SHAKE THE FUTURE.

CENTRALE NANTES

CSI Report, 1st year.

Development of a Finite Volume scheme on an adaptively refined Cartesian grid in the presence of air/water interfaces.

Tommaso Zanelli

17/06/2020

Grid-flow solver

- Formerly WCCH (Weakly-compressible cartesian hydrodynamics)
- Developed by Nextflow Software and LHEEA of École Centrale de Nantes
- High order finite volume discretization
- Adaptively refined multi-level Cartesian grid
- Immersed boundary method
- Written in Fortran 90
- MPI (Message Passing Interface) for parallelization

Grid-flow solver

Weakly-compressible formulation:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0 \\ \dfrac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \boldsymbol{f} \end{cases}$$

$$p = \frac{\rho_0\, c_0^2}{\gamma} \left[\left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right]$$

Incompressible formulation:

$$\begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) = -\dfrac{\nabla p}{\rho} + \dfrac{1}{\rho} \nabla \cdot \boldsymbol{\tau} + \boldsymbol{f} \\ \nabla \cdot \boldsymbol{u} = 0 \end{cases}$$

Grid-flow solver

Weakly-compressible formulation:

- Fully explicit
- Optimal scalability
- Small Δt required for stability
- Compressibility effects are not of interest

Incompressible formulation:

- Semi-implicit
- Projection scheme requires solution of a linear system (PETSc)
- Less stringent stability requirements on Δt

Grid-flow solver
Projection scheme

Chorin-Temam projection scheme (Classic[1][2]):

1. Prediction step: $$\boldsymbol{u}^* = \boldsymbol{u}^n + \Delta t \boldsymbol{RHS}^n$$

2. Pressure Poisson equation: $$\nabla^2 p^{n+1} = \frac{\rho}{\Delta t} \nabla \cdot \boldsymbol{u}^*$$

3. Velocity correction step: $$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \frac{\Delta t}{\rho} \nabla p^{n+1}$$

[1]: Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. Mathematics of computation, 22(104):745–762, 1968.
[2]: Roger Temam. Sur l'approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires (ii). Archive for rational mechanics and analysis, 33(5):377–385, 1969.

Grid-flow solver
Projection scheme

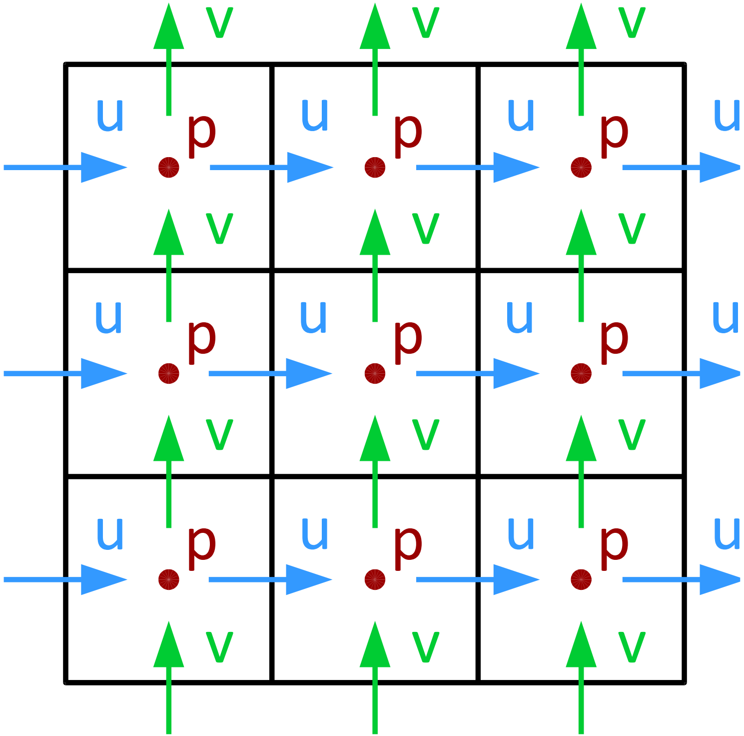
Chorin-Temam projection scheme (Incremental[3]):

1. Prediction step: $$\boldsymbol{u}^* = \boldsymbol{u}^n - \frac{\Delta t}{\rho}\nabla p^n + \Delta t \boldsymbol{RHS}^n$$
2. Pressure correction Poisson equation: $$\nabla^2 \phi = \frac{\rho}{\Delta t}\nabla \cdot \boldsymbol{u}^*$$
3. Velocity correction step: $$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \frac{\Delta t}{\rho}\nabla \phi$$
4. Pressure correction step: $$p^{n+1} = p^n + \phi$$

[3]: Jean-Luc Guermond, Peter Minev, and Jie Shen. An overview of projection methods for incompressible flows. Computer methods in applied mechanics and engineering, 195(44-47):6011–6045, 2006.

Grid-flow solver

Staggered grid[4]



[4]: Francis H Harlow and J Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. The physics of fluids, 8(12):2182–2189, 1965.

Grid-flow solver

Time integration

Set of Conservation Equations:

$$\frac{\partial \boldsymbol{W}}{\partial t} + \nabla \cdot \boldsymbol{\Psi}\left(\boldsymbol{W}\right) = \boldsymbol{S}$$

Time integration:

- Forward Euler

$$\frac{\boldsymbol{W}^{n+1} - \boldsymbol{W}^{n}}{\Delta t} + \nabla \cdot \boldsymbol{\Psi}\left(\boldsymbol{W}^{n}\right) = \boldsymbol{S}^{n}$$

- Fourth-order Runge-Kutta

$$\begin{aligned}
\boldsymbol{k}_1 &= -\nabla \cdot \boldsymbol{\Psi}(\boldsymbol{W}^n) + \boldsymbol{S}\left(t^n\right) \\
\boldsymbol{k}_2 &= -\nabla \cdot \boldsymbol{\Psi}\left(\boldsymbol{W}^n + \frac{1}{2}\Delta t\, \boldsymbol{k}_1\right) + \boldsymbol{S}\left(t^{n+\frac{1}{2}}\right) \\
\boldsymbol{k}_3 &= -\nabla \cdot \boldsymbol{\Psi}\left(\boldsymbol{W}^n + \frac{1}{2}\Delta t\, \boldsymbol{k}_2\right) + \boldsymbol{S}\left(t^{n+\frac{1}{2}}\right) \\
\boldsymbol{k}_4 &= -\nabla \cdot \boldsymbol{\Psi}(\boldsymbol{W}^n + \Delta t\, \boldsymbol{k}_3) + \boldsymbol{S}^n\left(t^{n+1}\right) \\
\boldsymbol{W}^{n+1} &= \boldsymbol{W}^n \quad + \Delta t\left(\frac{1}{6}\boldsymbol{k}_1 + \frac{1}{3}\boldsymbol{k}_2 + \frac{1}{3}\boldsymbol{k}_3 + \frac{1}{6}\boldsymbol{k}_4\right)
\end{aligned}$$

