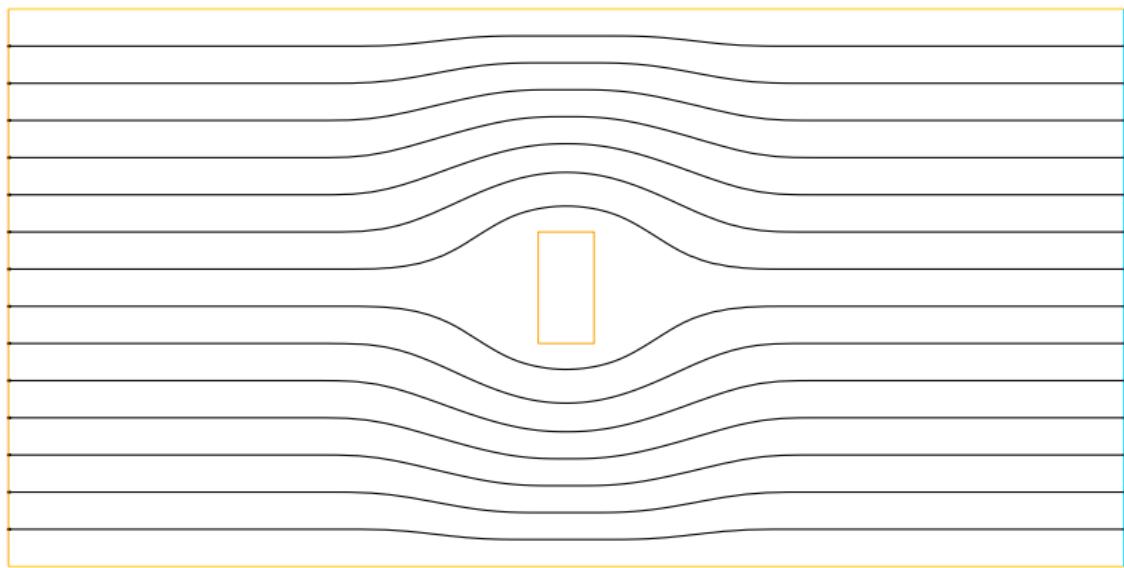
Lid-driven cavity - before interface tracking



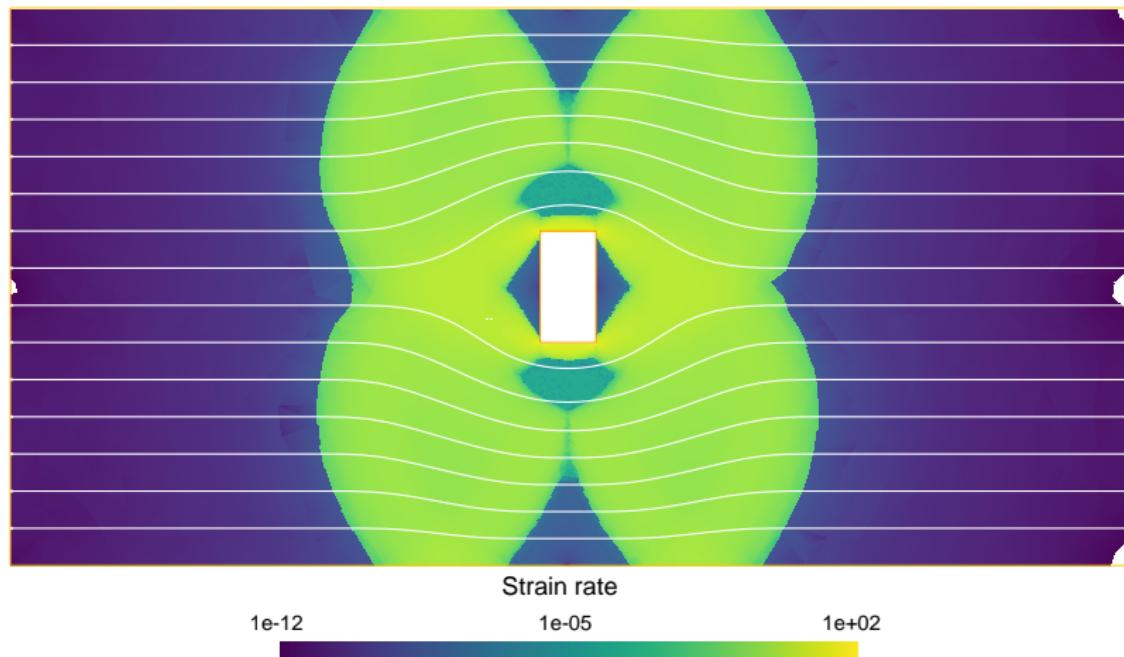
Lid-driven cavity - after interface tracking



