The Jacobian Analytical Method (JAM)

MIGUEL A. HERRADA

ETSI, UNIVERSITY OF SEVILLE

Index

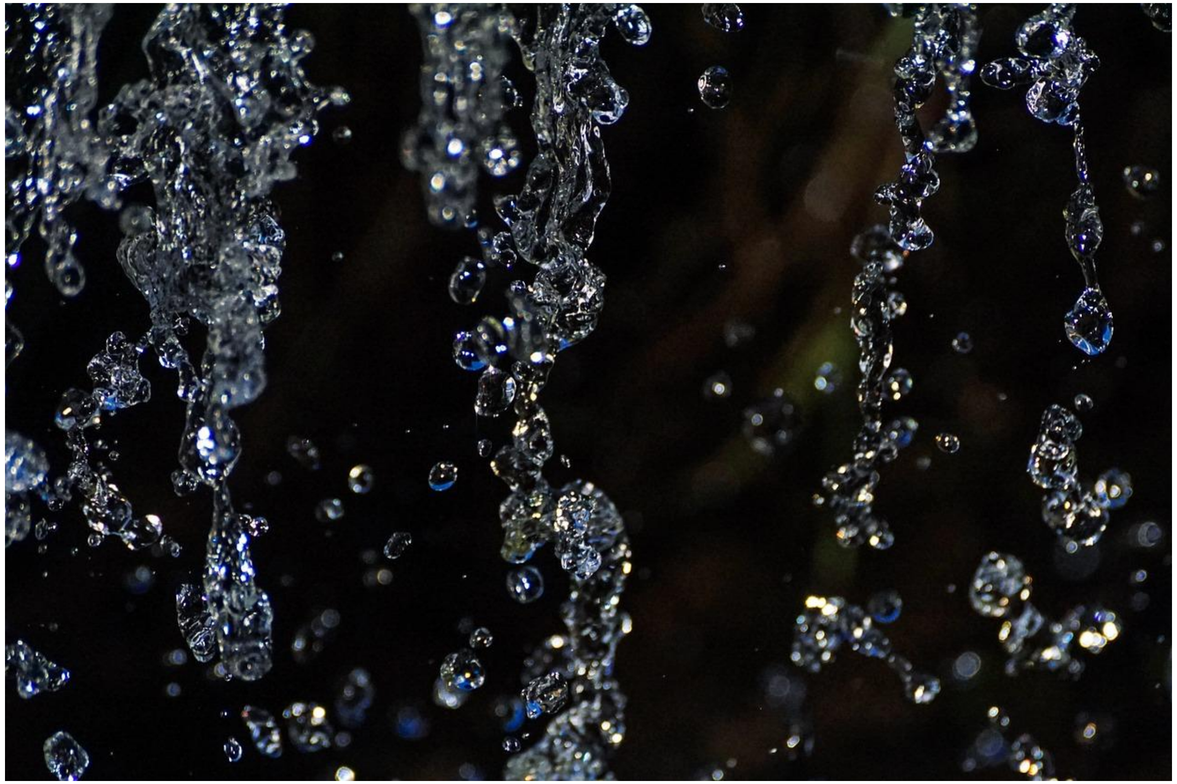
1. Introduction
2. JAM description and applications
3. Example. Burgers' problem
4. Implementation using JAM

Introduction

Complex non-linear problems

On the instability of jets (1878)

Lord Rayleigh



Introduction

- Nonlinear system:

$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}.$ x vector of N unknows.

- Solving the system using the Newton Method:

$\boldsymbol{DF}(\boldsymbol{x_o})\Delta \boldsymbol{x} = -\boldsymbol{F}(\boldsymbol{x_o}) \rightarrow \boldsymbol{x}_{new} = \boldsymbol{x}_o + \Delta \boldsymbol{x}.$ “x_o” guess solution *while* $|\Delta x| > \varepsilon$

Problems with the Jacobian Matrix **DF and Function F**

1. In the case of a mapped domain, it is not easy to obtain the expressions that are the basis of the function F.
2. If the matrix is obtained by applying the derivatives numerically, it is a dense matrix.
3. Once the matrix has been obtained numerically, it is difficult to separate it into pieces according to certain criteria, such as which is the temporal or spatial part of the matrix.

The Jacobian Analytical Method (JAM)

<u>*A numerical method to study the dynamics of capillary fluid systems*</u>

MA Herrada, JM Montanero - Journal of Computational Physics, 2016

J.M. Montanero

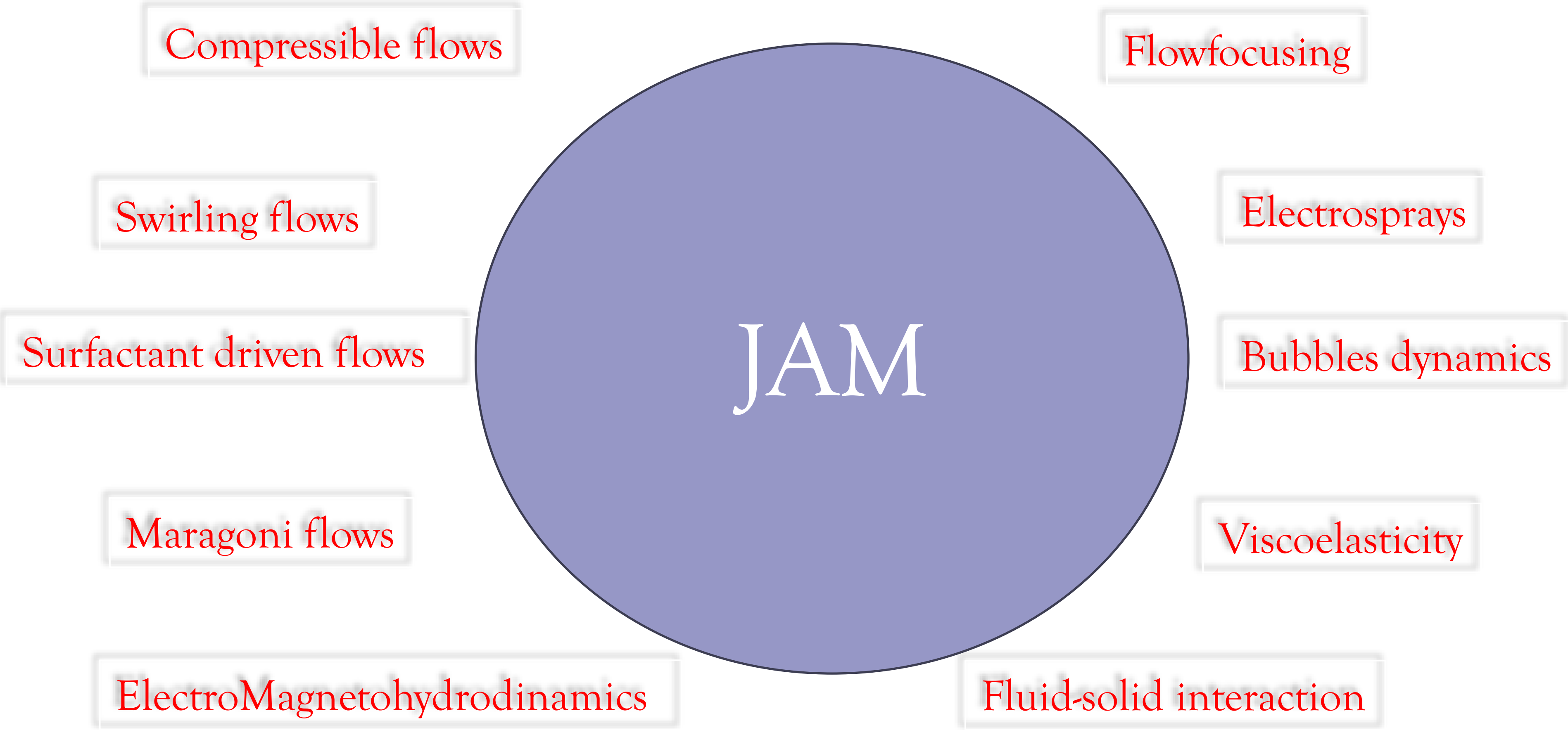
The key elements of the method

1. Used of a symbolic toolbox to compute the analytical Jacobians.
2. Used of sparse collocation matrices and analytical Jacobians to mount the numerical Jacobian matrix.
3. Extreme flexibility for using the Jacobian matrix to construct a generalised eigenvalue problem, allowing the study of the global stability for the nonlinear problem.
4. Use of analytical or elliptical mappings for accurate interphase tracking.

