Simulating viscoelastic flows with Basilisk

J.M. López-Herrera
Universidad de Sevilla

S. Popinet, A. Castrejón-Pita & E. Vega

***BGUM2017**, November 15th -16th, Princeton, EE.UU.*

Summary

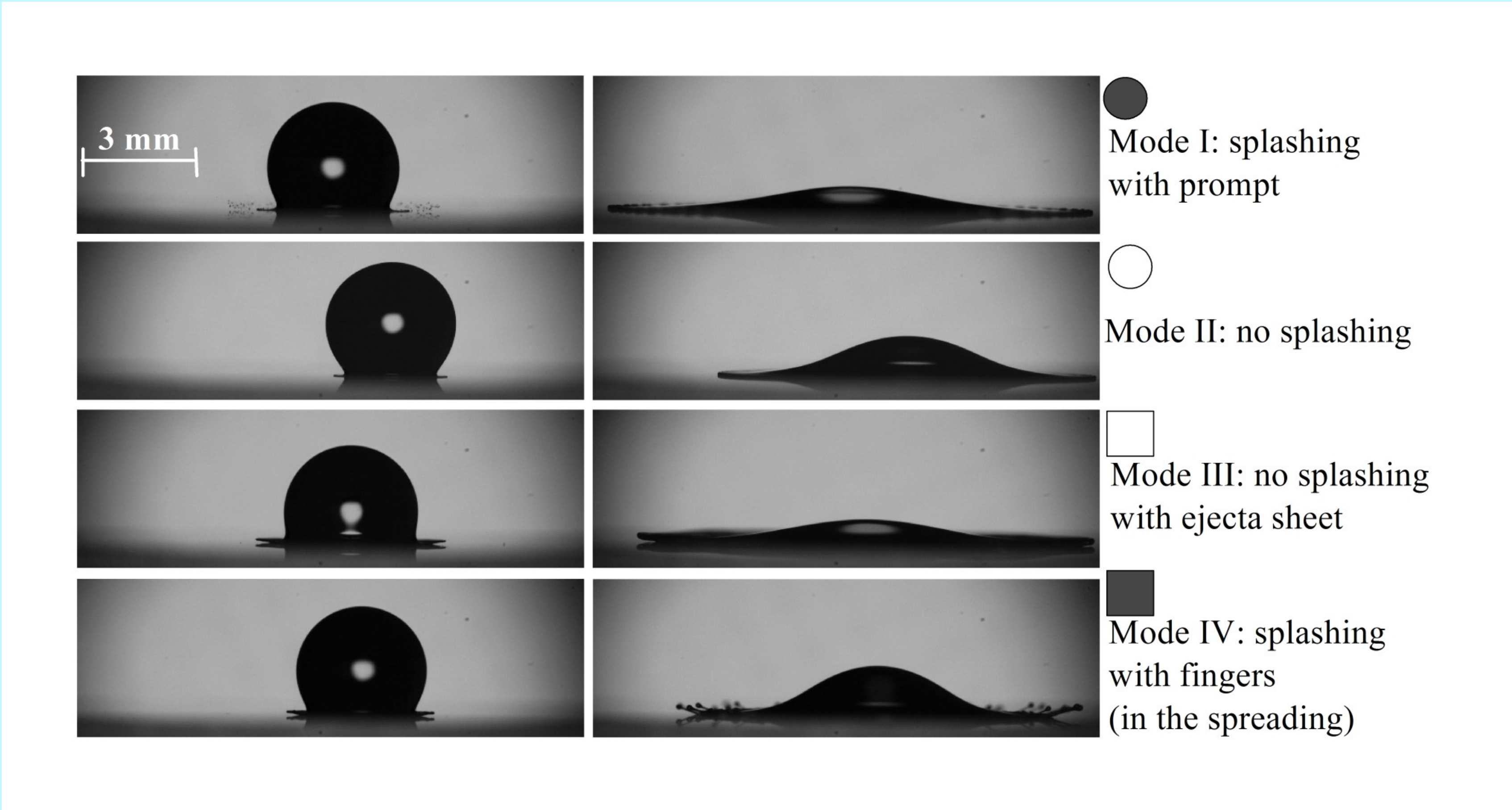
1. Motivation
2. Equations
3. Numerical Scheme
4. Test problems
5. Further improvements

Summary

1. **Motivation**
2. Equations
3. Numerical Scheme
4. Test problems
5. Further improvements

Motivation

Impact on slightly viscoelastic droplet

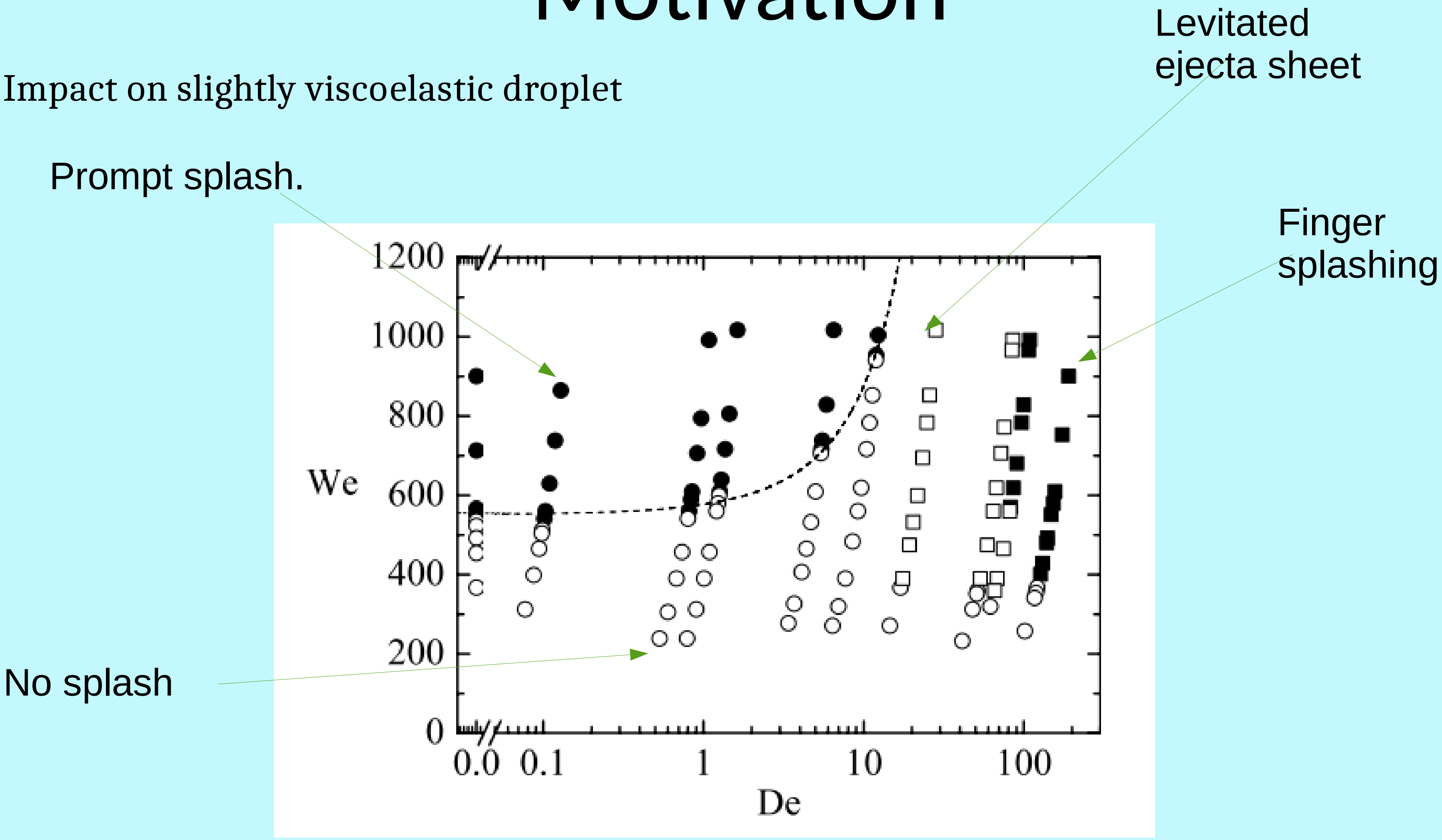


Water + polyacrylamide (PAA)