Flow around an obstacle



Flow around an obstacle

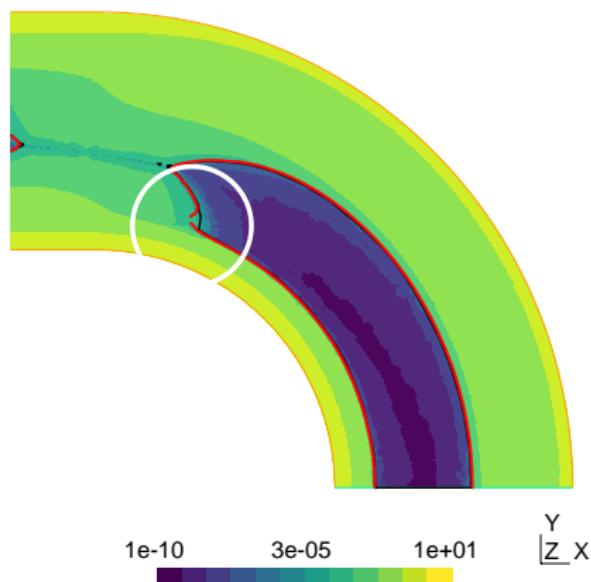


Flow around an obstacle - Solid body motion

Table of Contents

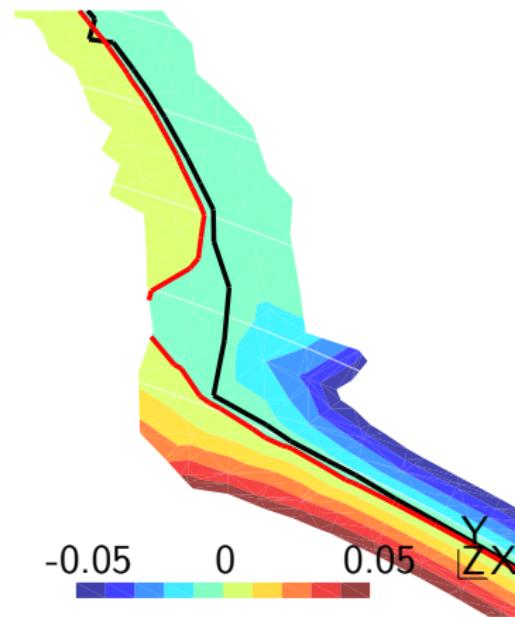
- 1 The context: what is a yield stress fluid ?
- 2 The motivation: improve flow simulations with interface tracking
- 3 The method: interface tracking algorithm
- 4 Numerical results
- 5 Future work

Corners of solid regions



Strain rate of the flow in a curved channel

Corners of solid regions

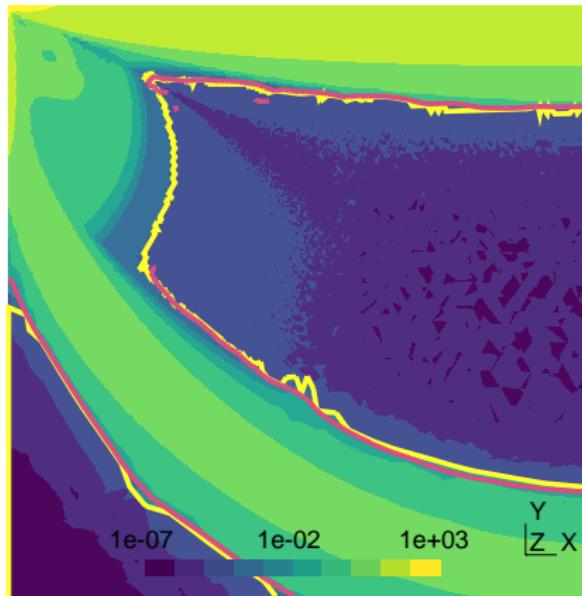


Virtual field ϕ , interface predictor/corrector in black/red.

