Towards self-similar solvers:

An application to surface tension driven flows

N. CAILLER[†], R. WUNENBURGER[†] & A. ANTKOWIAK[†]

[†]Institut Jean Le Rond ∂'Alembert, CNRS UMR 7190, Sorbonne Université, Paris 75005, France

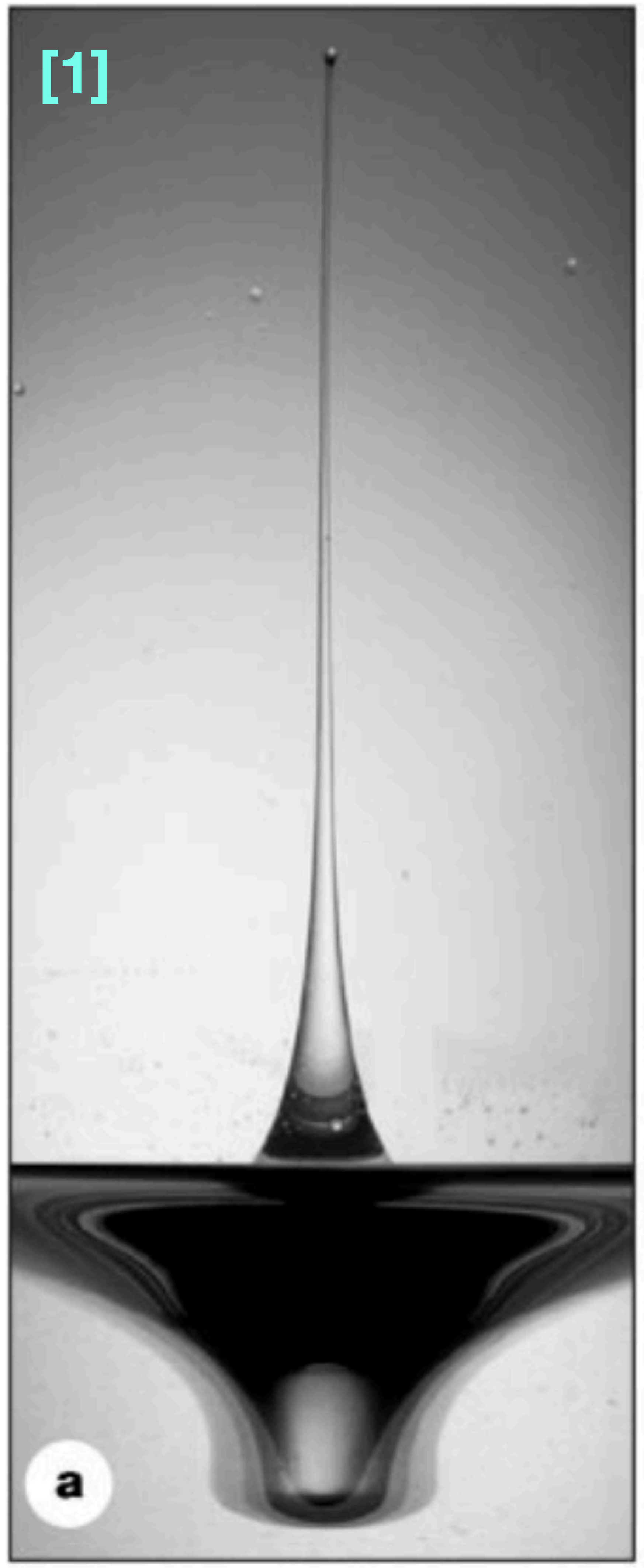
BASILISK (GERRIS) USERS' MEETING

* * *

5-7th July 2023

Experimental near-singular distortion of the free surface.

Balance between ***inertia*** *and* ***surface tension***.



Finite-time singularities
(collapsing Faraday Waves)

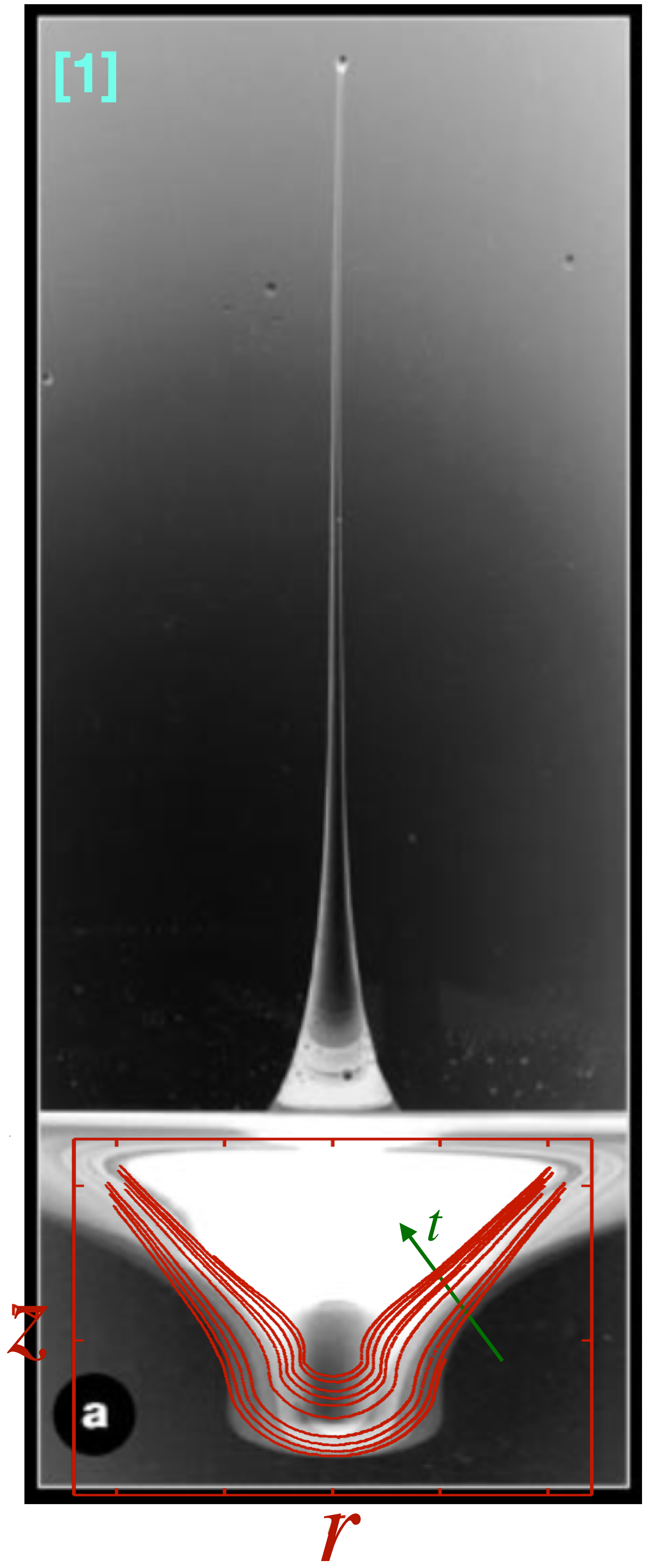
[1] Zeff *et al.*, *Singularity dynamics in curvature collapse and jet eruption on fluid surface*. Nature **403** (2000)

I.1 - Visualizing Scale Invariances (1/2)

Experimental near-singular distortion of the free surface.

*Balance between **inertia** and **surface tension**.*

*Collapsing cavity profiles seem to have the **same shape**.*



[1] Zeff *et al.*, *Singularity dynamics in curvature collapse and jet eruption on fluid surface.* Nature **403** (2000)

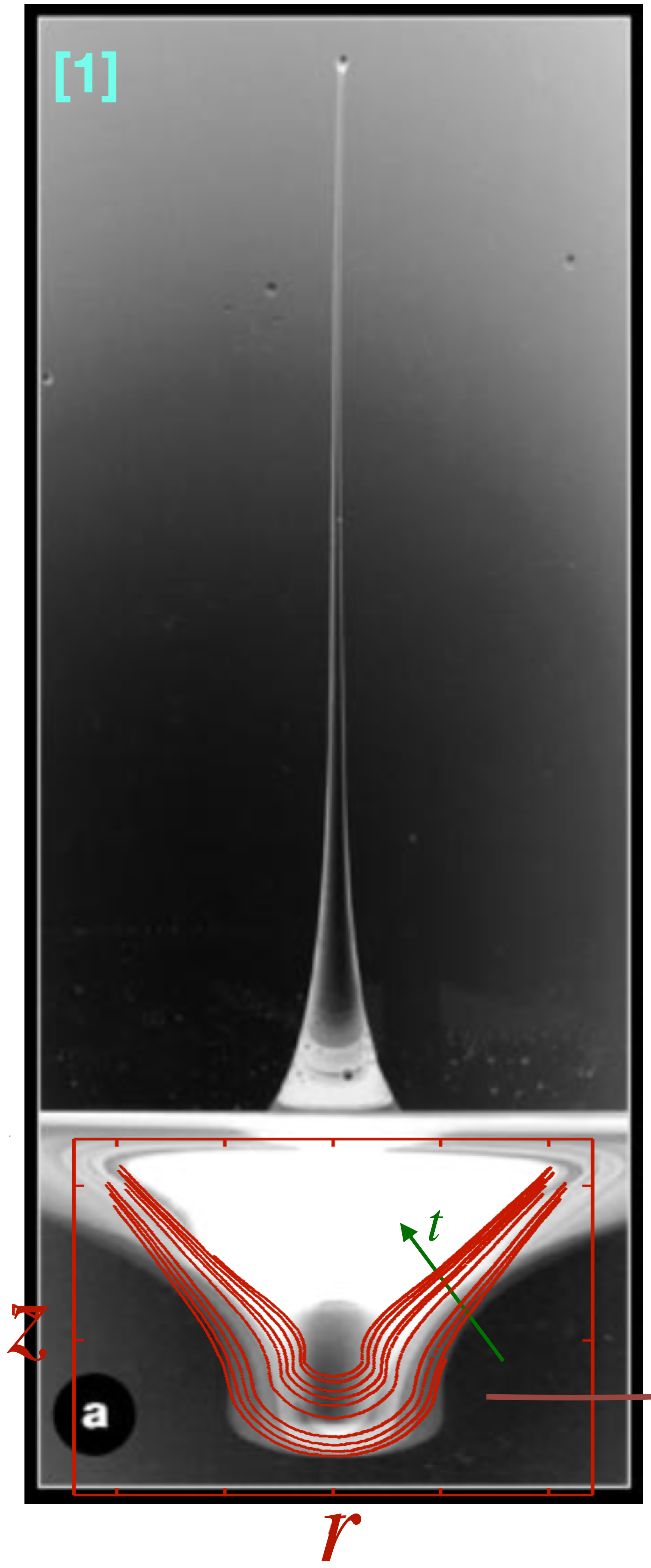
I.1 - Visualizing Scale Invariances (1/2)

Experimental near-singular distortion of the free surface.

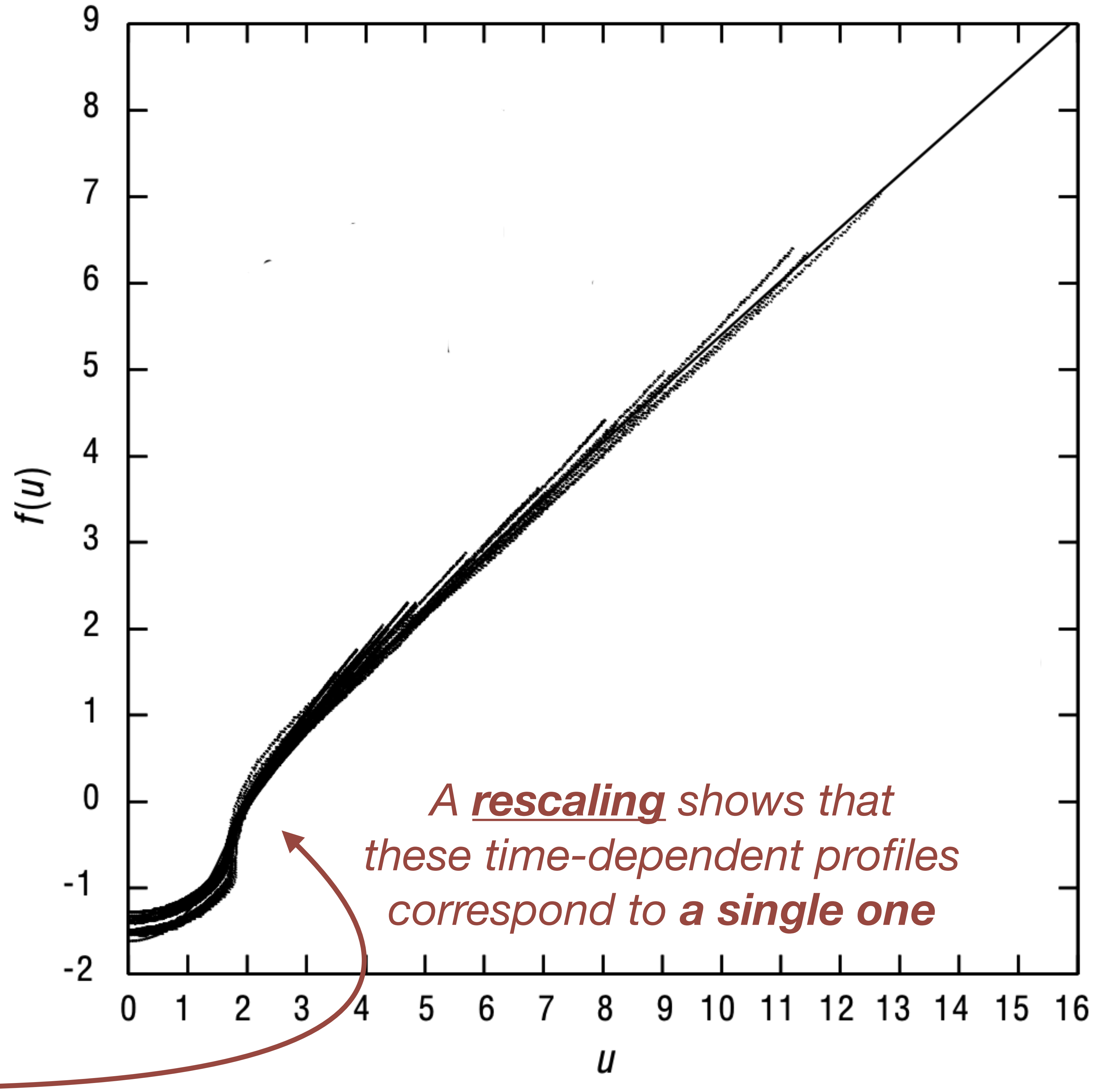
Balance between ***inertia*** *and* ***surface tension***.

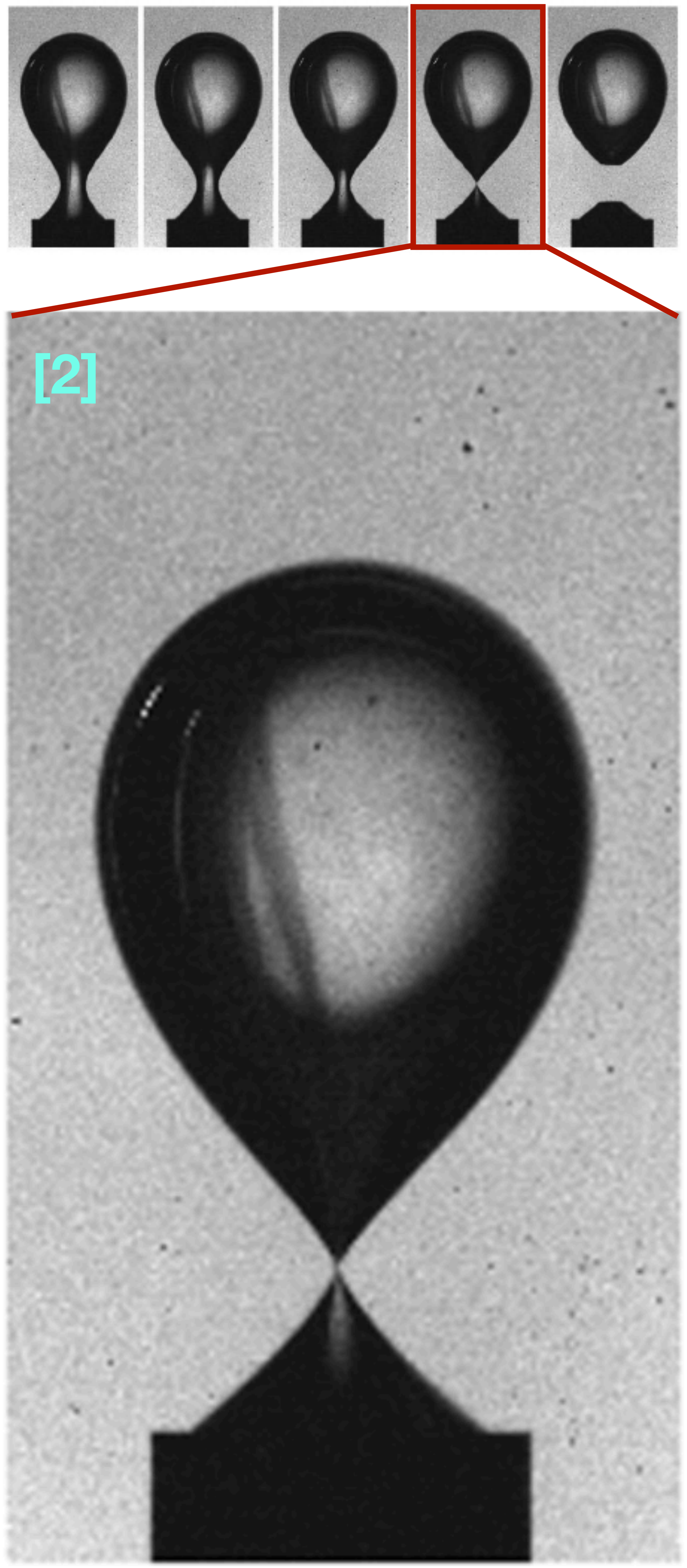
Collapsing cavity profiles seem to have the ***same shape****.*

<u>Scale invariance</u> with *homotheties*



[1] Zeff *et al.*, *Singularity dynamics in curvature collapse and jet eruption on fluid surface.* <u>Nature</u> **403** (2000)