Limitations

1. Simply connected domains.
2. Not easy to parallelize.
3. Expensive for 3D problems.
4. No easy to explain!

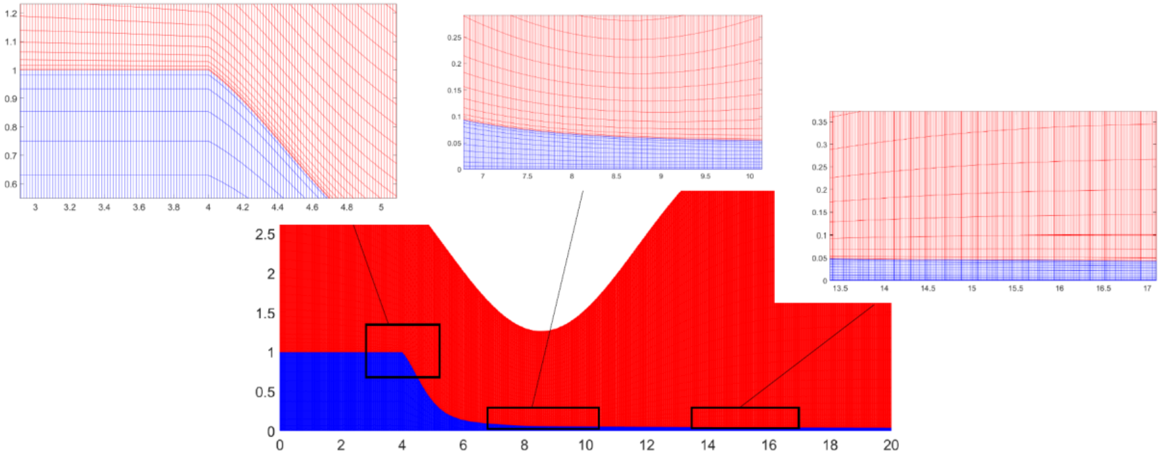
Applications



Applications

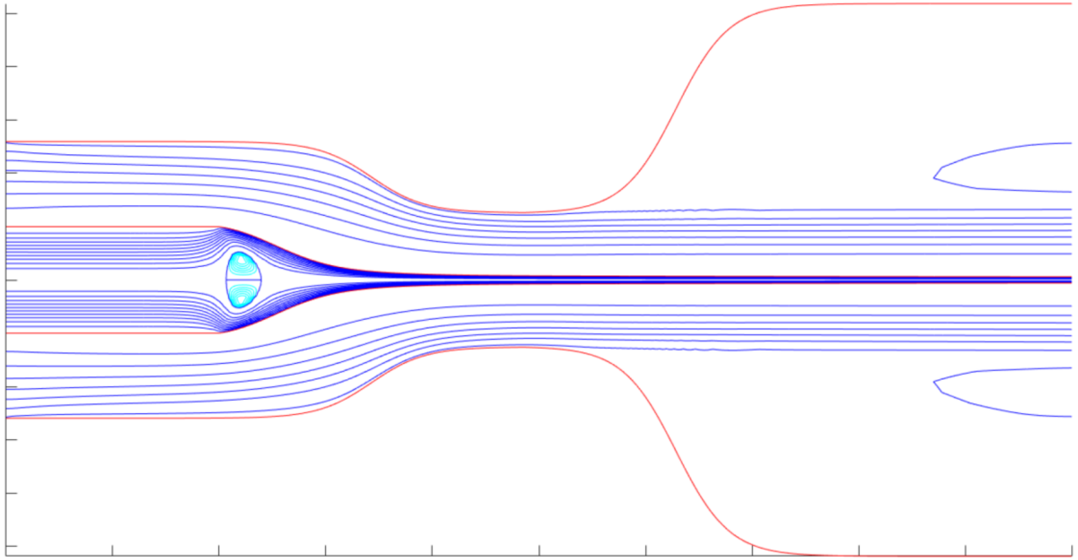
Flowfocusing

Mesh

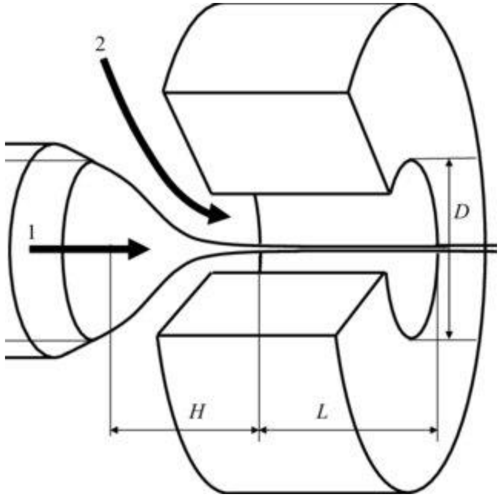


S. Blanco-Trejo, M.A. Herrada, A.M. Gañán-Calvo, A. Rubio, M.G. Cabezas, J.M. Montanero, Whipping in gaseous flow focusing, *International Journal of Multiphase Flow*, **130**, 2020,

Steady Solution



A. Gañán-Calvo PRL. (1998)

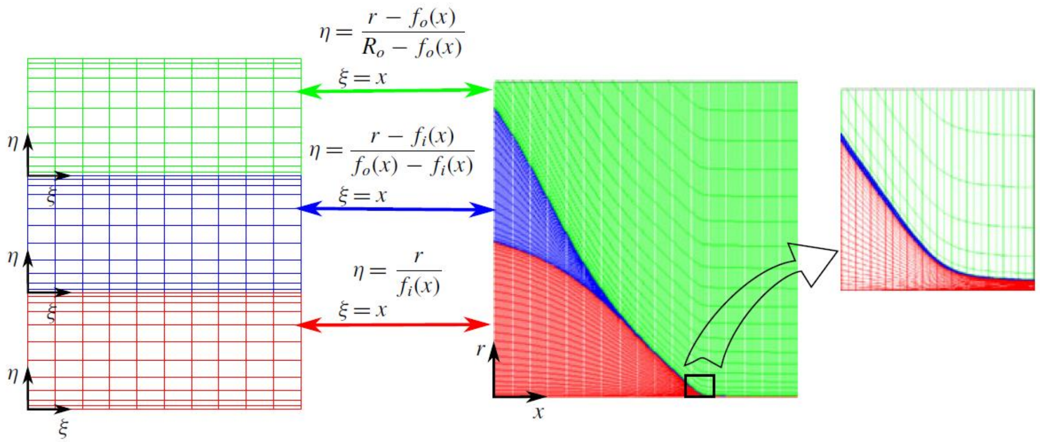


A. Gañán-Calvo

Applications

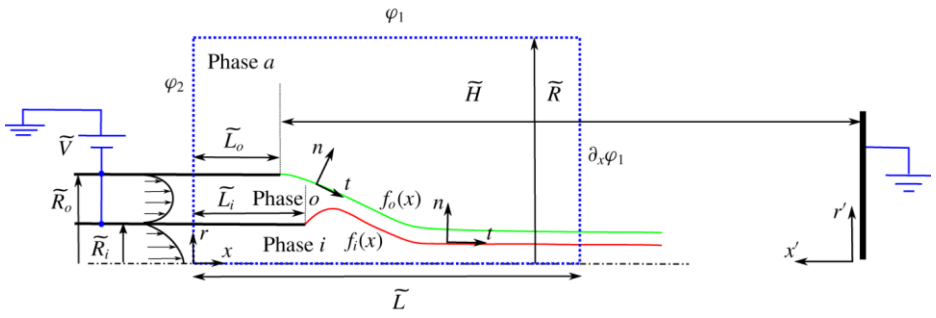
Electrosprays

Mesh

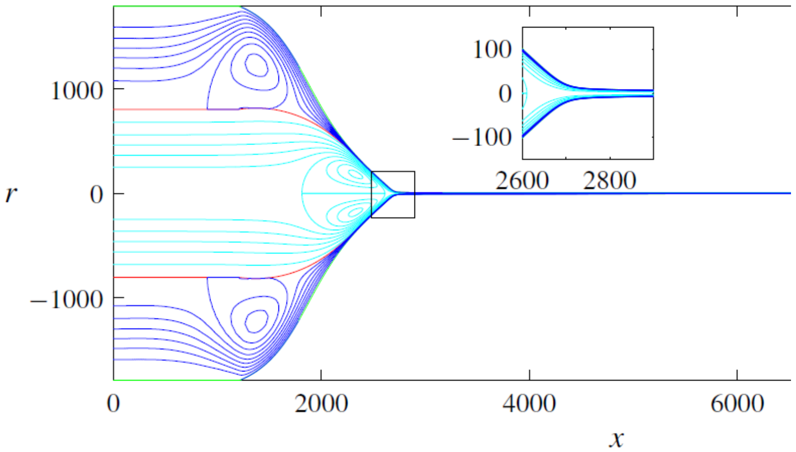


LÓPEZ-HERRERA, J., HERRADA, M., GAMERO-CASTAÑO, M., & GAÑÁN-CALVO, A. (2020). A numerical simulation of coaxial electrosprays. *Journal of Fluid Mechanics*, 885, A15.

Problem setup



Steady Solution

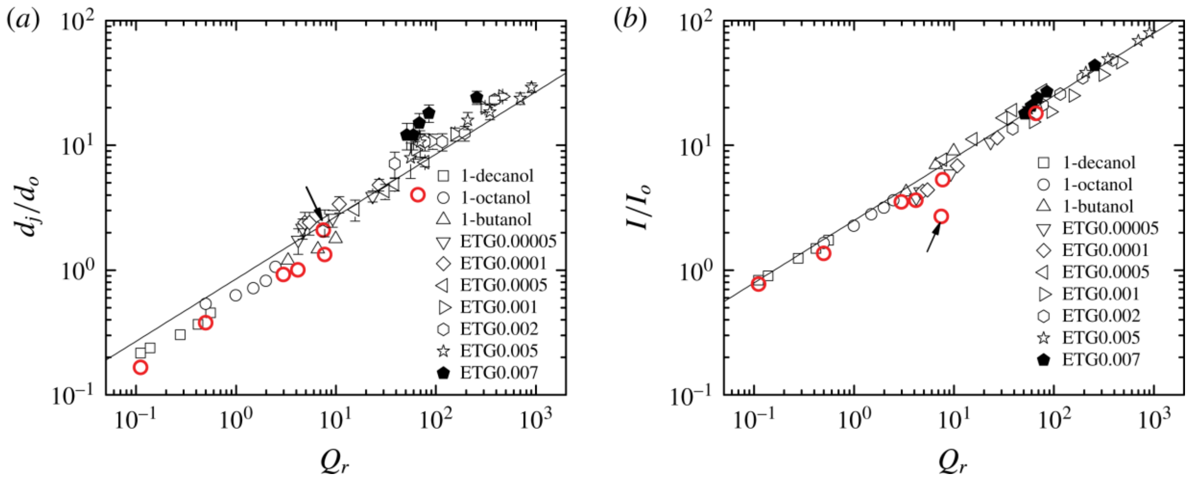


J.M. López-Herrera

Applications

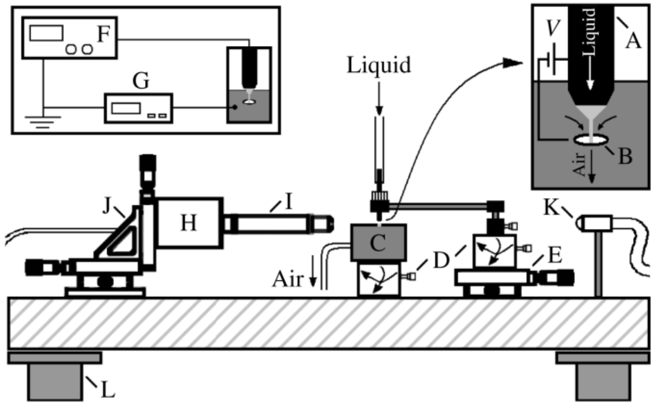
Electrosprays II

Comparison with experiments

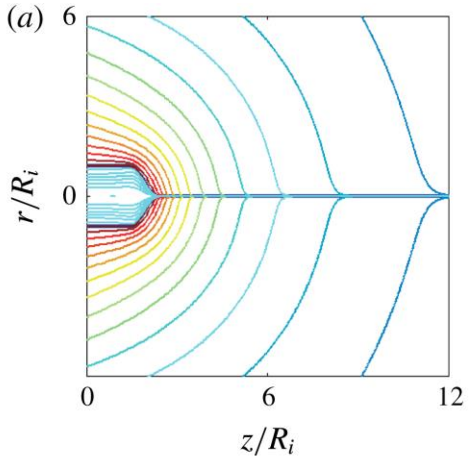


Ponce-Torres, A., Rebollo-Muñoz, N., Herrada, M., Gañán-Calvo, A., & Montanero, J. (2018). The steady cone-jet mode of electrospraying close to the minimum volume stability limit. *Journal of Fluid Mechanics*, **857**, 142-172.