Yield tolerance parameter

- ▶ $\dot{\gamma}(\mathbf{v}^h)$ never exactly 0, but from $\sim 10^{-12}$ to $\sim 10^{-5}$
- ▶ Interface predictor selects elements with $\dot{\gamma}(\mathbf{v}^h) < 10^{-4}$
- ▶ Would prefer algorithm predicts interface autonomously

Yield tolerance parameter



Strain rate of the lid-driven cavity flow, predictor/corrector in yellow/red

Key points

- ▶ Solid-liquid interface of yield stress fluids
- ▶ Predictor-corrector method to track this interface
- ▶ Bingham model can generate pressure jumps

Code is open-source: github.com/vinzphenix/Bingham_fluid

Equations

- Conservation laws for incompressible fluids:

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega$$

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{f} \quad \text{in } \Omega$$

Equations

- Conservation laws for incompressible fluids, with low Reynolds:

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega$$

$$0 = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{f} \quad \text{in } \Omega$$

Equations

- Conservation laws for incompressible fluids:

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega$$

$$0 = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{f} \quad \text{in } \Omega$$

- Constitutive law

$$\dot{\gamma} = 0 \quad \text{if } \tau < \tau_0$$

$$\boldsymbol{\tau} = (K + \tau_0/\dot{\gamma}) \dot{\gamma} \quad \text{if } \tau \geq \tau_0$$

Equations

- Conservation laws for incompressible fluids:

$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega$$

$$0 = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{f} \quad \text{in } \Omega$$

- Constitutive law

$$\dot{\gamma} = 0 \quad \text{if } \tau < \tau_0$$

$$\boldsymbol{\tau} = (K + \tau_0/\dot{\gamma}) \dot{\gamma} \quad \text{if } \tau \geq \tau_0$$

- Boundary conditions

$$\mathbf{v} = \mathbf{U} \quad \text{on } \partial\Omega_D$$

$$\hat{\mathbf{n}} \cdot \boldsymbol{\sigma} = \mathbf{g} \quad \text{on } \partial\Omega_N$$

Energy functional

System of PDE's is equivalent to

$$\mathbf{v} = \arg \min_{\mathbf{u} \in \mathcal{V}} \mathcal{J}(\mathbf{u})$$

$$\mathcal{V} = \left\{ \mathbf{u} \in H^1(\Omega)^d \mid \int_{\Omega} q \nabla \cdot \mathbf{u} \, dx = 0 \quad \forall q \in L^2(\Omega), \mathbf{u} = \mathbf{U} \text{ on } \partial\Omega_D \right\}$$

$$\mathcal{J}(\mathbf{u}) = \underbrace{\frac{K}{2} \int_{\Omega} \|\dot{\gamma}(\mathbf{u})\|^2 \, dx}_{\text{Viscous}} + \underbrace{\tau_0 \int_{\Omega} \|\dot{\gamma}(\mathbf{u})\| \, dx}_{\text{Yield}} - \underbrace{\int_{\Omega} \mathbf{f} \cdot \mathbf{u} \, dx}_{\text{Body forces}} - \underbrace{\int_{\partial\Omega_N} \mathbf{g} \cdot \mathbf{u} \, ds}_{\text{Boundary forces}}$$

Finite element method: a stable discretization ?