Vega, E.J. & Castrejón-Pita, A.A., (2017).. Experiments in Fluids, 58(5), p.57

Motivation

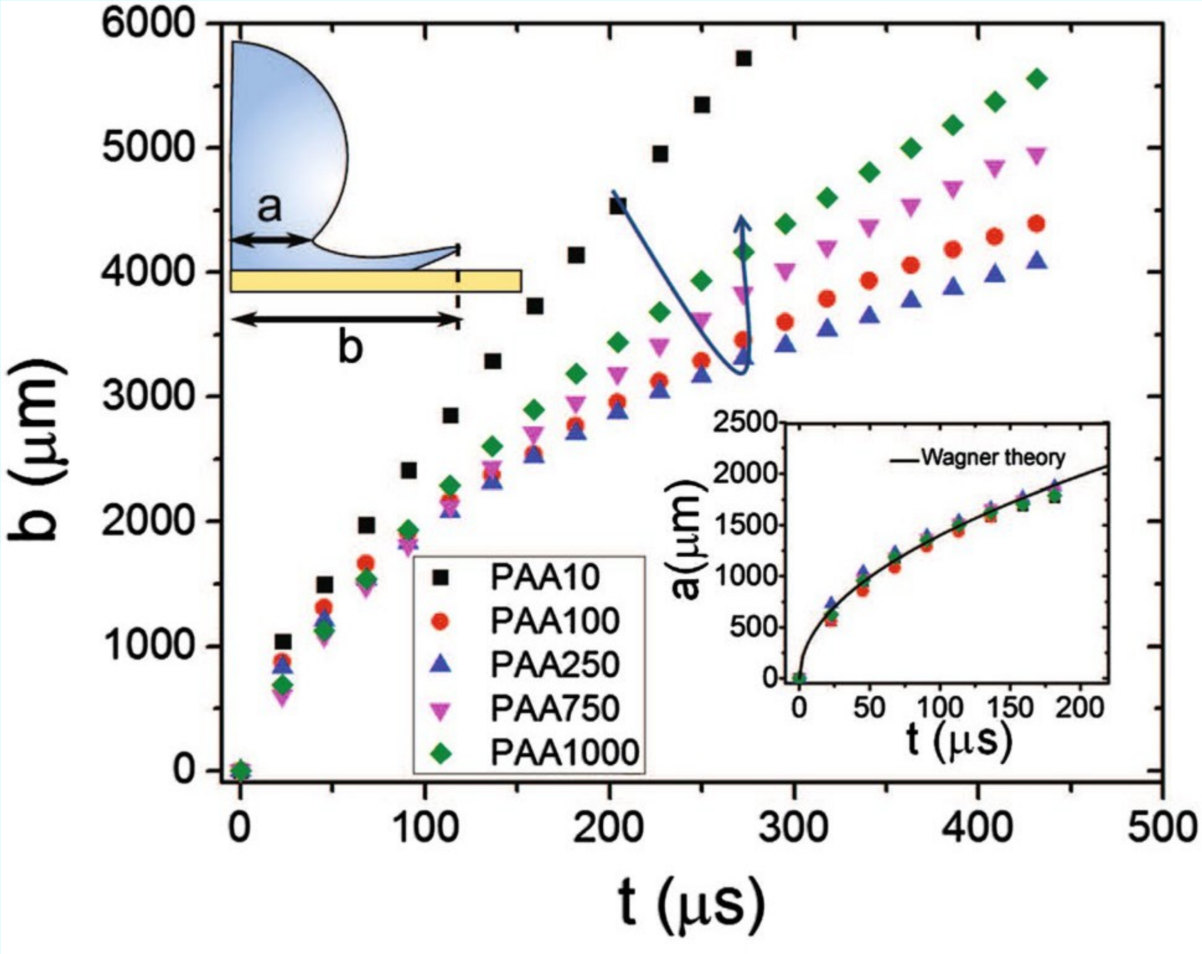
Impact on slightly viscoelastic droplet



Vega, E.J. & Castrejón-Pita, A.A., (2017).. Experiments in Fluids, 58(5), p.57

Motivation

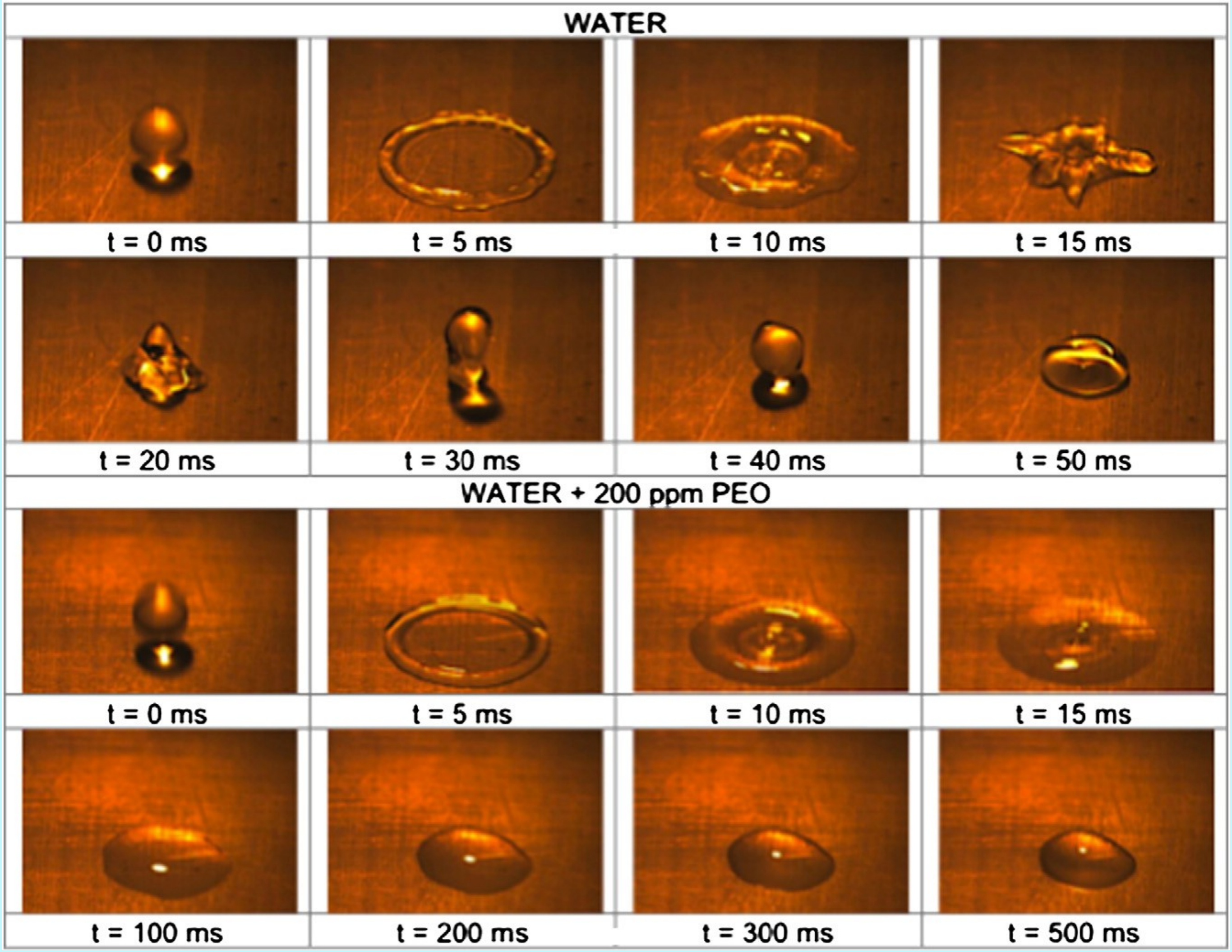
Impact on slightly viscoelastic droplet



Vega, E.J. & Castrejón-Pita, A.A., (2017).. Experiments in Fluids, 58(5), p.57

Motivation

Impact on slightly viscoelastic droplet



Bertola, V. (2013). *Adv. in Colloid and Interf. Sci.*, *193–194*, 1–11

Summary

1. Motivation

2. **Equations**

   1. **Constitutive models.**
   2. **Boundary Conditions**
   3. **The High Weissemberg number problem.**
   4. **Solutions to the HWNP. Conformation kernels.**

3. Numerical Scheme

4. Test problems

5. Further improvements

Equations

$$\nabla \cdot \mathbf{u} = 0,$$

$$\rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \gamma \kappa \mathbf{n} \delta_s + \rho g$$

$$\boldsymbol{\tau} = \boldsymbol{\tau_s} + \boldsymbol{\tau_p},$$

$$\boldsymbol{\tau}_s = 2\mu_s \mathbf{D} = \mu_s(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$$

$$\boldsymbol{\tau_p} = \frac{\mu_p \mathbf{f_S}(\mathbf{A})}{\lambda}$$

Conformation tensor

Polymeric Stress

Solvent Stress

Equations

$$\underbrace{\partial_t \mathbf{A} + \nabla \cdot (\mathbf{u}\mathbf{A}) - \mathbf{A} \cdot \nabla \mathbf{u} - \nabla \mathbf{u}^T \cdot \mathbf{A}} = -\frac{\mathbf{f_R}(\mathbf{A})}{\lambda}$$

$\overset{\triangledown}{\mathbf{A}}$ = Upper Convective Derivative

$\nabla \mathbf{u}|_{ij} = \partial_i u_j.$ Attention, some people swap the indexes!

CONSTITUTIVE MODELS

	Oldroyd B	FENE-P	FENE-CR	linear PTT
$\mathbf{f_R}(\mathbf{A})$	$\mathbf{A} - \mathbf{I}$	$\frac{\mathbf{A}}{1-tr(\mathbf{A})/L^2} - \mathbf{I}$	$\frac{\mathbf{A}-\mathbf{I}}{1-tr(\mathbf{A})/L^2}$	$(1+\varepsilon\, tr(\mathbf{A}-\mathbf{I}))(\mathbf{A}-\mathbf{I})$
$\mathbf{f_S}(\mathbf{A})$	$\mathbf{A} - \mathbf{I}$	$\frac{\mathbf{A}}{1-tr(\mathbf{A})/L^2} - \mathbf{I}$	$\frac{\mathbf{A}-\mathbf{I}}{1-tr(\mathbf{A})/L^2}$	$\mathbf{A} - \mathbf{I}$

$\mathbf{f_R}(\mathbf{A})$ and $\mathbf{f_S}(\mathbf{A})$ are the relaxation and the stress functions.

Equations

Some issues

- I have not a clear what means **A** from the point of view of the physics**.** Do not ask me!
- The constitutive equation has some conservative property to be preserved? I do not think so!
- The equations are of hyperbolic nature. Except for the advection term the time evolution of **A** at a certain point does not depend on the value of **A** at its vicinity. So boundary conditions are irrelevant unless the fluid enters in the computational domain.

Equations

For the Oldroyd-B constitutive model since $\mathbf{f_R}(\mathbf{A}) = \mathbf{f_S}(\mathbf{A}) = \mathbf{A} - \mathbf{I}$

$$\overset{\triangledown}{\mathbf{A}} = -\frac{\mathbf{f_R}(\mathbf{A})}{\lambda}$$

$$\boldsymbol{\tau_p} + \lambda \overset{\triangledown}{\boldsymbol{\tau_p}} = 2\mu_p \mathbf{D}$$

Boundary Conditions

Wall:

$$\tau_{nn} = 0, \quad \text{and} \quad \partial_t(\tau_{nt}) + \lambda\tau_{nt} = \mu_p \partial_n(u_t)$$

$$\tau_{\theta\theta} = 0$$

Axis of symmetry:

$$\partial_r \tau_{\theta\theta} = \partial_r \tau_{rr} = \partial_r \tau_{zz} = 0, \quad \text{and} \quad \tau_{rz} = 0\,.$$

M.F. Tomé *et al.* Journal of Non-Newtonian Fluid Mechanics, 141(2-3):148–166,(2007).

The High Weissemberg Number Problem

Numerical schemes have upper limits in λ .
(Wi is its dimensionless counterpart).

$$\phi_t + a(x)\phi_x - b(x)\phi = -\phi/Wi$$

$$\phi_j^{n+1} = \underbrace{\left[1 - \frac{a_j \Delta t}{\Delta x} + \Delta t \left(b_j - \frac{1}{Wi}\right)\right]}_{\text{Stable if} <= 1} \phi_j^n + \left[\frac{a_j \Delta t}{\Delta x}\right] \phi_{j-1}^n$$

Stable if <= 1

$$\Delta x \leq \frac{a_j}{b_j - Wi^{-1}}$$

Note that in corners:
Decelerations ($b_j < 0$)
And low velocities ($a_j \simeq 0$)

Fattal, R., & Kupferman, R. (2005). *Journal of Non-Newtonian Fluid Mechanics*, *126*(1), 23–37. http://doi.org/10.1016/j.jnnfm.2004.12.003

Solutions to the HWNP

Changes of variables = kernels

Log kernel

Fattal, R., & Kupferman, R. (2005)

$$\Psi = \log \mathbf{A} = \mathbf{R} \log \Lambda \, \mathbf{R}^T$$

$$(\nabla \mathbf{u})^T = \Omega + \mathbf{B} + \mathbf{N}\mathbf{A}^{-1}$$

$$\partial_t \Psi + \mathbf{u} \cdot \nabla \Psi - 2\mathbf{B} - (\Omega\Psi - \Omega\Psi) = -\frac{e^{-\Psi}}{\lambda} \mathbf{f_R}(e^{\Psi})$$

Square root kernel

Balci, et al, C. R. (2011).
http://doi.org/10.1016/j.jnnfm.2011.02.008

$$\mathbf{A} = \mathbf{b}\mathbf{b}^{\mathbf{T}}$$

$$\partial_t \mathbf{b} + \nabla \cdot (\mathbf{u}\mathbf{b}) = \mathbf{b} \cdot \nabla \mathbf{u} + \mathbf{a} \cdot \mathbf{b} - \frac{\mathbf{b}^{-1} \, \mathbf{f_R}(\mathbf{b} \cdot \mathbf{b}^T)}{\lambda}$$

Summary

1.Motivation

2.Equations

3.Numerical Scheme

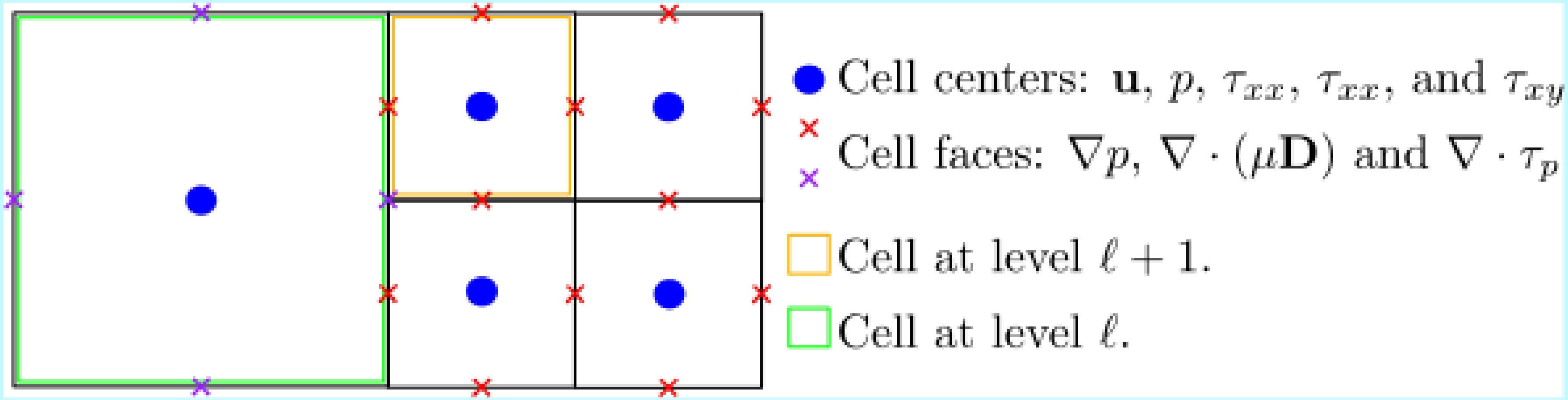
1. **Classic approach**
2. **Log kernel**
3. **Square root kernel**

4. Test problems

5.Further improvement

Numerical Scheme

- We use the incompressible Navier--Stokes centered formulation solver. (“centered.h”)
- Solvent stress is treated as it is, an standard viscosity term.
- Polymeric stresses are added as an acceleration. Stress components are defined at cell centers.



Numerical Scheme

Classic approach

$$\boldsymbol{\tau_p} + \lambda \overset{\nabla}{\boldsymbol{\tau_p}} = 2\mu_p \mathbf{D}$$

1. The stress components are advected explicitly with the BCG scheme,

$$\boldsymbol{\tau}_p^* = \boldsymbol{\tau}_p^{n-1/2} + \Delta t \, \nabla \cdot (\boldsymbol{\tau}_p^n \mathbf{u}^n)$$

2. The upper convective derivative is solved implicitly,

$$\left(1 + \frac{\lambda}{\Delta t}\right) \boldsymbol{\tau}_p^{n+1/2} - (\nabla \mathbf{u}^T)^n \, \boldsymbol{\tau}_p^{n+1/2} + \boldsymbol{\tau}_p^{n+1/2} \nabla \mathbf{u}^n = 2\mu_p D + \lambda \frac{\boldsymbol{\tau}_p^*}{\Delta t}$$

$$\mathbf{a}^{n+1/2} = \frac{\nabla \cdot \boldsymbol{\tau}_p^{n+1/2}}{\rho}$$

Numerical scheme

Log Kernel

We use a time-split scheme

$$\partial_t \Psi + \mathbf{u} \cdot \nabla \Psi = 0$$

$$\partial_t \Psi - 2\mathbf{B} - (\Omega\Psi - \Psi\Omega) = 0$$

$$\partial_t \Psi = \frac{e^{-\Psi}\mathbf{f}_{\mathbf{R}}(e^{\Psi})}{\lambda}$$

Hao, J. & Pan, T.W., (2007). Applied Mathematics Letters, 20(9), pp.988–993.