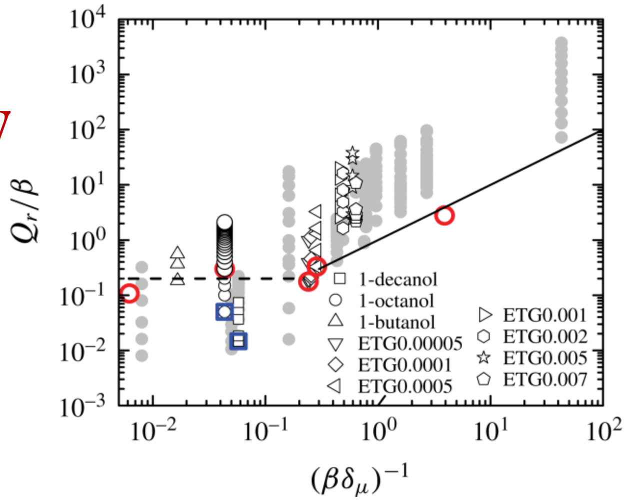
Problem setup



Basic flow



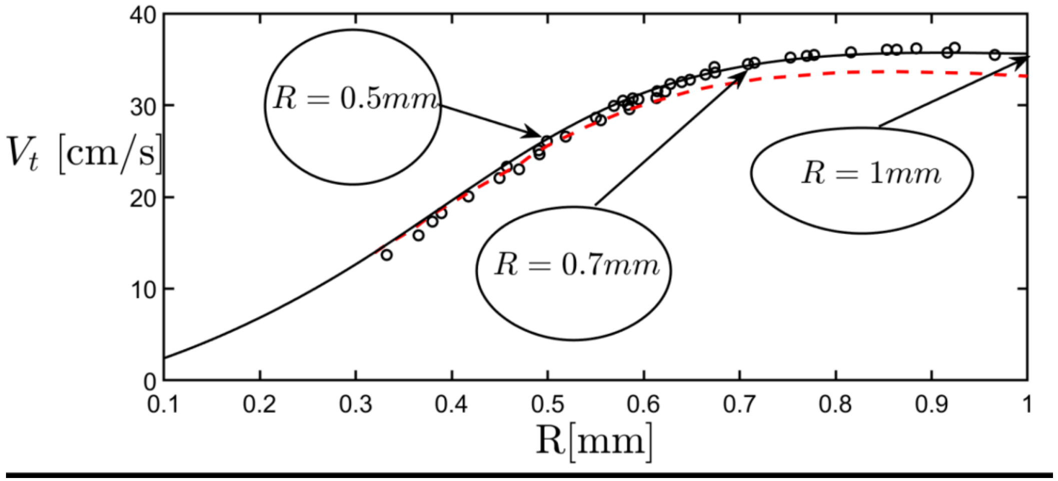
Stability



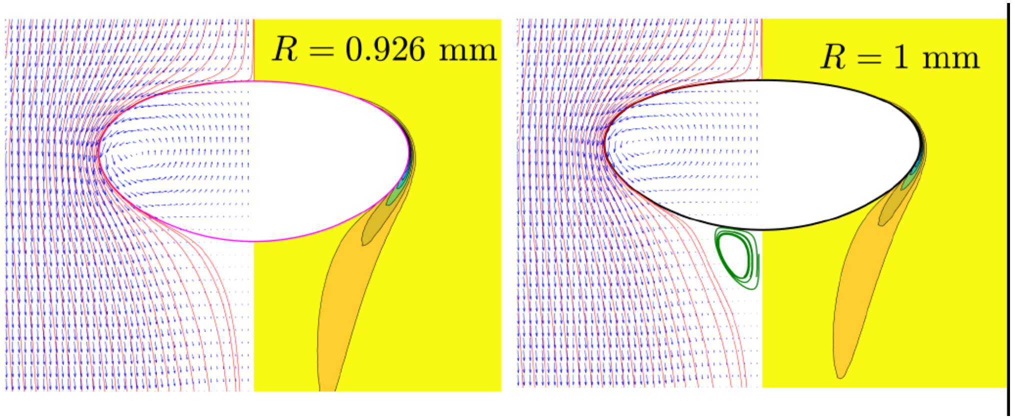
Applications

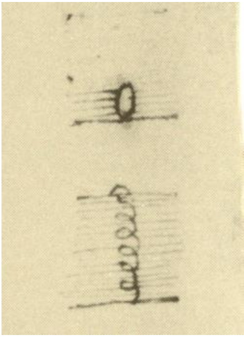
Bubbles dynamics

Rising Velocity



Steady Solutions

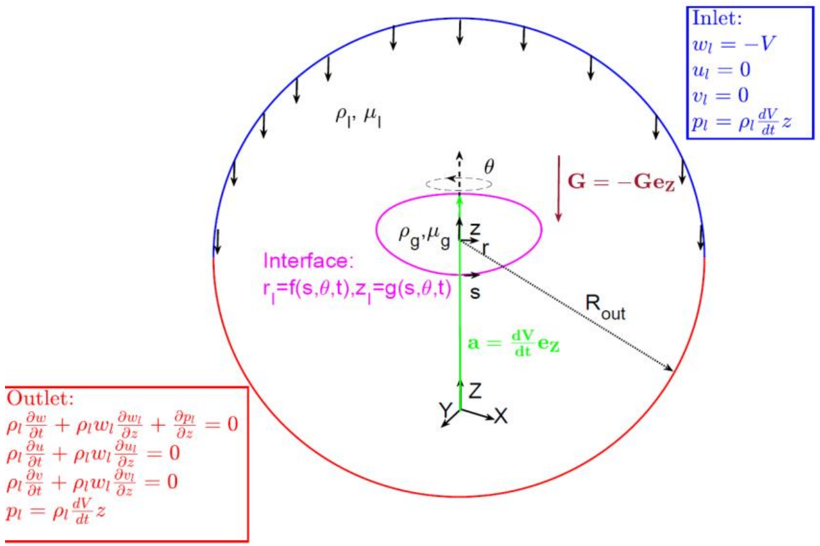




Leonardo da Vinci drawing (Codex Leicester)

HERRADA, M.A., EGGERS, J.G. (2023). Path instability of an air bubble rising in water *PNAS*, **120**, e2216830120.

Problem setup

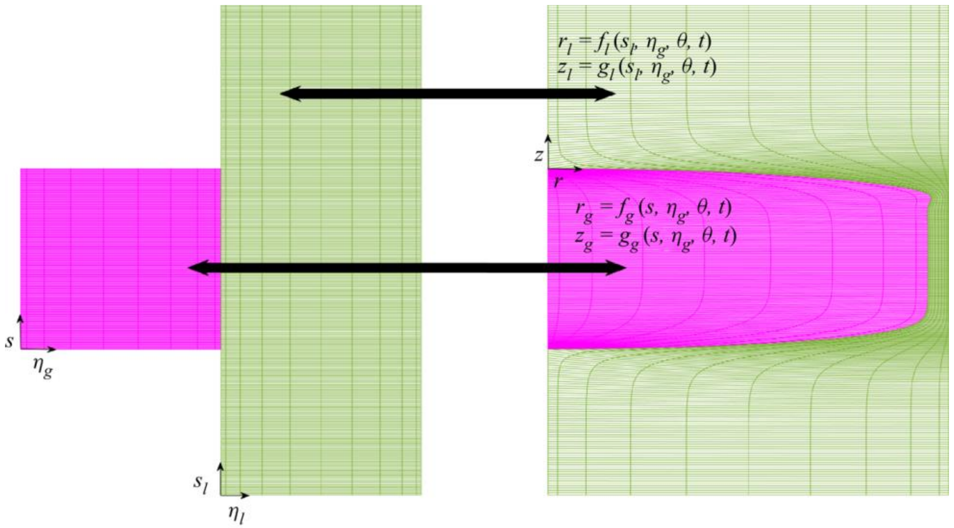


Jens Eggers

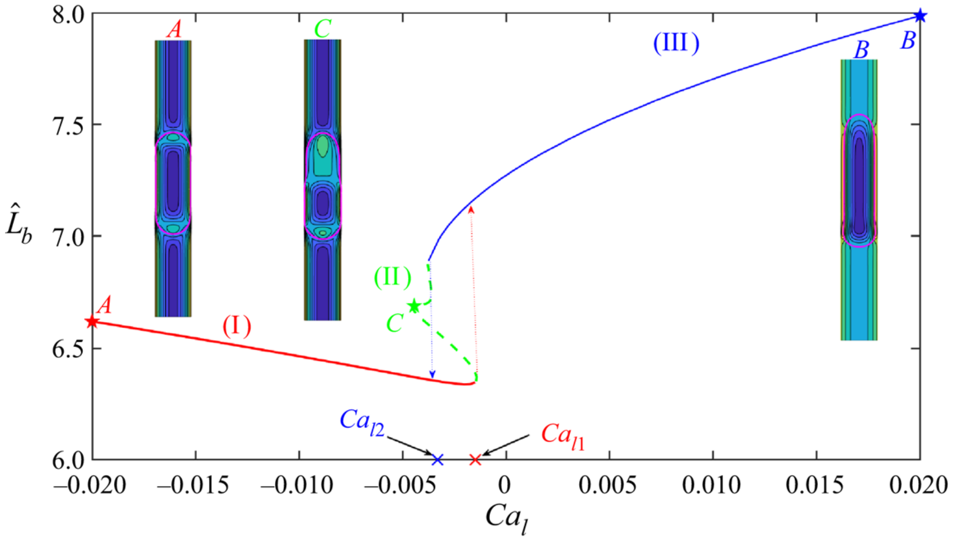
Applications

Bubbles dynamics (II)

Mesh

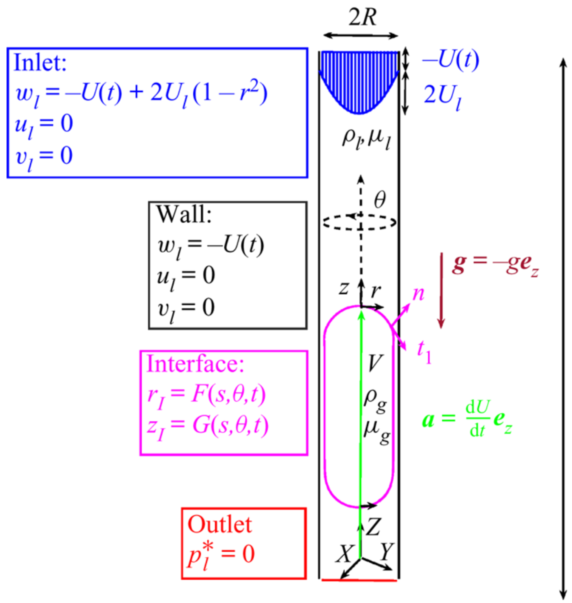


Steady Solutions



HERRADA, M., YU, Y., & STONE, H. (2023). *Global stability analysis of bubbles rising in a vertical capillary with an external flow*. Journal of Fluid Mechanics, **958**, A45.