Grid-flow solver

Finite Volume formulation

Set of Conservation Equations:

$$\frac{\partial \boldsymbol{W}}{\partial t} + \nabla \cdot \boldsymbol{\Psi}\left(\boldsymbol{W}\right) = \boldsymbol{S}$$

Integration over control volume:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_i} \boldsymbol{W}\,\mathrm{d}\Omega - \int_{\partial\Omega_i} \boldsymbol{W}\left(\boldsymbol{u}_{\Omega_i}\cdot\boldsymbol{n}\right)\mathrm{d}S + \int_{\partial\Omega_i} \boldsymbol{\Psi}\left(\boldsymbol{W}\right)\cdot\boldsymbol{n}\mathrm{d}S = \int_{\partial\Omega_i} \boldsymbol{S}\mathrm{d}\Omega$$

Volume averages:

$$\overline{\boldsymbol{W}}_i = \frac{1}{\Omega_i}\int_{\Omega_i} \boldsymbol{W}\,\mathrm{d}\Omega\,; \qquad \overline{\boldsymbol{S}}_i = \frac{1}{\Omega_i}\int_{\Omega_i} \boldsymbol{S}\,\mathrm{d}\Omega\,;$$

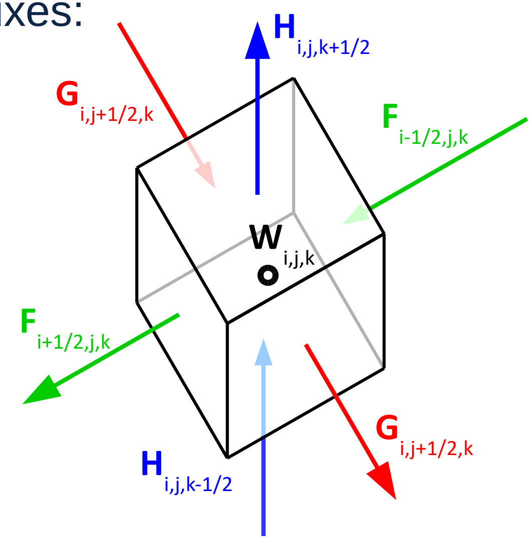
$$\int_{\partial\Omega_i} \boldsymbol{\Psi}\left(\boldsymbol{W}\right)\cdot\boldsymbol{n}\mathrm{d}S = \sum_{j=1}^{M}\int_{A_{ij}} \boldsymbol{\Psi}\left(\boldsymbol{W}\right)\cdot\boldsymbol{n}\mathrm{d}S = \sum_{j=1}^{M} \boldsymbol{F}_{ij}\,A_{ij}$$

Grid-flow solver
Finite Volume formulation

Set of Conservation Equations:

$$\frac{\partial \boldsymbol{W}}{\partial t} + \nabla \cdot \boldsymbol{\Psi}\left(\boldsymbol{W}\right) = \boldsymbol{S}$$

Control volume (3D) and fluxes:



Grid-flow solver
Finite Volume formulation

Set of Conservation Equations:

$$\frac{\partial \boldsymbol{W}}{\partial t} + \nabla \cdot \boldsymbol{\Psi}\left(\boldsymbol{W}\right) = \boldsymbol{S}$$

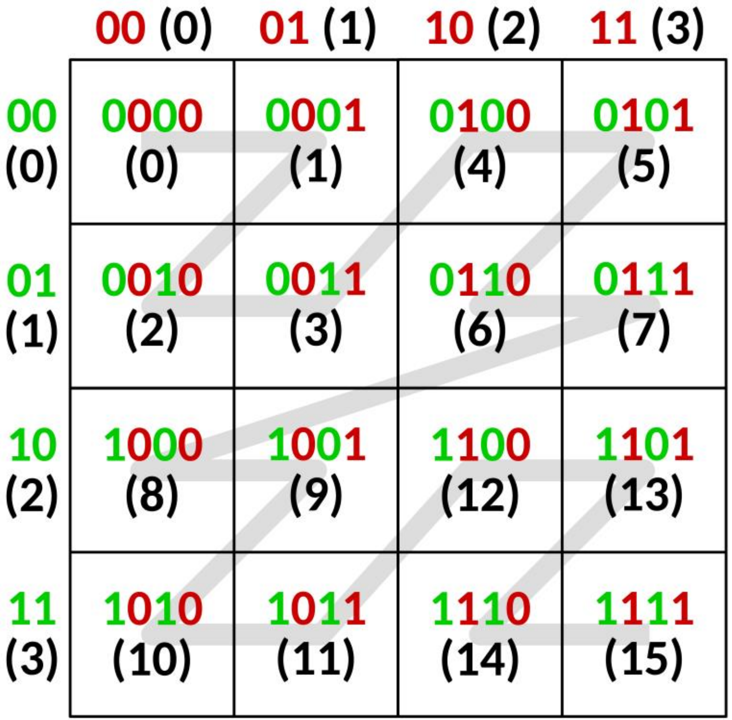
Discrete formulation (3D Cartesian grid):

$$\begin{aligned}\frac{\mathrm{d}\overline{\boldsymbol{W}}_{i,j,k}}{\mathrm{d}t} = &-\frac{\cancel{\Delta y \Delta z}}{\Delta x\, \cancel{\Delta y \Delta z}}\left(\boldsymbol{F}_{i+\frac{1}{2},j,k} - \boldsymbol{F}_{i-\frac{1}{2},j,k}\right) - \\ &-\frac{\cancel{\Delta z \Delta x}}{\Delta y\, \cancel{\Delta z \Delta x}}\left(\boldsymbol{G}_{i,j+\frac{1}{2},k} - \boldsymbol{G}_{i,j-\frac{1}{2},k}\right) - \\ &-\frac{\cancel{\Delta x \Delta y}}{\Delta z\, \cancel{\Delta x \Delta y}}\left(\boldsymbol{H}_{i,j,k+\frac{1}{2}} - \boldsymbol{H}_{i,j,k-\frac{1}{2}}\right) + \overline{\boldsymbol{S}}_{i,j,k}\end{aligned}$$