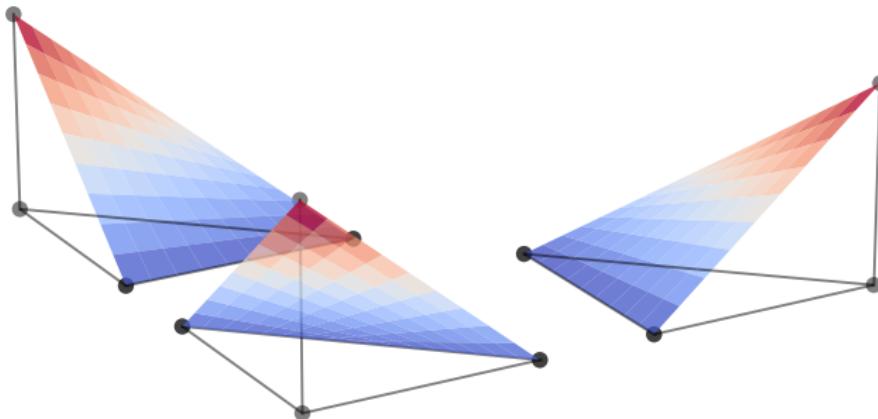
Discretize velocity-pressure fields, over a mesh of triangles, with either:

- ▶ Mini element: $(\mathcal{P}_1^C \oplus \mathcal{B}_3^C) - \mathcal{P}_1^C$
- ▶ Taylor-Hood element: $\mathcal{P}_2^C - \mathcal{P}_1^C$

Finite element method: a stable discretization ?

Discretize velocity-pressure fields, over a mesh of triangles, with either:

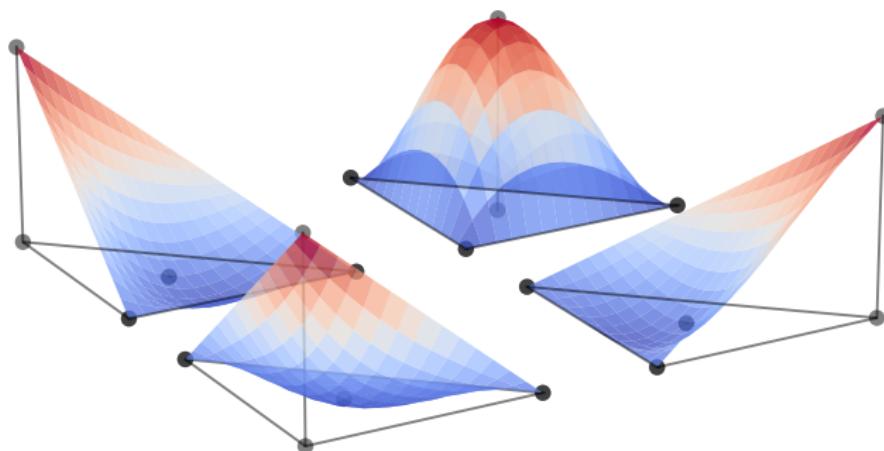
- ▶ Mini element: $(\mathcal{P}_1^C \oplus \mathcal{B}_3^C) - \mathcal{P}_1^C$
- ▶ Taylor-Hood element: $\mathcal{P}_2^C - \mathcal{P}_1^C$



Finite element method: a stable discretization ?

Discretize velocity-pressure fields, over a mesh of triangles, with either:

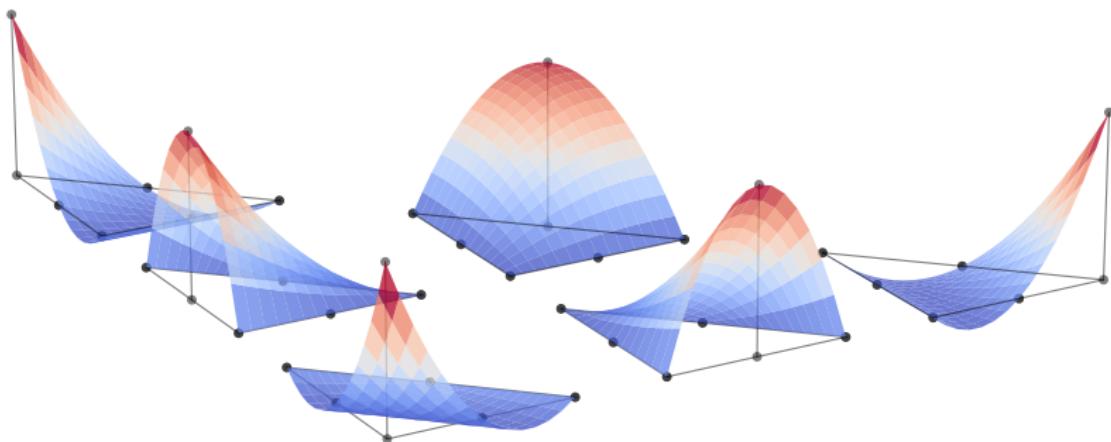
- ▶ Mini element: $(\mathcal{P}_1^C \oplus \mathcal{B}_3^C) - \mathcal{P}_1^C$
- ▶ Taylor-Hood element: $\mathcal{P}_2^C - \mathcal{P}_1^C$



Finite element method: a stable discretization ?