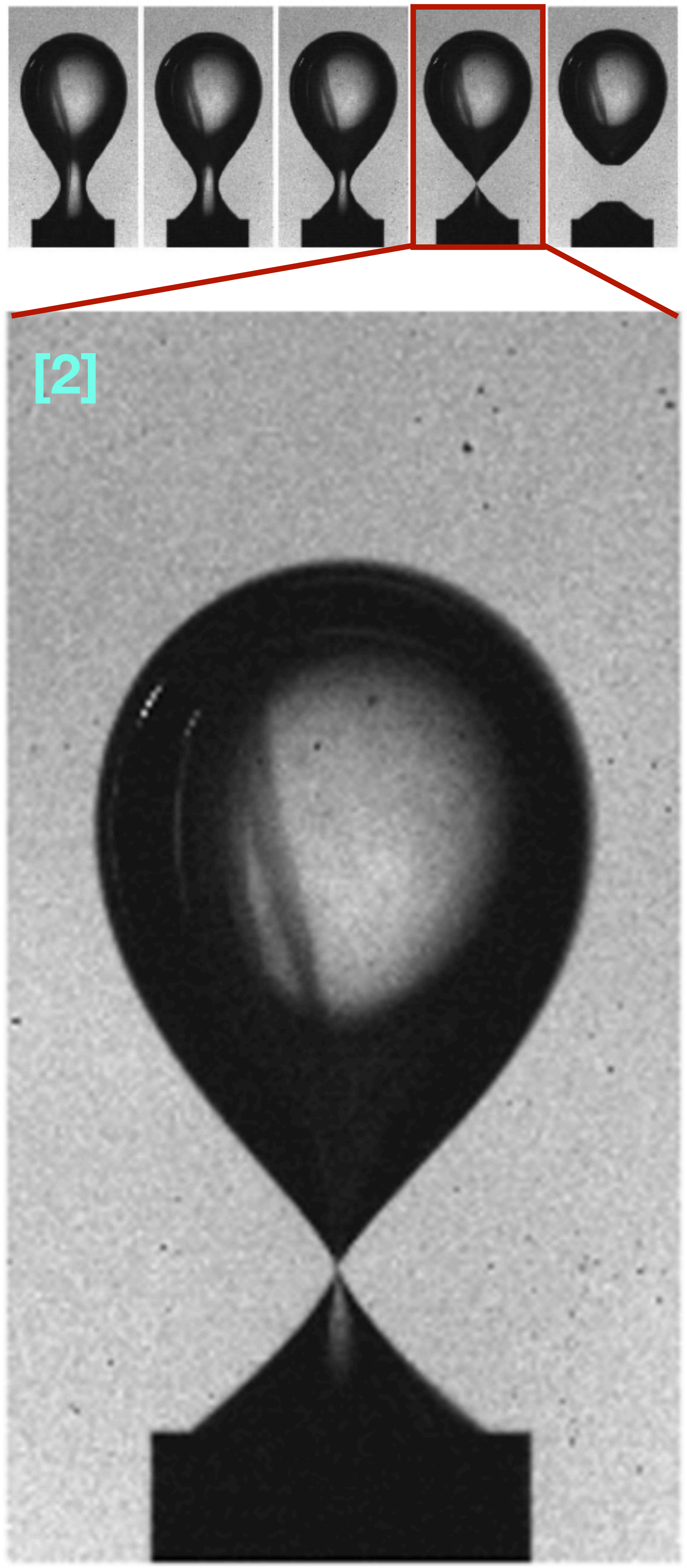


Bubble ***pinch-offs***

*A wide variety of **3D** physical problems highly **non-linear** involving locally **conical shapes**.*

Lengths scales evolving between 10^{-2} *and* 10^{-9} m.

[2] Thoroddsen *et al.*, *Experiments on bubble pinch-off.* Phys. Fluids **19** (2007)



Bubble **pinch-offs**

*A wide variety of **3D** physical problems highly **non-linear** involving locally **conical shapes**.*

Lengths scales evolving between 10^{-2} *and* 10^{-9} m.

⇒ Scanning all these scales requires high computational costs (*cf.* **[3], [4]**).

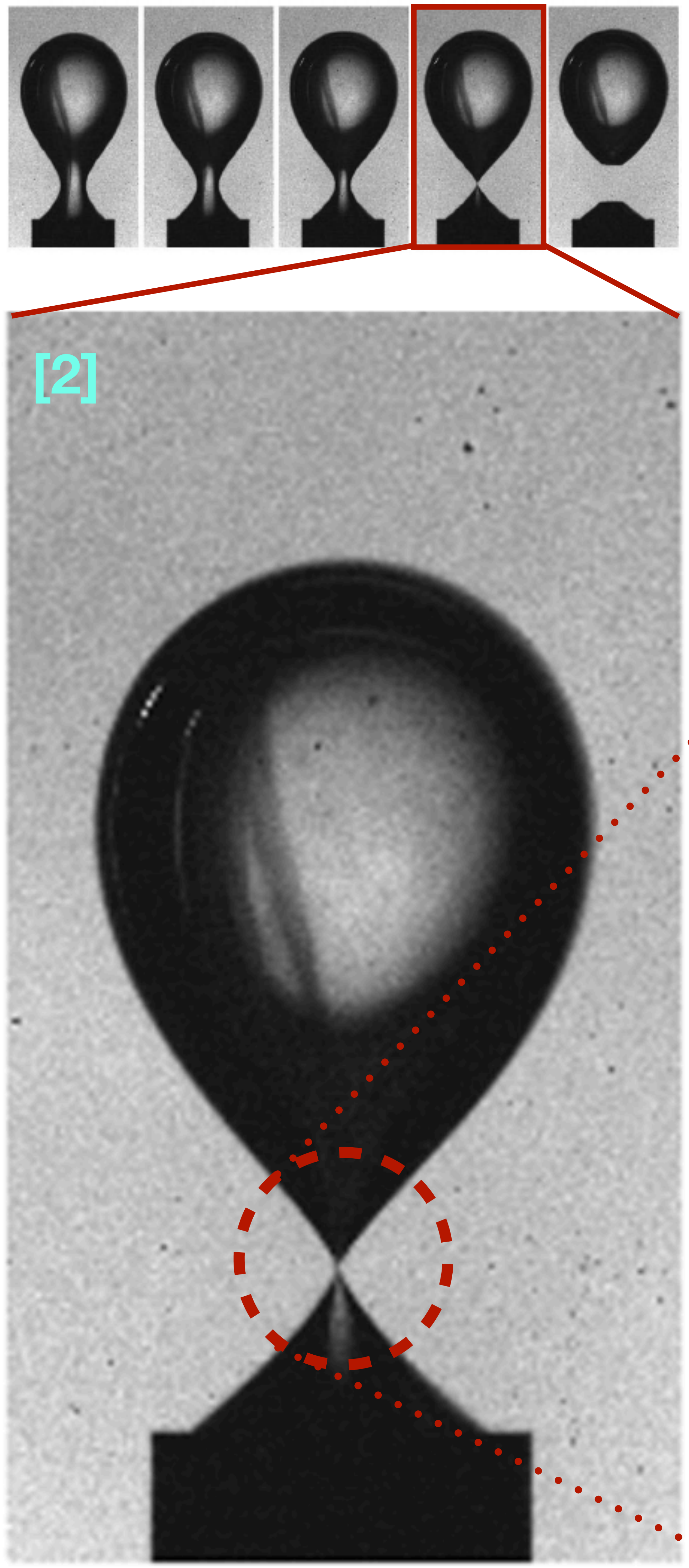
[2] Thoroddsen *et al.*, *Experiments on bubble pinch-off.* Phys. Fluids **19** (2007)

[3] A. M. Gañán-Calvo, *Revision of Bubble Bursting: Universal Scaling Laws of Top Jet Drop Size and Speed.* PRL **119** (2017)

[4] A. Berny, *Étude numérique de l'éclatement d'une bulle à la surface de différents liquides.* PhD Thesis (2020)

I - Introduction

I.1 - Visualizing Scale Invariances (2/2)



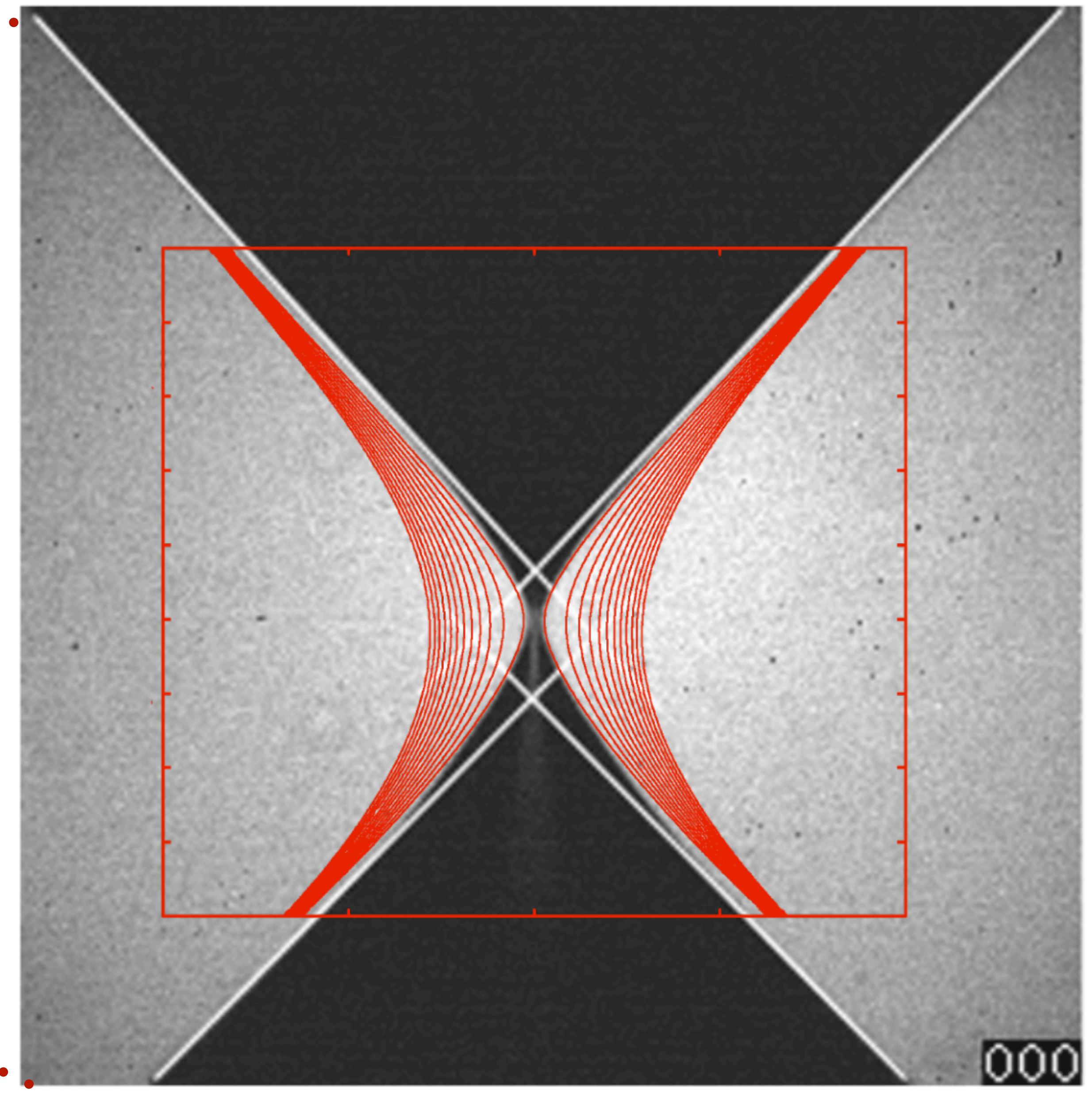
Bubble **pinch-offs**

*A wide variety of **3D** physical problems highly **non-linear** involving locally **conical shapes**.*

Lengths scales evolving between 10^{-2} *and* 10^{-9} m.

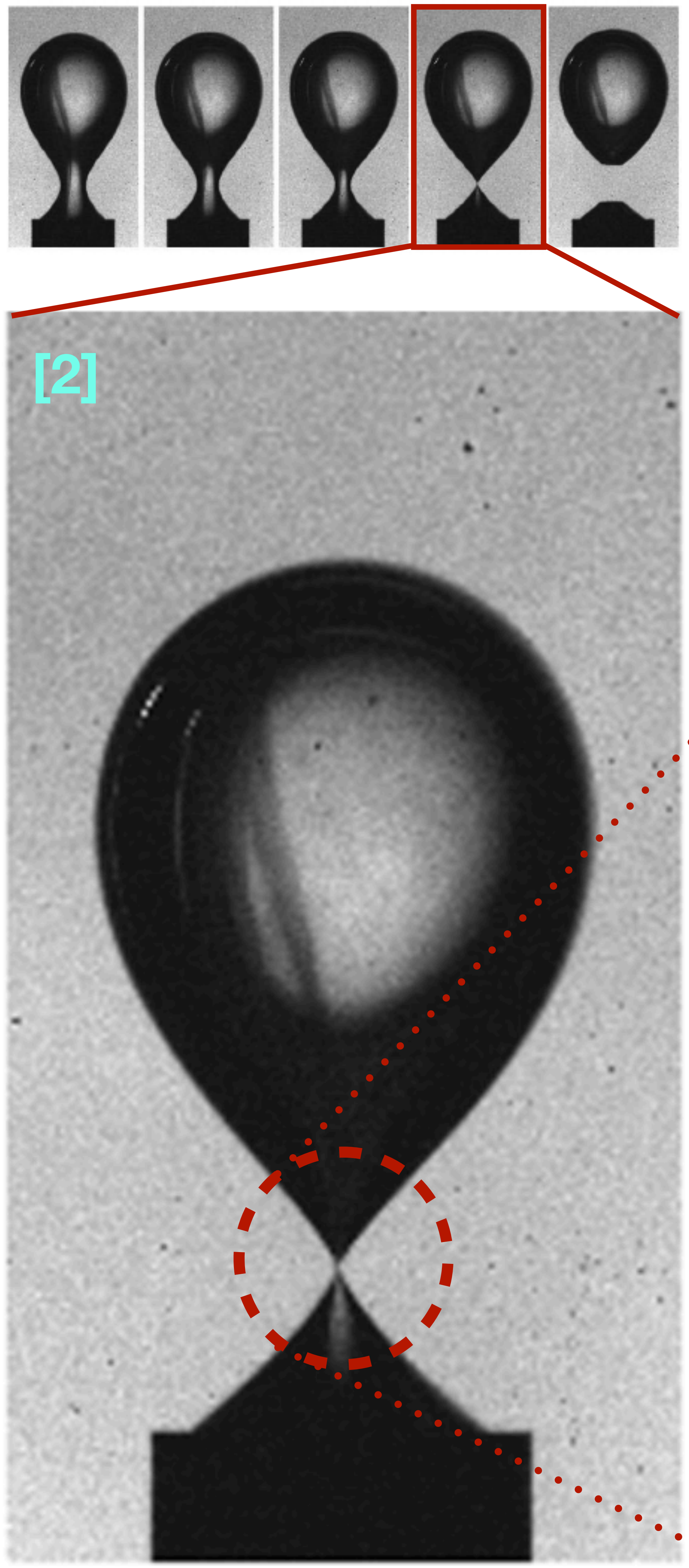
⇒ Scanning all these scales requires high computational costs (*cf.* **[3], [4]**).

[2] Thoroddsen *et al.*, *Experiments on bubble pinch-off.* Phys. Fluids **19** (2007)



[3] A. M. Gañán-Calvo, *Revision of Bubble Bursting: Universal Scaling Laws of Top Jet Drop Size and Speed.* PRL **119** (2017)

[4] A. Berny, *Étude numérique de l'éclatement d'une bulle à la surface de différents liquides.* PhD Thesis (2020)

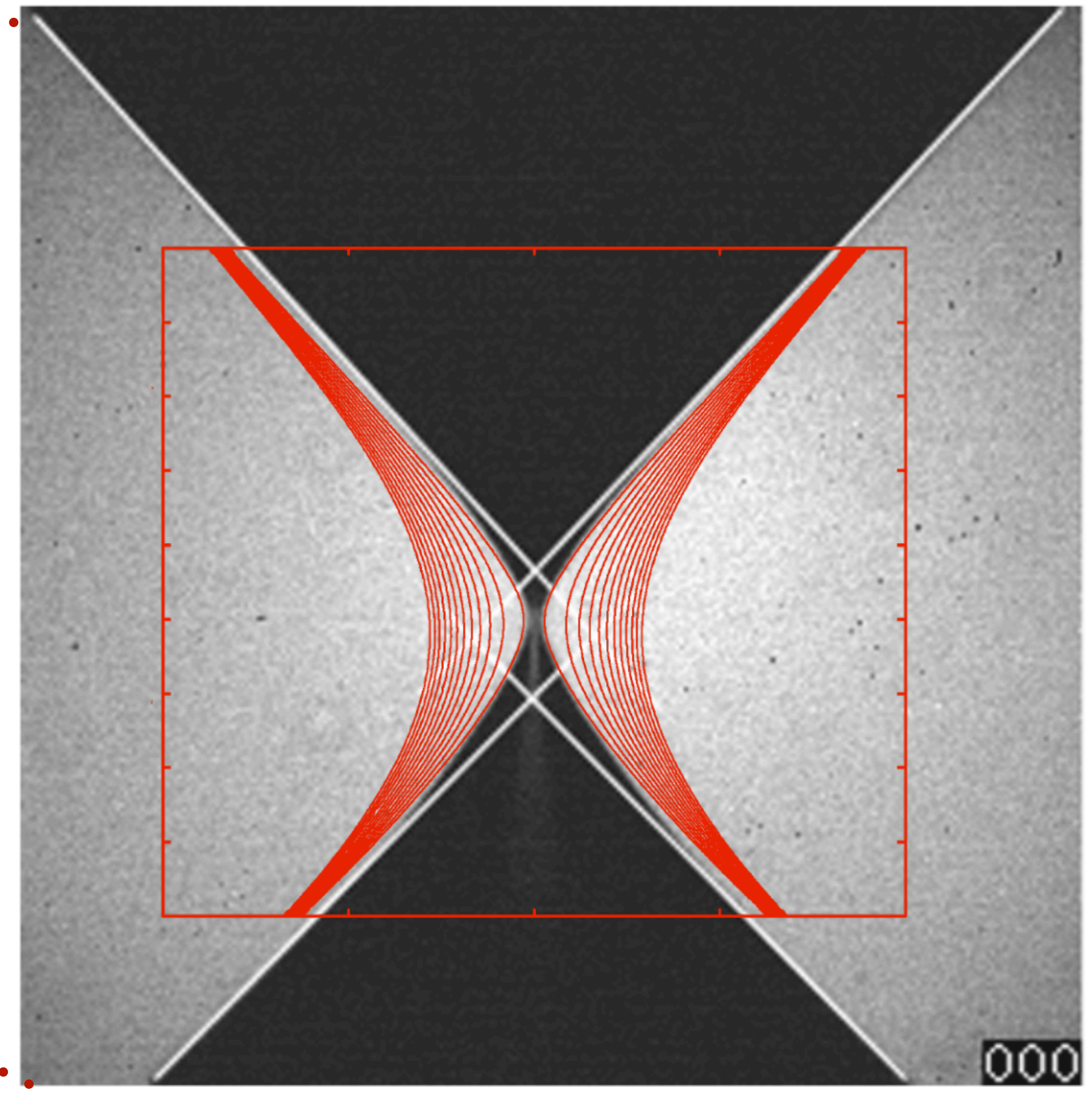


A wide variety of ***3D*** *physical problems highly* ***non-linear*** *involving locally* ***conical shapes***.