Grid-flow solver

Adaptive Mesh Refinement (AMR)

Z-Order curve:



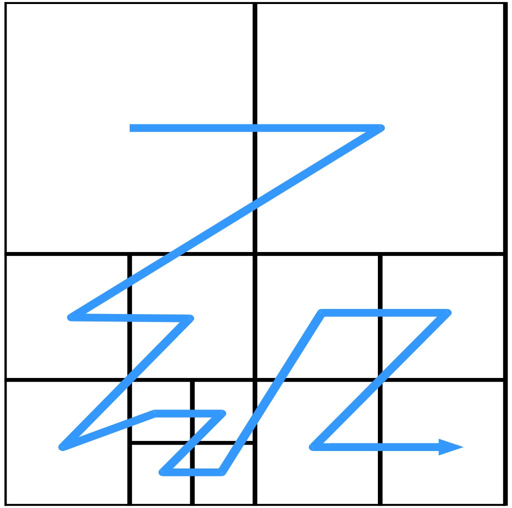
SHAKE THE FUTURE.

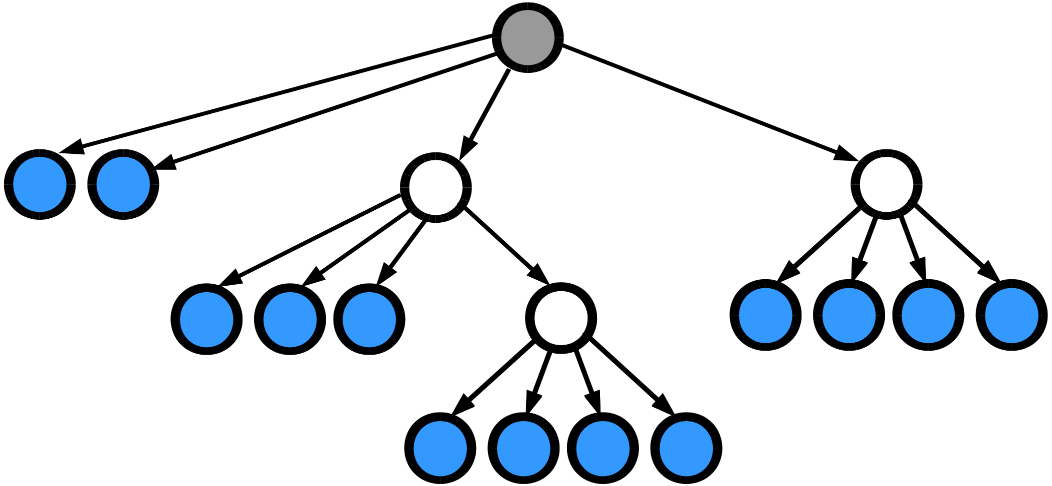
CENTRALE NANTES

Grid-flow solver

Adaptive Mesh Refinement (AMR)

Multi-level grid and quadtree:

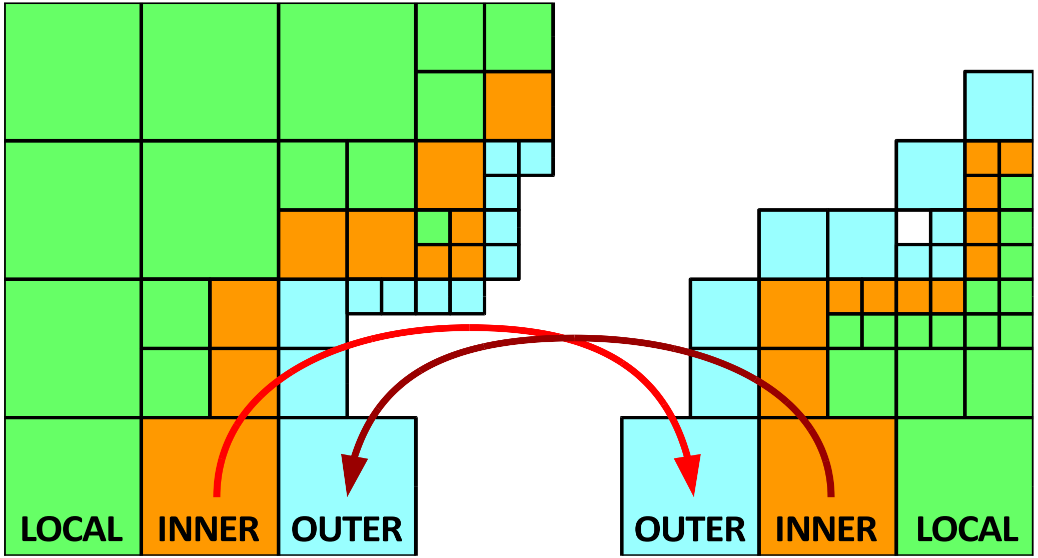




Grid-flow solver
Adaptive Mesh Refinement (AMR)

- Domain partitioned *recursively* into blocks, starting from a single one.
- Each block can have four (2D) or eight (3D) *children* blocks.
- Partitioning is continued until the desired level of refinement is reached.
- Adjacent blocks must differ by at most one level of refinement.
- Blocks are further divided into a fixed number of cells.

Grid-flow solver
Parallelization



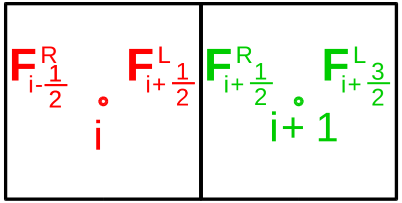
Grid-flow solver
Adaptive Mesh Refinement (AMR)

- An operation is performed on the inner blocks first.
- The updated values at the inner blocks are sent to the neighbour processes.
- The operation is performed on the local blocks
- The updated inner block values from the neighbour processes are received and used to fill the outer blocks.