Discretize velocity-pressure fields, over a mesh of triangles, with either:

- ▶ Mini element: $(\mathcal{P}_1^C \oplus \mathcal{B}_3^C) - \mathcal{P}_1^C$
- ▶ Taylor-Hood element: $\mathcal{P}_2^C - \mathcal{P}_1^C$



Finite element approximation

- Finite element approximation:

$$\mathbf{v}^h(\mathbf{x}) = \sum_{\text{nodes } j} \mathbf{v}_j \phi_j(\mathbf{x})$$

- Discrete functional:

$$\begin{aligned} \mathcal{J}(\mathbf{v}^h) = & \sum_i \sum_g \omega_g \left[\frac{K}{2} \|\dot{\gamma}(\mathbf{v}^h)\|^2 + \tau_0 \|\dot{\gamma}(\mathbf{v}^h)\| - \mathbf{f} \cdot \mathbf{v}^h \right] \det \frac{d\mathbf{x}}{d\xi}_{i,g} \\ & - \sum_e \sum_g \tilde{\omega}_g \mathbf{g} \cdot \mathbf{v}^h \det \frac{d\tilde{\mathbf{x}}}{d\xi}_{e,g} \end{aligned}$$

Finite element approximation

- Finite element approximation:

$$\mathbf{v}^h(\mathbf{x}) = \sum_{\text{nodes } j} \mathbf{v}_j \phi_j(\mathbf{x})$$

- Discrete functional, with additional variables $S_{i,g}$ and $T_{i,g}$:

$$\begin{aligned} \mathcal{J}(\mathbf{v}^h) = & \sum_i \sum_g \omega_g \left[\frac{K}{2} \underbrace{\|\dot{\gamma}(\mathbf{v}^h)\|^2}_{\leq S_{i,g}} + \tau_0 \underbrace{\|\dot{\gamma}(\mathbf{v}^h)\|}_{\leq T_{i,g}} - \mathbf{f} \cdot \mathbf{v}^h \right] \det \frac{d\mathbf{x}}{d\xi}_{i,g} \\ & - \sum_e \sum_g \tilde{\omega}_g \mathbf{g} \cdot \mathbf{v}^h \det \frac{d\tilde{\mathbf{x}}}{d\xi}_{e,g} \end{aligned}$$

Objective and constraints

$$\underset{\mathbf{v}_j, S_{i,g}, T_{i,g}}{\text{minimize}} \quad \sum_{i,g} \omega_g \left[\frac{K}{2} S_{i,g} + \tau_0 T_{i,g} - \mathbf{f} \cdot \mathbf{v}^h \right] \det \frac{d\mathbf{x}}{d\xi}_{i,g} - \sum_{e,g} \tilde{\omega}_g \mathbf{g} \cdot \mathbf{v}^h \det \frac{d\tilde{\mathbf{x}}}{d\xi}_{e,g}$$

such that $S_{i,g} \geq (2\partial_x u^h)^2 + (2\partial_y v^h)^2 + (\partial_y u^h + \partial_x v^h)^2 \quad \forall i, g$

$$T_{i,g} \geq \sqrt{(2\partial_x u^h)^2 + (2\partial_y v^h)^2 + (\partial_y u^h + \partial_x v^h)^2} \quad \forall i, g$$

$$0 = \sum_{i,g} \omega_g q_I|_{\mathbf{x}_g} (\partial_x u^h + \partial_y v^h) \det \frac{d\mathbf{x}}{d\xi}_{i,g} \quad \forall I$$