Lengths scales evolving between 10^{-2} *and* 10^{-9} m.

⇒ Scanning all these scales requires high computational costs (*cf.* **[3]**, **[4]**).

[2] Thoroddsen *et al.*, *Experiments on bubble pinch-off.* Phys. Fluids **19** (2007)



Bubble ***pinch-offs***

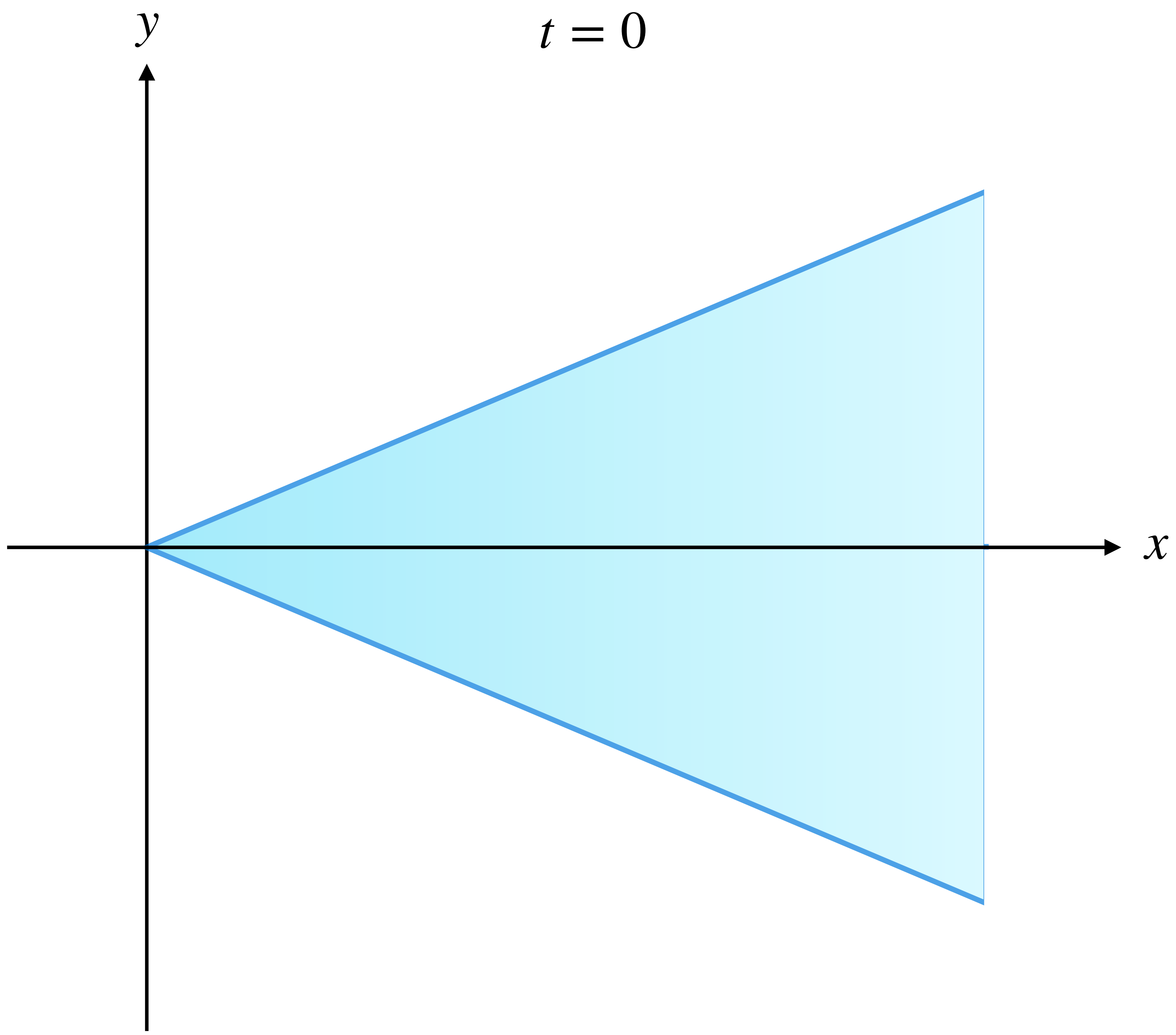
Again, scale invariance with cones seems to emerge, when looking at surface profile shapes!

[3] A. M. Gañán-Calvo, *Revision of Bubble Bursting: Universal Scaling Laws of Top Jet Drop Size and Speed.* PRL **119** (2017)

[4] A. Berny, *Étude numérique de l'éclatement d'une bulle à la surface de différents liquides.* PhD Thesis (2020)

II.A - Keller & Miksis [5] problem

Breaking of a liquid sheet assimilated to a wedge.

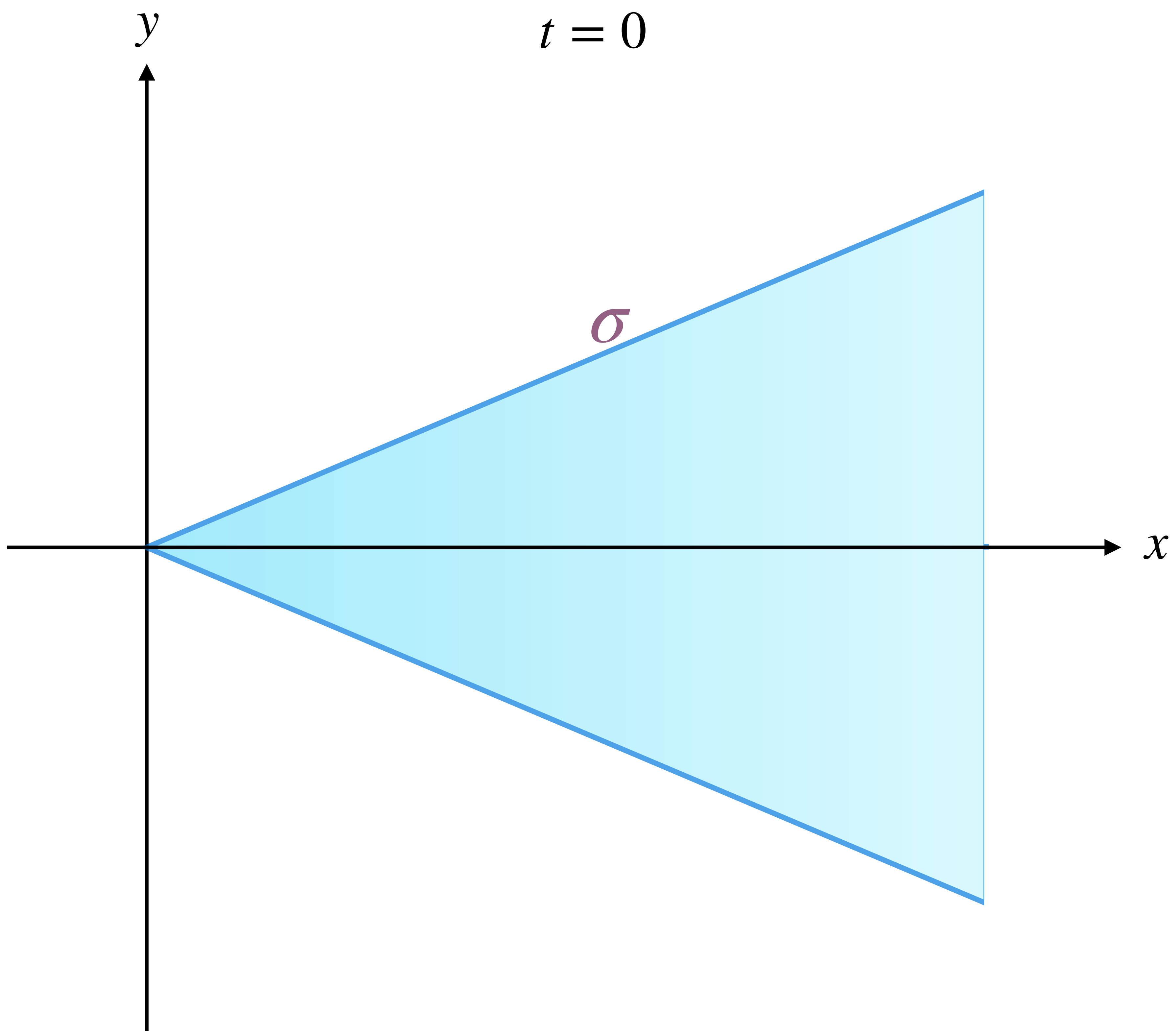


Cross-section of a tapered sheet of liquid (wedge)

[5] Keller and Miksis, *Surface Tension Driven Flows*. SIAM **43** (1983)

II.A - Keller & Miksis [5] problem

Breaking of a liquid sheet assimilated to a wedge.



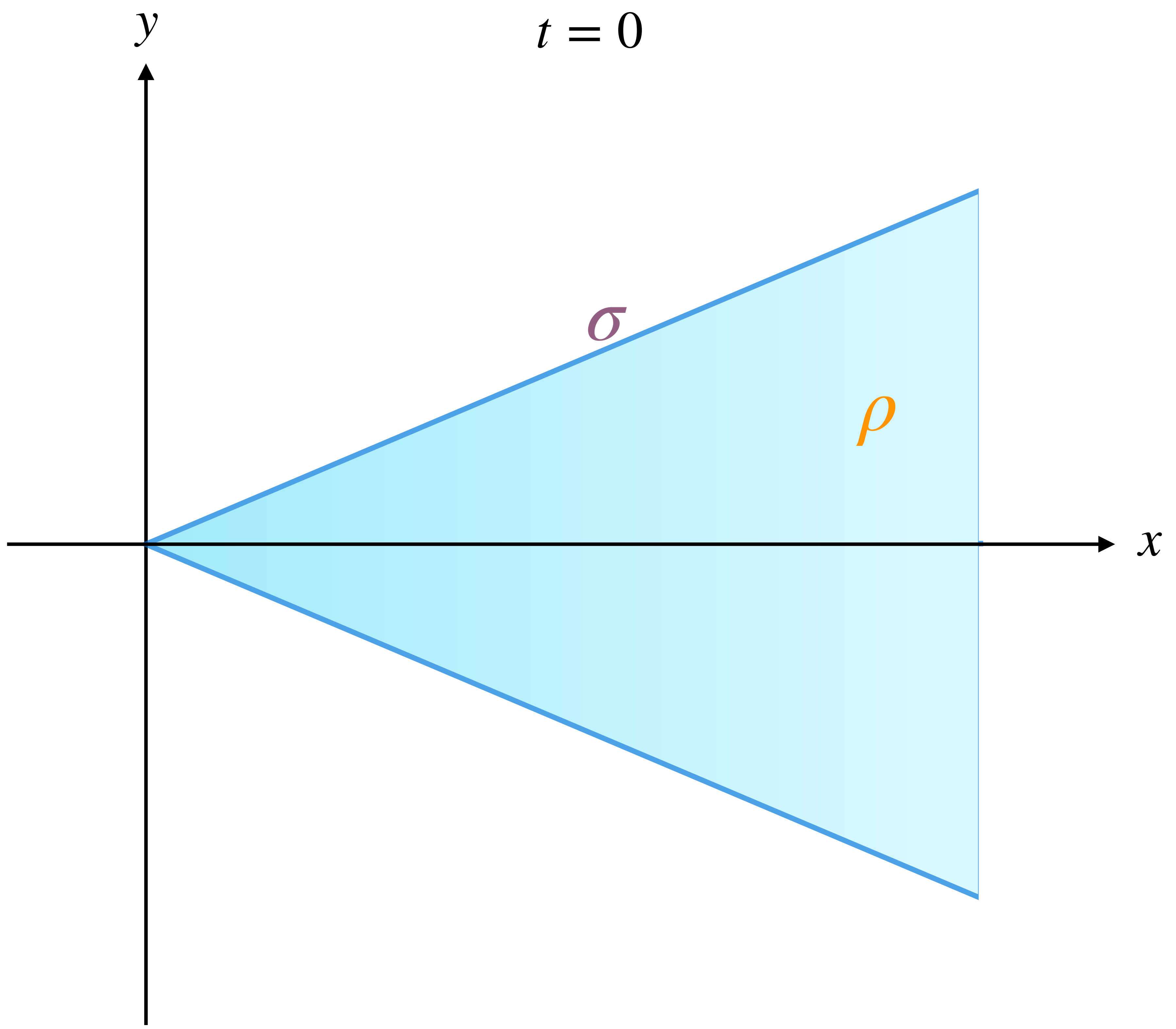
Cross-section of a tapered sheet of liquid (wedge)

σ - surface tension

[5] Keller and Miksis, *Surface Tension Driven Flows*. SIAM **43** (1983)

II.A - Keller & Miksis [5] problem

Breaking of a liquid sheet assimilated to a wedge.



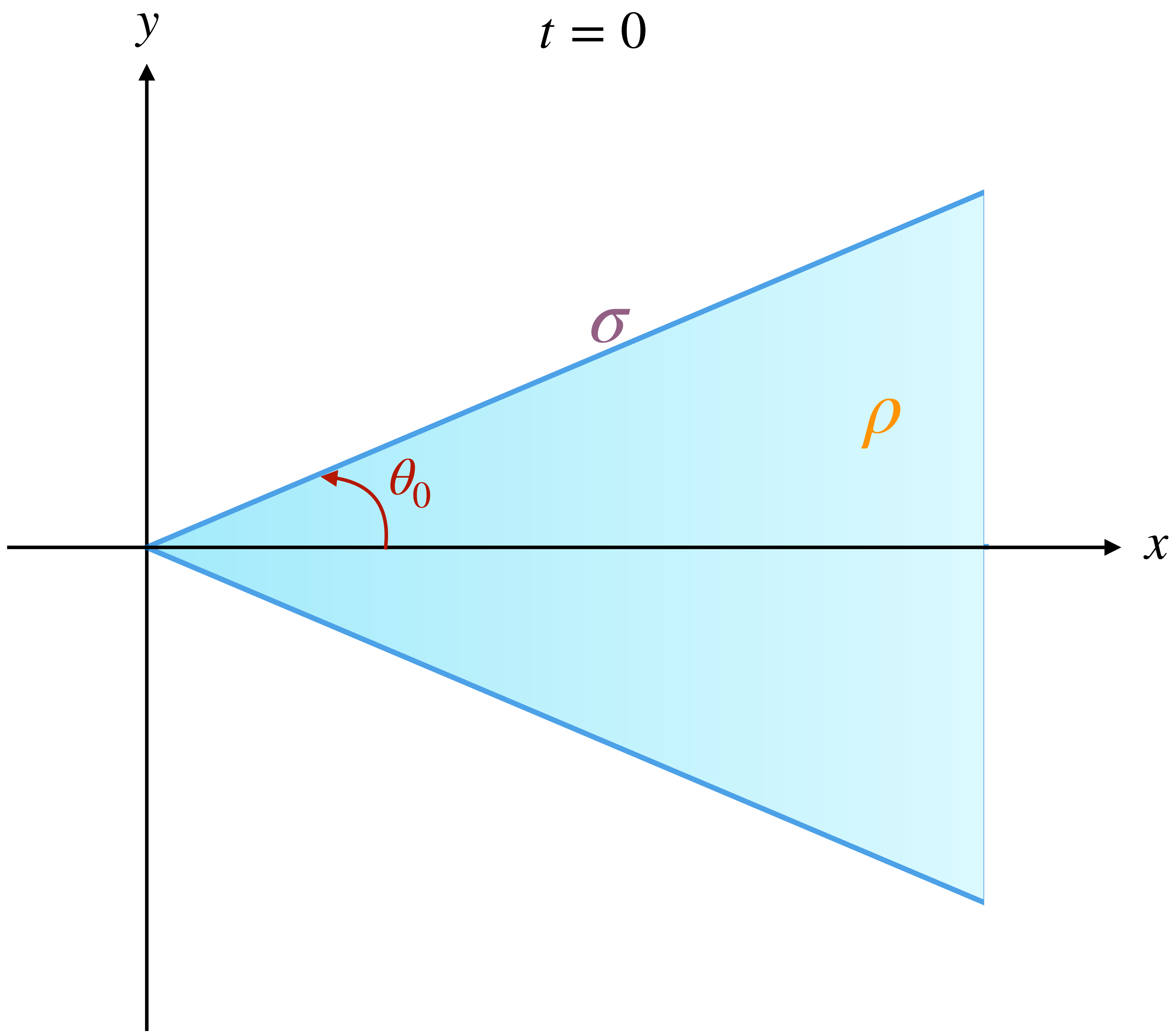
Cross-section of a tapered sheet of liquid (wedge)

σ - **surface tension**

ρ - **density**

[5] Keller and Miksis, *Surface Tension Driven Flows.* SIAM **43** (1983)

Breaking of a liquid sheet assimilated to a wedge.