Grid-flow solver

Numerical fluxes

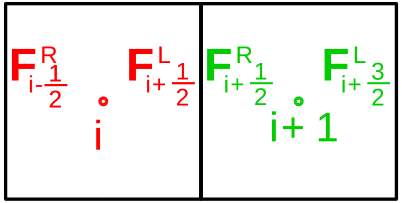
At each interface two reconstructions are performed:



In the incompressible formulation, the flux in each direction is chosen as

$$\begin{cases} \boldsymbol{F}_{i+\frac{1}{2},j,k} = \boldsymbol{F}^L_{i+\frac{1}{2},j,k} & \text{if } u_{i+\frac{1}{2},j,k} > 0 \\ \boldsymbol{F}_{i+\frac{1}{2},j,k} = \dfrac{1}{2}\left(\boldsymbol{F}^L_{i+\frac{1}{2},j,k} + \boldsymbol{F}^R_{i+\frac{1}{2},j,k}\right) & \text{if } u_{i+\frac{1}{2},j,k} = 0 \\ \boldsymbol{F}_{i+\frac{1}{2},j,k} = \boldsymbol{F}^R_{i+\frac{1}{2},j,k} & \text{if } u_{i+\frac{1}{2},j,k} < 0 \end{cases}$$

Grid-flow solver
Numerical fluxes – first order reconstruction



This consists in assigning the left and right fluxes as the values at the centre of the left and right cells respectively:

$$\boldsymbol{F}^L_{i+\frac{1}{2}} = \boldsymbol{\Psi}_i$$

$$\boldsymbol{F}^R_{i+\frac{1}{2}} = \boldsymbol{\Psi}_{i+1}$$

Grid-flow solver

Numerical fluxes – MUSCL[5] reconstruction

A quantity u is reconstructed as:

$$\begin{cases} u^R_{i-\frac{1}{2}} = u_i - \frac{1}{2}\Theta\left(\kappa_i\right)\left(u_{i+1} - u_i\right) \\ u^L_{i+\frac{1}{2}} = u_i + \frac{1}{2}\Theta\left(\kappa_i\right)\left(u_{i+1} - u_i\right) \end{cases}$$

Where $\Theta\left(\kappa\right)$ is called *limiter* and $\kappa_i = \dfrac{u_i - u_{i-1}}{u_{i+1} - u_i}$

One example is the *Minmod* limiter:

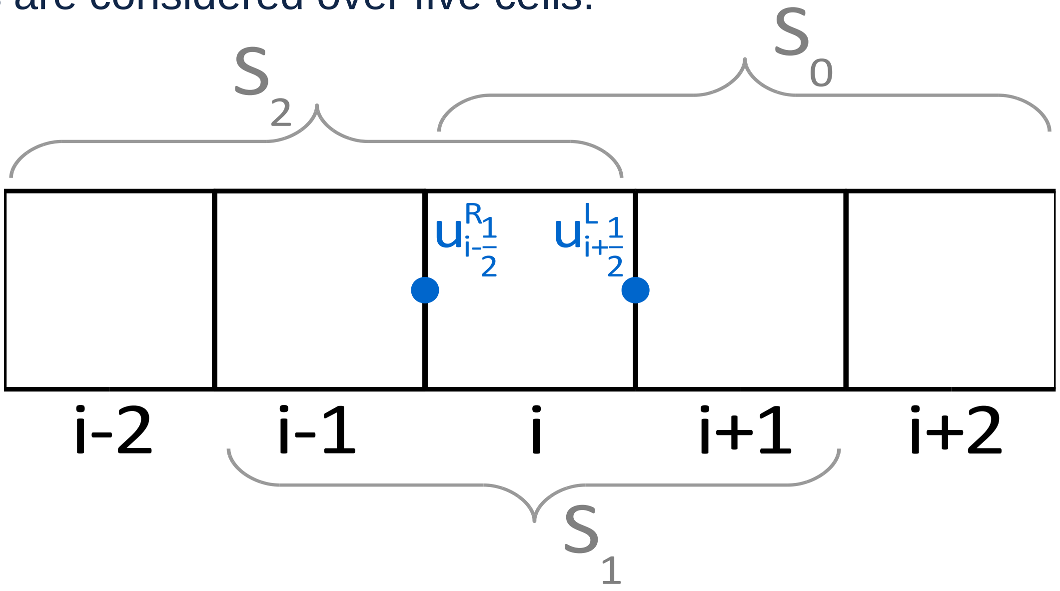
$$\Theta\left(\kappa\right) = \max\left[0,\ \min\left(1,\ \kappa\right)\right]$$

[5]: Pierre Bigay. Développement d’un solveur faiblement compressible sur maillage Cartésien pour les écoulements hydrodynamiques autour de corps. PhD thesis, 2015.

Grid-flow solver

Numerical fluxes – WENO reconstruction[6]

Three stencils are considered over five cells:



[6]: Louis Vittoz. Contributions au développement d’un solveur volumes finis sur grille cartésienne localement raffinée en vue d’application à l’hydrodynamique navale. PhD thesis, 2018.

Grid-flow solver

Numerical fluxes – WENO reconstruction

Smoothness indicators are defined over the stencils:

$$\begin{cases} \mathrm{IS}_0 = \frac{13}{12}\left(u_{i+2} - 2u_{i+1} + u_i\right)^2 + \frac{1}{4}\left(u_{i+2} - 4u_{i+1} + 3u_i\right)^2 \\ \mathrm{IS}_1 = \frac{13}{12}\left(u_{i+1} - 2u_i + u_{i-1}\right)^2 + \frac{1}{4}\left(u_{i+1} - u_{i-1}\right)^2 \\ \mathrm{IS}_2 = \frac{13}{12}\left(u_i - 2u_{i-1} + u_{i-2}\right)^2 + \frac{1}{4}\left(3u_i - 4u_{i-1} + u_{i-2}\right)^2 \end{cases}$$

And used to compute the weights:

$$\alpha_i = \frac{d_i}{\mathrm{IS}_i + \varepsilon} \qquad \omega_i = \frac{\alpha_i}{\sum_{j=1}^{3} \alpha_j} \qquad i = 0,\ 1,\ 2$$

With:

$$d_0^L = \frac{3}{10},\ d_1^L = \frac{3}{5},\ d_2^L = \frac{1}{10}, \quad d_0^R = \frac{1}{10},\ d_1^R = \frac{3}{5},\ d_2^R = \frac{3}{10}$$

Grid-flow solver
Numerical fluxes – WENO reconstruction

The reconstructed values are:

$$\begin{cases} u^L_{i+\frac{1}{2}} = \omega^L_0 \left(\frac{1}{3}u_i + \frac{5}{6}u_{i+1} - \frac{1}{6}u_{i+2}\right) + \omega^L_1 \left(-\frac{1}{6}u_{i-1} + \frac{5}{6}u_i + \frac{1}{3}u_{i+1}\right) + \\ \quad + \omega^L_2 \left(\frac{1}{3}u_{i-2} - \frac{7}{6}u_{i-1} + \frac{11}{6}u_i\right) \\ u^R_{i-\frac{1}{2}} = \omega^R_0 \left(\frac{11}{6}u_i - \frac{7}{6}u_{i+1} + \frac{1}{3}u_{i+2}\right) + \omega^R_1 \left(\frac{1}{3}u_{i-1} + \frac{5}{6}u_i - \frac{1}{6}u_{i+1}\right) + \\ \quad + \omega^R_2 \left(-\frac{1}{6}u_{i-2} + \frac{5}{6}u_{i-1} + \frac{1}{3}u_i\right) \end{cases}$$