Problem setup

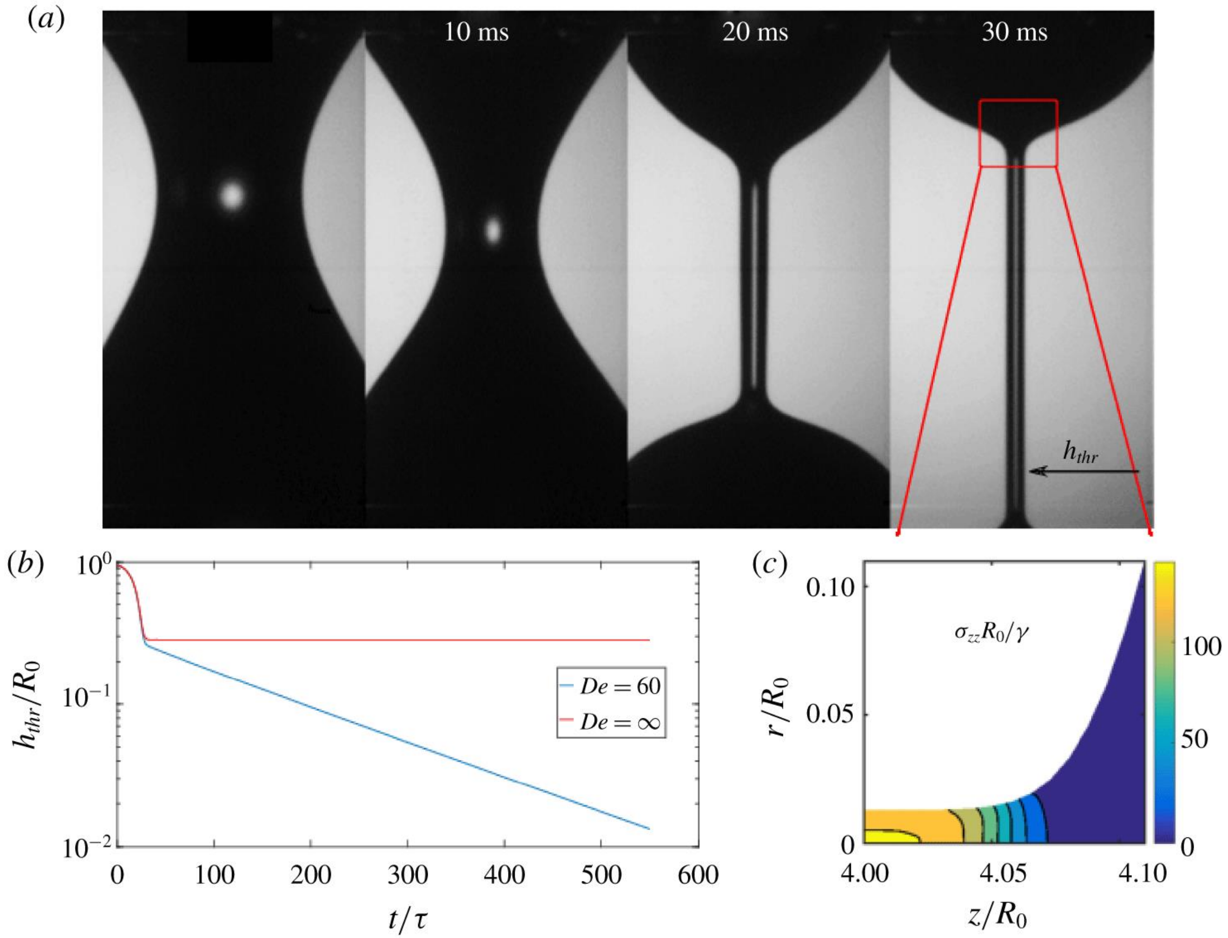


Howard Stone

Applications

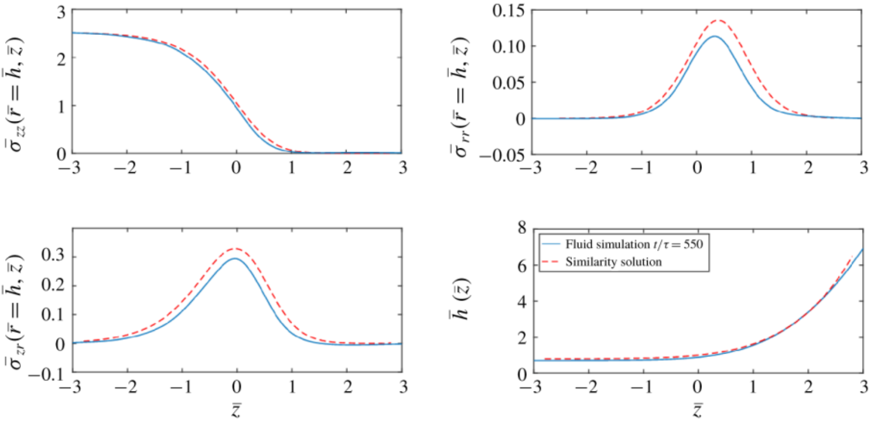
Viscoelasticity

Unteady Solutions



EGGERS, J., HERRADA, M., & SNOEIJER, J. (2020). *Self-similar breakup of polymeric threads as described by the Oldroyd-B model*. Journal of Fluid Mechanics, 887, A19.

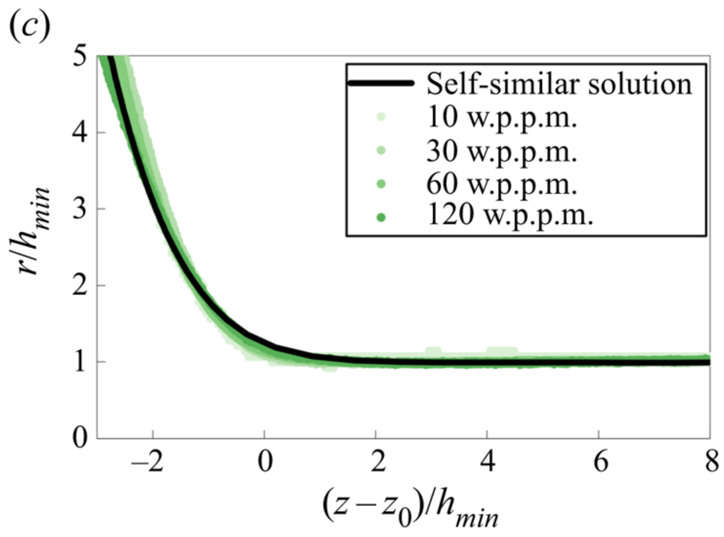
Self-similar profiles



Jacco Snoeijer

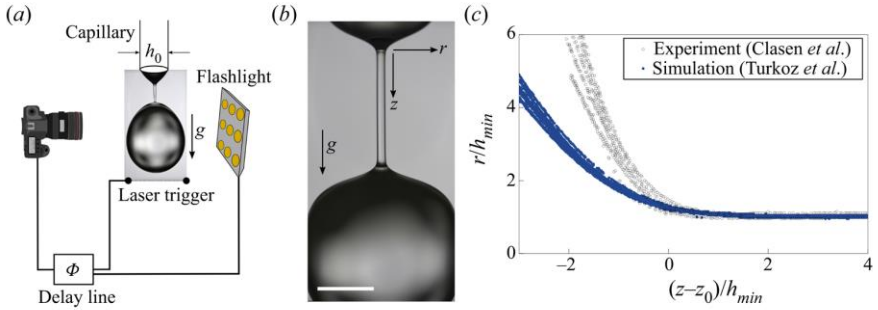
Applications

Comparison with experiments



Deblais, A., Herrada, M., Eggers, J., & Bonn, D. (2020). Self-similarity in the breakup of very dilute viscoelastic solutions. Journal of Fluid Mechanics, **904**, R2

Problem-setup

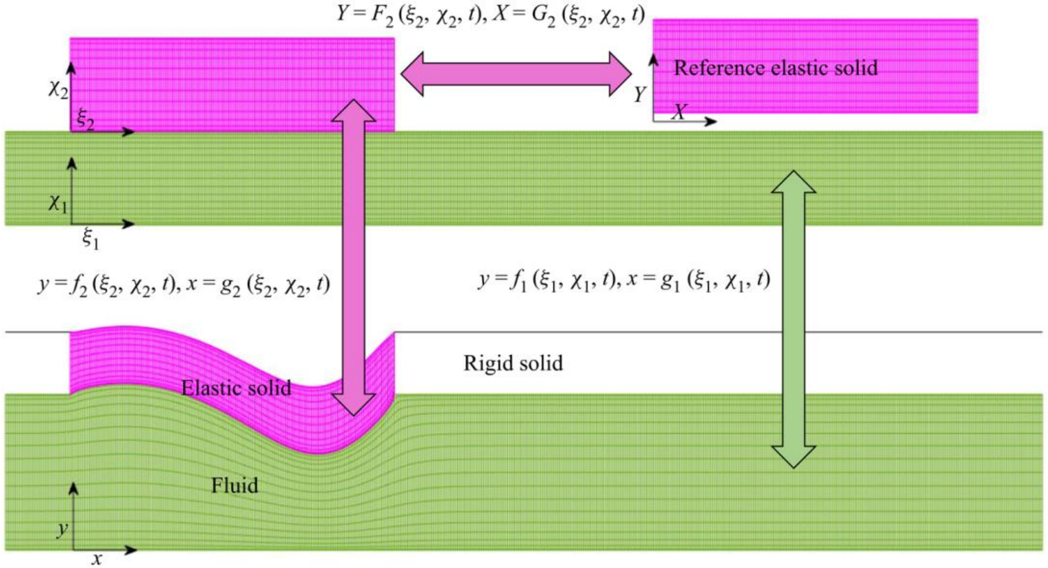


Daniel Bonn

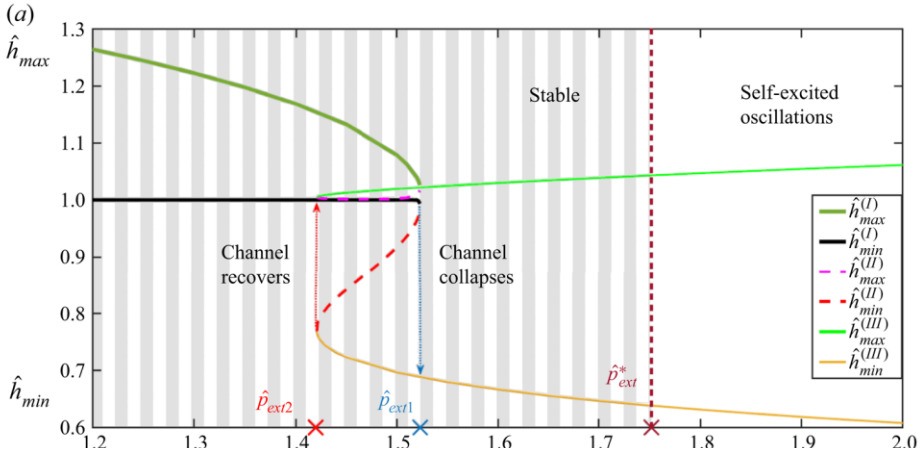
Applications

Fluid-solid interaction

Mesh



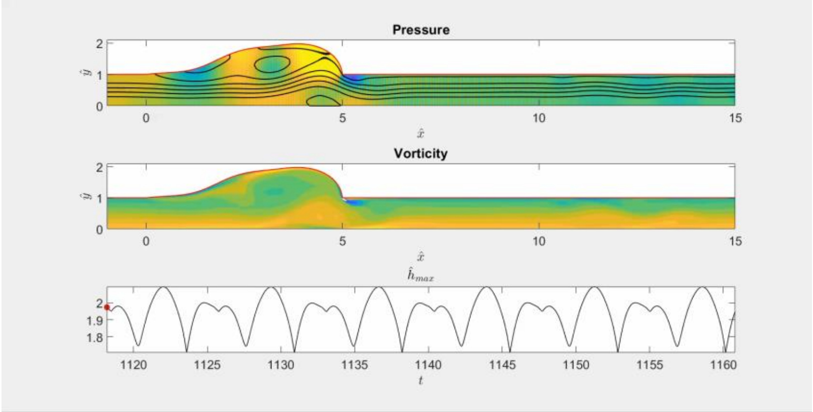
Basic flows



HERRADA, M., BLANCO-TREJO, S., EGGERS, J., & STEWART, P. (2022). Global stability analysis of flexible channel flow with a hyperelastic wall. *Journal of Fluid Mechanics*, **934**, A28.