Cross-section of a tapered sheet of liquid (wedge)

σ - surface tension

ρ - density

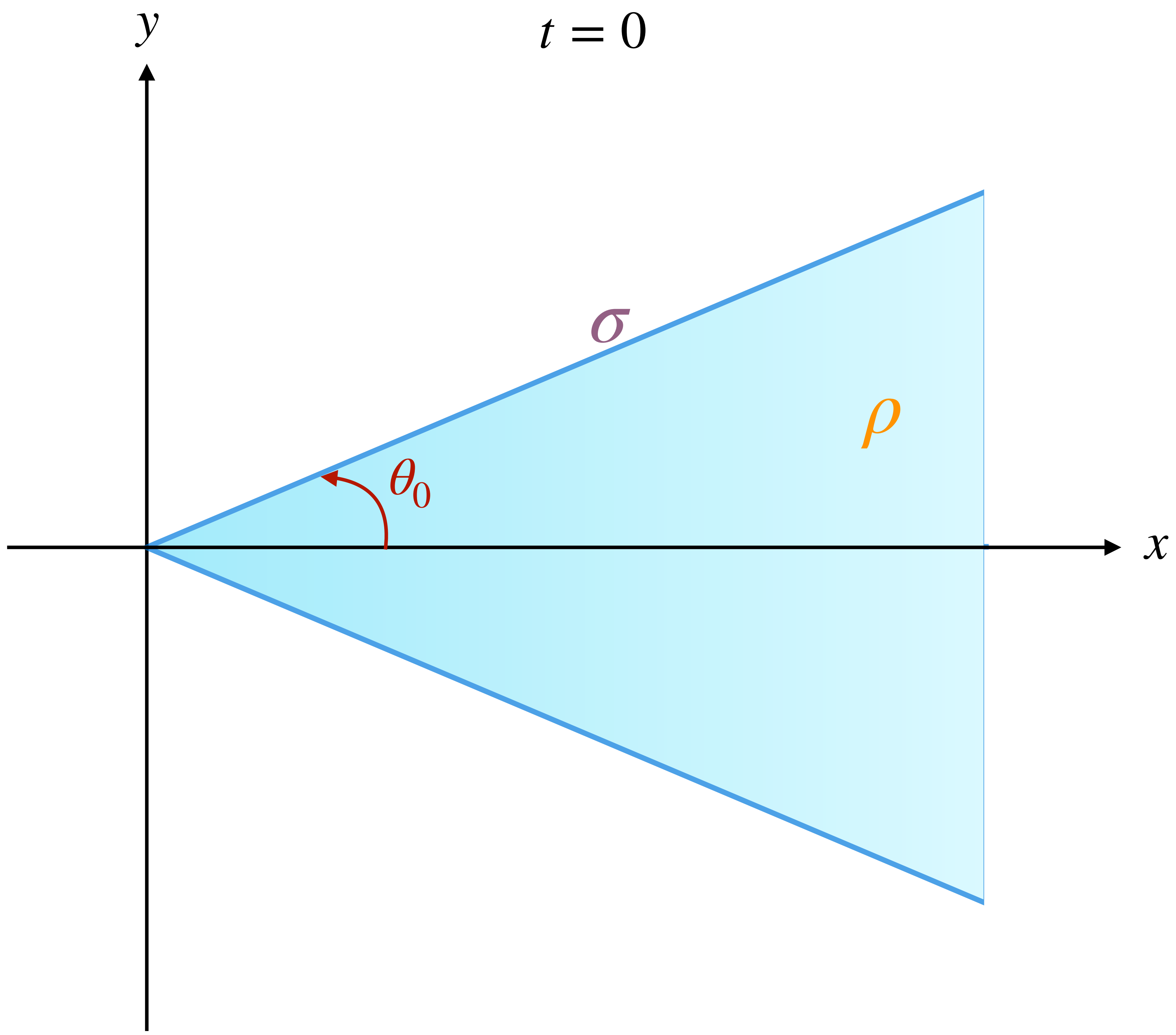
θ_0 - initial and far-field angle

[5] Keller and Miksis, *Surface Tension Driven Flows.* SIAM **43** (1983)

II.A - Keller & Miksis [5] problem

Breaking of a liquid sheet assimilated to a wedge.

Recoil near the vertex of the wedge under surface tension effects.



Cross-section of a tapered sheet of liquid (wedge)

σ - **surface tension**

ρ - **density**

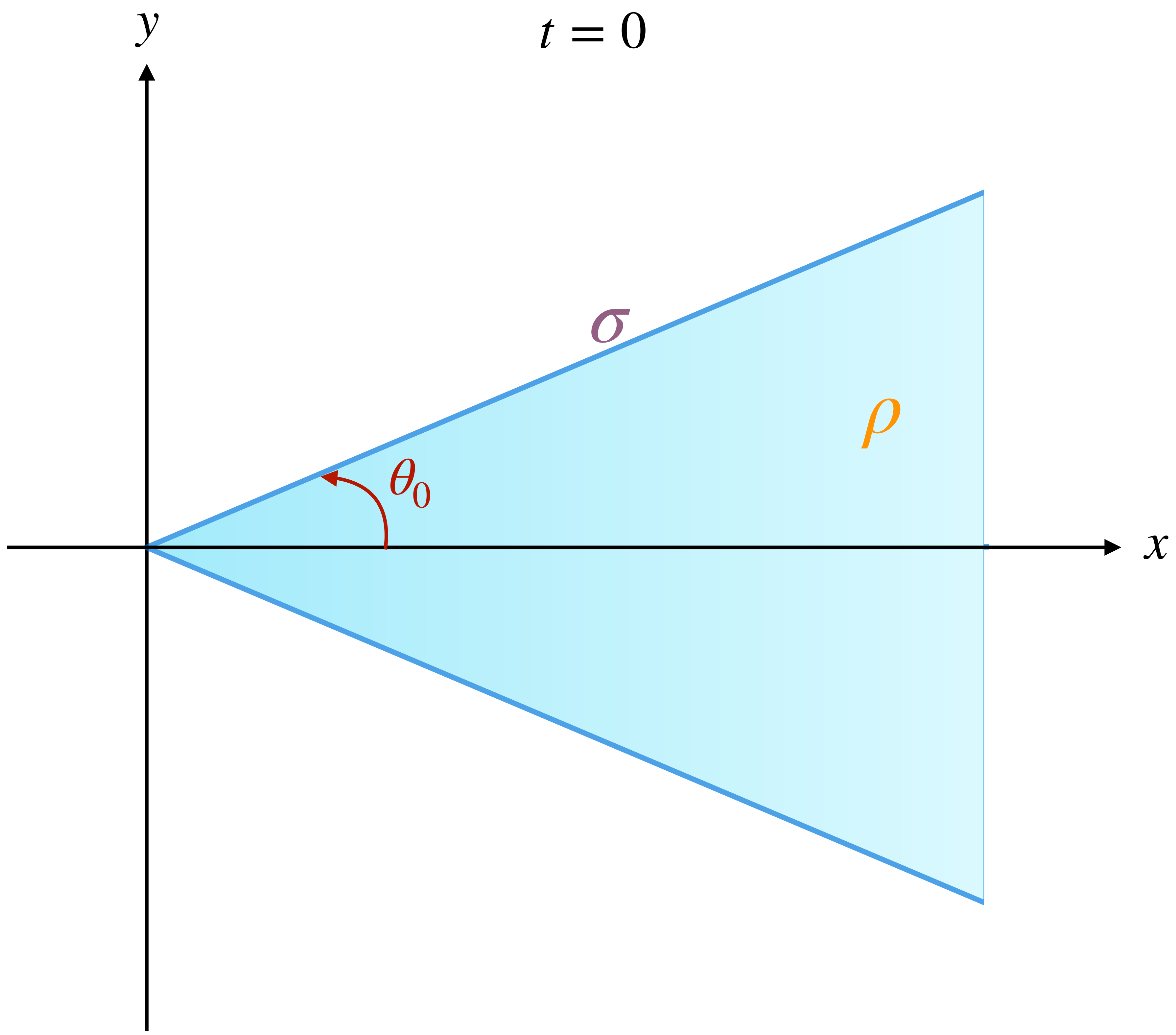
θ_0 - **initial and far-field angle**

[5] Keller and Miksis, *Surface Tension Driven Flows.* SIAM **43** (1983)

Breaking of a liquid sheet assimilated to a wedge.

Recoil near the vertex of the wedge under surface tension effects.

No viscosity taken into account.



Cross-section of a tapered sheet of liquid (wedge)

σ **- surface tension**

ρ **- density**

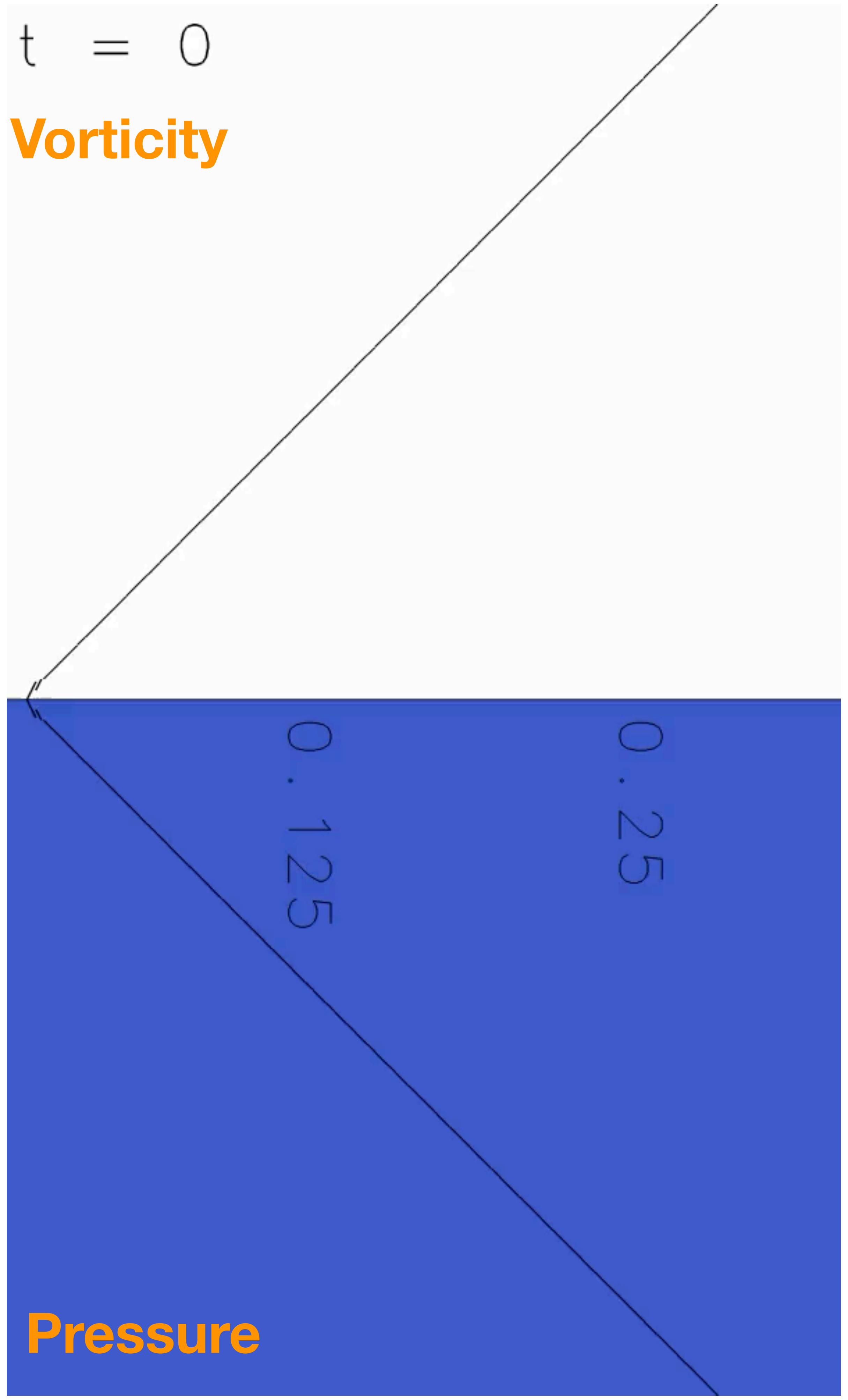
θ_0 **- initial and far-field angle**

[5] Keller and Miksis, *Surface Tension Driven Flows*. SIAM **43** (1983)

What is happening if we are simulating this study case?

$$N = 7$$
$$N_{max} = 8$$
$$L_0 = 1$$
$$\theta_0 = 45°$$
$$\beta_0 = 90°$$
$$\rho_{fluid} = 1$$
$$\rho_{gas} = 10^{-3}$$
$$\sigma = 1$$

Global Parameters



[5] **Keller and Miksis**, *Surface Tension Driven Flows*. SIAM **43** (1983)

II.A - Keller & Miksis [5] problem

What is happening if we are simulating this study case?

$N = 7$

$N_{max} = 8$

$L_0 = 1$

$\theta_0 = 45°$

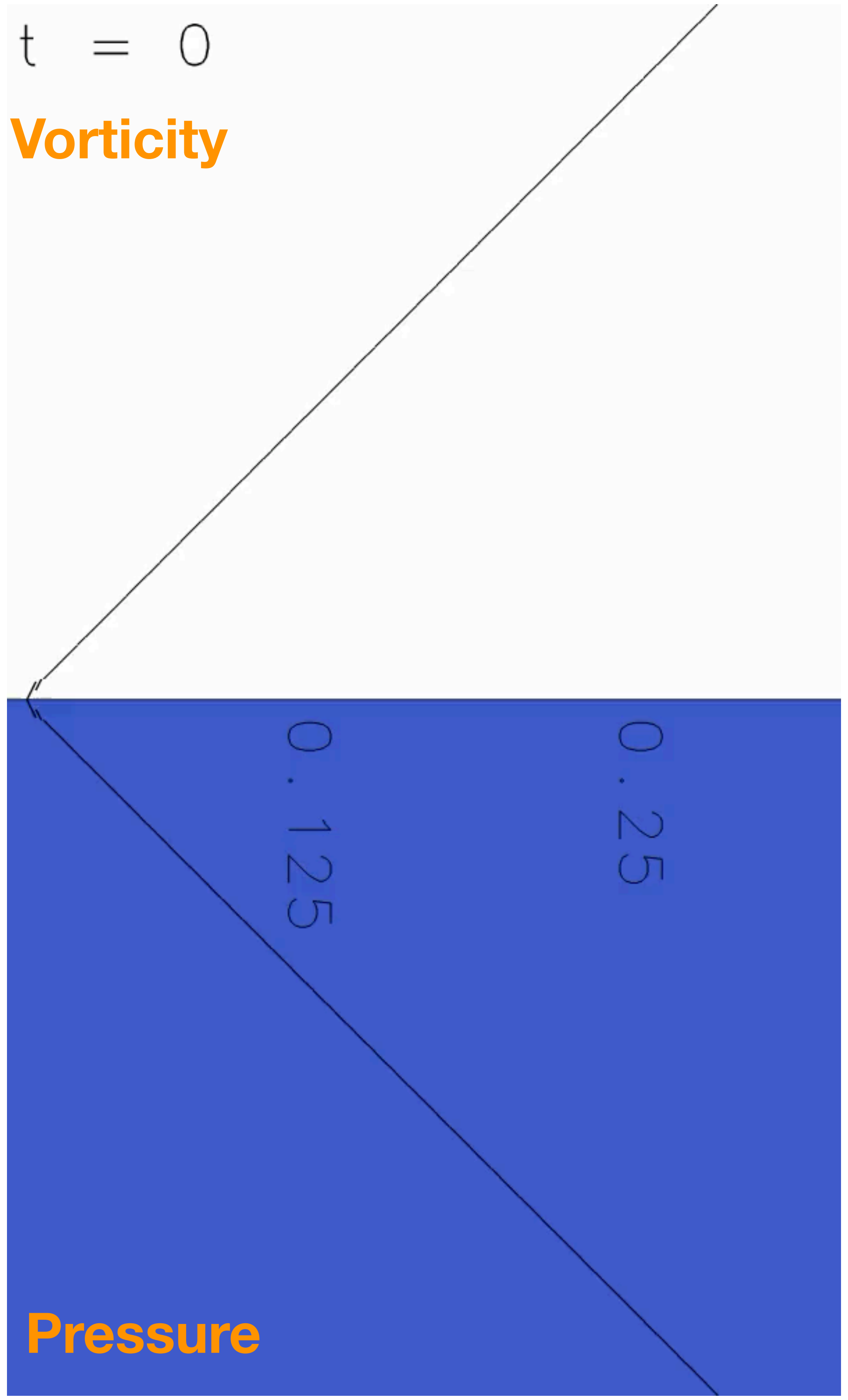
$\beta_0 = 90°$

$\rho_{fluid} = 1$

$\rho_{gas} = 10^{-3}$

$\sigma = 1$

Global Parameters



[5] Keller and Miksis, *Surface Tension Driven Flows.* SIAM **43** (1983)

What is happening if we are simulating this study case?