Grid-flow solver

Immersed Boundary[6]

Momentum in a cell is given as the sum of a solid and a fluid contribution:

$$\left.\frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t}\right|^n = (1 - H_i)\left.\frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t}\right|^n_{fluid} + H_i\left.\frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t}\right|^n_{solid}$$

With H_i the volume fraction occupied by the solid, and:

$$\left.\frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t}\right|^n_{fluid} \approx \boldsymbol{RHS}_i^n \qquad \left.\frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t}\right|^n_{solid} \approx \boldsymbol{a}_i^n$$

This allows for the definition of a source term $\boldsymbol{S}_i^n$, such that:

$$\left.\frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t}\right|^n \approx (1 - H_i)\,\boldsymbol{RHS}_i^n + H_i\boldsymbol{a}_i^n = \boldsymbol{RHS}_i^n + \boldsymbol{S}_i^n$$

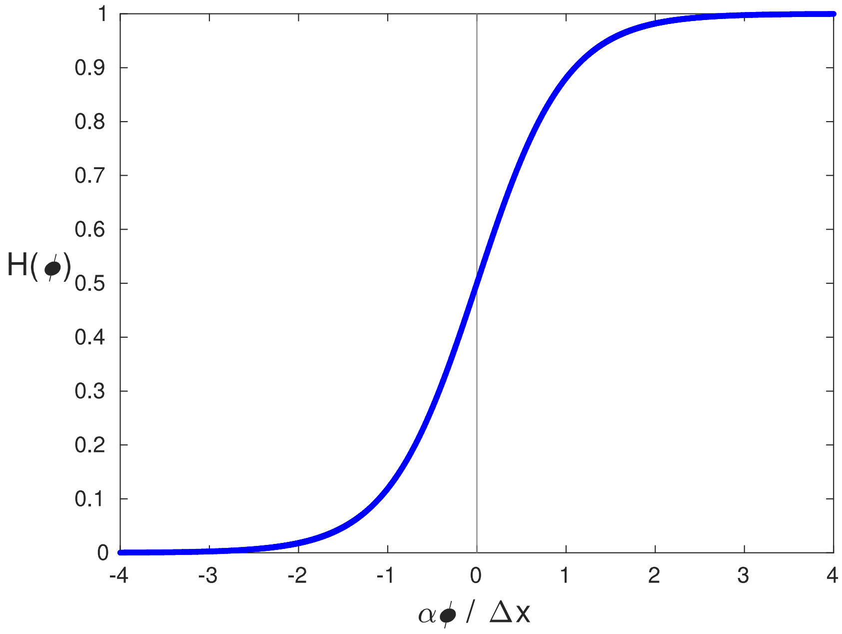
[6]: Louis Vittoz. Contributions au développement d’un solveur volumes finis sur grille cartésienne localement raffinée en vue d’application à l’hydrodynamique navale. PhD thesis, 2018.

Grid-flow solver
Immersed Boundary

The volume fraction is regularized close to the interface:

$$H(\phi) = \frac{1}{2}\left[1 + \tanh\left(\alpha \frac{\phi}{\Delta x}\right)\right]$$

Where ϕ is the signed distance.



Multiphase Flows

Problem breakdown[7]:

1. Numerical description for the location and shape of the boundary

2. Evolution algorithm for the boundary

3. Scheme for imposing boundary conditions at the interface

[7]: Cyril W Hirt and Billy D Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. Journal of computational physics, 39(1):201–225, 1981.

Multiphase Flows

1. Boundary numerical description

- Lagrangian methods
- Volume of Fluid
- Level Set

Multiphase Flows

1. Boundary numerical description

Lagrangian methods:

A set of massless particles is introduced and advected with the fluid.
The particles are used to tag either:

- One of the two fluids: *Volume tracking* (i.e. Marker and Cell[4])
- The interface itself: *Surface tracking*

[4]: Francis H Harlow and J Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. The physics of fluids, 8(12):2182–2189, 1965.

Multiphase Flows

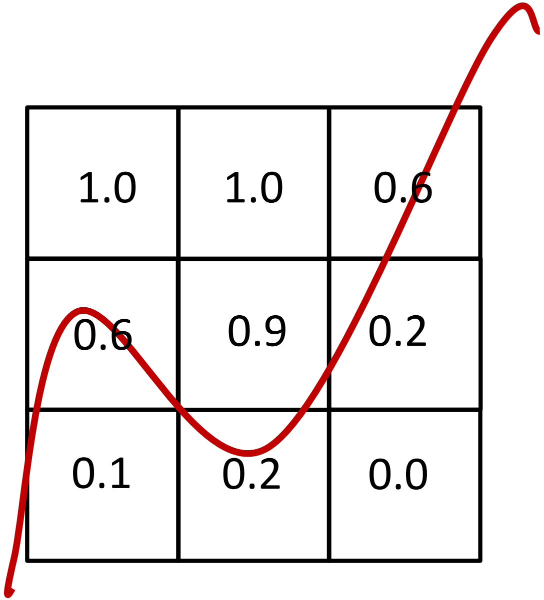
1. Boundary numerical description

Volume of Fluid[7]:

- The *volume fraction* $\alpha \in [0, 1]$ is defined as the portion of a cell filled by one particular fluid.

- α is advected with the fluid.

- An algorithm is needed to reconstruct the shape and positon of the interface from α.



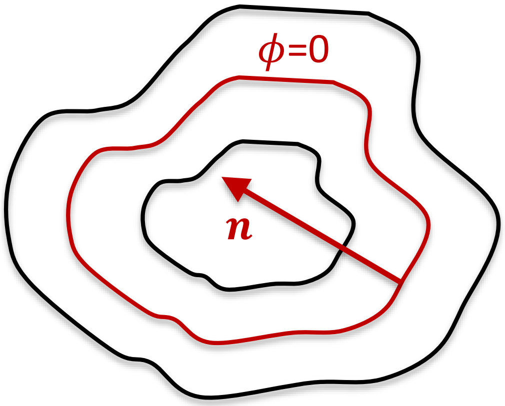
[7]: Cyril W Hirt and Billy D Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. Journal of computational physics, 39(1):201–225, 1981.