Numerical scheme

Log Kernel

Given $\tau_p^{n-1/2}$ and $\mathbf{u}^n$

1.- $\tau_p^{n-1/2} = \frac{\lambda}{\mu_p}\mathbf{f_S}(\mathbf{A}^{n-1/2})$ where $\mathbf{f_{S,R}}(\mathbf{A}) = \eta_{S,R}(\nu_{S,R}\mathbf{A} - \mathbf{I})$

2.- Diagonalize $\mathbf{A} = \mathbf{R}\,\Lambda\,\mathbf{R}^T$, calculate $\Psi^{n-1/2} = \mathbf{R}\,\log(\Lambda)\,\mathbf{R}^T\Big|^{n-1/2}$

3.-Calculate B^n and Ω^n

4.-The log-conformation tensor is advected using the BCG scheme.

$$\Psi^* = \Psi^{n-1/2} - \Delta t \nabla \cdot (\mathbf{u}^n \Psi^n)$$

5.- Ψ is upper advected

$$\Psi^{**} = \Psi^* + \Delta t(2\mathbf{B}^n + \Omega^n \Psi^{n-1/2} - \Psi^{n-1/2}\Omega^n)$$

6.- Obtain $\mathbf{A}^{**}$,

$$\Psi^{**} = \mathbf{R}\,\log(\Lambda)\,\mathbf{R}^T\Big|^{**} \quad \mathbf{A}^{**} = \mathbf{R}\,(\Lambda)\,\mathbf{R}^T\Big|^{**}.$$

7.-Finally, $\mathbf{A}^{n+1/2}$ is calculated analytically,

$$\mathbf{A}^{n+1/2} = \mathbf{A}^{**}\, e^{-\eta_R \nu_R \Delta t/\lambda} + (1 - e^{-\eta_R \nu_R \Delta t/\lambda})\frac{\mathbf{I}}{\nu_R}$$

8.- Finally, $\tau_p^{n+1/2} = \frac{\mu_p}{\lambda}\mathbf{f_R}(\mathbf{A}^{n+1/2}) = \frac{\mu_p}{\lambda}\eta_R(\nu_R \mathbf{A}^{n+1/2} - \mathbf{I})$

Numerical scheme

Square Root Kernel

Given $\mathbf{b}^{n-1/2}$ and $\mathbf{u}^n$

1.-The log-conformation tensor is advected using the BCG scheme.

$$\mathbf{b}^* = \mathbf{b}^{n-1/2} + \Delta t \, \nabla \cdot (\mathbf{b}^n \mathbf{u}^n)$$

2.- The rest of the equation is linearized and solved implicitly,

$$\frac{\mathbf{b}^{n+1/2}}{\Delta t} - \mathbf{b}^{n+1/2} \nabla \mathbf{u}^n - \mathbf{a}^n \cdot \mathbf{b}^{n+1/2} + \frac{\eta_R \nu_R}{\lambda} \mathbf{b}^{n+1/2} = \frac{\mathbf{b}^*}{\Delta t} + \frac{\eta_R}{\lambda} (\mathbf{b}^{-1})^{n-1/2}$$

3.- Finally, the polymeric stress is computed from $\mathbf{b}^{n+1/2}$

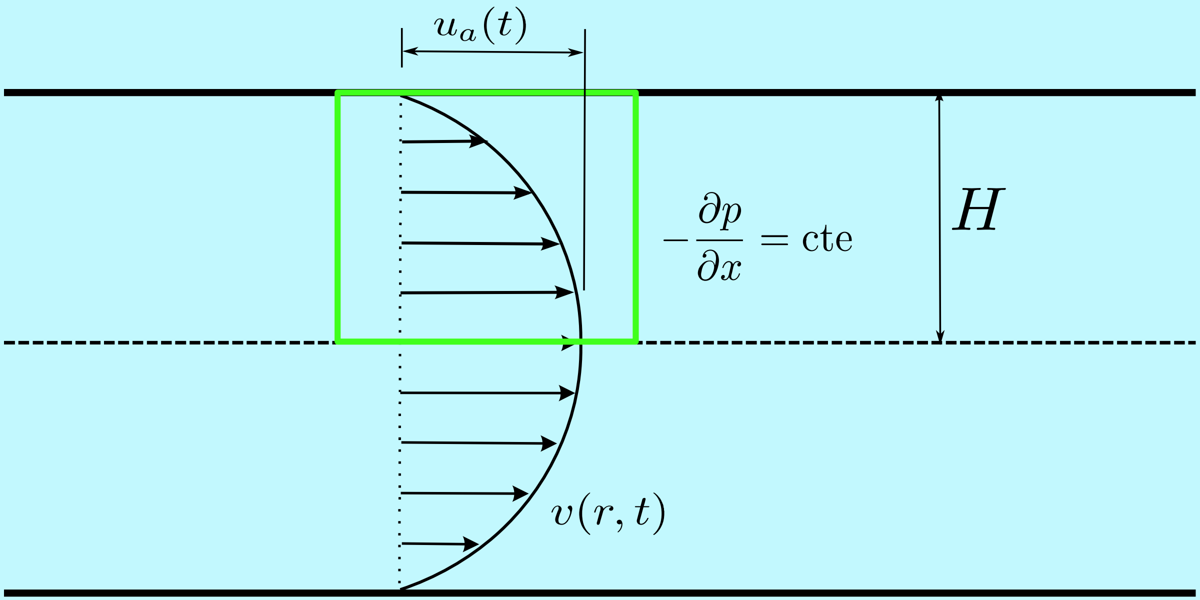
$$\mathbf{A}^{n+1/2} = (\mathbf{b}\mathbf{b}^{\mathbf{T}})^{n+1/2} \quad \text{and} \quad \tau_p^{n+1/2} = \frac{\mu_p}{\lambda} \mathbf{f_S}(\mathbf{A}^{n+1/2})$$

Summary

1. Motivation
2. Equations
3. Numerical Scheme
4. **Test problems**
   1. **Poiseuille Flow**
   2. **Drop in Couette flow**
   3. **Lid cavity**
   4. **Drop splashing**
5. Further improvements

Testing problems

Poiseuille flow



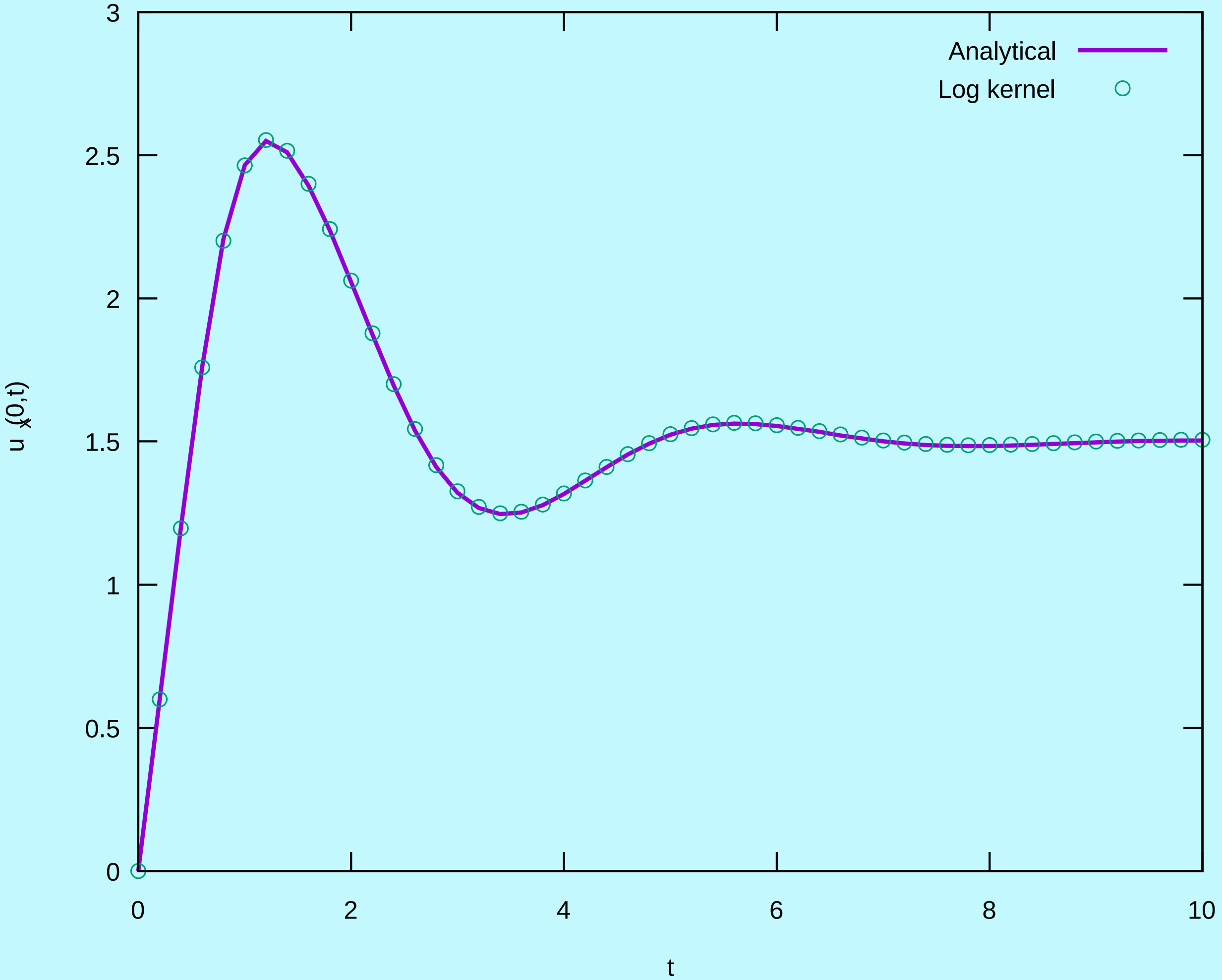
$$u(y,t) = 1.5(1-y^2) - 48\sum_{k=1}^{\infty}\frac{\sin((1+t)n/2)}{n^3}e^{\alpha_n t/2}G(t)$$

$$n = (2k-1)\pi,\ \alpha_n = 1+\beta\,E\,n^2/4 \text{ and } G(t) = \sinh(\theta_n t/2) + \tfrac{\gamma_n}{\theta_n}\cosh(\theta_n t/2)$$

$$\theta_n = \sqrt{\alpha_n^2 - E\,n^2} \quad \text{and} \quad \gamma_n = 1 - \frac{2-\beta}{4}\,E\,n^2$$