$N = 7$

$N_{max} = 8$

$L_0 = 1$

$\theta_0 = 45°$

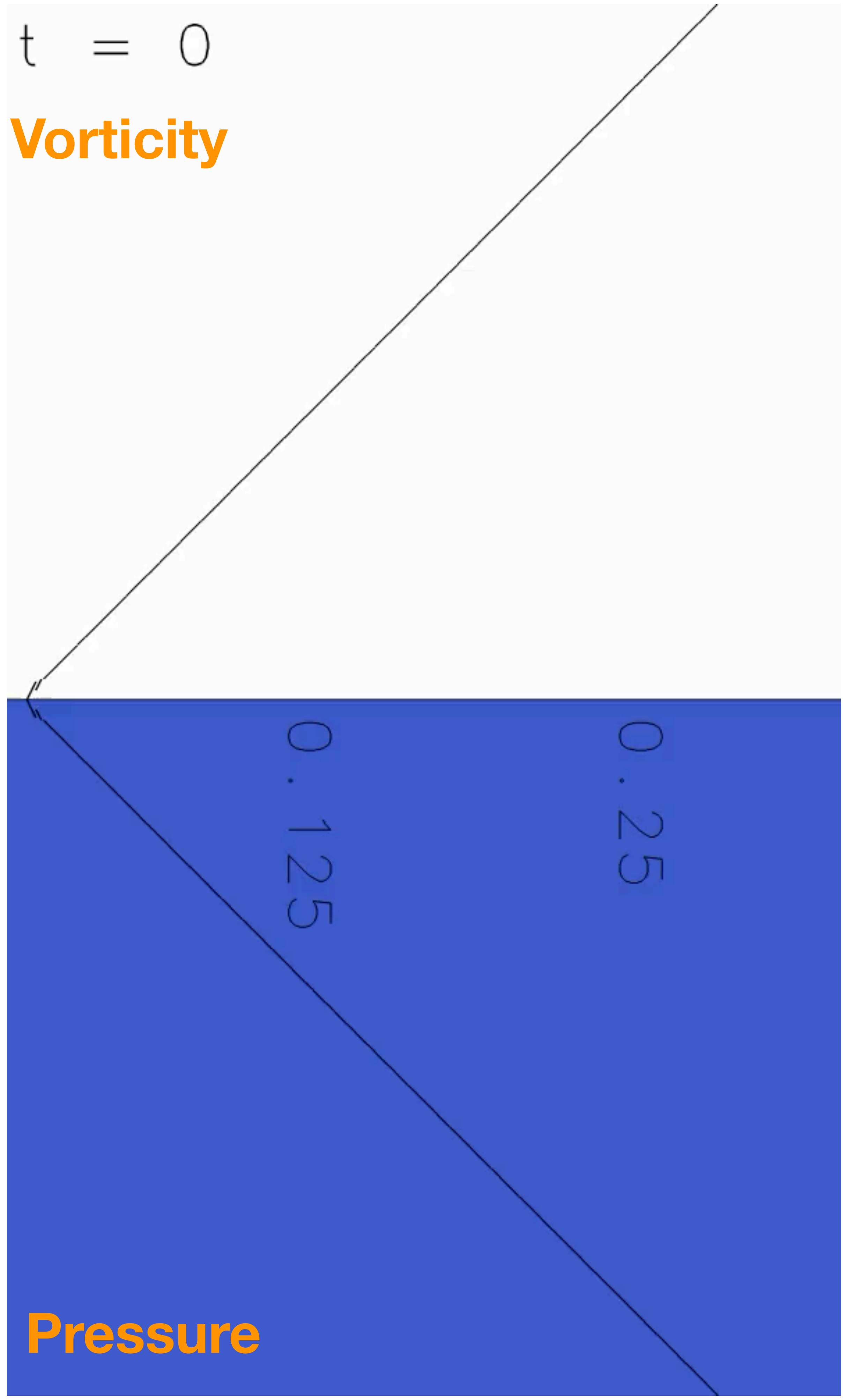
$\beta_0 = 90°$

$\rho_{fluid} = 1$

$\rho_{gas} = 10^{-3}$

$\sigma = 1$

Global Parameters



As shown in the video ⇒ too many scales to be scanned!

[5] Keller and Miksis, *Surface Tension Driven Flows.* SIAM **43** (1983)

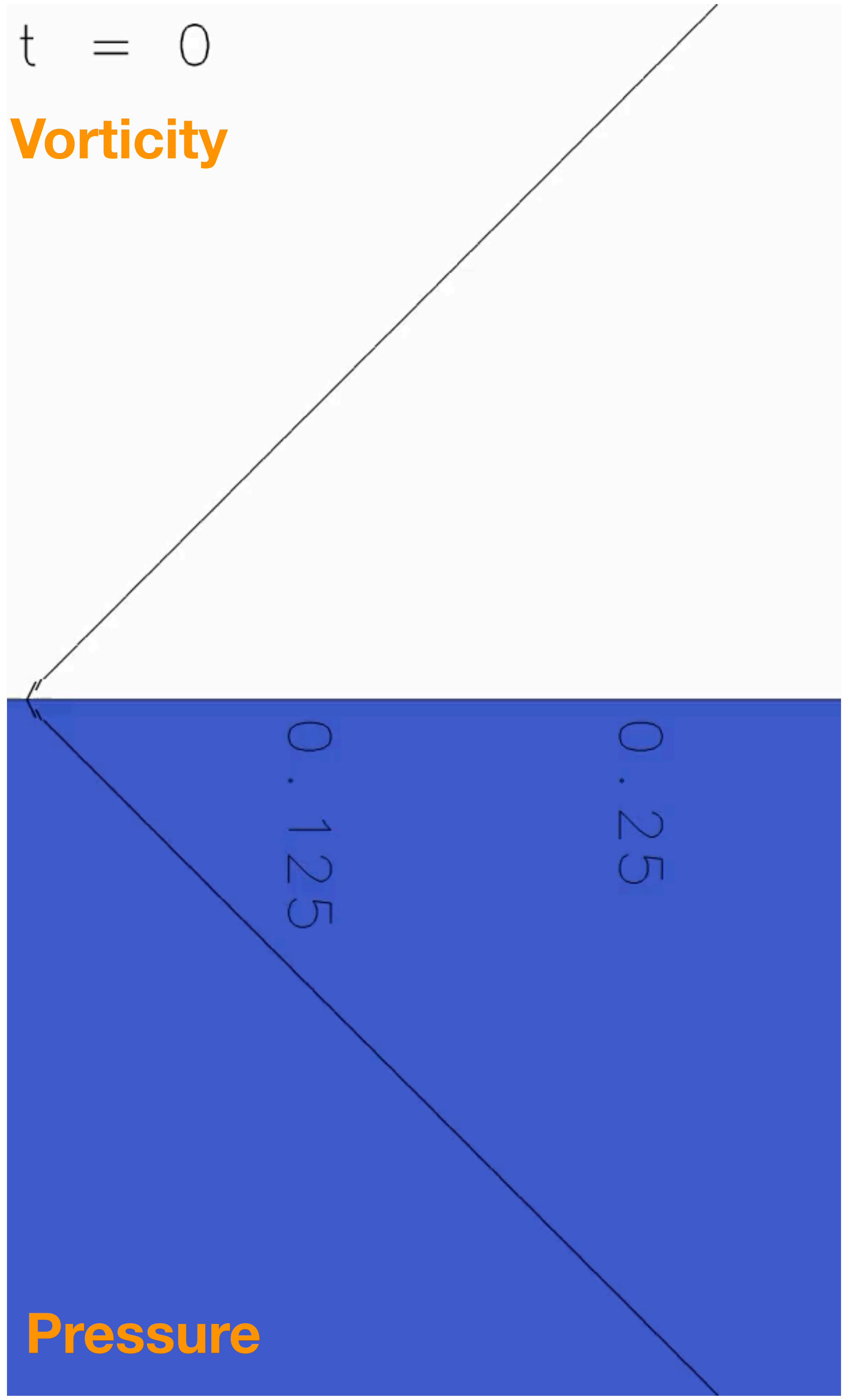
II.A - Keller & Miksis [5] problem

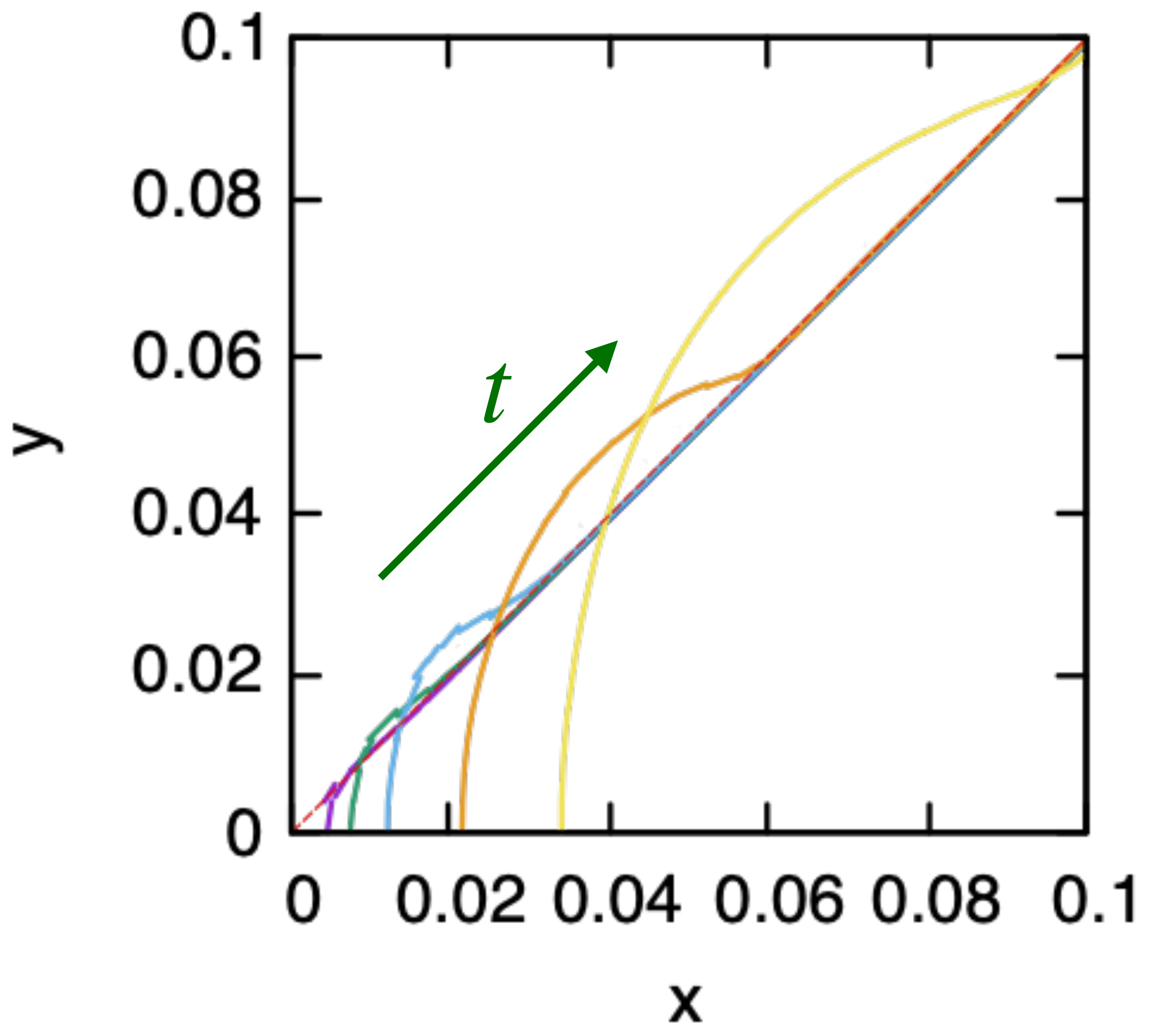
II.2 - *Simulation in the Physical Space*

What is happening if we are simulating this study case?

$N = 7$
$N_{max} = 8$
$L_0 = 1$
$\theta_0 = 45°$
$\beta_0 = 90°$
$\rho_{fluid} = 1$
$\rho_{gas} = 10^{-3}$
$\sigma = 1$

Global Parameters





In the physical space, a *similar pattern* is observed, signature of a **scale invariance**?

As shown in the video ⇒ too many scales to be scanned!

[5] Keller and Miksis, *Surface Tension Driven Flows.* SIAM **43** (1983)

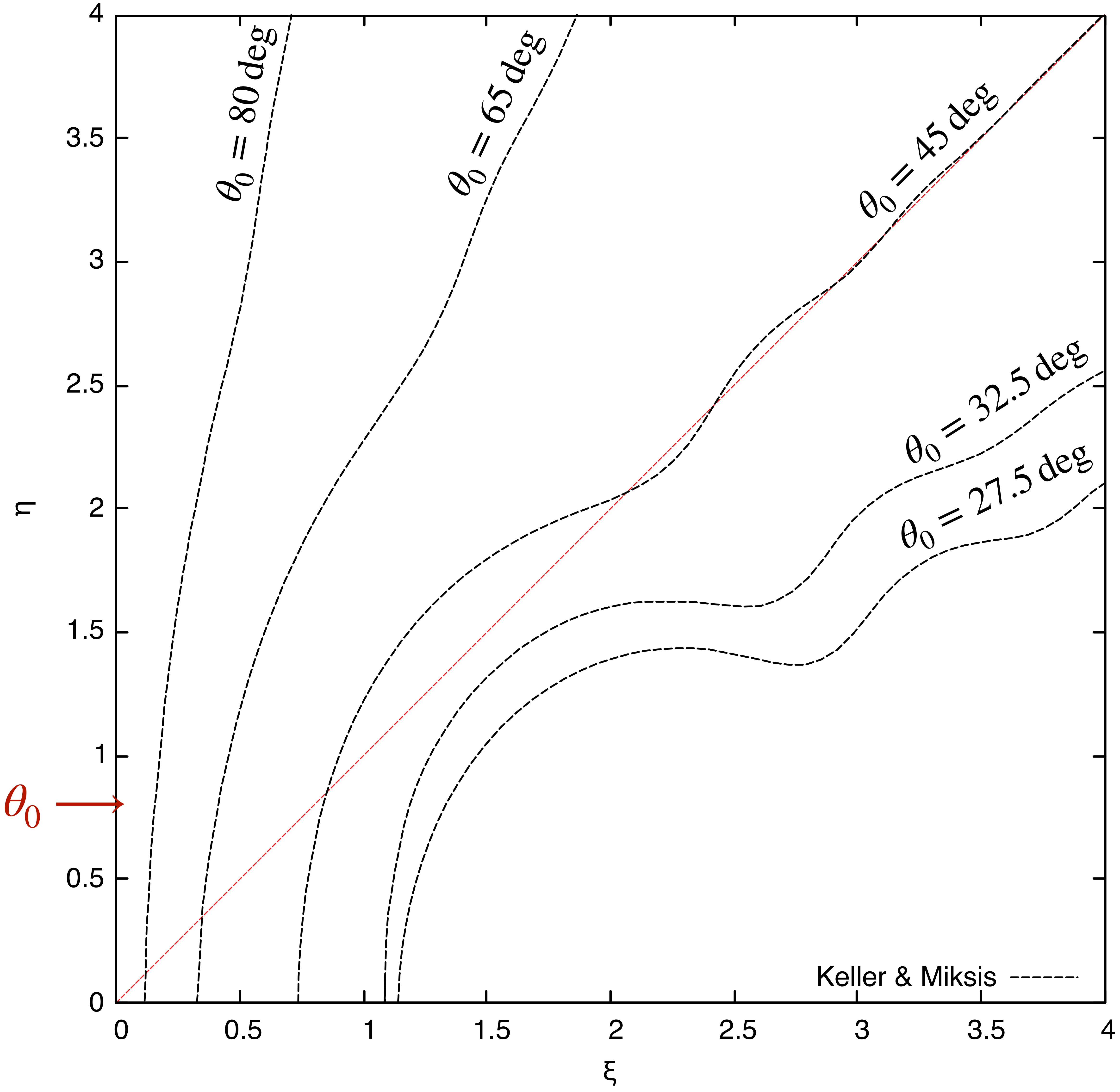
II.A - Keller & Miksis [5] problem

- Keller and Miksis determined theoretically the *scale invariance*
- Their numerical results were obtained:
  - by using a *Boundary Integral Method*;
  - searching for a **stationary solution**

To that end, they relied on 2 main assumptions:

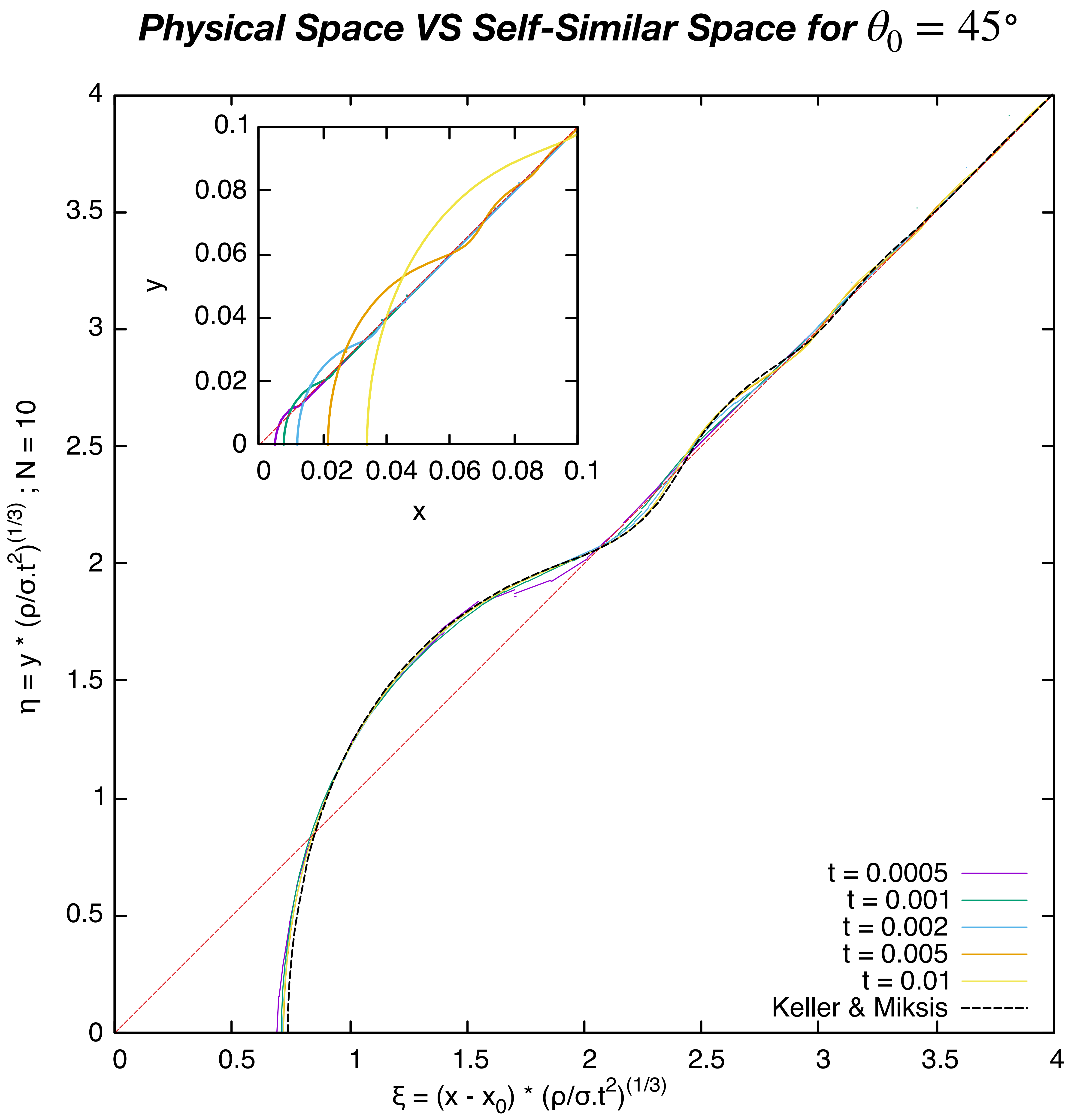
1. The fluid is **irrotational**: *potential theory*
2. The fluid is **inviscid**.