Multiphase Flows

1. Boundary numerical description

Level Set[8]:

- The *Level Set* field ϕ is defined so that the interface between the fluids is its zero-level curve: $\partial\Omega = \{\boldsymbol{x} | \, \phi\,(\boldsymbol{x},\, t) = 0\}$



- ϕ is advected with the fluid.

- Geometric properties of the interface can be computed from ϕ:

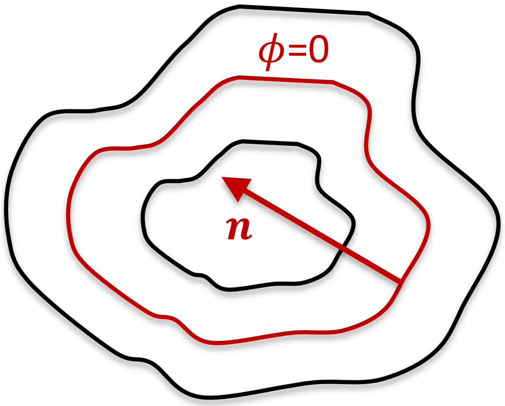
$$\boldsymbol{n} = \frac{\nabla\phi}{|\nabla\phi|} \qquad \kappa = -\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$$

[8]: Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. Journal of computational physics, 79(1):12–49, 1988.

Multiphase Flows

1. Boundary numerical description

Level Set:



- The Level Set is usually defined as the *signed distance* from the interface. This implies:

$$|\nabla\phi| = 1$$

- After advection the level curves of ϕ may no longer be parallel.

- Redistantiation[9]: ϕ can be corrected with few fictive time steps of the equation:

$$\frac{\partial d}{\partial \tau} + \mathrm{sign}\left(\phi\right)\left(|\nabla d| - 1\right) = 0$$

[9]: Daniel Hartmann, Matthias Meinke, and Wolfgang Schröder. Differential equation based constrained reinitialization for level set methods. Journal of Computational Physics, 227(14):6821–6845, 2008.

Multiphase Flows

1. Boundary numerical description

CLSVOF[10] (Conservative Level Set – Volume of Fluid):

- The Level Set offers a better description of the interface geometry than the Volume of Fluid, but has worst mass conservation properties.

- The two approaches can be coupled by:

$$\alpha = \frac{1}{\Omega_i} \int_{\Omega_i} \mathrm{H}(\phi)\, \mathrm{d}\Omega \qquad \text{where} \qquad \mathrm{H}(\phi) = \begin{cases} 1 & \text{if} \quad \phi > 0 \\ 0 & \text{if} \quad \phi \leq 0 \end{cases}$$

and used to correct each other.

[10]: Thibaut Ménard. Développement d'une méthode Level Set pour le suivi d'interface. Application de la rupture de jet liquide. 2007.

Multiphase Flows

2. Evolution algorithm for the boundary

- For Eulerian approaches (Level Set, VOF) the advection equation

$$\frac{\mathrm{D}\phi}{\mathrm{D}t} = \frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0$$

describes the movement of the interface.

- For the VOF method, a specialized advection scheme may be needed to ensure that $\alpha \in [0, 1]$.

- Advection in finite arithmetics may cause excessive diffusion of the interface.

Multiphase Flows

3. Boundary conditions at the interface

- At the interface between the two fluids there is a discontinuity in fluid properties (density, viscosity) and stresses:

$$[\boldsymbol{\tau n} - p\boldsymbol{n}] = -\sigma\kappa\boldsymbol{n}$$

where σ is the surface tension.

- For viscous flows velocity is generally considered as continuous.

- Surface tension can be applied as a discontinuity in pressure or as an impulsive force in the momentum equation, in the latter case the pressure is considered as continuous.

Multiphase Flows

3. Boundary conditions at the interface

Discontinuity management strategies:

- *Delta functions*: the discontinuity is smeared using a regularized Heaviside function $H(\phi)$

$$\begin{cases} \rho(\phi) = \rho_1 + (\rho_2 - \rho_1) H(\phi) \\ \mu(\phi) = \mu_1 + (\mu_2 - \mu_1) H(\phi) \\ \boldsymbol{f}_{sf}(\phi) = \sigma\kappa(\phi) \nabla H \end{cases}$$

- *Immersed interface*: the computational stencils are modified to account for the discontinuity.

- *Ghost fluid*[11]: each fluid is extended into the other by one (or more) cells, the ghost values are determined imposing the jump conditions.

[11]: Ronald P Fedkiw, Tariq Aslam, Barry Merriman, Stanley Osher, et al. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). Journal of computational physics, 152(2):457–492, 1999.

Implementation

Level Set advection
Weakly-compressible

Recovered by Guillaume Oger from the work of Amélie Bardin[12]

Variable density
Incompressible

Level Set advection
Incompressible

Multiphase flow
Incompressible

New developments

[12]: Amélie Bardin. Développement de méthodes d’interface dans un solveur hydrodynamique basé sur une résolution volumes finis sur maillage cartésien en hypothèse faiblement compressible. PhD thesis, 2015.

Implementation

Method summary

- Viscosity and surface tension are not considered.
- The conservative form of the advection equation for ϕ is used.
- The method can be summarized in the following passages:

1. Update the Level Set:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = -\nabla \cdot (\boldsymbol{u}^n \phi^n)$$

2. Compute the intermediate velocity field:

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} = -\nabla \cdot (\boldsymbol{u}^n \otimes \boldsymbol{u}^n) + \boldsymbol{g}$$

Implementation

Method summary

3. Compute the density at the cell faces:

$$\rho\left(\phi^{n+1}\right) = \rho_1 + \frac{\rho_2 - \rho_1}{2}\left[1 + \tanh\left(\alpha\frac{\phi^{n+1}}{\Delta x}\right)\right]$$

4. Solve the variable coefficient Poisson equation for the pressure:

$$\nabla \cdot \left(\frac{1}{\rho\left(\phi^{n+1}\right)}\nabla p^{n+1}\right) = \frac{1}{\Delta t}\nabla \cdot \boldsymbol{u}^*$$

5. Apply the pressure gradient to correct the vector field:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \frac{\Delta t}{\rho\left(\phi^{n+1}\right)}\nabla p^{n+1}$$

Implementation
Level Set Advection

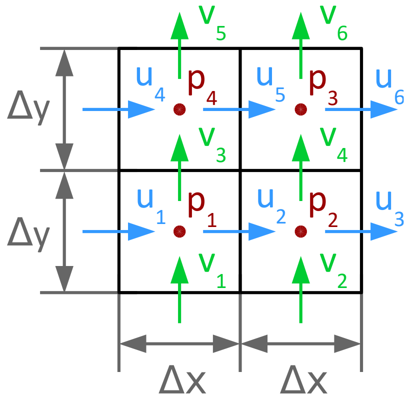
- In the incompressible formulation the Level Set is placed at the cell centres while the velocity is placed at the cell faces.
- In the earlier version of the code (2019) on which this work started, no high-order reconstruction scheme was available to cell-centred variables in the incompressible formulation.
- In earlier tests, a first order reconstruction had to be used for the Level Set.
- The multipasic scheme was adapted to a more recent (2020) version of the code, removing this restriction.
- Later tests employ the WENO and MUSCL reconstruction schemes for the Level Set.