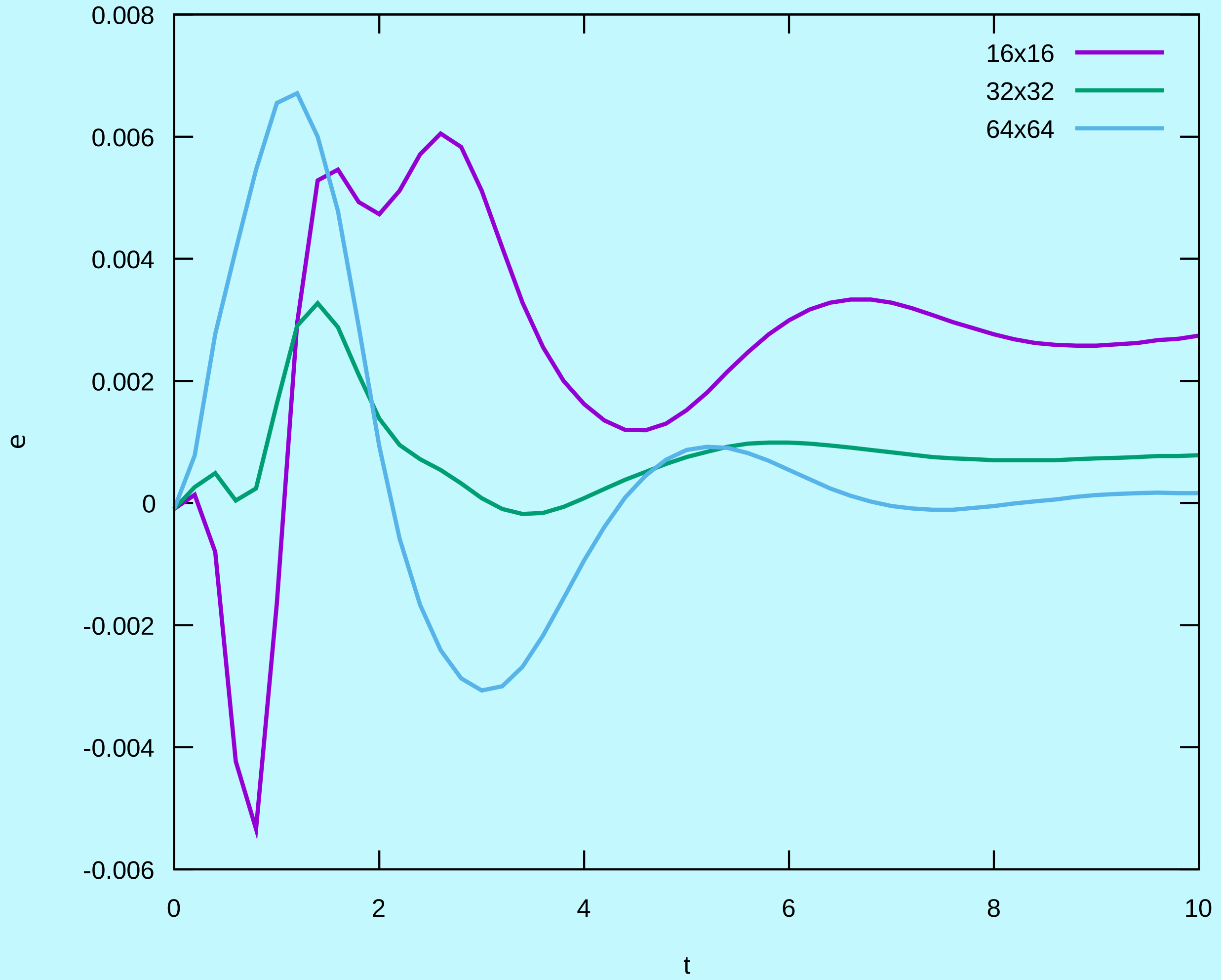
Testing problems

Poiseuille flow



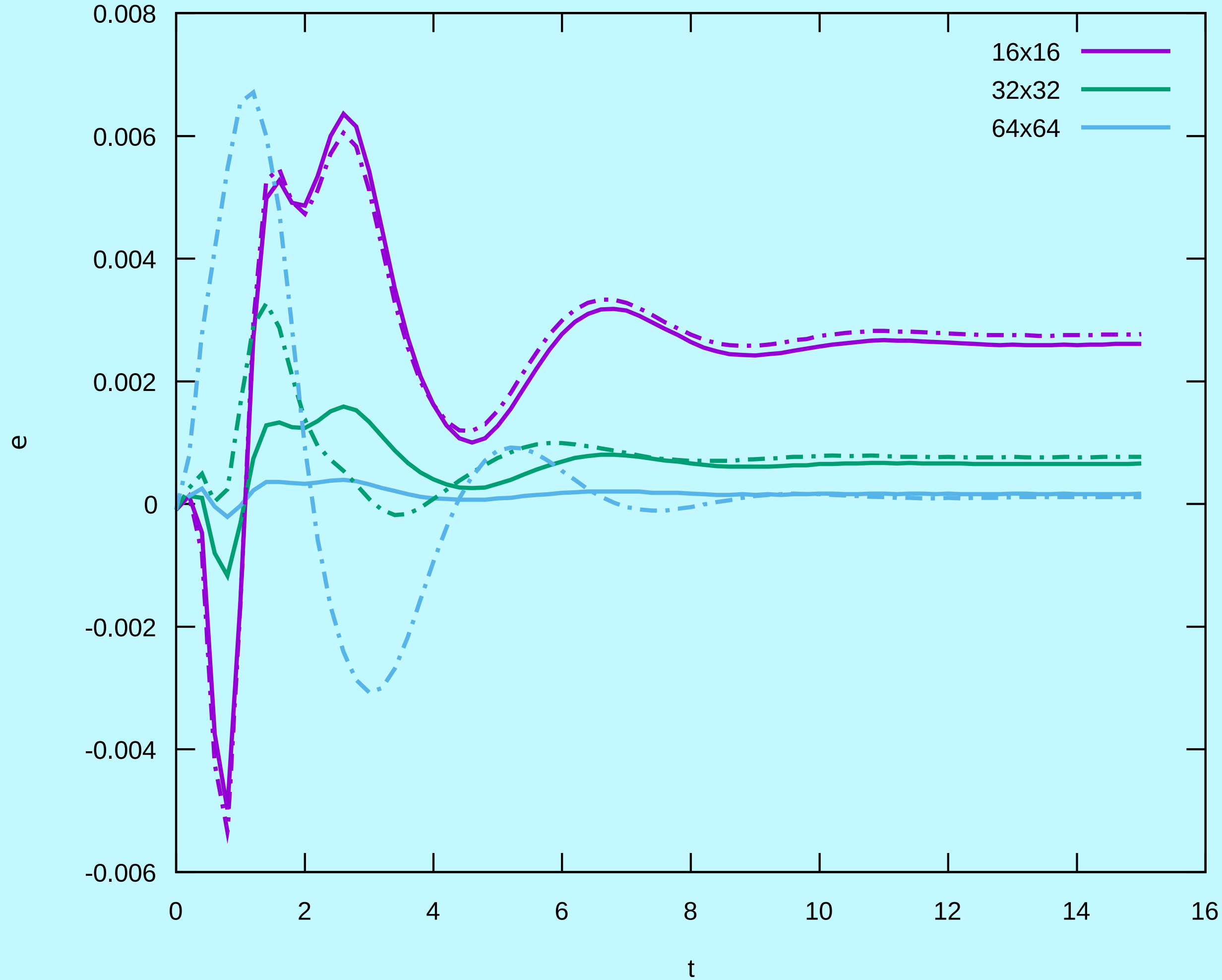
Testing problems

Poiseuille flow



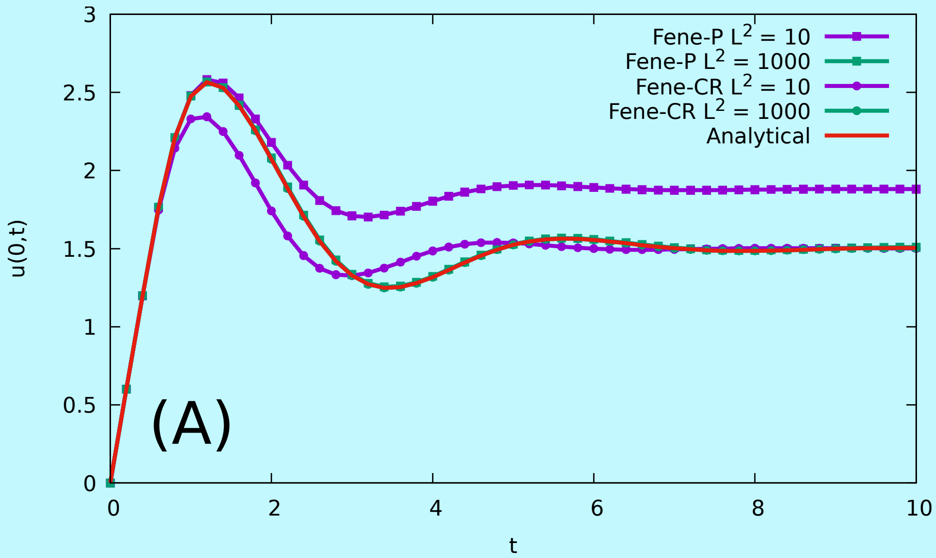
Testing problems

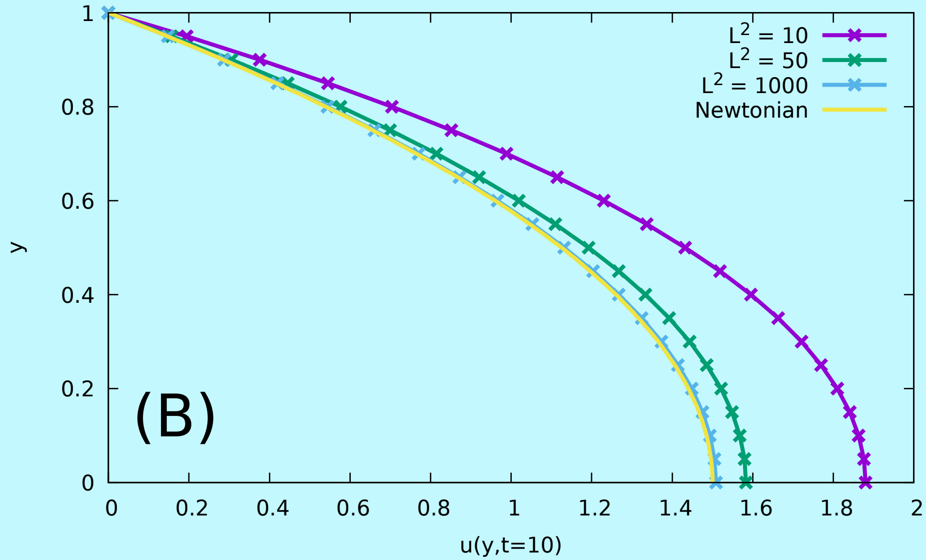
Poiseuille flow



Testing problems

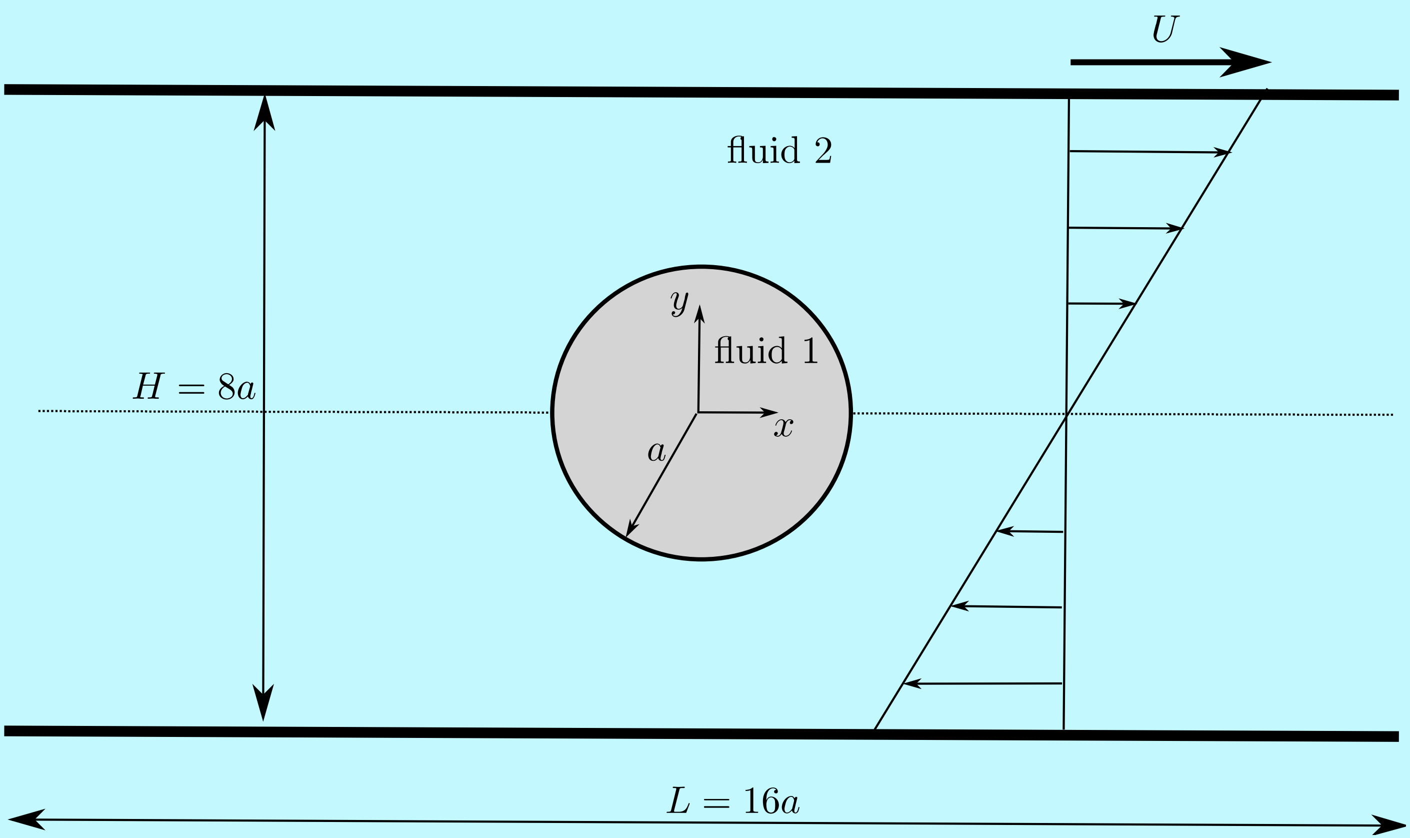
Poiseuille flow (FENE model)