STATIONARY SOLUTIONS FOR DIFFERENT WEDGE ANGLES θ_0 →

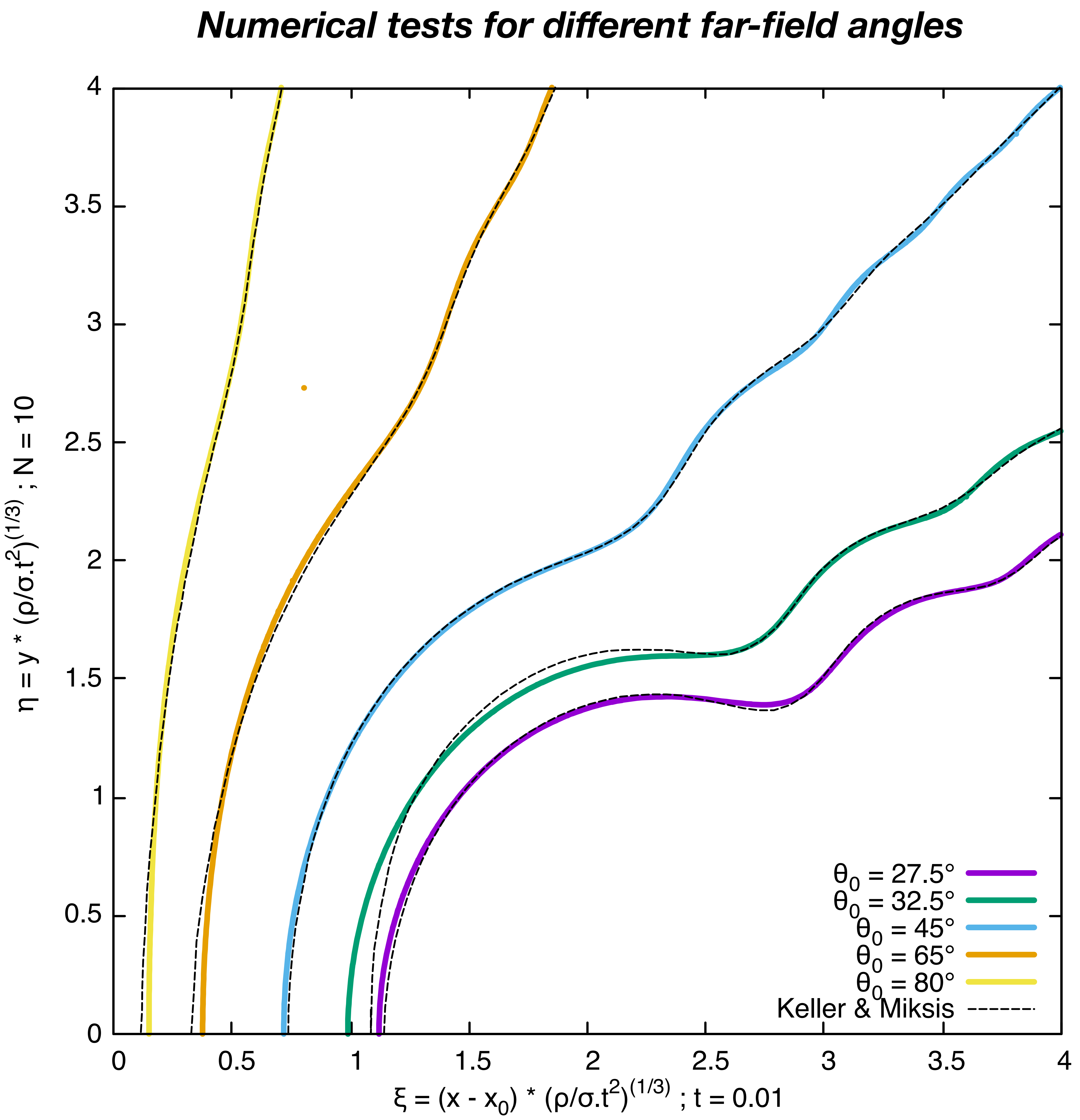


[5] **Keller and Miksis**, *Surface Tension Driven Flows*. SIAM **43** (1983)

II.A - Keller & Miksis [5] problem



II.4 - Rescaled simulations: invariance



[5] Keller and Miksis, *Surface Tension Driven Flows.* SIAM **43** (1983)

1. **Change of scale** on each quantity x_i involved in the problem, under the form $x_i \mapsto x_i^* x_i'$, where x_i' is the *transformed variable* and x_i^* is a *scaling factor.*

1. **Change of scale** on each quantity x_i involved in the problem, under the form $x_i \mapsto x_i^* x_i'$, where x_i' is the *transformed variable* and x_i^* is a *scaling factor.*

1. **Change of scale** on each quantity x_i involved in the problem, under the form $x_i \mapsto x_i^* x_i'$, where x_i' is the *transformed variable* and x_i^* is a *scaling factor.*

2. **Find the scaling group that leaves the problem invariant:**
   After changing scales, the problem written for x_i' is formally identical to the one written for x_i; this implies ***constraints***.

1. **Change of scale** on each quantity x_i involved in the problem, under the form $x_i \mapsto x_i^* x_i'$, where x_i' is the *transformed variable* and x_i^* is a *scaling factor.*

2. **Find the scaling group that leaves the problem invariant:** After changing scales, the problem written for x_i' is formally identical to the one written for x_i; this implies ***constraints***.

1. **Change of scale** on each quantity x_i involved in the problem, under the form $x_i \mapsto x_i^* x_i'$, where x_i' is the *transformed variable* and x_i^* is a *scaling factor.*

2. **Find the scaling group that leaves the problem invariant:**
   After changing scales, the problem written for x_i' is formally identical to the one written for x_i; this implies ***constraints***.

3. **Choice of free parameters:**
   Counting the number of *independent relationships* R and *scaling factors* S:
   $F = R - S$ free parameters letting invariant the problem.

II.5 - The Scale Invariance Method

1. **Change of scale** on each quantity x_i involved in the problem, under the form $x_i \mapsto x_i^* x_i'$, where x_i' is the *transformed variable* and x_i^* is a *scaling factor.*

2. **Find the scaling group that leaves the problem invariant:** After changing scales, the problem written for x_i' is formally identical to the one written for x_i; this implies ***constraints***.

3. **Choice of free parameters:** Counting the number of *independent relationships* R and *scaling factors* S:

   $F = R - S$ free parameters letting invariant the problem.

1. **Change of scale** on each quantity x_i involved in the problem, under the form $x_i \mapsto x_i^* x_i'$, where x_i' is the *transformed variable* and x_i^* is a *scaling factor.*

2. **Find the scaling group that leaves the problem invariant:**
   After changing scales, the problem written for x_i' is formally identical to the one written for x_i; this implies ***constraints***.

3. **Choice of free parameters:**
   Counting the number of *independent relationships* R and *scaling factors* S:
   $F = R - S$ free parameters letting invariant the problem.

4. **Consequences for the solution:**
   As the original problem is invariant to these changes of scale, so is its solution.

Simplified cartesian Navier-Stokes eqns:

$$\partial_x u = -\,\partial_y v$$

$$\partial_t u + u\,\partial_x u + v\,\partial_y u = -\,\partial_x(p/\rho)$$

$$\partial_t v + u\,\partial_x v + v\,\partial_y v = -\,\partial_y(p/\rho)$$

Initial Condition (free-surface shape):

$$f(x, y, 0) = y\cot\theta_0 - x$$

Kinematic Condition:

$$\partial_t f + u\partial_x f = 0$$

$$\partial_t f + v\partial_x f = 0$$

Dynamic Condition:

$$p(x, y, t)\Big|_{f=0} = \sigma\kappa$$

where κ is the *curvature.*

Simplified cartesian Navier-Stokes eqns:

$$\partial_x u = -\partial_y v$$

$$\partial_t u + u\,\partial_x u + v\,\partial_y u = -\partial_x(p/\rho)$$

$$\partial_t v + u\,\partial_x v + v\,\partial_y v = -\partial_y(p/\rho)$$

Initial Condition (free-surface shape):

$$f(x,y,0) = y\cot\theta_0 - x$$

Kinematic Condition:

$$\partial_t f + u\partial_x f = 0$$

$$\partial_t f + v\partial_x f = 0$$

Dynamic Condition:

$$p(x,y,t)\Big|_{f=0} = \sigma\kappa$$

where κ is the *curvature.*

DECOMPOSITION

$$x = x^* x', \quad y = y^* y', \quad t = t^* t', \quad u = u^* u',$$

$$v = v^* v', \quad p = p^* p', \quad \rho = \rho^* \rho', \quad \sigma = \sigma^* \sigma'$$

Simplified cartesian Navier-Stokes eqns:

$$\partial_x u = -\partial_y v$$
$$\partial_t u + u\,\partial_x u + v\,\partial_y u = -\partial_x (p/\rho)$$
$$\partial_t v + u\,\partial_x v + v\,\partial_y v = -\partial_y (p/\rho)$$

Initial Condition (free-surface shape):

$$f(x, y, 0) = y \cot \theta_0 - x$$

Kinematic Condition:

$$\partial_t f + u \partial_x f = 0$$
$$\partial_t f + v \partial_x f = 0$$

Dynamic Condition:

$$p(x, y, t)\Big|_{f=0} = \sigma\kappa$$

where κ is the *curvature.*

DECOMPOSITION

$$x = x^* x', \quad y = y^* y', \quad t = t^* t', \quad u = u^* u',$$
$$v = v^* v', \quad p = p^* p', \quad \rho = \rho^* \rho', \quad \sigma = \sigma^* \sigma'$$

To conserve the scales and let ***invariant*** *the problem, the following constraints have to be respected:*

$$\frac{u^*}{v^*} \overset{1}{=} \frac{x^*}{y^*} \qquad \frac{u^*}{t^*} \overset{2}{=} \frac{u^{*2}}{x^*} \overset{3}{=} \frac{1}{x^*}\frac{p^*}{\rho^*} \qquad p^* \overset{4}{=} \frac{\sigma^*}{x^*} \qquad x^* \overset{5}{=} y^*$$

Simplified cartesian Navier-Stokes eqns:

$$\partial_x u = -\partial_y v$$
$$\partial_t u + u\,\partial_x u + v\,\partial_y u = -\partial_x (p/\rho)$$
$$\partial_t v + u\,\partial_x v + v\,\partial_y v = -\partial_y (p/\rho)$$

Initial Condition (free-surface shape):

$$f(x, y, 0) = y \cot\theta_0 - x$$

Kinematic Condition:

$$\partial_t f + u \partial_x f = 0$$
$$\partial_t f + v \partial_x f = 0$$

Dynamic Condition:

$$p(x, y, t)\Big|_{f=0} = \sigma\kappa$$

where κ is the *curvature.*

DECOMPOSITION

$$x = x^* x', \quad y = y^* y', \quad t = t^* t', \quad u = u^* u',$$
$$v = v^* v', \quad p = p^* p', \quad \rho = \rho^* \rho', \quad \sigma = \sigma^* \sigma'$$

To conserve the scales and let ***invariant*** *the problem, the following constraints have to be respected:*

$$\frac{u^*}{v^*} \overset{1}{=} \frac{x^*}{y^*} \qquad \frac{u^*}{t^*} \overset{2}{=} \frac{u^{*2}}{x^*} \overset{3}{=} \frac{1}{x^*}\frac{p^*}{\rho^*} \qquad p^* \overset{4}{=} \frac{\sigma^*}{x^*} \qquad x^* \overset{5}{=} y^*$$

8 scaling factors
5 independent relations $\Rightarrow$ **3 *free param.*** $\Rightarrow$ (σ^*, ρ^*, t^*)

Simplified cartesian Navier-Stokes eqns:

$$\partial_x u = -\partial_y v$$
$$\partial_t u + u\,\partial_x u + v\,\partial_y u = -\partial_x (p/\rho)$$
$$\partial_t v + u\,\partial_x v + v\,\partial_y v = -\partial_y (p/\rho)$$

Initial Condition (free-surface shape):

$$f(x, y, 0) = y \cot\theta_0 - x$$

Kinematic Condition:

$$\partial_t f + u\partial_x f = 0$$
$$\partial_t f + v\partial_x f = 0$$

Dynamic Condition:

$$p(x, y, t)\Big|_{f=0} = \sigma\kappa$$

where κ is the *curvature.*

DECOMPOSITION

$$x = x^* x', \quad y = y^* y', \quad t = t^* t', \quad u = u^* u',$$
$$v = v^* v', \quad p = p^* p', \quad \rho = \rho^* \rho', \quad \sigma = \sigma^* \sigma'$$

*To conserve the scales and let **invariant** the problem, the following constraints have to be respected:*

$$\frac{u^*}{v^*} \overset{1}{=} \frac{x^*}{y^*} \qquad \frac{u^*}{t^*} \overset{2}{=} \frac{u^{*2}}{x^*} \overset{3}{=} \frac{1}{x^*}\frac{p^*}{\rho^*} \qquad p^* \overset{4}{=} \frac{\sigma^*}{x^*} \qquad x^* \overset{5}{=} y^*$$

8 scaling factors
5 independent relations $\Rightarrow$ **3 *free param.*** $\Rightarrow$ (σ^*, ρ^*, t^*)

Variables can be expressed according to this choice of parameters:

$$x^* = y^* = \left(\frac{\sigma^* t^{*2}}{\rho^*}\right)^{1/3} \qquad u^* = v^* = \left(\frac{\sigma^*}{\rho^* t^*}\right)^{1/3} \qquad p^* = \rho^* \left(\frac{\sigma^*}{\rho^* t^*}\right)^{2/3}$$

III.1 - Motivation for computing directly into the Self-Similar Space

- **Self-similar solution:** *stationary* solution in a particular rescaled space, thanks to the study of scale invariances throughout the physical problem

- Previous slides: only *rescaling* ⇒ all the needed scales have to be simulated

- Simulate directly into the self-similar space = all the scales have become $O(1)$
  ⇒ overcoming numerous numerical obstacles (*meshes*)

- To include easily **viscosity** effects:
  not done in papers **[5]**, **[6]**, **[7]** using *potential theory*

[5] Keller and Miksis, *Surface Tension Driven Flows*. SIAM **43** (1983)

[6] Day *et al.*, *Self-Similar Capillary Pinchoff of an Inviscid Fluid*. PRL **80** (1998)

[7] Sierou and Lister, *Self-similar recoil of inviscid drops*. Physics of Fluids **19** (2004)

- We search for the most general solution to the *N-S.* equations while using **scale invariant variables**, under the form $f\left(\tilde{X}, \tilde{Y}, \tau\right)$, where:

$$\tau = \ln t$$

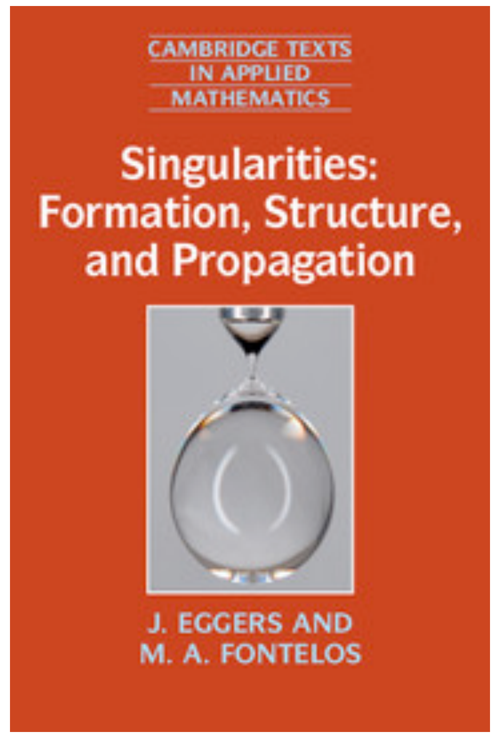
following **[8] (*logarithmic time*).**

III - Self-Similar Navier-Stokes solver with *Basilisk*

- We search for the most general solution to the *N-S.* equations while using **scale invariant variables**, under the form $f\left(\tilde{X}, \tilde{Y}, \tau\right)$, where:

$$\tau = \ln t$$

following **[8] (*logarithmic time*).**



[8] Eggers and Fontelos, *Singularities: Formation, Structure, and Propagation.* Cambridge Texts of Applied Mathematics (2015)

III - Self-Similar Navier-Stokes solver with *Basilisk*

III.2 - Strategy for finding S-I. Solutions

- We search for the most general solution to the *N-S.* equations while using **<u>scale invariant variables</u>**, under the form $f\left(\tilde{X}, \tilde{Y}, \tau\right)$, where:

$$\tau = \ln t$$

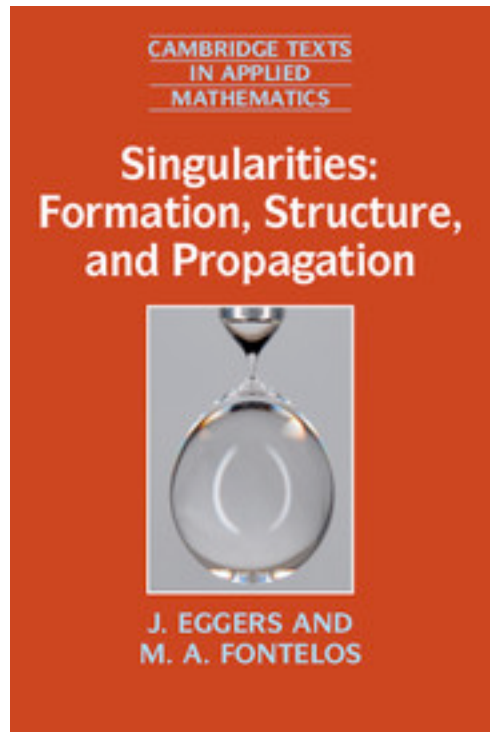
following **[8] (*logarithmic time*).**

- The introduction of the variable τ:

  $\Rightarrow$ gives an *evolution equation* **<u>in</u>** the self-similar space, solved with *Basilisk*;

  $\Rightarrow$ while the problem remains *unsteady*, the solution is <u>NOT</u> *scale invariant*;

  $\Rightarrow$ *Basilisk* is used to study **<u>how</u>** the numerical solution converges or not towards a *scale invariant solution*, *i.e.* a solution when $\partial_\tau f \to 0$ condition is met.