Implementation

Variable coefficient Poisson Equation

Divergence matrices $\boldsymbol{D}_x$; $\boldsymbol{D}_y$; $\boldsymbol{D}_z$ are constructed so that they perform centred finite differences on face-centred variables, i.e.:

$$\boldsymbol{D}_x = \frac{\Delta y \cancel{\Delta x}}{\cancel{\Delta x}} \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$



The gradient matrices are obtained as their transpose:

$$\boldsymbol{G}_x = \mathrm{diag}\left\{\frac{1}{V_i}\right\} \boldsymbol{D}_x^T \,; \qquad \boldsymbol{G}_y = \mathrm{diag}\left\{\frac{1}{V_i}\right\} \boldsymbol{D}_y^T \,; \qquad \boldsymbol{G}_z = \mathrm{diag}\left\{\frac{1}{V_i}\right\} \boldsymbol{D}_z^T$$

Implementation

Variable coefficient Poisson Equation

In the monophasic case, the Laplacian and the right-hand side term of the Poisson equation for the pressure are computed as:

$$\boldsymbol{L} = \boldsymbol{D}_x \boldsymbol{G}_x + \boldsymbol{D}_y \boldsymbol{G}_y + \boldsymbol{D}_z \boldsymbol{G}_z$$

$$\boldsymbol{b} = \frac{\rho}{\Delta t} \left(\boldsymbol{D}_x \boldsymbol{u} + \boldsymbol{D}_y \boldsymbol{v} + \boldsymbol{D}_z \boldsymbol{w} \right)$$

Implementation

Variable coefficient Poisson Equation

In the multiphase scheme, the Level Set is interpolated at the cell faces and used to compute the density. The diagonal matrices

$$\mathrm{diag}\left\{\frac{1}{\rho}\right\}_x; \qquad \mathrm{diag}\left\{\frac{1}{\rho}\right\}_y; \qquad \mathrm{diag}\left\{\frac{1}{\rho}\right\}_z$$

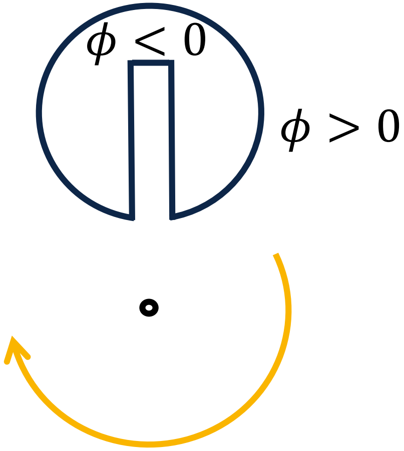
Are built and used to obtain the Laplacian matrix and the right-hand side term of the Poisson equation:

$$\boldsymbol{L} = \boldsymbol{D}_x \mathrm{diag}\left\{\frac{1}{\rho}\right\}_x \boldsymbol{G}_x + \boldsymbol{D}_y \mathrm{diag}\left\{\frac{1}{\rho}\right\}_y \boldsymbol{G}_y + \boldsymbol{D}_z \mathrm{diag}\left\{\frac{1}{\rho}\right\}_z \boldsymbol{G}_z$$

$$\boldsymbol{b} = \frac{1}{\Delta t}\left(\mathrm{diag}\left\{\frac{1}{\rho}\right\}_x \boldsymbol{D}_x \boldsymbol{u} + \mathrm{diag}\left\{\frac{1}{\rho}\right\}_y \boldsymbol{D}_y \boldsymbol{v} + \mathrm{diag}\left\{\frac{1}{\rho}\right\}_z \boldsymbol{D}_z \boldsymbol{w}\right)$$

Results

Level-Set avection – Zalesak disk[13]



Domain: $[0,1] \times [0,1]$

Velocity field: $\begin{cases} u = -\Omega(y - y_0) \\ v = \quad \Omega(\mathrm{x} - \mathrm{x}_0) \end{cases}$

With:
$x_0 = 0.5$
$y_0 = 0.5$
$\Omega = 2\pi$

Base grid: 64×64 cells

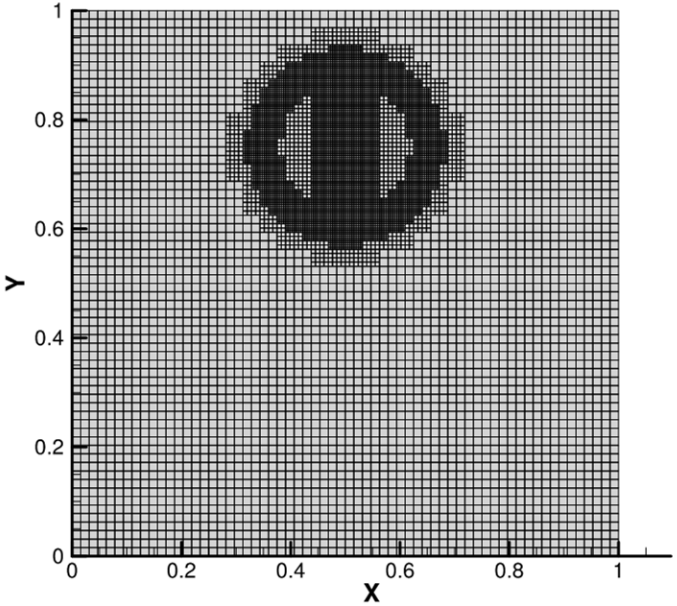
Dynamic AMR

min. cell size: $3.90625 \cdot 10^{-2}$

[13]: Steven T. Zalesak. Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids. Journal of computational physics, 31(3):335–362, 1979.