$$\mathbf{v}_j = \mathbf{U} \quad \forall j \in \partial\Omega_D$$

Reformulation for conic optimization

$$\underset{\mathbf{v}_j, S_{i,g}, T_{i,g}}{\text{minimize}} \quad \sum_{i,g} \omega_g \left[\frac{K}{2} S_{i,g} + \tau_0 T_{i,g} - \mathbf{f} \cdot \mathbf{v}^h \right] \det \frac{d\mathbf{x}}{d\xi}_{i,g} - \sum_{e,g} \tilde{\omega}_g \mathbf{g} \cdot \mathbf{v}^h \det \frac{d\mathbf{x}}{d\xi}_{e,g}$$

such that

$0 \preceq_{L_R^5} \left(S_{i,g}, \frac{1}{2}, \sqrt{2}\partial_x u^h, \sqrt{2}\partial_y v^h, \partial_y u^h + \partial_x v^h \right)$	$\forall i, g$
$0 \preceq_{L^4} \left(T_{i,g}, \sqrt{2}\partial_x u^h, \sqrt{2}\partial_y v^h, \partial_y u^h + \partial_x v^h \right)$	$\forall i, g$
$0 = \sum_{i,g} \omega_g q_I _{\mathbf{x}_g} (\partial_x u^h + \partial_y v^h) \det \frac{d\mathbf{x}}{d\xi}_{i,g}$	$\forall I$
$\mathbf{v}_j = \mathbf{U}$	$\forall j \in \partial\Omega_D$

Reformulation for conic optimization

$$\underset{\mathbf{V}_j, S_{i,g}, T_{i,g}}{\text{minimize}} \quad \sum_{i,g} \omega_g \left[\frac{K}{2} S_{i,g} + \tau_0 T_{i,g} - \mathbf{f} \cdot \mathbf{v}^h \right] \det \frac{d\mathbf{x}}{d\xi}_{i,g} - \sum_{e,g} \tilde{\omega}_g \mathbf{g} \cdot \mathbf{v}^h \det \frac{d\mathbf{x}}{d\xi}_{e,g}$$

such that

$$0 \preceq_{L_R^5} \left(S_{i,g}, \frac{1}{2}, \sqrt{2}\partial_x u^h, \sqrt{2}\partial_y v^h, \partial_y u^h + \partial_x v^h \right) \quad \forall i, g$$

$$0 \preceq_{L^4} \left(T_{i,g}, \sqrt{2}\partial_x u^h, \sqrt{2}\partial_y v^h, \partial_y u^h + \partial_x v^h \right) \quad \forall i, g$$

$$0 = \sum_{i,g} \omega_g q_I|_{\mathbf{x}_g} (\partial_x u^h + \partial_y v^h) \det \frac{d\mathbf{x}}{d\xi}_{i,g} \quad \forall I$$

$$\mathbf{v}_j = \mathbf{U} \quad \forall j \in \partial\Omega_D$$

- ▶ Objective is linear in variables $U_j, V_j, S_{i,g}, T_{i,g}$
- ▶ Equality constraints are linear
- ▶ Inequality constraints are proper cones