Peter Stewart

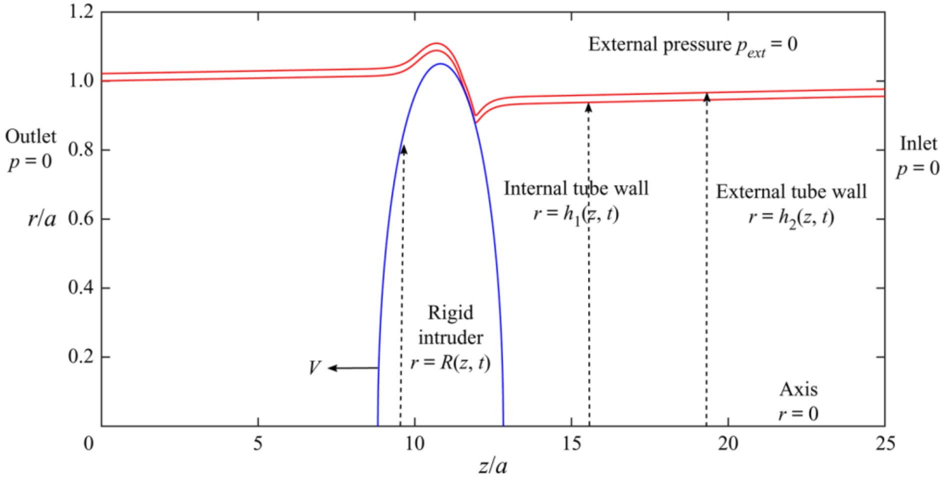
Unsteady simulations



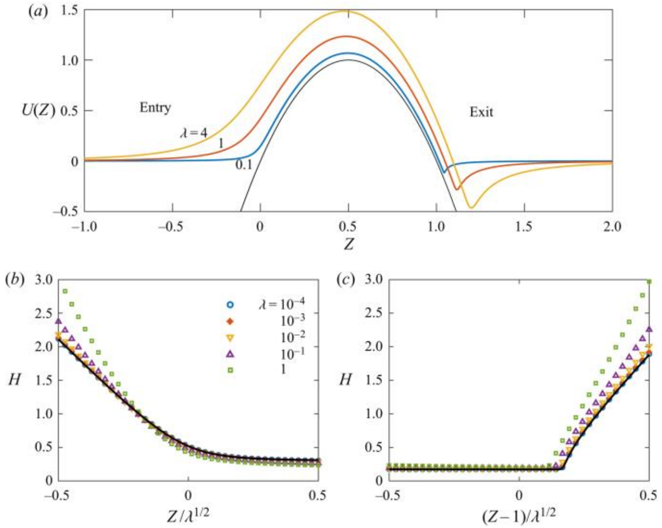
Fluid-solid interaction II

Applications

Numerical domain



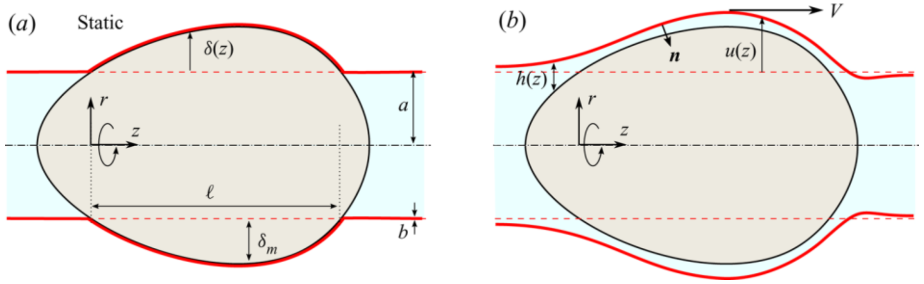
Basic flows



Bhargav Rallabandi

RALLABANDI, B., EGGERS, J., HERRADA, M., & STONE, H. (2021). Motion of a tightly fitting axisymmetric object through a lubricated elastic tube. *Journal of Fluid Mechanics*, **926**, A27.

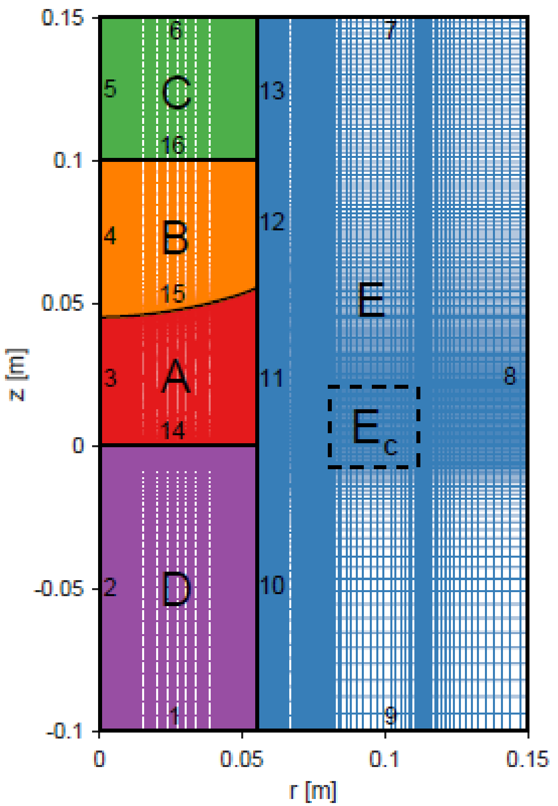
Problem setup



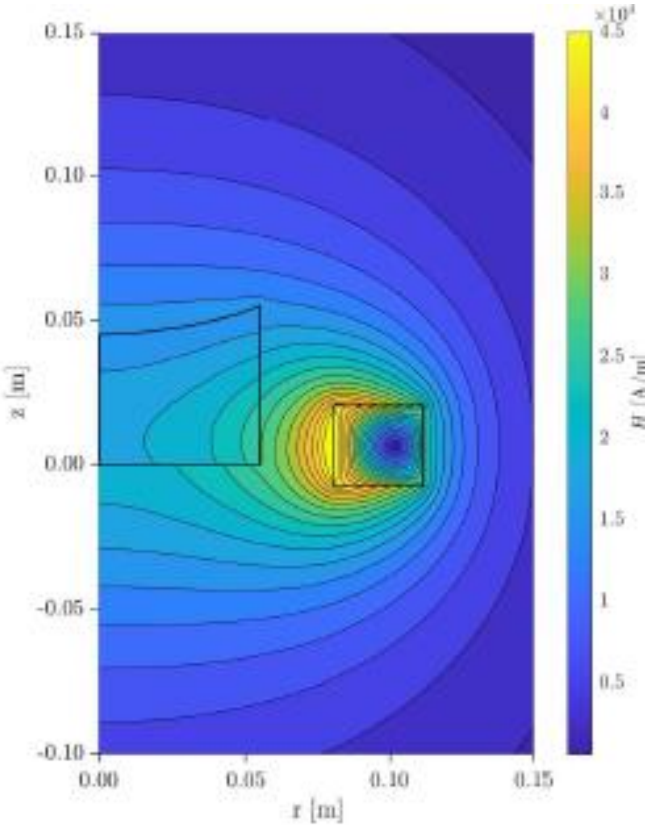
Applications

Electro/Magneto hydrodinamics

Mesh

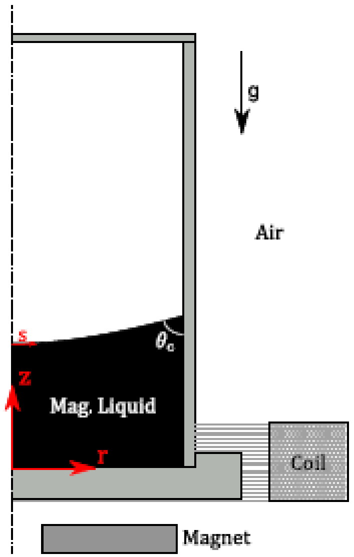


Steady Solution



Á Romero-Calvo, MA Herrada, G Cano-Gómez, H Schaub (2022). Fully coupled interface-tracking model for axisymmetric ferrohydrodynamic flows. *Applied Mathematical Modelling*. **111**, 836-861.