[8] Eggers and Fontelos, *Singularities: Formation, Structure, and Propagation.* <u>Cambridge Texts of Applied Mathematics</u> (2015)

Self-similar variables

$$\tilde{X} = \left(\frac{\rho}{\sigma t^2}\right)^{1/3} x \quad ; \quad \tilde{Y} = \left(\frac{\rho}{\sigma t^2}\right)^{1/3} y$$

$$\tilde{\mathbf{X}} := \begin{pmatrix} \tilde{X} \\ \tilde{Y} \end{pmatrix} \quad ; \quad \tau = \ln(t)$$

Self-similar functions

$$\mathbf{u}(x, y, t) = \left(\frac{\sigma}{\rho t}\right)^{1/3} \tilde{\mathbf{U}}\left[\tilde{X}(x, t), \tilde{Y}(y, t), \tau(t)\right]$$

$$p(x, y, t) = \rho \left(\frac{\sigma}{\rho t}\right)^{2/3} \tilde{P}\left[\tilde{X}(x, t), \tilde{Y}(y, t), \tau(t)\right]$$

Special Notations

- The substitution $\tilde{\nu} = \nu \left(\rho/\sigma\right)^{2/3}$
- The notation simplification $\nabla \equiv \nabla_{(\tilde{X}, \tilde{Y})}$
- The free surface $\tilde{F}$
- The curvature $\tilde{\kappa}$

Self-similar variables

$$\tilde{X} = \left(\frac{\rho}{\sigma t^2}\right)^{1/3} x \quad ; \quad \tilde{Y} = \left(\frac{\rho}{\sigma t^2}\right)^{1/3} y$$

$$\tilde{\mathbf{X}} := \begin{pmatrix} \tilde{X} \\ \tilde{Y} \end{pmatrix} \quad ; \quad \tau = \ln(t)$$

Self-similar functions

$$\mathbf{u}(x,y,t) = \left(\frac{\sigma}{\rho t}\right)^{1/3} \tilde{\mathbf{U}}\left[\tilde{X}(x,t), \tilde{Y}(y,t), \tau(t)\right]$$

$$p(x,y,t) = \rho \left(\frac{\sigma}{\rho t}\right)^{2/3} \tilde{P}\left[\tilde{X}(x,t), \tilde{Y}(y,t), \tau(t)\right]$$

Special Notations

- The substitution $\tilde{\nu} = \nu \left(\rho/\sigma\right)^{2/3}$
- The notation simplification $\nabla \equiv \nabla_{(\tilde{X},\tilde{Y})}$
- The free surface $\tilde{F}$
- The curvature $\tilde{\kappa}$

Self-similar N-S. Equations

$$\nabla \cdot \tilde{\mathbf{U}} = 0 \qquad \partial_\tau \tilde{\mathbf{U}} + \nabla \cdot \left(\tilde{\mathbf{U}} \otimes \tilde{\mathbf{U}}\right) = -\nabla \tilde{P} + \tilde{\nu}\left(\nabla^2 \tilde{\mathbf{U}}\right) e^{-\tau/3} \underbrace{-\tilde{\mathbf{U}}}_{\text{source term}} + \overbrace{\frac{2}{3} \nabla \cdot \left(\tilde{\mathbf{X}} \otimes \tilde{\mathbf{U}}\right)}^{\text{new advection term}}$$

Self-similar variables

$$\tilde{X} = \left(\frac{\rho}{\sigma t^2}\right)^{1/3} x \quad ; \quad \tilde{Y} = \left(\frac{\rho}{\sigma t^2}\right)^{1/3} y$$

$$\tilde{\mathbf{X}} := \begin{pmatrix}\tilde{X} \\ \tilde{Y}\end{pmatrix} \quad ; \quad \tau = \ln(t)$$

Self-similar functions

$$\mathbf{u}(x,y,t) = \left(\frac{\sigma}{\rho t}\right)^{1/3} \tilde{\mathbf{U}}\left[\tilde{X}(x,t), \tilde{Y}(y,t), \tau(t)\right]$$

$$p(x,y,t) = \rho \left(\frac{\sigma}{\rho t}\right)^{2/3} \tilde{P}\left[\tilde{X}(x,t), \tilde{Y}(y,t), \tau(t)\right]$$

Special Notations

- The substitution $\tilde{\nu} = \nu \left(\rho/\sigma\right)^{2/3}$
- The notation simplification $\nabla \equiv \nabla_{(\tilde{X},\tilde{Y})}$
- The free surface $\tilde{F}$
- The curvature $\tilde{\kappa}$

Self-similar N-S. Equations

$$\nabla \cdot \tilde{\mathbf{U}} = 0 \qquad \partial_\tau \tilde{\mathbf{U}} + \nabla \cdot \left(\tilde{\mathbf{U}} \otimes \tilde{\mathbf{U}}\right) = -\nabla \tilde{P} + \tilde{\nu}\left(\nabla^2 \tilde{\mathbf{U}}\right) e^{-\tau/3} \underbrace{-\tilde{\mathbf{U}}}_{\text{source term}} \overbrace{+\frac{2}{3}\nabla \cdot \left(\tilde{\mathbf{X}} \otimes \tilde{\mathbf{U}}\right)}^{\text{new advection term}}$$

Init. Cond. propagated

$$t \to 0 \Rightarrow \tilde{F}\left(\tilde{X}, \tilde{Y}\right) \xrightarrow[\|\tilde{\mathbf{x}}\| \to \infty]{} \tilde{X}\tan(\theta_0) - \tilde{Y}$$

Kinematic Condition

$$\tilde{F}_\tau + \left(\tilde{\mathbf{U}} \cdot \nabla\right)\tilde{F} - \frac{2}{3}\left(\tilde{\mathbf{X}} \cdot \nabla\right)\tilde{F} = 0$$

Far-field Condition

$$\tilde{\mathbf{U}} \xrightarrow[\|\tilde{\mathbf{x}}\| \to \infty]{} \mathbf{0}$$

Dynamic Condition

$$\tilde{P}(\tilde{X}, \tilde{Y}, \tau)\Big|_{\tilde{F}=0} = \tilde{\kappa}$$

III - Self-Similar Navier-Stokes solver with *Basilisk*

Two major additions:

1. A new *advection velocity* $\tilde{\mathbf{\Lambda}} := \tilde{\mathbf{U}} - (2/3)\,\tilde{\mathbf{X}} \neq \tilde{\mathbf{U}}$ the *advected velocity*
2. A *source term* $-\tilde{\mathbf{U}}$ in the RHS of the self-similar formulation of the N-S. equations

Libraries used:

```
centered.h -> bcg.h
two-phase.h -> vof.h
contact.h
tension.h
```

⟶ ***Modularity** of Basilisk can be exploited to tackle these changes!*

III - Self-Similar Navier-Stokes solver with *Basilisk*

Libraries used:

```
centered.h -> bcg.h
two-phase.h -> vof.h
contact.h
tension.h
```

⟶ ***Modularity** of Basilisk can be exploited to tackle these changes!*

Two major additions:

1. A new *advection velocity* $\tilde{\mathbf{\Lambda}} := \tilde{\mathbf{U}} - (2/3)\,\tilde{\mathbf{X}} \neq \tilde{\mathbf{U}}$ the *advected velocity*
2. A *source term* $-\tilde{\mathbf{U}}$ in the RHS of the self-similar formulation of the N-S. equations

In `centered.h`

Two major additions:

1. A new *advection velocity* $\tilde{\mathbf{\Lambda}} := \tilde{\mathbf{U}} - (2/3)\,\tilde{\mathbf{X}} \neq \tilde{\mathbf{U}}$ the *advected velocity*
2. A *source term* $-\tilde{\mathbf{U}}$ in the RHS of the self-similar formulation of the N-S. equations

Libraries used:

```
centered.h -> bcg.h
two-phase.h -> vof.h
contact.h
tension.h
```

⟶ ***Modularity** of Basilisk can be exploited to tackle these changes!*

In `centered.h`

- Initialization of a position vector `xi[]` and a face advection velocity vector `lambdaf[]` in **`event init`**.

III - Self-Similar Navier-Stokes solver with *Basilisk*

Libraries used:

`centered.h -> bcg.h`
`two-phase.h -> vof.h`
`contact.h`
`tension.h`

⟶ ***Modularity** of Basilisk can be exploited to tackle these changes!*

Two major additions:

1. A new *advection velocity* $\tilde{\Lambda} := \tilde{U} - (2/3)\,\tilde{X} \neq \tilde{U}$ the *advected velocity*
2. A *source term* $-\tilde{U}$ in the RHS of the self-similar formulation of the N-S. equations

In `centered.h`

- Initialization of a position vector `xi[]` and a face advection velocity vector `lambdaf[]` in **`event init`**.
- Set the correct velocity for the CFL with `lambdaf[]` in **`event stability`**.

Libraries used:

```
centered.h -> bcg.h
two-phase.h -> vof.h
contact.h
tension.h
```

⟶ ***Modularity*** *of Basilisk can be exploited to tackle these changes!*

Two major additions:

1. A new *advection velocity* $\tilde{\mathbf{\Lambda}} := \tilde{\mathbf{U}} - (2/3)\,\tilde{\mathbf{X}} \neq \tilde{\mathbf{U}}$ the *advected velocity*
2. A *source term* $-\tilde{\mathbf{U}}$ in the RHS of the self-similar formulation of the N-S. equations

In `centered.h`

- Initialization of a position vector `xi[]` and a face advection velocity vector `lambdaf[]` in **`event init`**.
- Set the correct velocity for the CFL with `lambdaf[]` in **`event stability`**.
- The face advected velocity $\tilde{\mathbf{U}}_f$ is predicted with the face advection velocity $\tilde{\mathbf{\Lambda}}_f$ in the **`prediction()`** function.

III - Self-Similar Navier-Stokes solver with *Basilisk*

Two major additions:

1. A new *advection velocity* $\tilde{\boldsymbol{\Lambda}} := \tilde{\mathbf{U}} - (2/3)\,\tilde{\mathbf{X}}$ $\neq \tilde{\mathbf{U}}$ the *advected velocity*
2. A *source term* $-\tilde{\mathbf{U}}$ in the RHS of the self-similar formulation of the N-S. equations

Libraries used:

```
centered.h -> bcg.h
two-phase.h -> vof.h
contact.h
tension.h
```

⟶ ***Modularity*** *of Basilisk can be exploited to tackle these changes!*

In `centered.h`

- Initialization of a position vector `xi[]` and a face advection velocity vector `lambdaf[]` in **`event init`**.
- Set the correct velocity for the CFL with `lambdaf[]` in **`event stability`**.
- The face advected velocity $\tilde{\mathbf{U}}_f$ is predicted with the face advection velocity $\tilde{\boldsymbol{\Lambda}}_f$ in the **`prediction()`** function.
- The predicted face advected velocity $\tilde{\mathbf{U}}_f$ is then projected to be *divergence* free and used for updating the face advection velocity $\tilde{\boldsymbol{\Lambda}}_f$ needed for the *BCG advection scheme* (BCG is **NOT** modified!) in the **`event advection_term`**:

```
event advection_term (i++,last)
{
  if (!stokes) {
    [...]
    advection ((scalar *){u}, lambdaf, dt, (scalar *){g});
  }
}
```

III - Self-Similar Navier-Stokes solver with *Basilisk*

Libraries used:

centered.h –> bcg.h
two-phase.h –> vof.h
contact.h
tension.h

⟶ ***Modularity** of Basilisk can be exploited to tackle these changes!*

Two major additions:

1. A new *advection velocity* $\tilde{\mathbf{\Lambda}} := \tilde{\mathbf{U}} - (2/3)\,\tilde{\mathbf{X}} \neq \tilde{\mathbf{U}}$ the *advected velocity*
2. A *source term* $-\tilde{\mathbf{U}}$ in the RHS of the self-similar formulation of the N-S. equations

In `centered.h`

- Initialization of a position vector `xi[]` and a face advection velocity vector `lambdaf[]` in **`event init`**.
- Set the correct velocity for the CFL with `lambdaf[]` in **`event stability`**.
- The face advected velocity $\tilde{\mathbf{U}}_f$ is predicted with the face advection velocity $\tilde{\mathbf{\Lambda}}_f$ in the **`prediction()`** function.
- The predicted face advected velocity $\tilde{\mathbf{U}}_f$ is then projected to be *divergence* free and used for updating the face advection velocity $\tilde{\mathbf{\Lambda}}_f$ needed for the *BCG advection scheme* (BCG is **NOT** modified!) in the **`event advection_term`**:

```
event advection_term (i++,last)
{
  if (!stokes) {
    [...]
    advection ((scalar *){u}, lambdaf, dt, (scalar *){g});
  }
}
```

In `vof.h`

We call the face vector `lambdaf[]` instead of `uf[]`
and replace each occurrence of the last one by the first one

III - Self-Similar Navier-Stokes solver with *Basilisk*

Libraries used:

`centered.h -> bcg.h`
`two-phase.h -> vof.h`
`contact.h`
`tension.h`

⟶ ***Modularity** of Basilisk can be exploited to tackle these changes!*

Two major additions:

1. A new *advection velocity* $\tilde{\mathbf{\Lambda}} := \tilde{\mathbf{U}} - (2/3)\,\tilde{\mathbf{X}} \neq \tilde{\mathbf{U}}$ the *advected velocity*
2. A *source term* $-\tilde{\mathbf{U}}$ in the RHS of the self-similar formulation of the N-S. equations

In `centered.h`

- Initialization of a position vector `xi[]` and a face advection velocity vector `lambdaf[]` in **`event init`**.
- Set the correct velocity for the CFL with `lambdaf[]` in **`event stability`**.
- The face advected velocity $\tilde{\mathbf{U}}_f$ is predicted with the face advection velocity $\tilde{\mathbf{\Lambda}}_f$ in the **`prediction()`** function.
- The predicted face advected velocity $\tilde{\mathbf{U}}_f$ is then projected to be *divergence* free and used for updating the face advection velocity $\tilde{\mathbf{\Lambda}}_f$ needed for the *BCG advection scheme* (BCG is **NOT** modified!) in the **`event advection_term`**:

```
event advection_term (i++,last)
{
  if (!stokes) {
    [...]
    advection ((scalar *){u}, lambdaf, dt, (scalar *){g});
  }
}
```

In `vof.h`

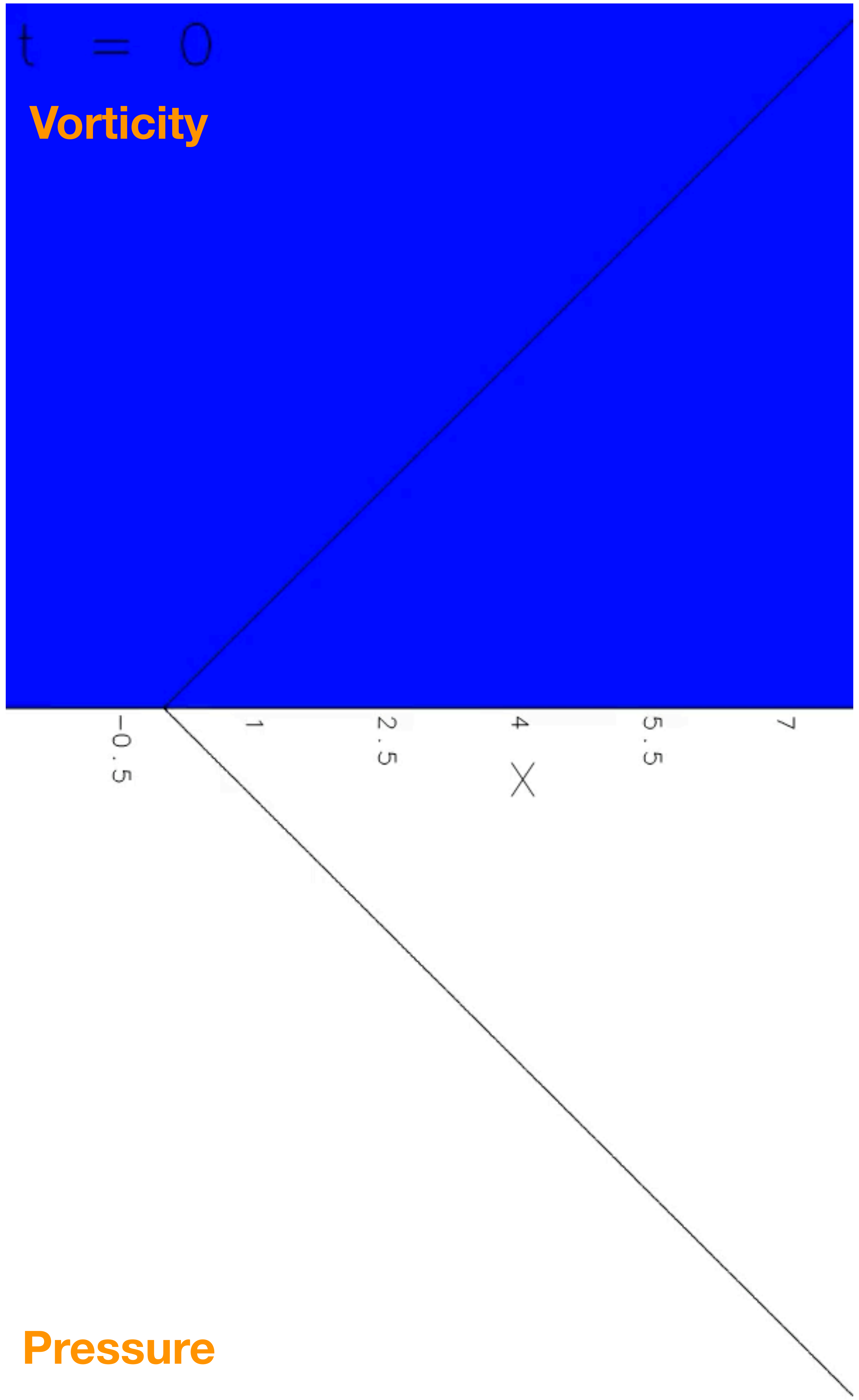
We call the face vector `lambdaf[]` instead of `uf[]`
and replace each occurrence of the last one by the first one