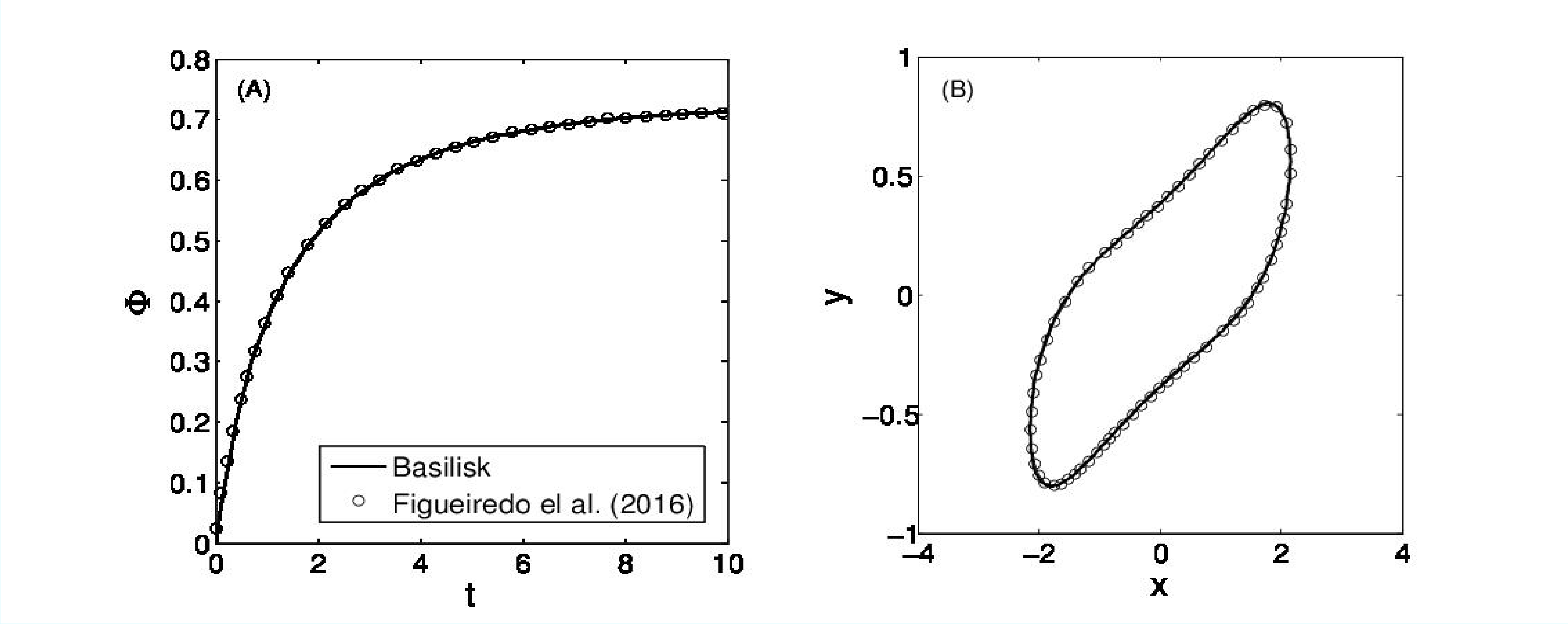
Numerical scheme

Drop in Couette flow



Numerical scheme

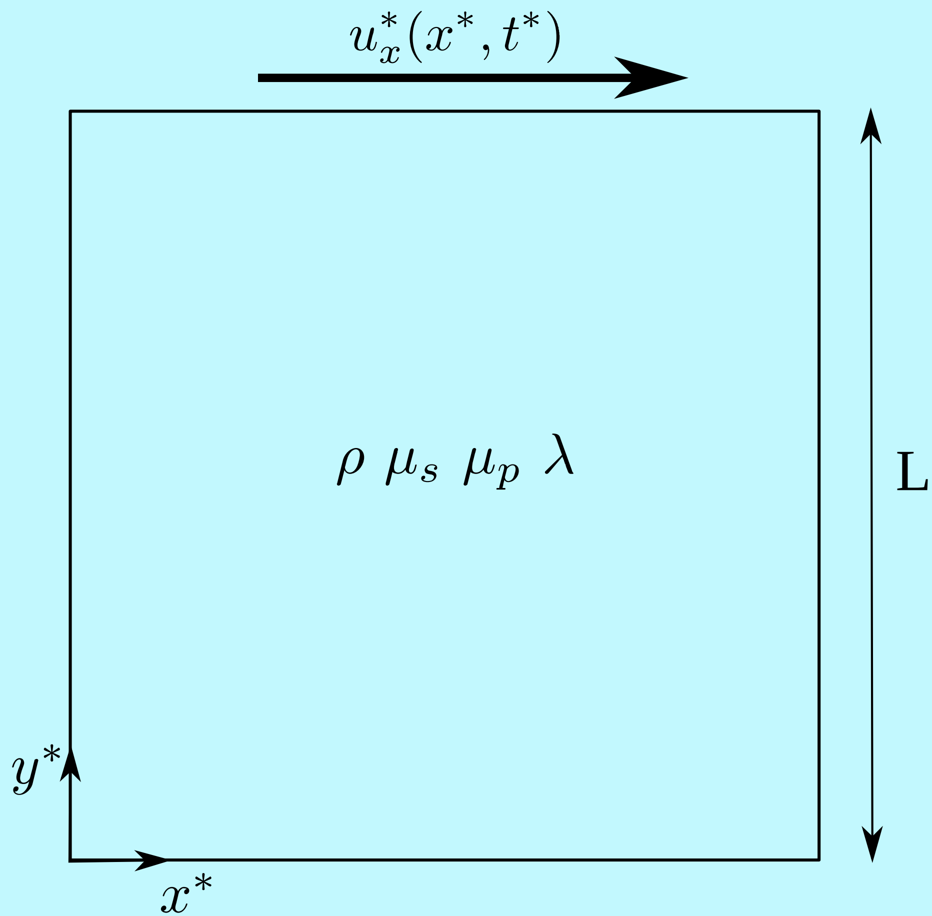
Drop in Couette flow



Testing problems

Lid cavity

$$u_x(x,t) = 8\left[1 + \tanh\left(8t - 4\right)\right] x^2 (1-x)^2$$

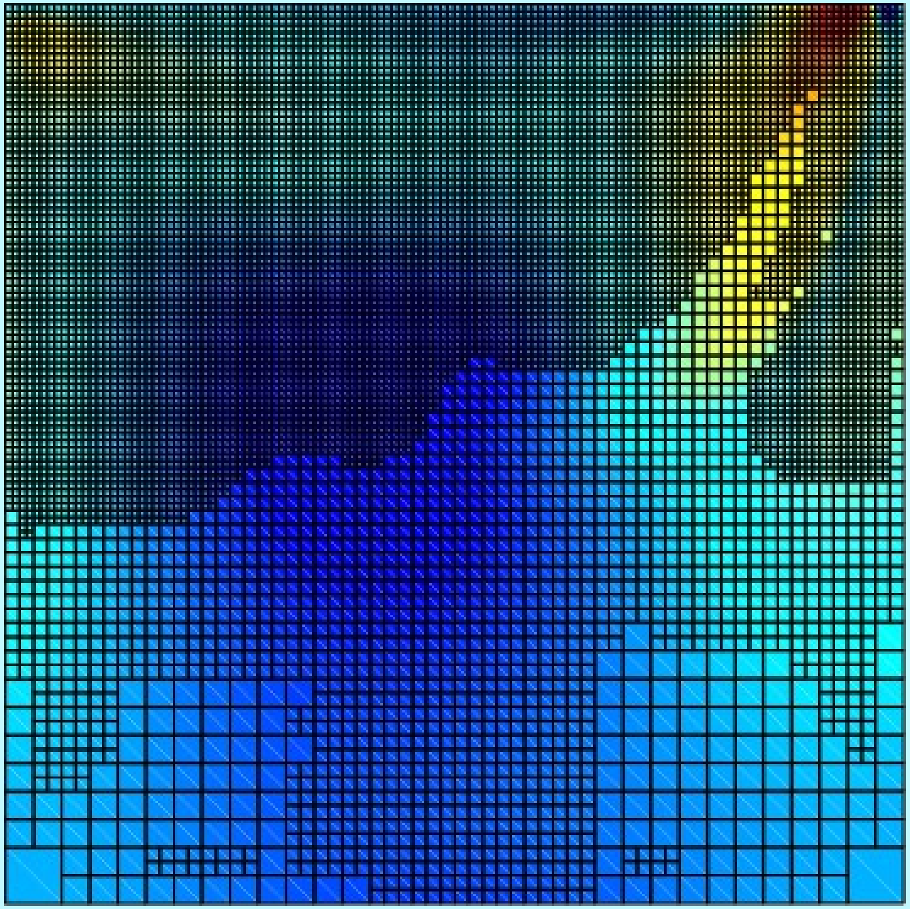


$$Wi = \frac{\lambda U_o}{L} \qquad Re = \frac{\rho U_o L}{\mu_o}$$

$$\beta = \frac{\mu_s}{\mu_o} \qquad \mu_o = \mu_s + \mu_p$$

Testing problems

Lid cavity



Distribution of Ψ_{xx}

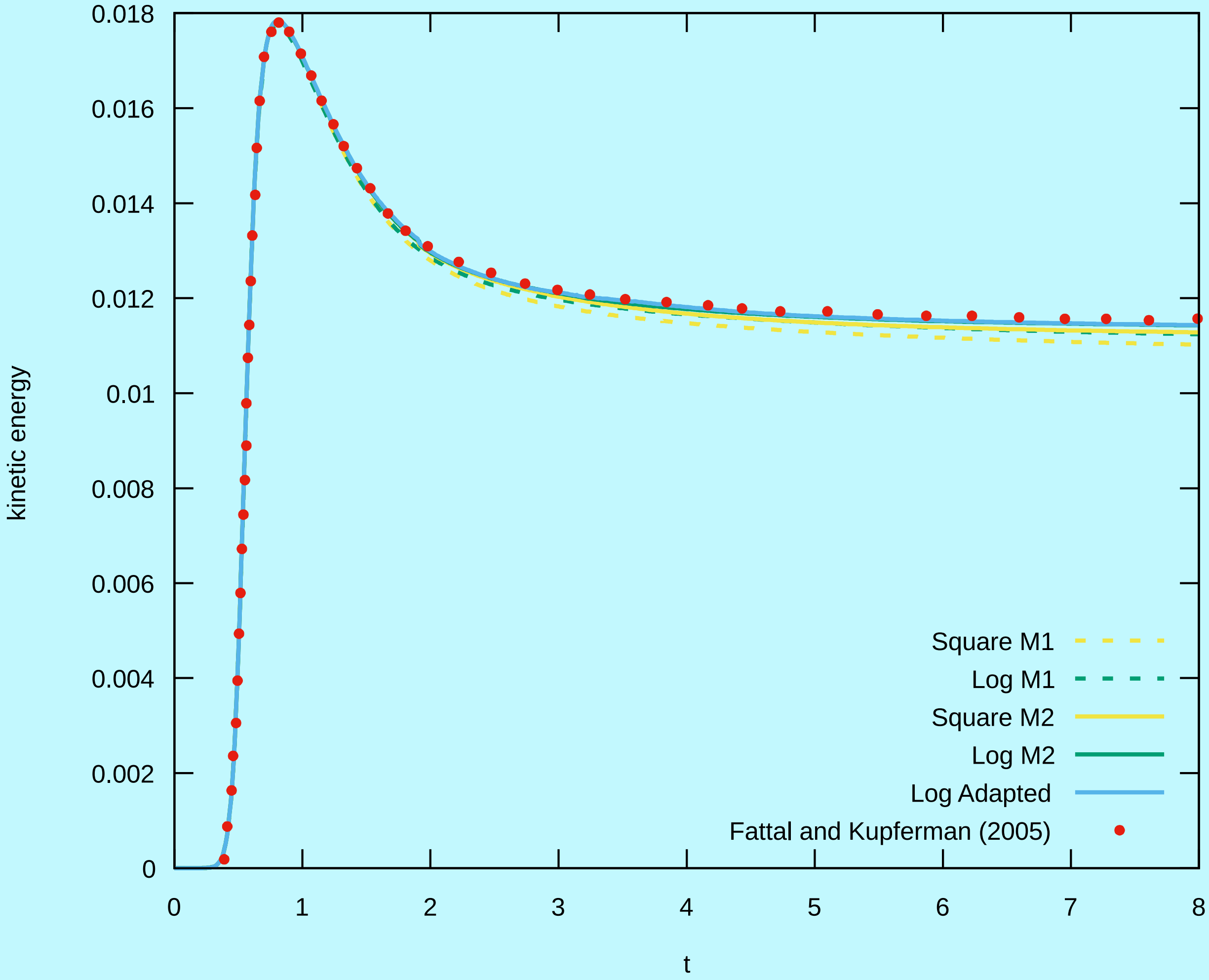
Adaption on velocity components each 50 steps

```
adapt_wavelet
 ({u.x, u.y}, (double[]){5e-4,5e-4}, 7, 5);
```

M1: Uniform 64 x 64 (Level 6)
M2: Uniform 128 x 128 (Level 7)