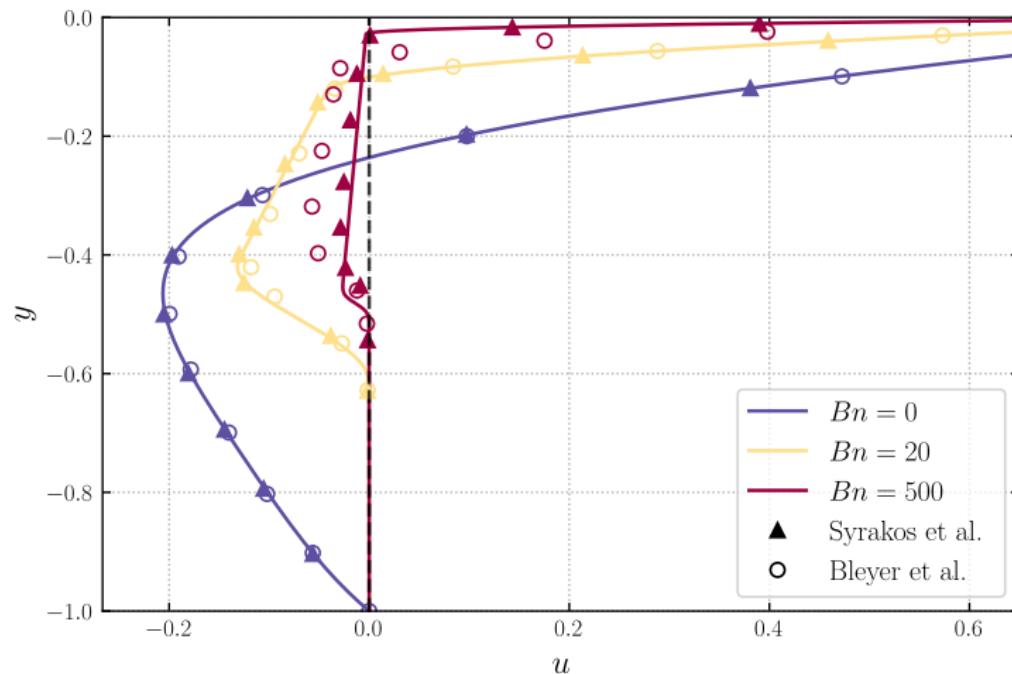
Interior-point method

- ▶ Encode each constraint with barrier g_n
- ▶ Minimize a modified functional $\mathcal{J} + \mu \sum g_n$
- ▶ Alternatively:
 - ▶ Perform Newton iterations (stay optimal)
 - ▶ Decrease $\mu \rightarrow 0$ (minimize original objective \mathcal{J})
- ▶ Proper cones provide special barriers (self-concordant)
- ▶ Solution obtained with Newton–Raphson method, within accuracy ϵ , after $\mathcal{O}(\sqrt{\nu} \log \frac{1}{\epsilon})$ iterations, where $\nu \propto$ number of variables

Interior-point method - example

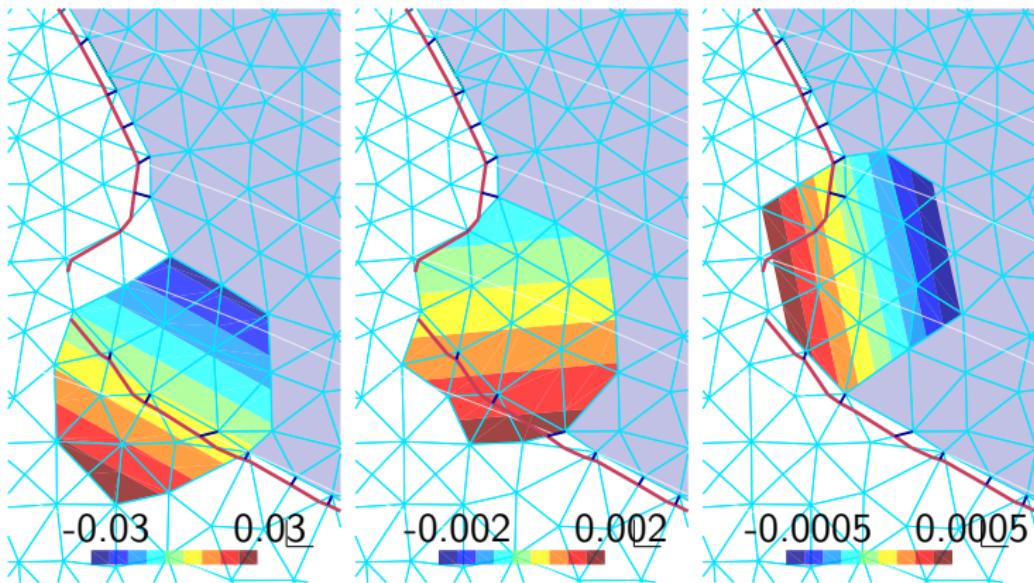
Minimize x with linear inequalities over x and y