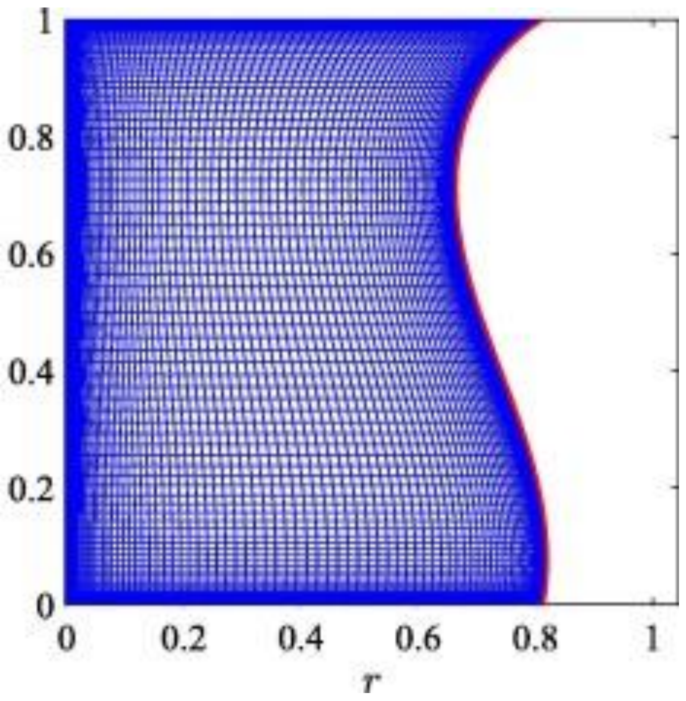
Problem setup



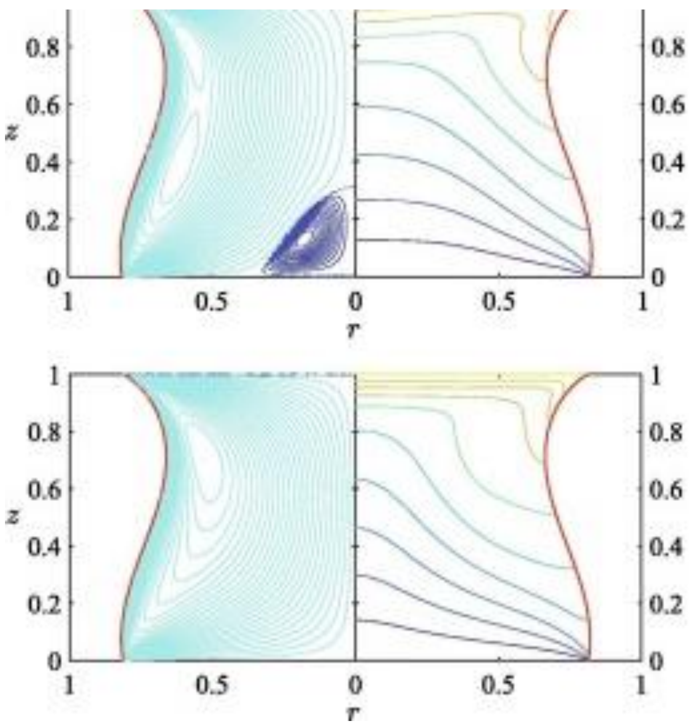
Álvaro Romero-Calvo

Applications

Mesh

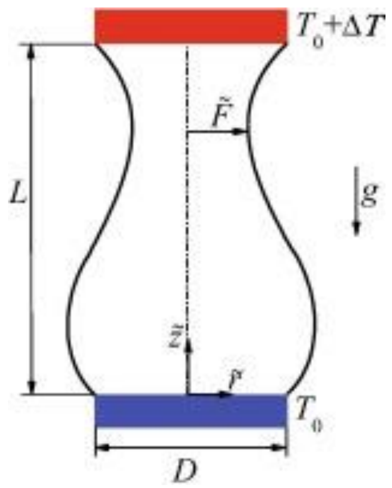


Steady Solution



LUIS M. CARRIÓN, MIGUEL A. HERRADA, JOSÉ M. MONTANERO, 2020 Influence of the dynamical free surface deformation on the stability of thermal convection in high-Prandtl-number liquid bridges, IJHMT, **146**,

Problem setup



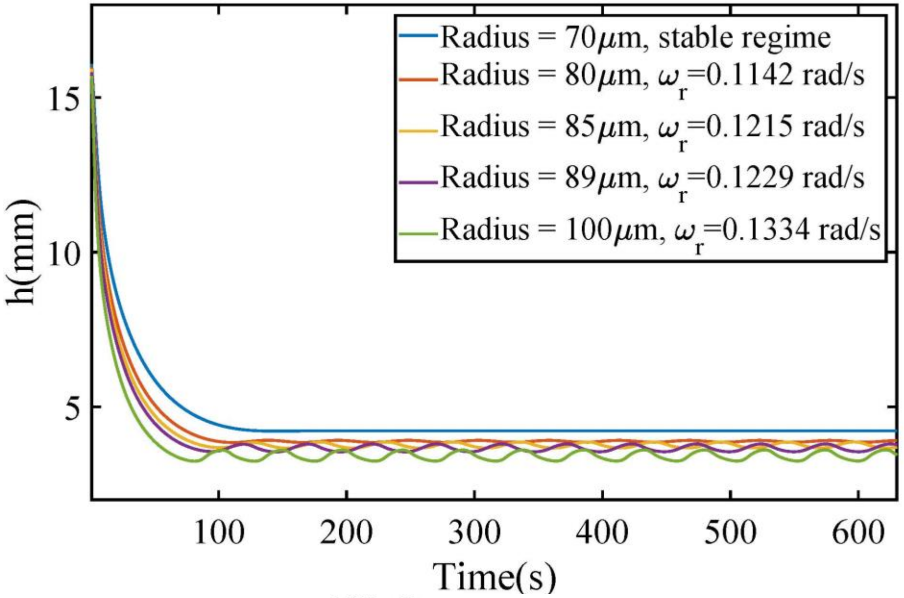
Luis Carrión

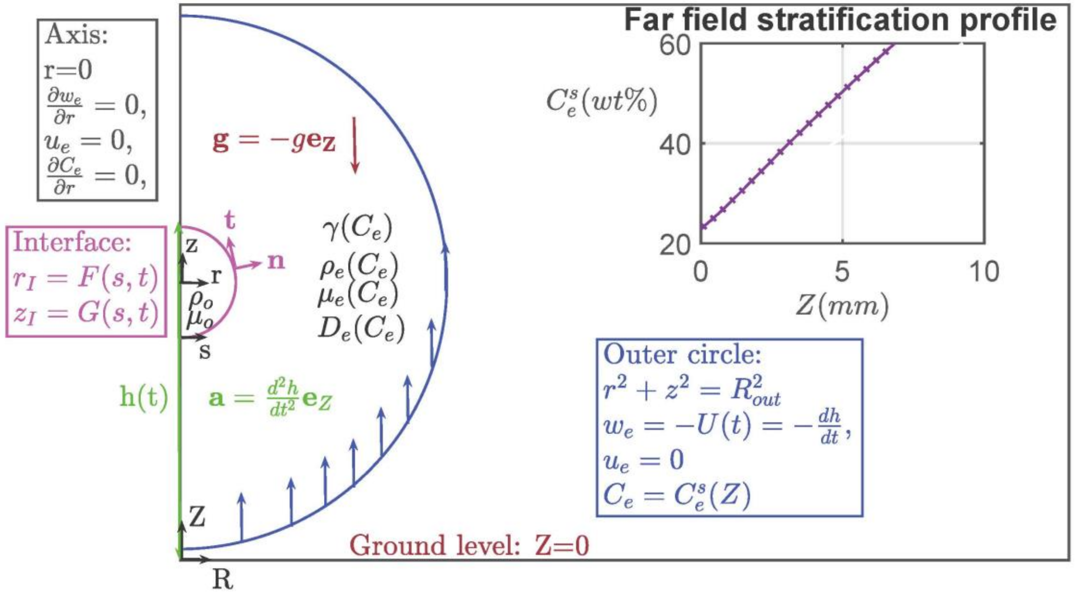
Applications

Marangoni Flows(II)

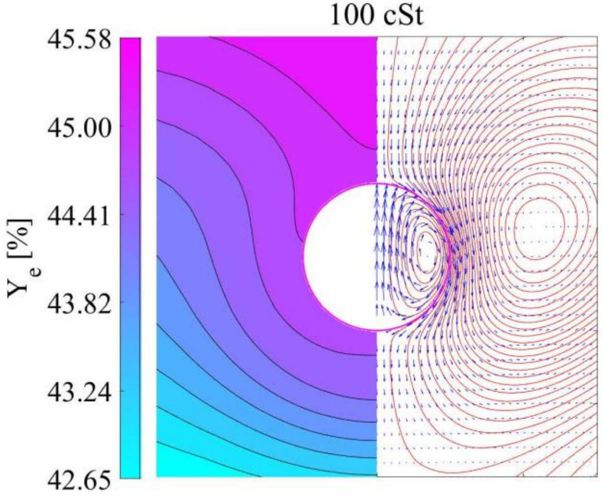
M. A. Herrada J. M. Montanero and Luis M. Carrión(2023). Dynamics of a silicon drop submerged in a stratified ethanol-water bath *Phys. Rev. Fluids* ***(accepted)***