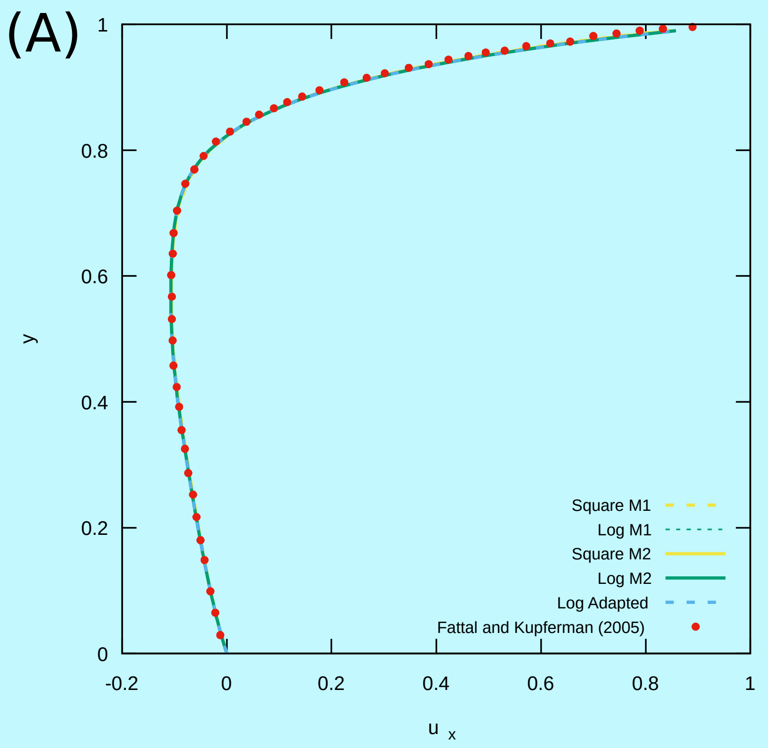
Testing problems

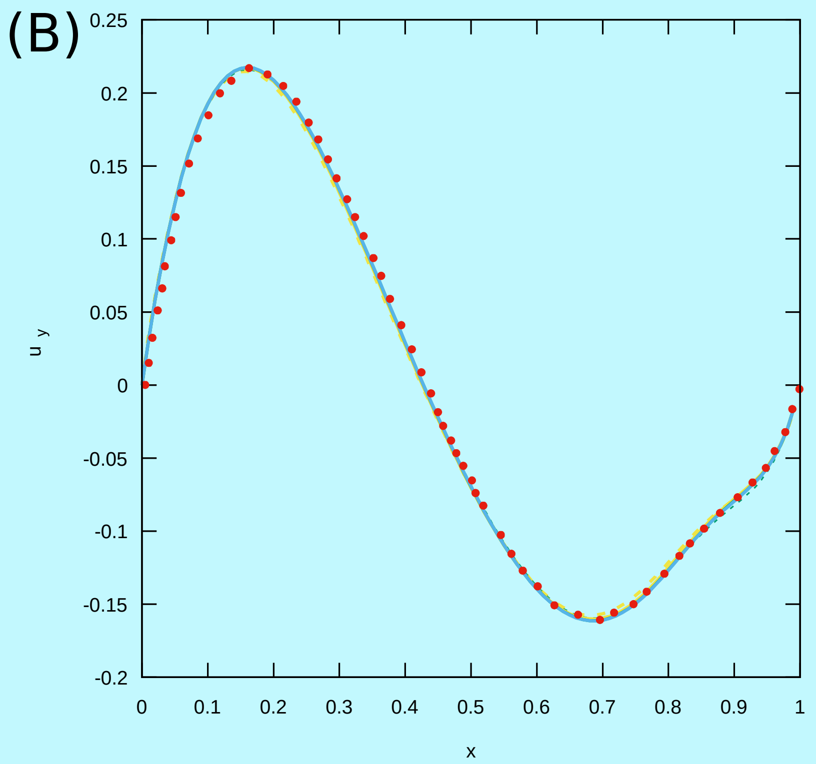
Lid cavity



Testing problems

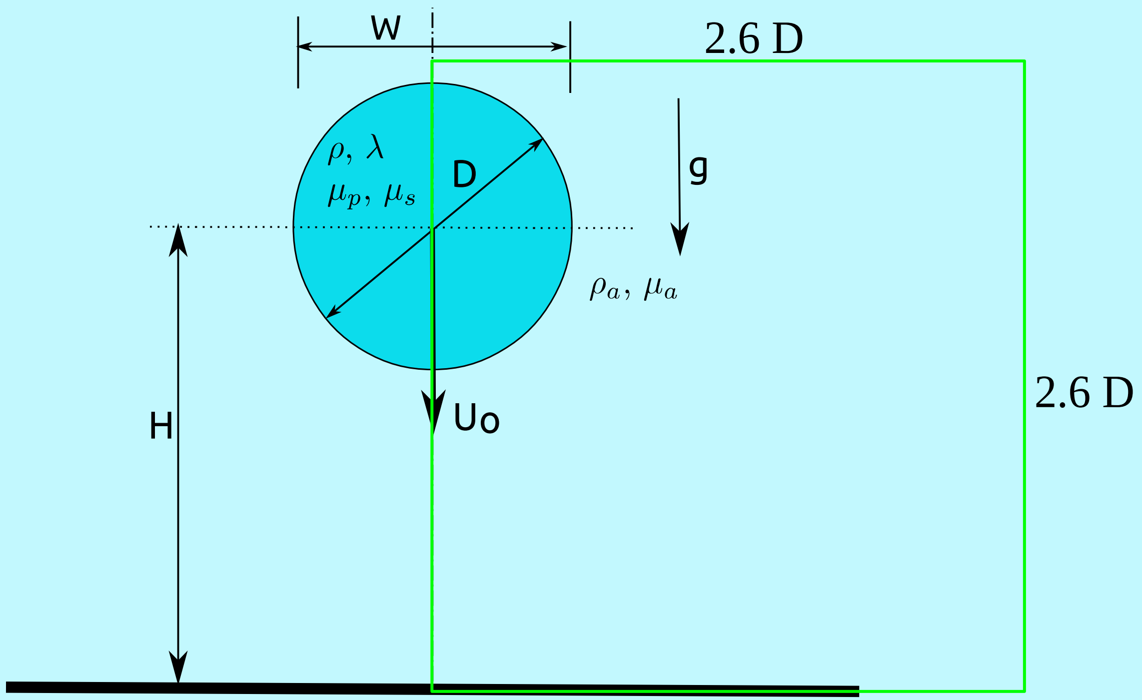
Lid cavity





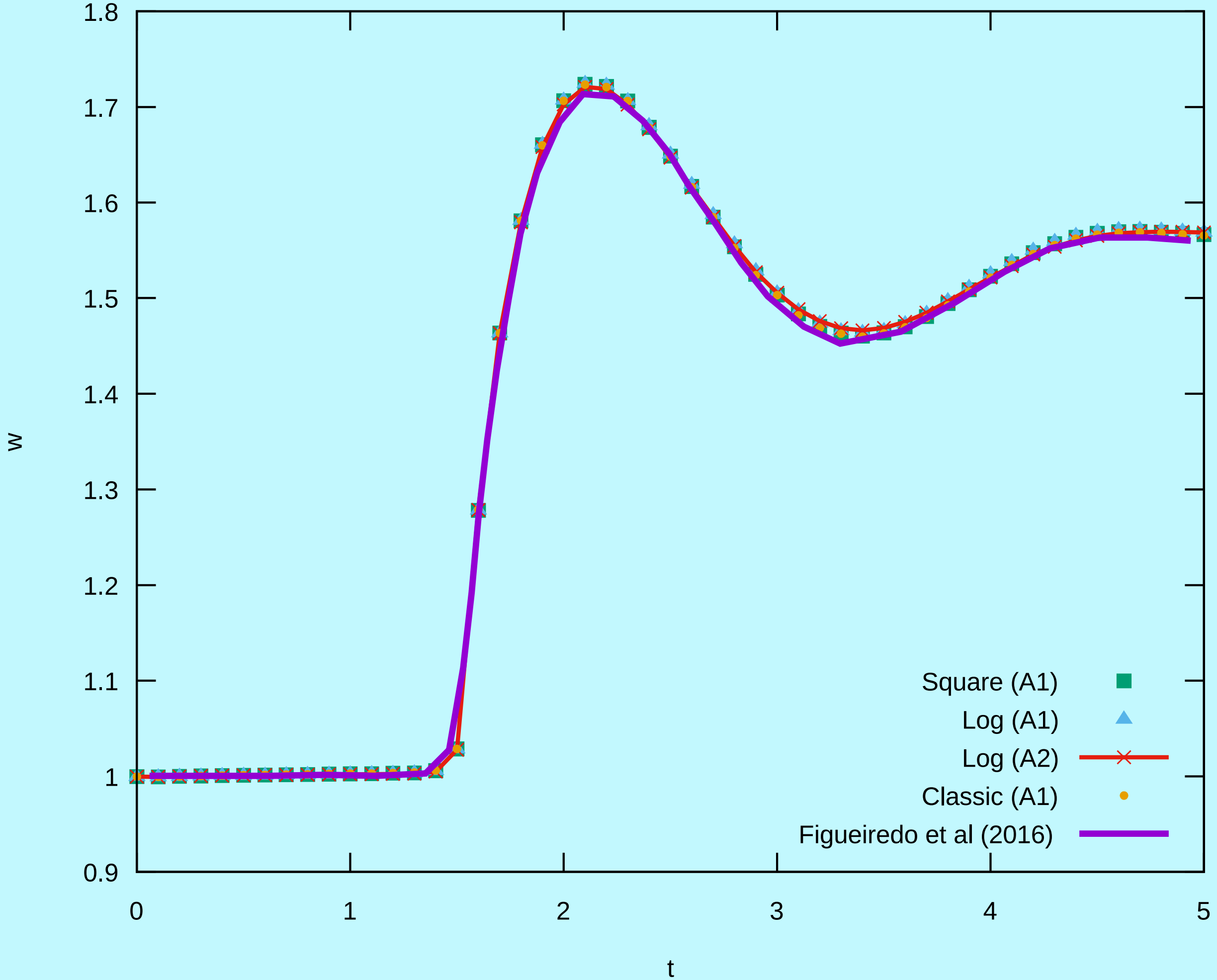
Testing problems

Drop impingement



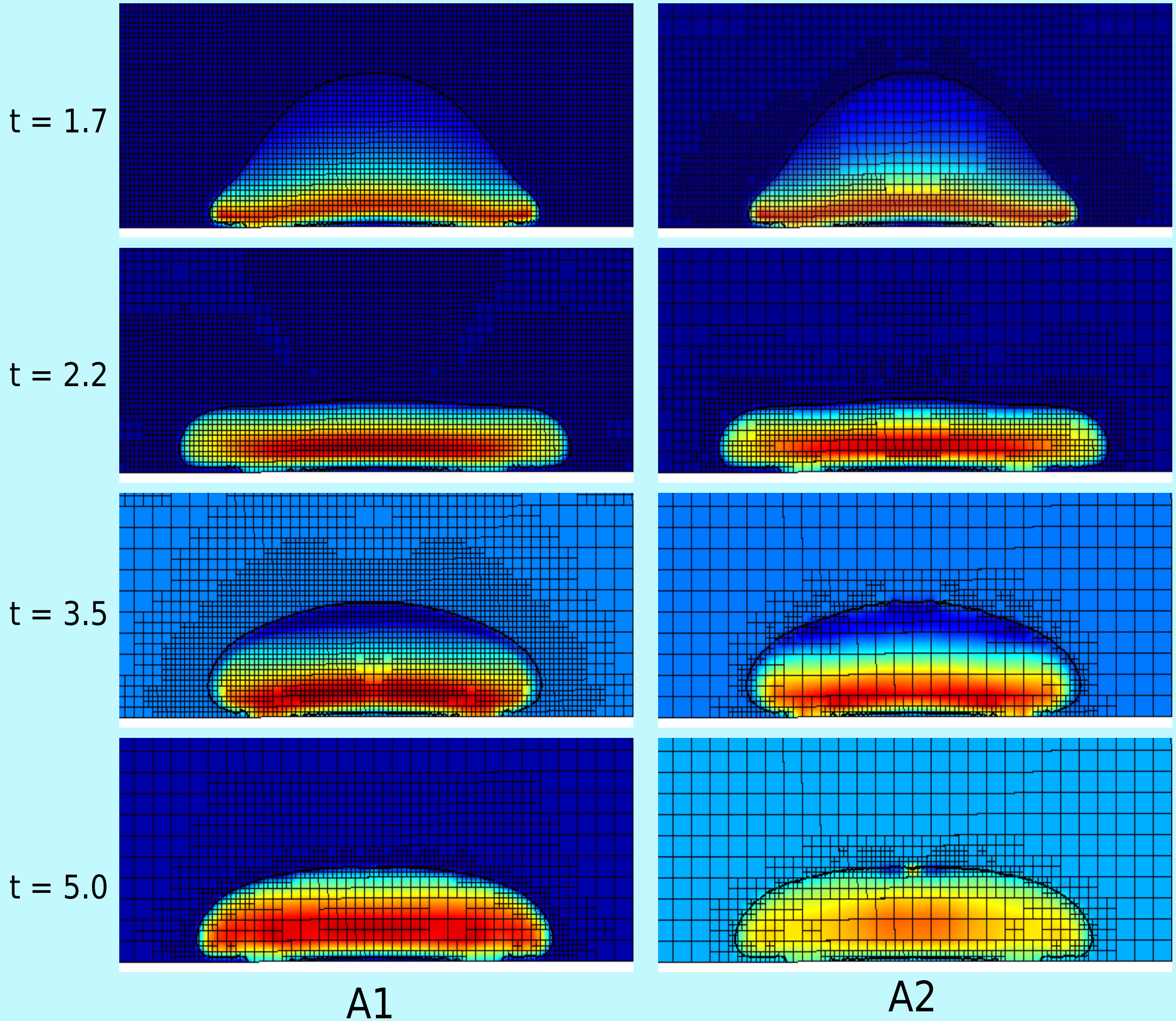
Testing problems

Drop impingement



Testing problems

Drop impingement



Testing problems

Drop impingement

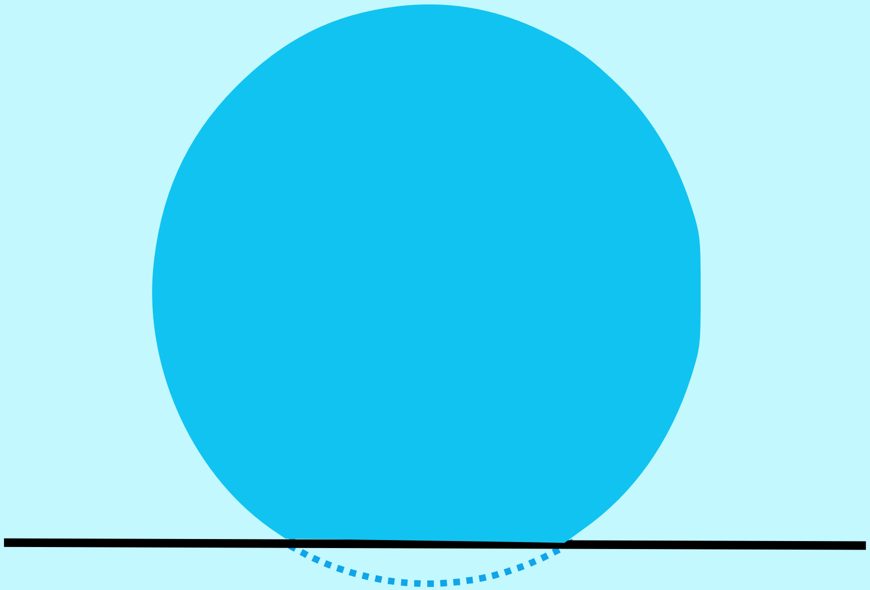
Water + PAA (1000 ppm)