In `user-file.c`

The *source term* $-\tilde{\mathbf{U}}$ in the RHS of the self-similar formulation of the N-S. equations is simply an update of the **`event acceleration`**

$N = 8$
$N_{max} = 9$
$L_0 = 12$
$\theta_0 = 45°$
$\beta_0 = 90°$
$\rho_{fluid} = 1$
$\rho_{gas} = 10^{-3}$
$\mu_{fluid} = 10^{-3}$
$\mu_{gas} = 10^{-5}$
$\sigma = 1$

Global Parameters



III - Self-Sim. N.-S.solver with *Basilisk*

$N = 8$

$N_{max} = 9$

$L_0 = 12$

$\theta_0 = 45°$

$\beta_0 = 90°$

$\rho_{fluid} = 1$

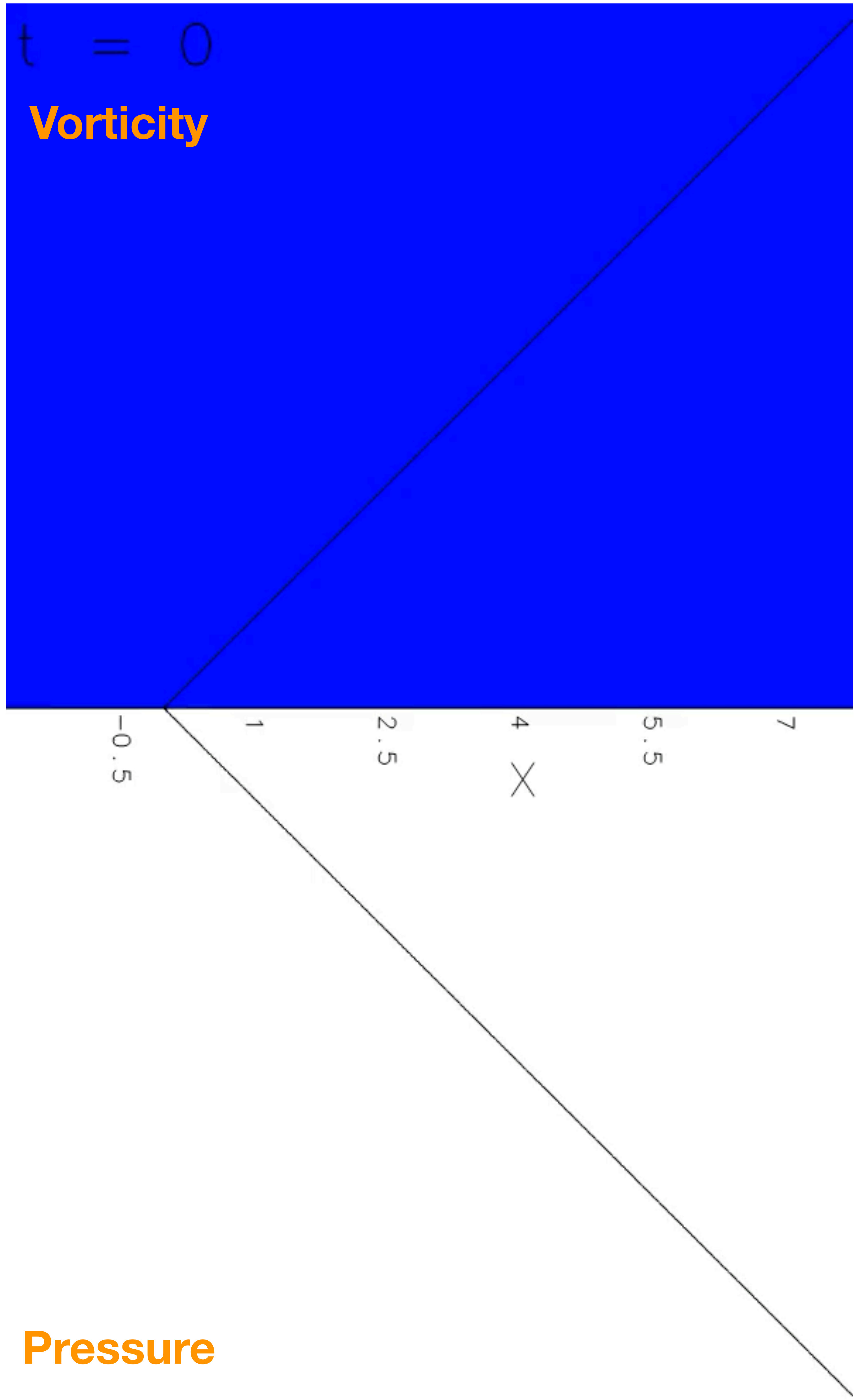
$\rho_{gas} = 10^{-3}$

$\mu_{fluid} = 10^{-3}$

$\mu_{gas} = 10^{-5}$

$\sigma = 1$

Global Parameters



$N = 8$

$N_{max} = 9$

$L_0 = 12$

$\theta_0 = 45°$

$\beta_0 = 90°$

$\rho_{fluid} = 1$

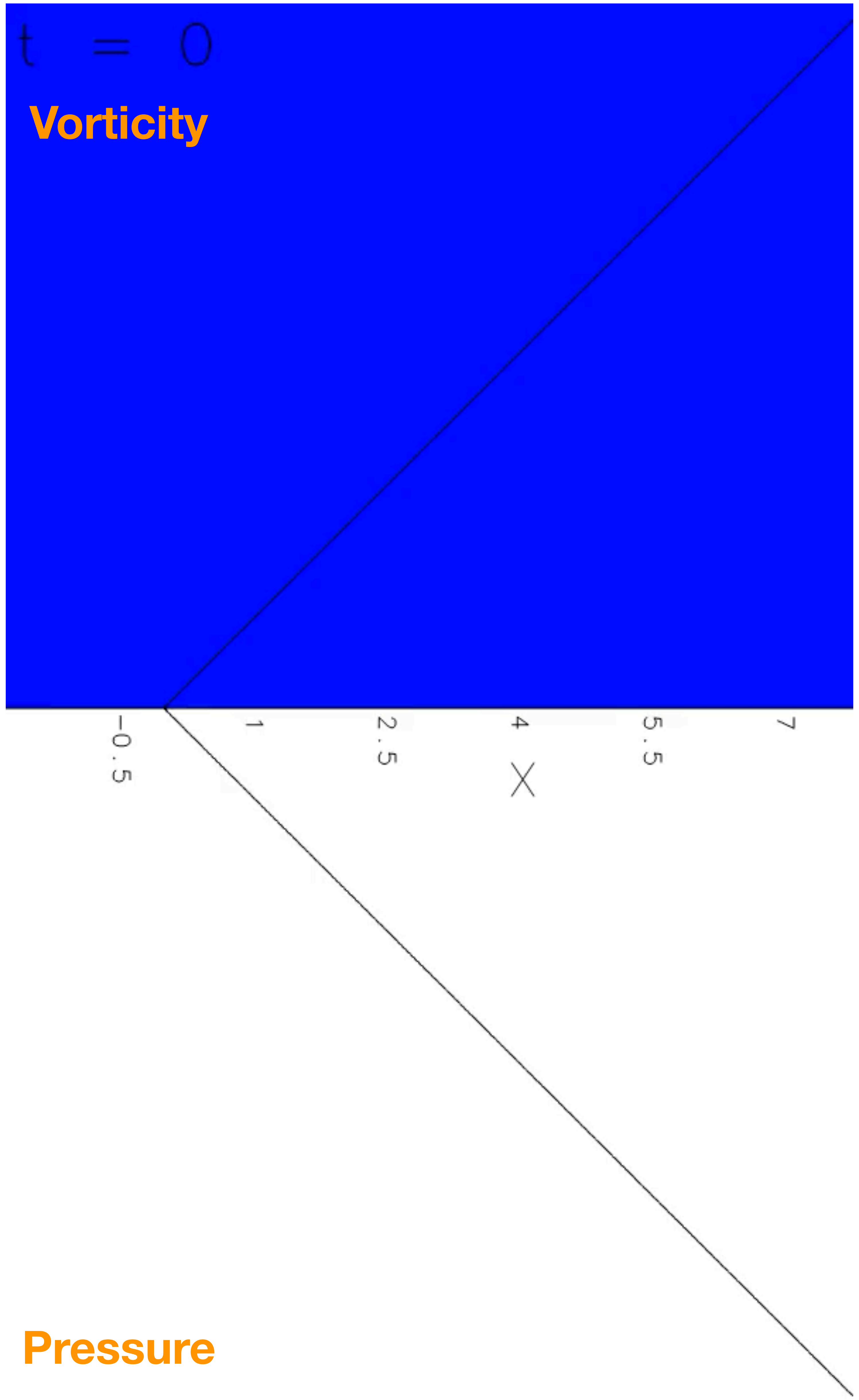
$\rho_{gas} = 10^{-3}$

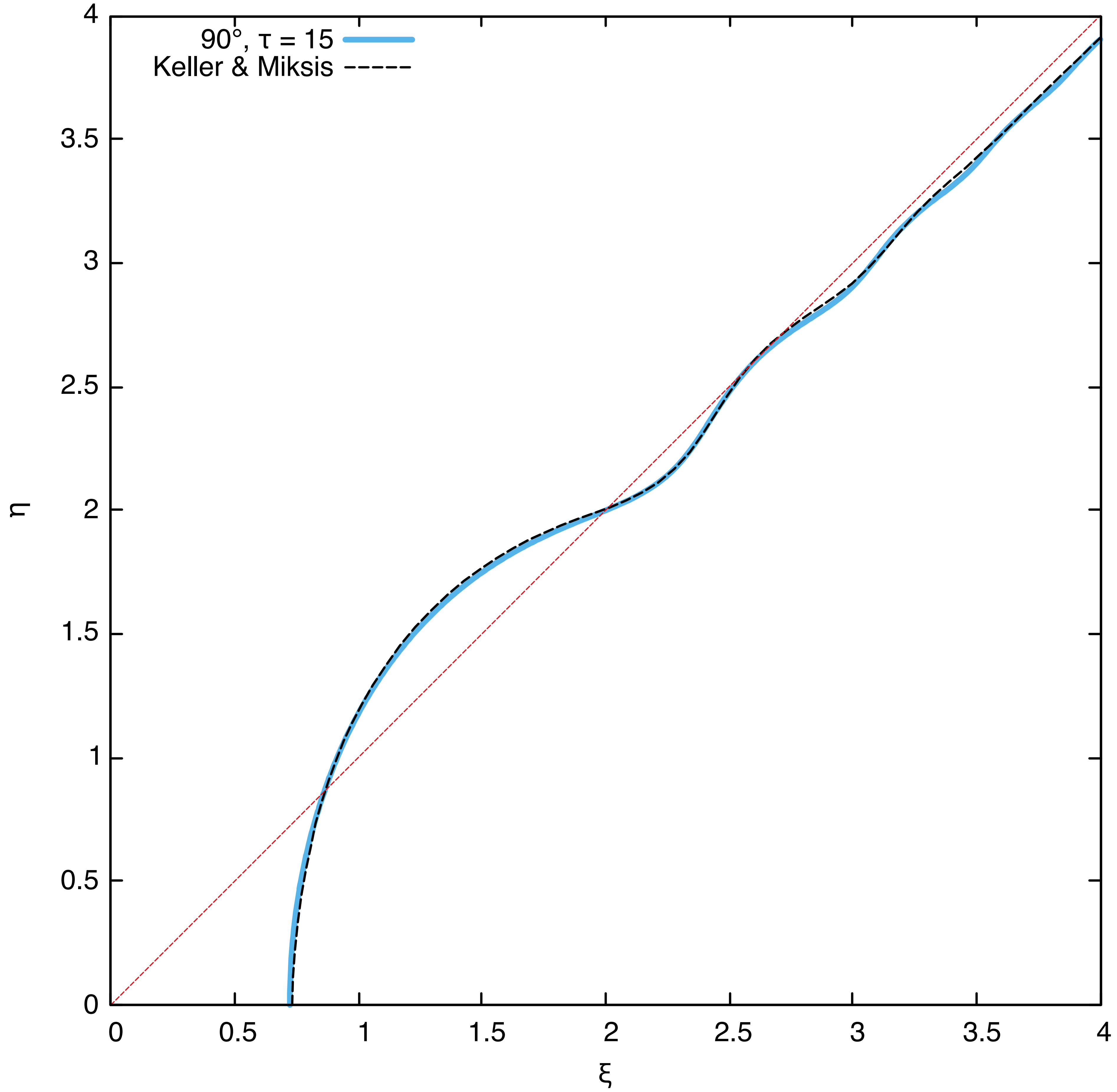
$\mu_{fluid} = 10^{-3}$

$\mu_{gas} = 10^{-5}$

$\sigma = 1$

Global Parameters





Time-profile showing the achieved convergence towards the self-similar solution

V - Conclusions & Perspectives

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

- Moving into self-similar space $\Rightarrow$ we recover the *Navier-Stokes* equations, with two additional terms: a ***new advection velocity***, and a ***source term***.

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

- Moving into self-similar space $\Rightarrow$ we recover the *Navier-Stokes* equations, with two additional terms: a ***new advection velocity***, and a ***source term***.

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

- Moving into self-similar space $\Rightarrow$ we recover the *Navier-Stokes* equations, with two additional terms: a ***new advection velocity***, and a ***source term***.

- Use of a *Navier-Stokes solver* allows us to "activate" the **viscosity** as we wish = better approximation to physical reality than the methods used up to now†.

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

- Moving into self-similar space $\Rightarrow$ we recover the *Navier-Stokes* equations, with two additional terms: a ***new advection velocity***, and a ***source term***.

- Use of a *Navier-Stokes solver* allows us to "activate" the **viscosity** as we wish = better approximation to physical reality than the methods used up to now†.

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

- Moving into self-similar space $\Rightarrow$ we recover the *Navier-Stokes* equations, with two additional terms: a ***new advection velocity***, and a ***source term***.

- Use of a *Navier-Stokes solver* allows us to "activate" the **viscosity** as we wish = better approximation to physical reality than the methods used up to now†.

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

- Moving into self-similar space $\Rightarrow$ we recover the *Navier-Stokes* equations, with two additional terms: a ***new advection velocity***, and a ***source term***.

- Use of a *Navier-Stokes solver* allows us to "activate" the **viscosity** as we wish = better approximation to physical reality than the methods used up to now†.

†*boundary integrals in potential theory with the assumption of an inviscid fluid*.

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

- Moving into self-similar space $\Rightarrow$ we recover the *Navier-Stokes* equations, with two additional terms: a ***new advection velocity***, and a ***source term***.

- Use of a *Navier-Stokes solver* allows us to "activate" the **viscosity** as we wish = better approximation to physical reality than the methods used up to now†.

- Extend the study to the **axisymmetric case**: more complex due to far-field boundary conditions with corrective terms *(asymptotic expansions)*.

†*boundary integrals in potential theory with the assumption of an inviscid fluid.*

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

- Moving into self-similar space $\Rightarrow$ we recover the *Navier-Stokes* equations, with two additional terms: a ***new advection velocity***, and a ***source term***.

- Use of a *Navier-Stokes solver* allows us to "activate" the **viscosity** as we wish = better approximation to physical reality than the methods used up to now†.

†*boundary integrals in potential theory with the assumption of an inviscid fluid*.

- Extend the study to the **axisymmetric case**: more complex due to far-field boundary conditions with corrective terms *(asymptotic expansions)*.

V - Conclusions & Perspectives

- **Self-similar framework** ⟶ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

- Moving into self-similar space ⇒ we recover the *Navier-Stokes* equations, with two additional terms: a ***new advection velocity***, and a ***source term***.

- Use of a *Navier-Stokes solver* allows us to "activate" the **viscosity** as we wish = better approximation to physical reality than the methods used up to now†.

† *boundary integrals in potential theory with the assumption of an inviscid fluid.*

- Extend the study to the **axisymmetric case**: more complex due to far-field boundary conditions with corrective terms *(asymptotic expansions)*.

- Study **finite-time singularities**:
  - observe the formation of *liquid jets*;
  - understand selection mechanisms by direct *perturbation* in the self-similar space of these fields.

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.
- Moving into self-similar space $\Rightarrow$ we recover the *Navier-Stokes* equations, with two additional terms: a ***new advection velocity***, and a ***source term***.
- Use of a *Navier-Stokes solver* allows us to "activate" the **viscosity** as we wish = better approximation to physical reality than the methods used up to now†.

† *boundary integrals in potential theory with the assumption of an inviscid fluid.*

- Extend the study to the **axisymmetric case**: more complex due to far-field boundary conditions with corrective terms *(asymptotic expansions)*.
- Study **finite-time singularities**:
  - observe the formation of *liquid jets*;
  - understand selection mechanisms by direct *perturbation* in the self-similar space of these fields.

V - Conclusions & Perspectives

- **Self-similar framework** $\longrightarrow$ right scale for following the dynamics of the problem + resolving certain numerical obstacles.

- Moving into self-similar space $\Rightarrow$ we recover the *Navier-Stokes* equations, with two additional terms: a ***new advection velocity***, and a ***source term***.

- Use of a *Navier-Stokes solver* allows us to "activate" the **viscosity** as we wish = better approximation to physical reality than the methods used up to now†.

†*boundary integrals in potential theory with the assumption of an inviscid fluid*.

- Extend the study to the **axisymmetric case**: more complex due to far-field boundary conditions with corrective terms *(asymptotic expansions)*.

- Study **finite-time singularities**:
  - observe the formation of *liquid jets*;
  - understand selection mechanisms by direct *perturbation* in the self-similar space of these fields.