Lid-driven cavity - comparison with the literature



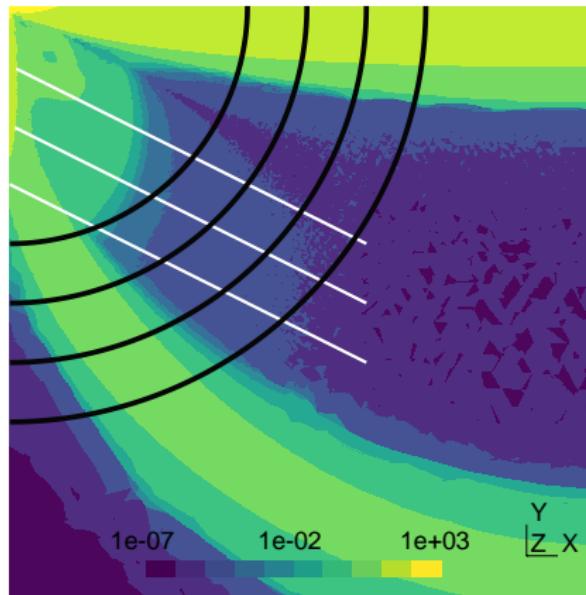
Lid-driven cavity - Solid body motion

Corners of solid regions



Successive linear approximations around the interface predictor

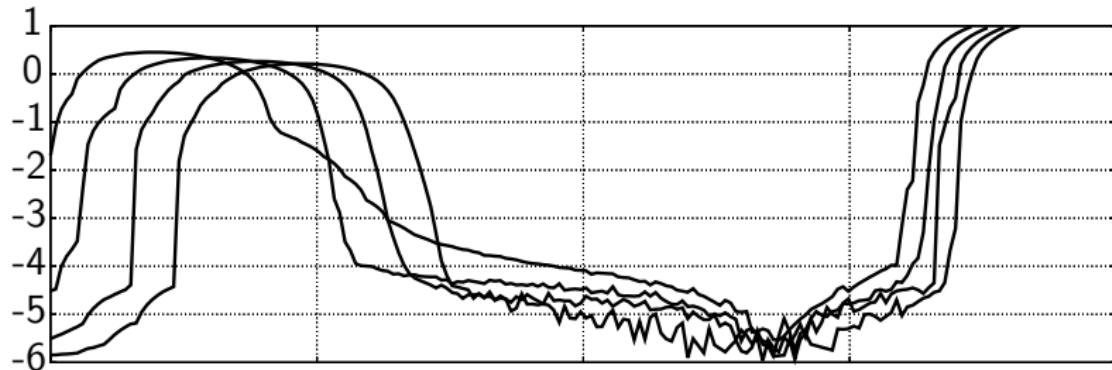
Yield tolerance parameter



Parametric curves

Yield tolerance parameter

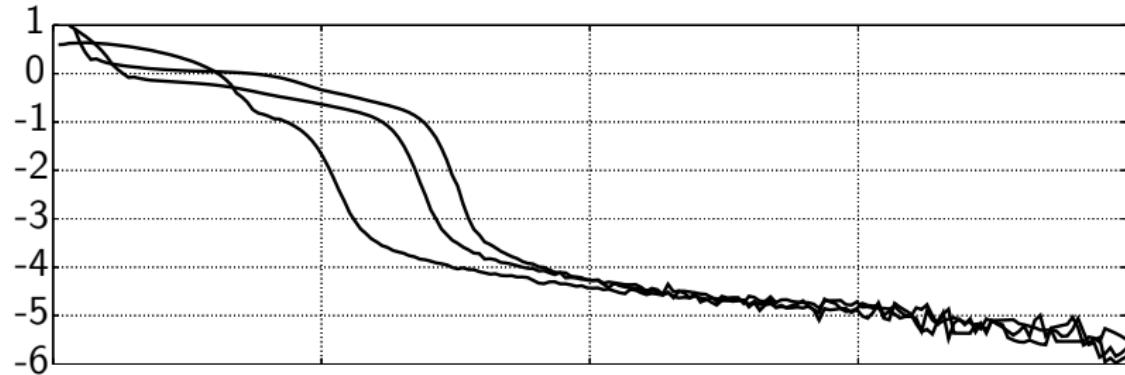
$\log_{10} \dot{\gamma}$ — circles



Evaluation of the strain rate along the parametric curves

Yield tolerance parameter

$\log_{10} \dot{\gamma}$ — lines



Evaluation of the strain rate along the parametric curves