$$Re_p = \frac{\rho U D}{\mu_p} = 602$$

$$We = \frac{\rho D U^2}{\sigma} = 760$$

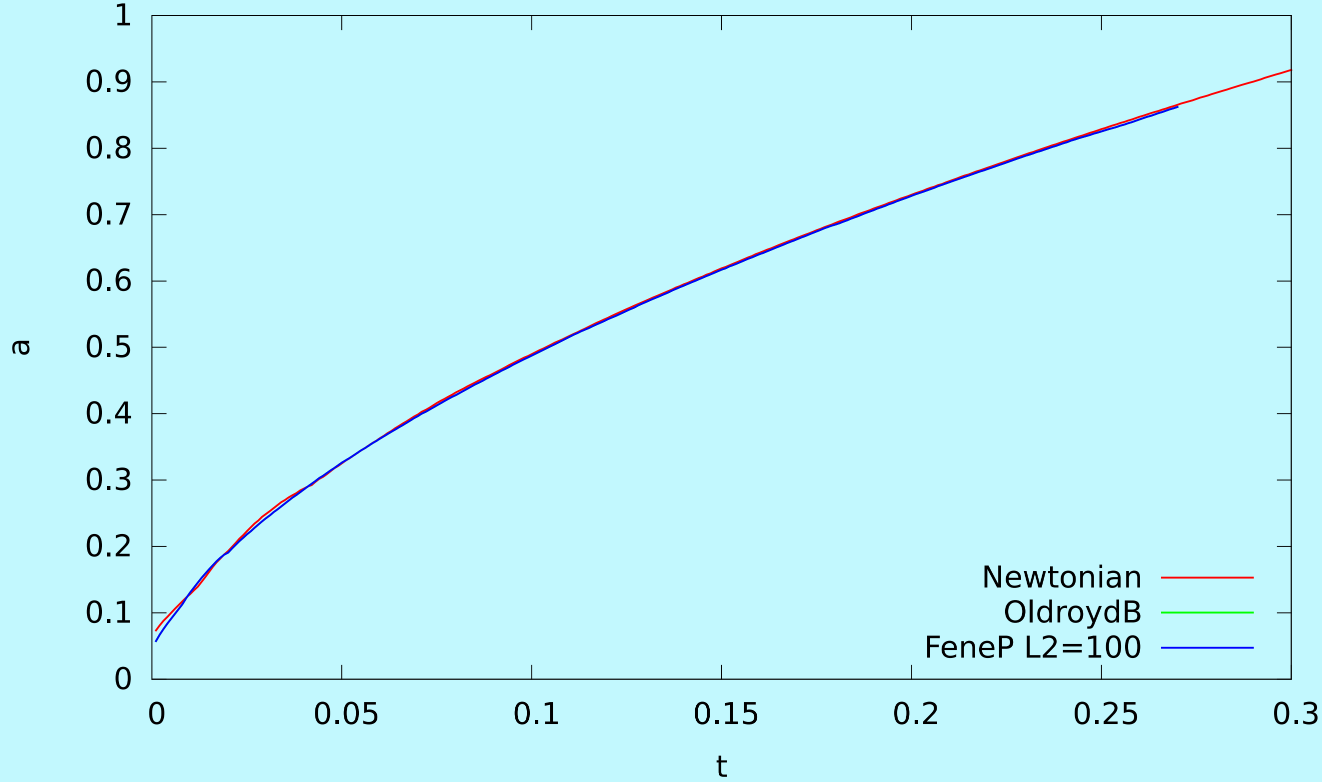
$$Re_s = \frac{\rho U D}{\mu_s} = 13383$$

$$De = \frac{\lambda U}{D} = 174.51$$



Testing problems

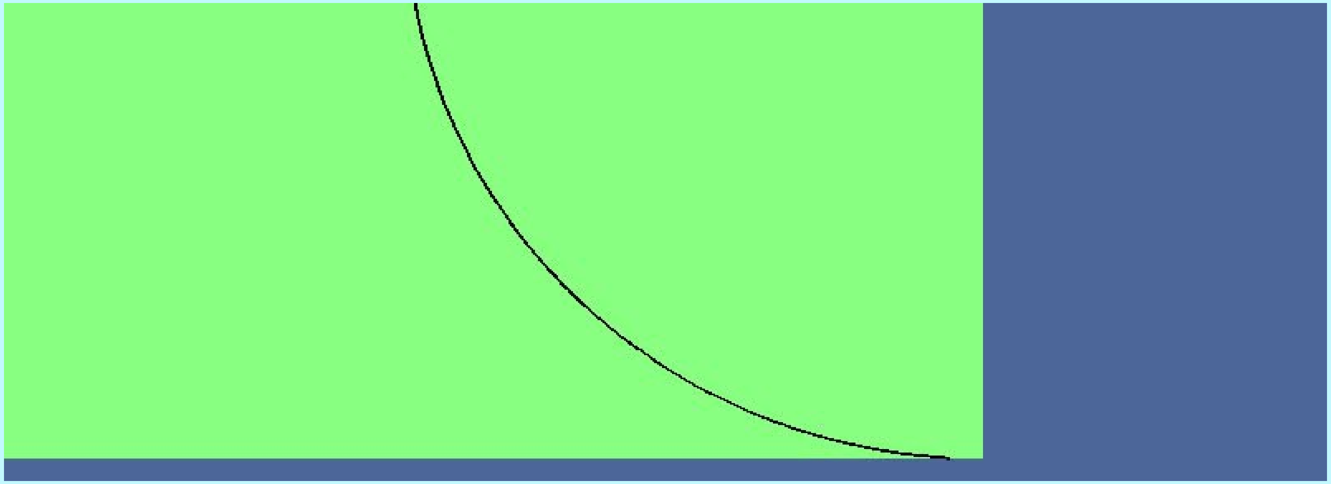
Drop impingement



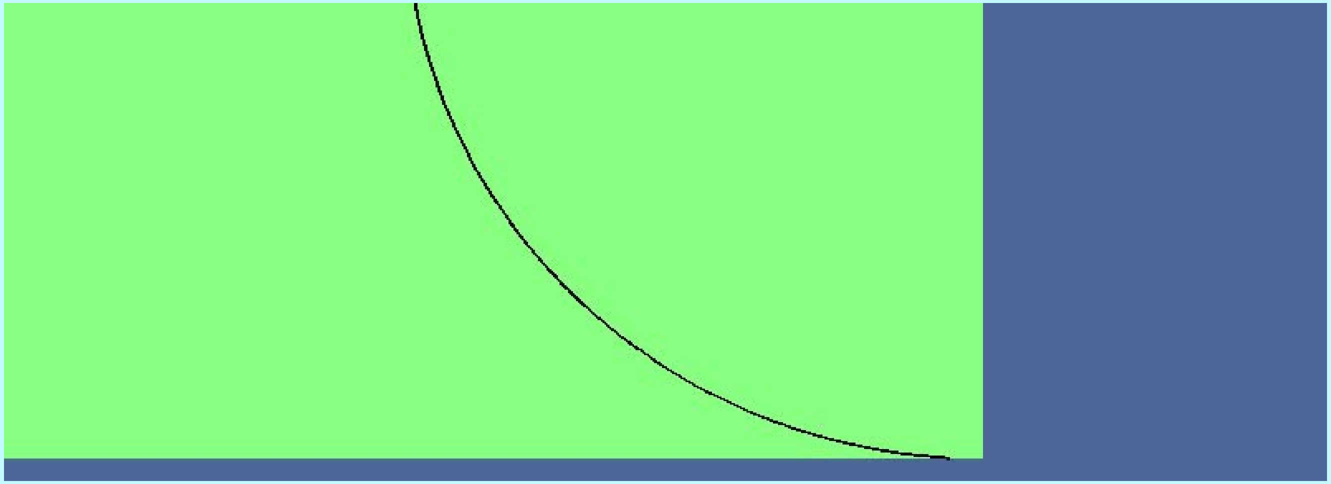
Testing problems

Drop impingement

f[left]=neuman(0);

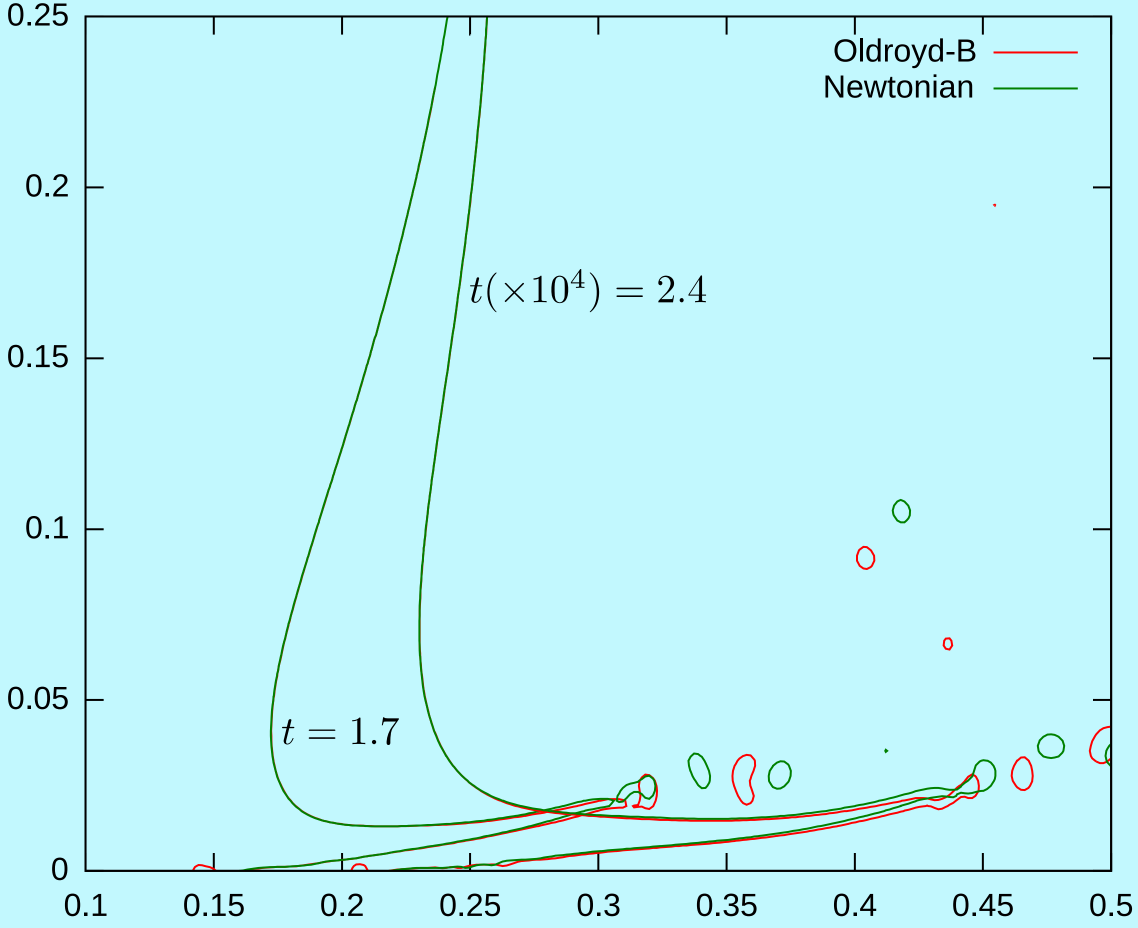


f[left]=dirichlet(1);