Bouncing drop





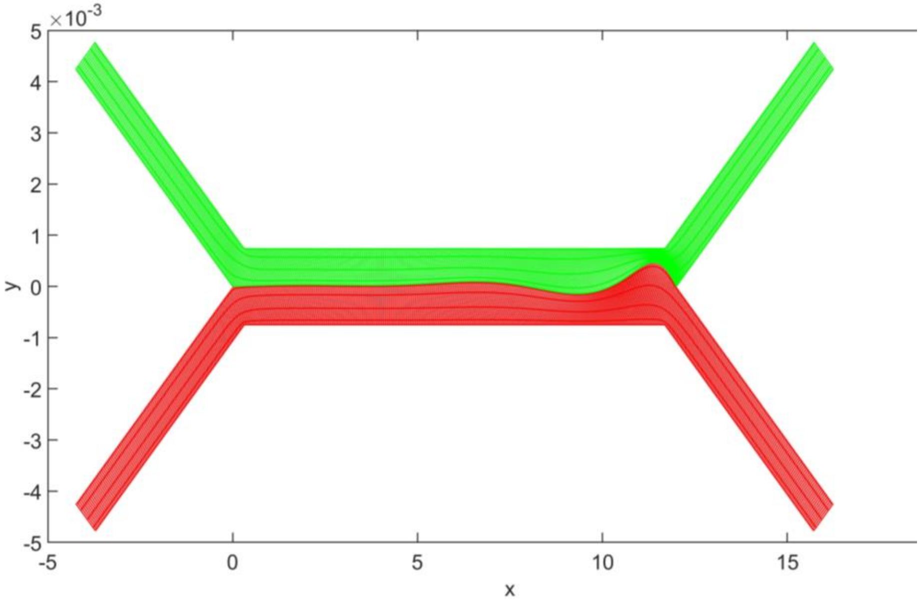
Steady Flow



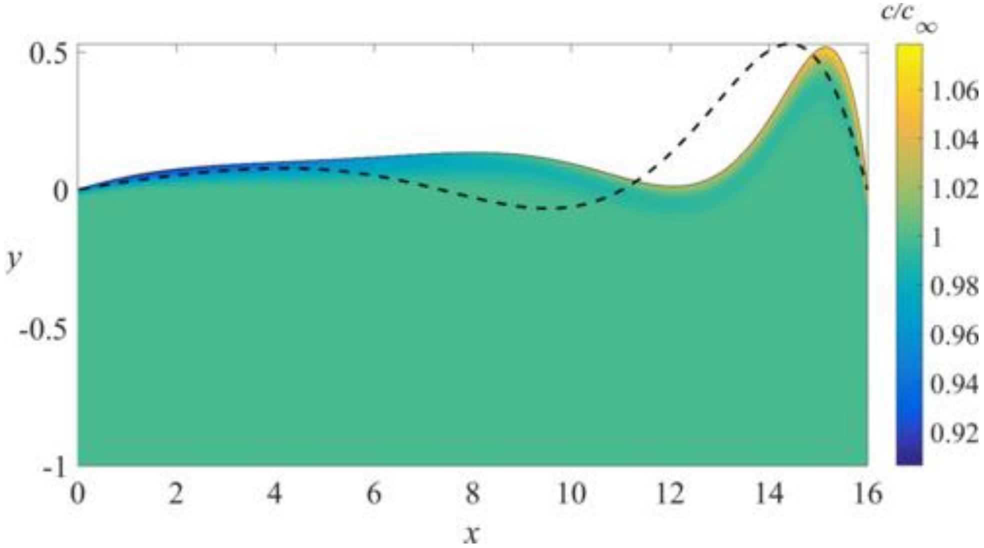
Applications

Surfactant driven flows

Mesh

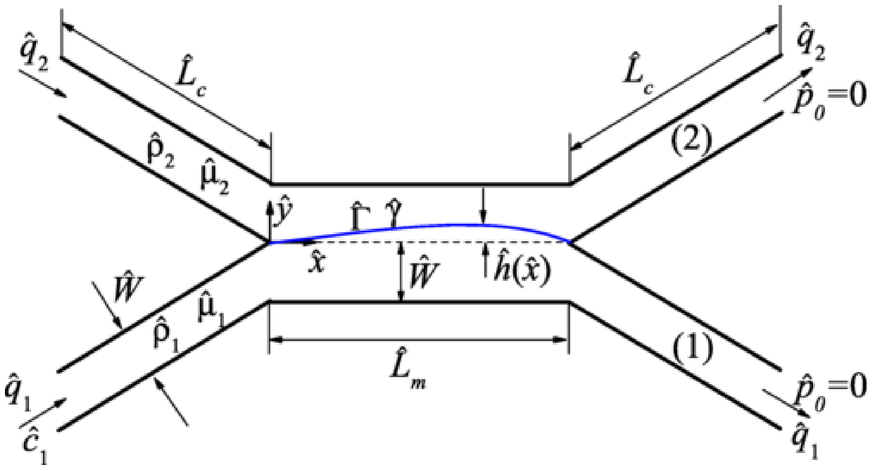


Steady Flow



M. A. Herrada, A. Ponce-Torres, P. R. Kaneelil, A. A. Pahlavan, H. A. Stone, and J. M. Montanero (2022). Effect of a soluble surfactant on the linear stability of two-phase flows in a finite-length channel *Phys. Rev. Fluids* **7**, 114003

Problem setup

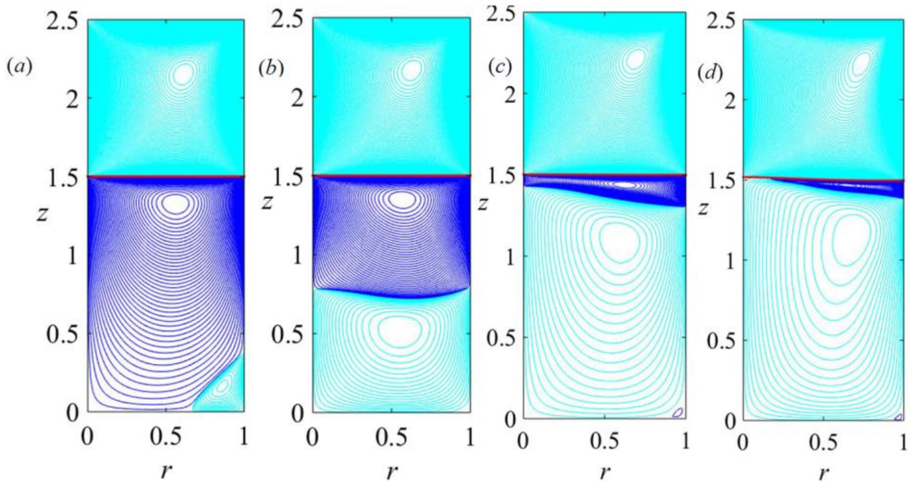


Alberto Ponce

Applications

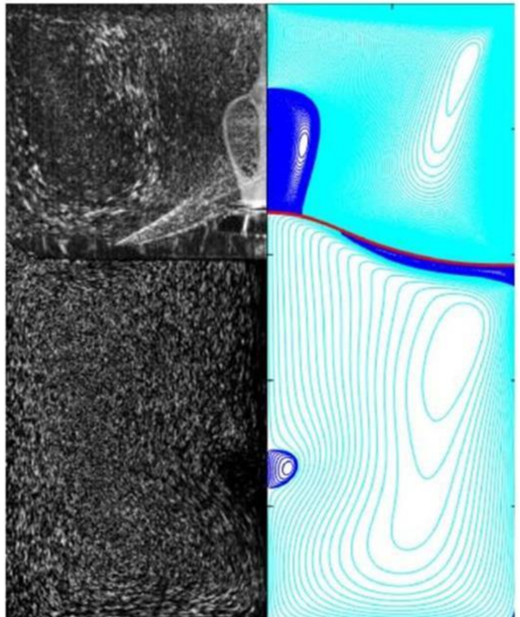
Swirling flows

Steady Solutions

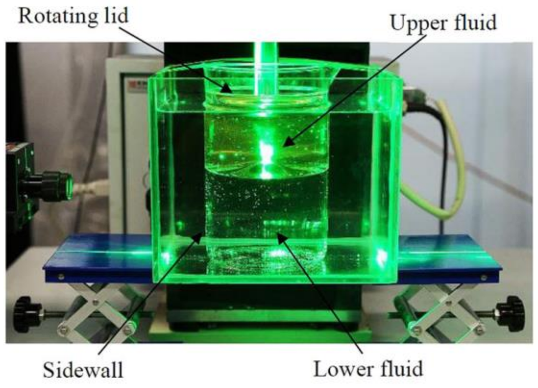


L. CARRIÓN, I. V. NAUMOV, B.R. SHARIFULLIN, M. A. HERRADA, V. N. SHTERN. 2020.Formation of dual vortex breakdown in a two-fluid confined flow. *Physics of Fluids* 1 32 (10): 104107

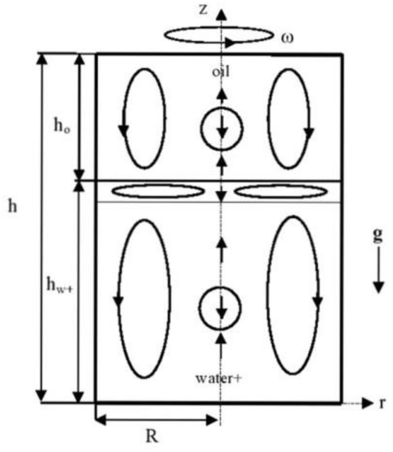
Comparison with experiments



Problem setup



(a)

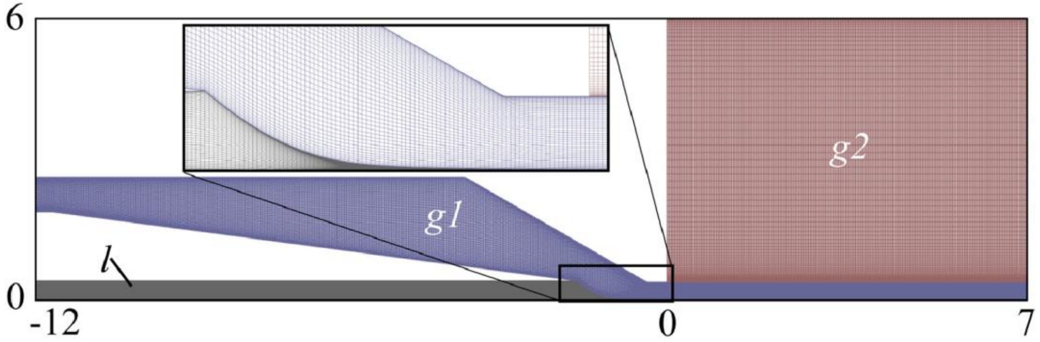


(b)

Vladimir Shtern

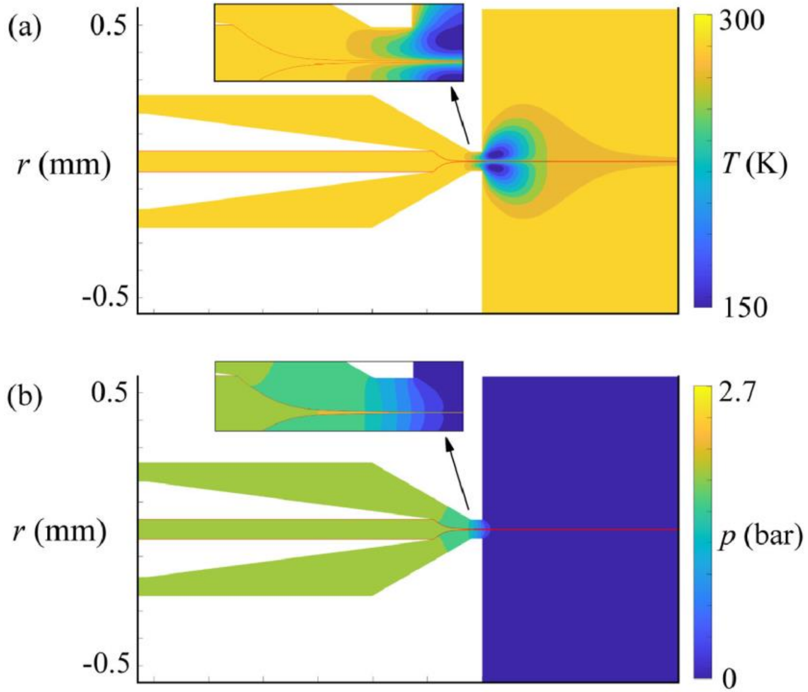
Applications

Mesh

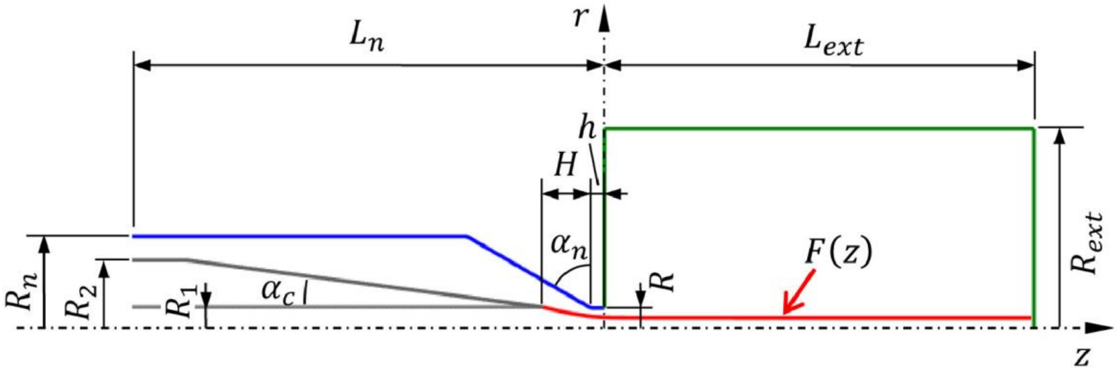


M. RUBIO, A. RUBIO, M.G. CABEZAS, M.A. HERRADA, A.M. GAÑÁN-CALVO, J.M. MONTANERO (2021) Transonic flow focusing: stability analysis and jet diameter, *International Journal of Multiphase Flow*, **142.**103720.

Steady Flow



Problem setup



Manuel Rubio

Example
Burger's problem

1)Velocity u: $\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\left[\frac{\partial^2 u}{\partial x^2}\right],\ 0 < x < 1, u(x = 0) = 1, u(x = 1) = -1.$

2) Stretching g: x=g(s,t), $0 \le s \le 1, g(s = 0) = 0\ \ , g(s = 1) = 1.$ $\quad Non\ singular\ \frac{\partial g}{\partial s} \ne 0$

Mapping

$$\frac{\partial}{\partial x} = \frac{\partial s}{\partial g}\frac{\partial}{\partial s},\quad \frac{\partial^2}{\partial x^2} = \frac{\partial s}{\partial g}\left(\frac{\partial s}{\partial g}\frac{\partial^2}{\partial s^2} - \frac{\partial^2 g}{\partial s^2}\left[\frac{\partial s}{\partial g}\right]^2\frac{\partial}{\partial s}\right),\quad \frac{\partial}{\partial t} = \frac{\partial s}{\partial g}\frac{\partial g}{\partial \tau}\frac{\partial}{\partial s} + \frac{\partial}{\partial \tau}$$

Nonlinear equations: Bulk and BC's

1) Equations for u: Bulk: $\frac{\partial u}{\partial \tau} + \left(u\frac{\partial s}{\partial g} - \frac{\partial s}{\partial g}\frac{\partial g}{\partial \tau}\right)\frac{\partial u}{\partial s} = \nu\left[\frac{\partial s}{\partial g}\left(\frac{\partial s}{\partial g}\frac{\partial^2 u}{\partial s^2} - \frac{\partial^2 g}{\partial s^2}\left[\frac{\partial s}{\partial g}\right]^2\frac{\partial u}{\partial s}\right)\right].\quad u(s = 0) = -1.\ u(s = 1) = 1.$

2) Equations for g: Bulk: $\frac{\partial g}{\partial s}\frac{\partial^2 g}{\partial s^2} = \left(\frac{\partial g}{\partial s}\right)^{3/2}\frac{\partial M}{\partial s}$ $, g(s = 0) = 0.\quad g(s = 1) = 1$ $\quad$ dx=dl(s,t)=$\left(\frac{\partial g}{\partial s}\right)^{3/2} = M(s,t)$

For this problem we choose: M(s, τ)= $\frac{1}{0.4+\alpha\left(\frac{\partial u}{\partial x}\right)^2}$, $\quad \alpha\ is\ a\ free\ parameter$

Implementation using JAM

Step 1: Identifying Equations and variables

2 Symbolic Variables : u, g.

3 Symbolic Equations: Bulk: FAAb(2) BC's: left : FAAl(2) and right: FAAr(2)

$$FAAb(1) = \frac{\partial \tau}{\partial g}\frac{\partial u}{\partial \tau} + u\frac{\partial s}{\partial g}\frac{\partial u}{\partial s} - \nu\left[\frac{\partial s}{\partial g}\left(\frac{\partial s}{\partial g}\frac{\partial^2 u}{\partial s^2} - \frac{\partial^2 g}{\partial s^2}\left[\frac{\partial s}{\partial g}\right]^2\frac{\partial u}{\partial s}\right)\right], \qquad FAAb(2) = \frac{\partial g}{\partial s}\frac{\partial^2 g}{\partial s^2} - \frac{\partial M}{\partial s}.$$

$$FAAl(1) = u - 1, \qquad FAAl(2) = g, \qquad FAAr(1) = u + 1,\ FAAr(2) = g - 1.$$

Step 2: Identifying vector of symbolic derivatives

Symbolic Vector: $x_s = \left[u, \frac{\partial u}{\partial s}, \frac{\partial^2 u}{\partial s^2}, \frac{\partial u}{\partial \tau}, g, \frac{\partial g}{\partial s}, \frac{\partial^2 g}{\partial s^2}, \frac{\partial g}{\partial \tau}\right]$: Vector size 8

Step 3: Computing Analytical Jacobians

3 Symbolic Jacobians: Bulk: dFAAb(2,8) BC: left dFAAl(2,8) and right dFAAr(2,8)

dFAAb=jacobian($FAAb$,x_s), dFAAl= jacobian($FAAl$,x_s), dFAAr= jacobian $FAAr$,x_s)

Implementation using JAM

Step 4: Saving Equations and Jacobians

matlabfunction tool has been used to storage the equations, for example for FAAb:

```
matlabFunction(FAAb,dFAAb,'file',[path_jacobian 'equationFAAb.m'],'vars',{s,t,xo,pa});
```

pa is a vector containing the parameters of the problem: pa(1)=ν and pa(2)= α

Step 5: Spatial Discretization

s is discretized in N points using second order central finite differences: $s_i = (i-1)\Delta s \ , i = 1:N$

$$\frac{\partial \Phi_i}{\partial s} = \frac{\Phi_{i+1} - \Phi_{i-1}}{2\Delta s}, \quad \frac{\partial^2 \Phi_i}{\partial s^2} = \frac{\Phi_{i+1} - 2\Phi_i + \Phi_{i-1}}{\Delta s^2},$$

Φ_i is the value of any the variable (u or g) at s_i

$$\frac{\partial \Phi_i}{\partial s} = \mathbf{ds}\Phi, \quad \frac{\partial^2 \Phi_i}{\partial s^2} = \mathbf{dss}\Phi :$$

ds and **dss** are NxN sparse matrices (Collocation matrices) KEY POINT!!

Implementation using JAM

Step 6: temporal discretization

The time is discretized using 2 order backwards finite differences

$\frac{\partial \Phi^{m}_{i}}{\partial \tau} = \frac{3\Phi^{m}_{i} - 4\Phi^{m-1}_{i} + \Phi^{m-2}_{i}}{2\Delta\tau}$, where Φ^{m}_{i} is the variable at the current time (τ) while Φ^{m-1}_{i} and Φ^{m-2}_{i} are the solutions at $(\tau - \Delta\tau)$ and $(\tau - 2\Delta\tau)$ respectively and $\Delta\tau$ is the time step.

⇩

$\frac{\partial \Phi_{i}}{\partial \tau} = \frac{3}{2\Delta\tau} \mathbf{I}\, \Phi + \frac{-4\Phi^{m-1}_{i} + \Phi^{m-2}_{i}}{2\Delta\tau}$, where **I** is the identity NxN matrix

Step 7: Creating the numerical guess solution

Vector: $x_o = [u_1, \ldots, u, uN, \ldots, g_N]$: Vector size 2N

Step 8: Evaluation of symbolic functions ($s= s_i$)

Bulk i=2:N-1

FAA(1:2,i)=**FAAb**(1:2,1:8,s_i) , **DFAA**(1:2,1:8,i) =**DFAAb** (1:2,1:8, s_i)

Left i=1

FAA(1:2,i)=**FAAl**(1:2,1:8, s_i) , **DFAA**(1:2,1:8,i) =**DFAAl** (1:2,1:8, s_i)

Right i=N

FAA(1:2,i)=**FAAr**(1:2,1:8, s_i) , **DFAA**(1:2,1:8,i) =**DFAAr** (1:2,1:8, s_i)

where **FAA** is2xN matrix and **dFAA** is 2x8xN array

Step 9: Assembly of the numerical Jacobian matrix

Jacobian Matrix **dF=a**= $\begin{pmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} \\ \mathbf{a}_{21} & \mathbf{a}_{22} \end{pmatrix}$, Matrix size 2Nx2N, Funtion **F=b=** $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ Vector size 2Nx1

Where each block are computed using **FAA, dFAA** and the collocation matrices. For k=1:2

$$\mathbf{a}_{k1} = \mathbf{dFAA}(k,1,:)\mathbf{I}+\mathbf{dFAA}(k,2,:)\mathbf{ds}+\mathbf{dFAA}(k,3,:)\mathbf{dss}+\frac{3}{2\Delta\tau}\mathbf{dFAA}(k,4,:)\mathbf{I}$$

$$\mathbf{a}_{k2} = \mathbf{dFAA}(k,5,:)\mathbf{I}+\mathbf{dFAA}(k,6,:)\mathbf{ds}+\mathbf{dFAA}(k,7,:)\mathbf{dss}+\frac{3}{2\Delta\tau}\mathbf{dFAA}(k,8,:)\mathbf{I}$$

$$\mathbf{b}_{k2} = \mathbf{FAA}(k,:)$$

JaM

Implementation using JAM

UNIVERSIDAD DE SEVILLA

Step 10: Solving the system

$\boldsymbol{DF}(\boldsymbol{x_o})\Delta \boldsymbol{x} = -\boldsymbol{F}(x_o) \rightarrow x_{new} = xo + \Delta x \; while \; |\Delta x| > \varepsilon$

For nonstationary problems, solve the nolinear problem and update the solution every time step.

Step 11: Eigen solver problem

1.Assuming a time dependence of the form: $\Phi(x,t)=\Phi_b(x)+\Delta\Phi_1(x)e^{-i\omega t}$ $\quad \frac{\Delta\Phi_1}{\Phi_b} \ll 1 \quad \omega = \omega_r + i\omega_i$

2. Split the Jacobian: $\boldsymbol{DF} = \boldsymbol{DFe}(\Phi_b) - i\omega \, \boldsymbol{DFt}(\Phi_b)$

3. Solve the generized eigen value problem: $\boldsymbol{DF}\Delta\Phi_1 \cong 0 \Longrightarrow \boldsymbol{DFe}(\Phi_b)\Delta\Phi_1 = i\omega \boldsymbol{DFt}(\Phi_b)\Delta\Phi_1$

Implementation using JAM

Matlab programs

Mainsteady.m A program to compute steady solutions +stability

Mainunsteady.m A program to compute unsteady solutions.

- blockA.m To write equations and save symbolic functions
- FAAb.m, FAAl.m, FAAr.m matlab functions generated by blockA.m
- finites2th.m To generate the collocation matrices
- matrixAB.m Assembly of numerical Jacobian matrix for the Newton method
- matrixABeigen.m Assembly of numerical matrices for the eigen value problem

JAM versus Fsolve

Mainfsolve.m to solve the problema using fsolve

Thank you very much for your attention!