Testing problems

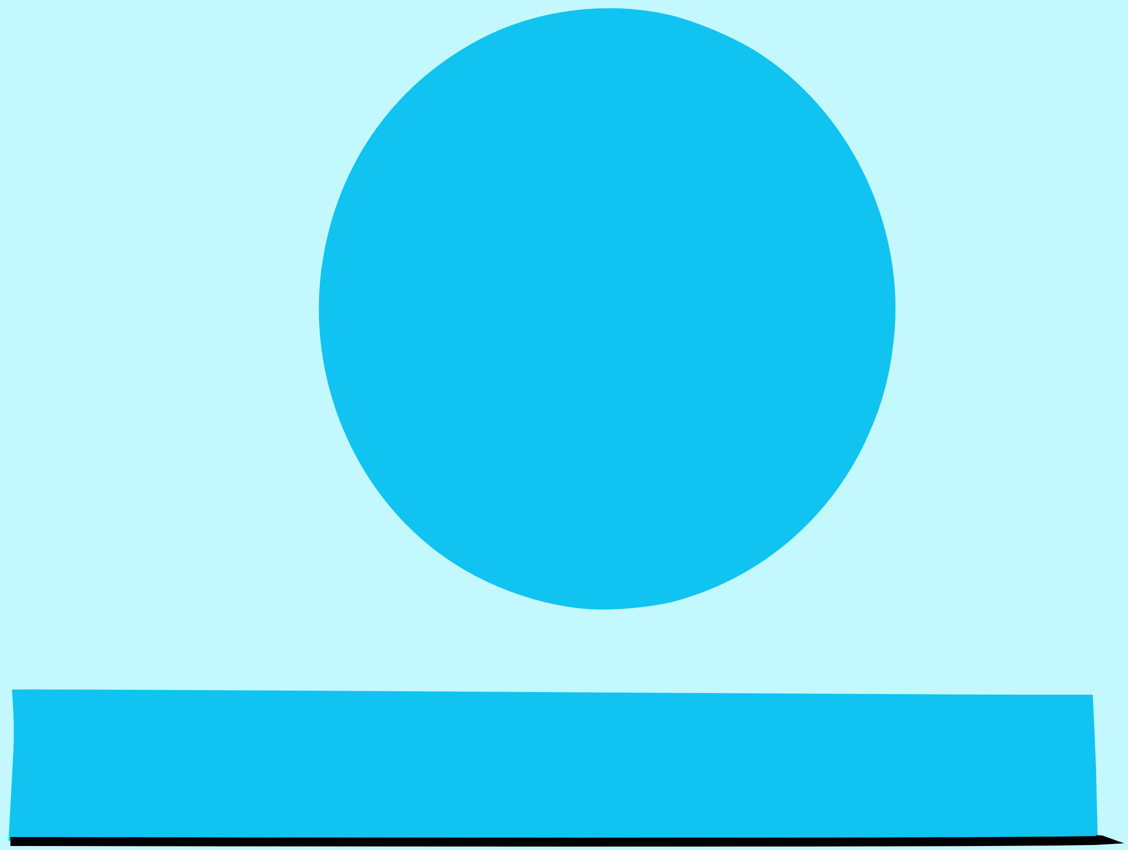
Drop impingement



Testing problems

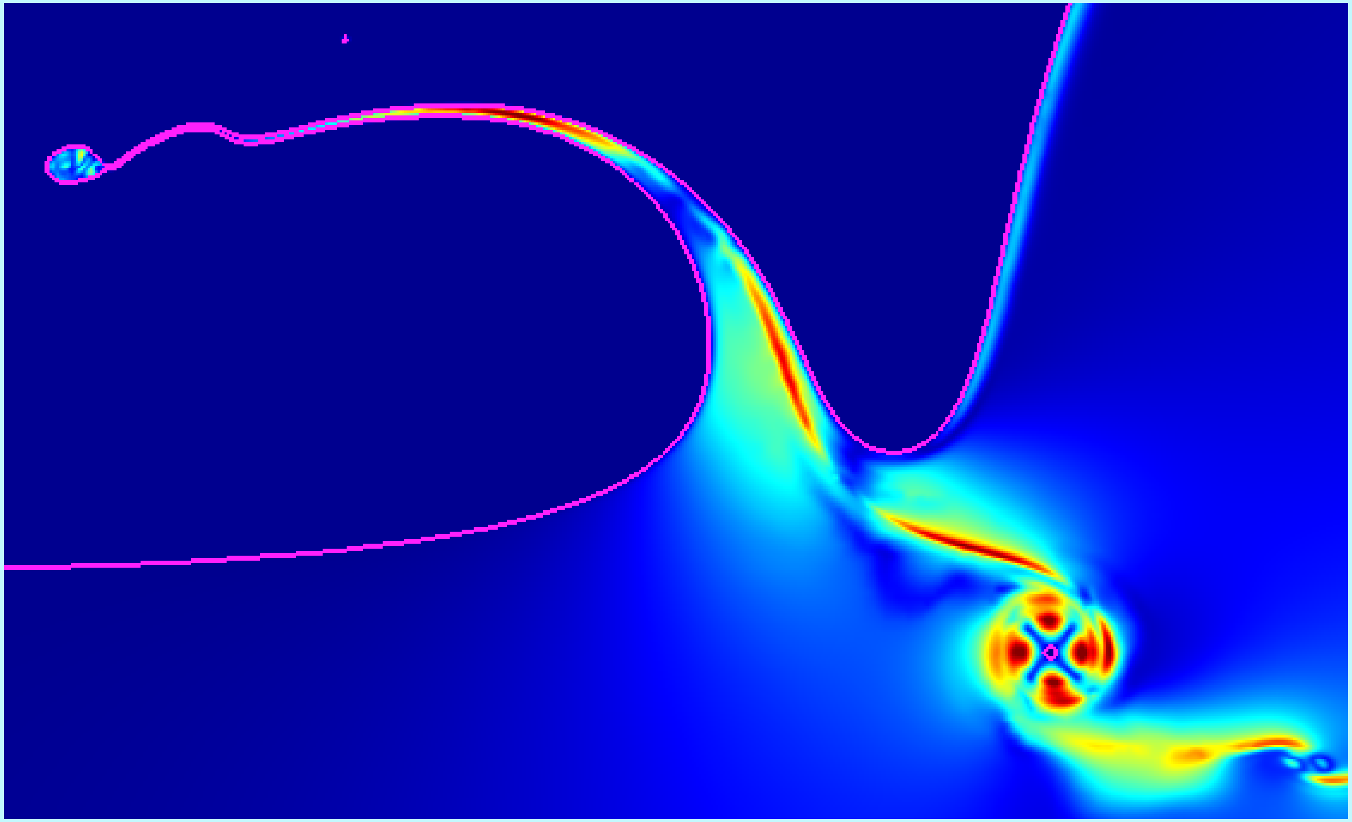
Drop impingement

Water + PAA (1000 ppm)



Testing problems

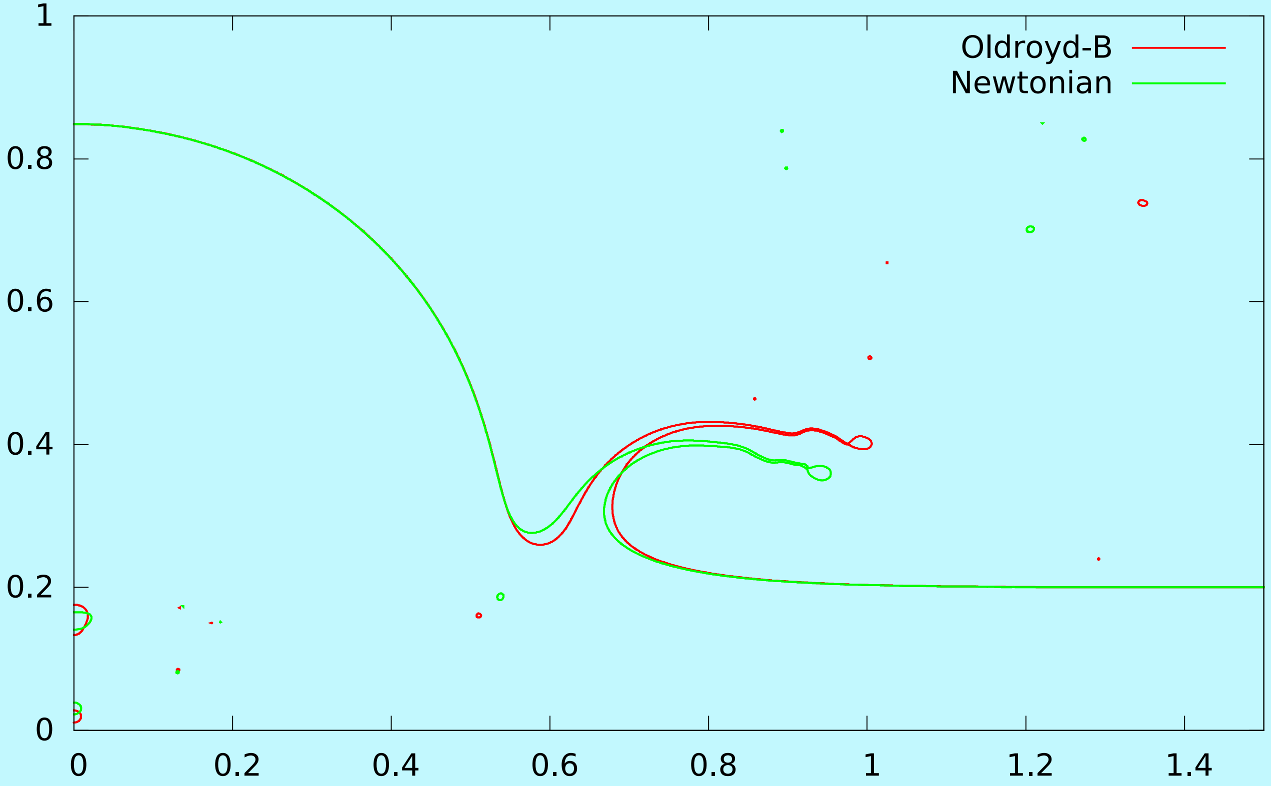
Drop impingement



$$||\Psi||_2 = \sqrt{\sum_{i,j} \Psi_{ij}^2}$$

Testing problems

Drop impingement



Summary

1.Motivation

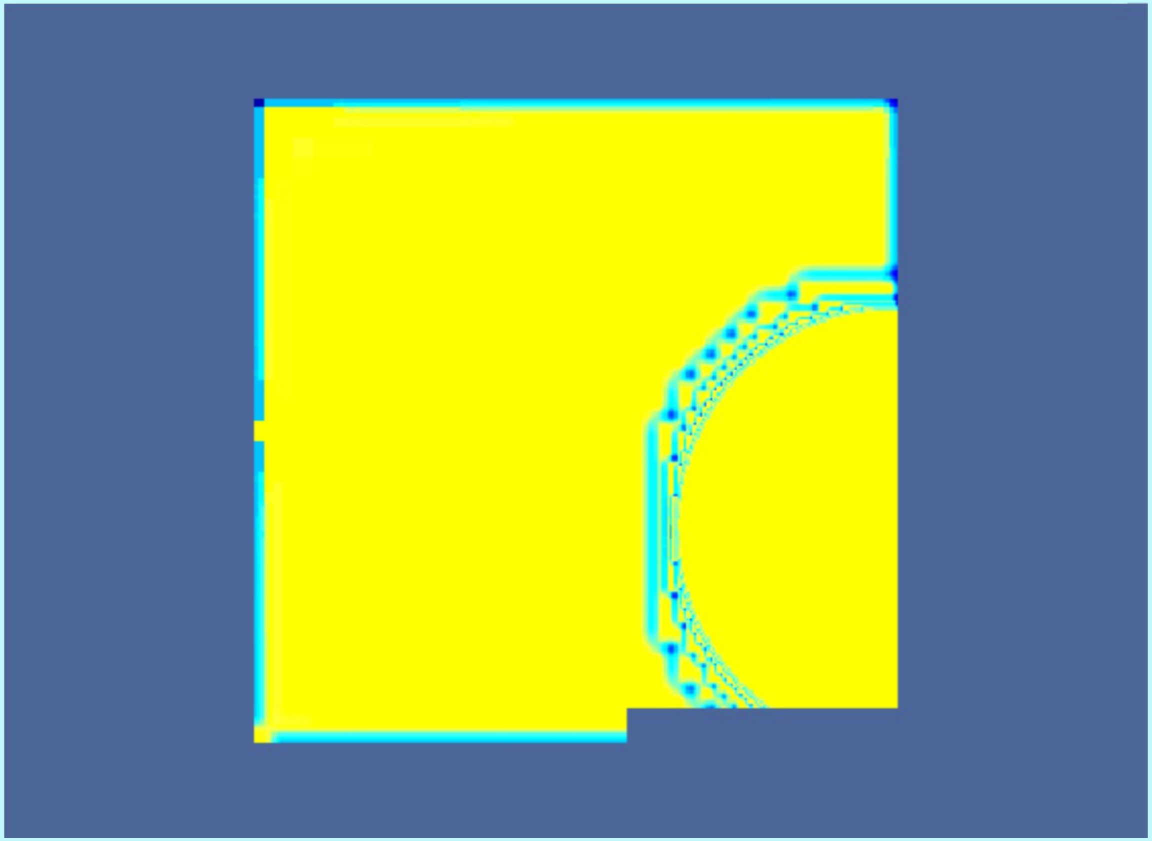
2.Equations

3.Numerical Scheme

4. Test problems

5.Further improvements

Problems



Further Improvements

Although the method seems to work reasonably well we can try further improvements.

- Move the off-diagonal terms to vertex. The problem with BC will be over.
- To use the WENO scheme for the advection. (High order scheme would help in the stabilization)
- Try the Both Side Diffusion (BSD)

$$\rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) - (\mu_s + \mu_p)(\nabla \mathbf{u} + \nabla \mathbf{u}^T) = \\ -\nabla p - \mu_p(\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \nabla \cdot \tau_p$$

- And of course, 3D

Simulating viscoelastic flows with Basilisk

J.M. López-Herrera
Universidad de Sevilla

S. Popinet, A. Castrejón-Pita & E. Vega

BGUM2017, *November 15th -16th, Princeton, EE.UU.*