Results
Level-Set avection – Zalesak disk

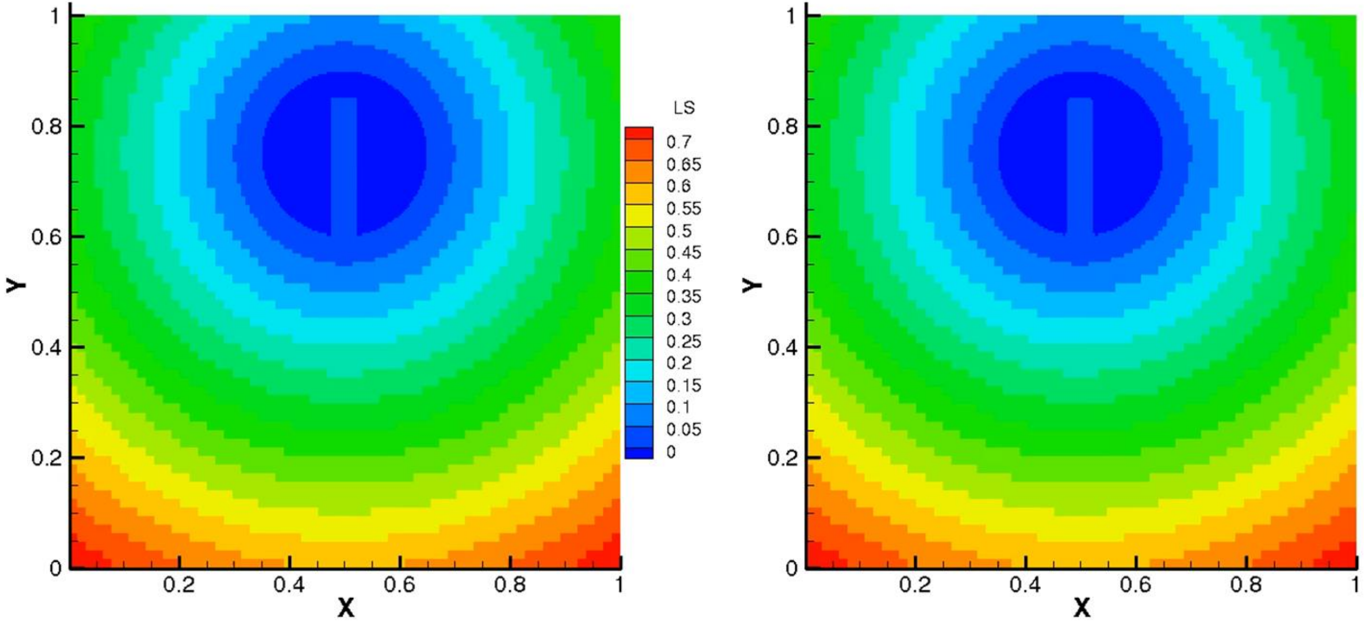
Dynamic AMR:



Results

Level-Set avection – Zalesak disk

1st order accurate reconstruction (2019) | WENO reconstruction (2020)



Results
PETSc Reinitialization test

- Variable density means the Laplacian matrix for the pressure Poisson equation needs to be reinitialized at every time iteration. This comes at a performance cost.

- The Taylor-Green vortex test case was run with and without reinitializing the solver.

- The test was run on two 12-core Intel Xeon E5-2680v3.

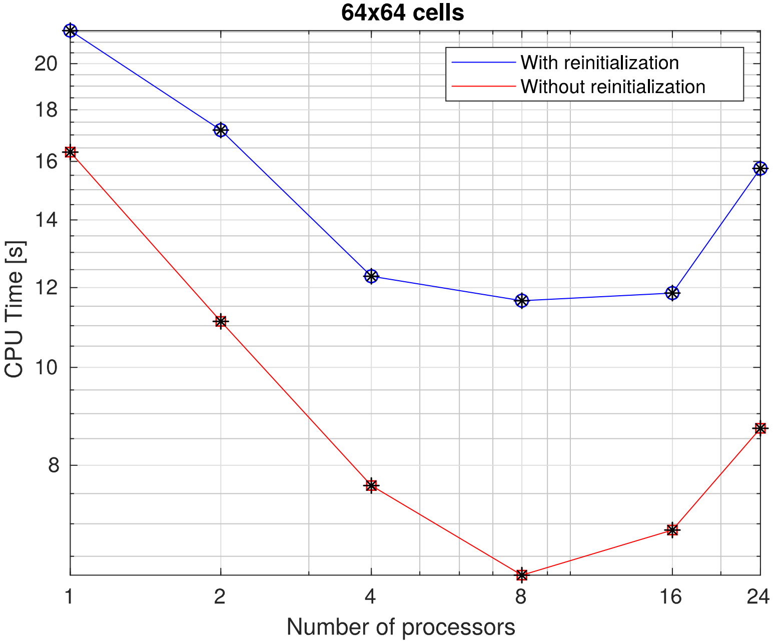
Results

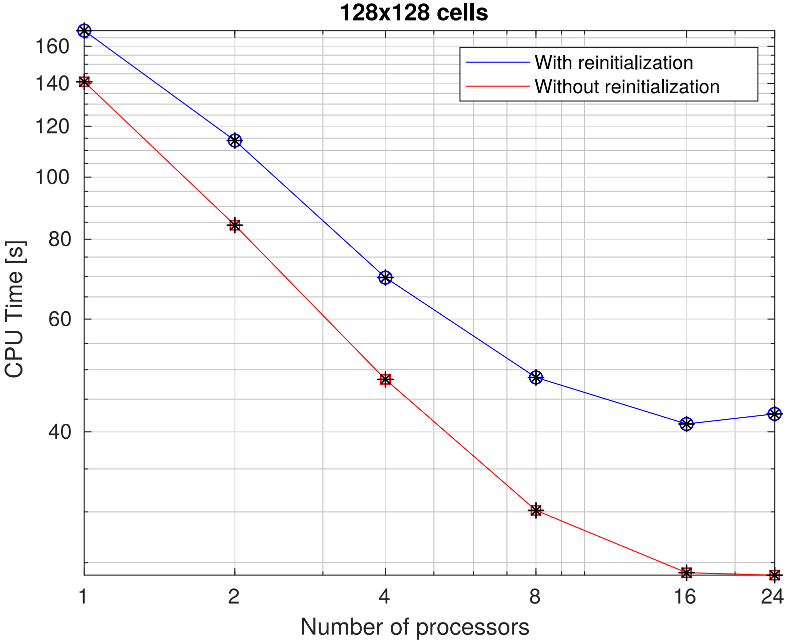
PETSc Reinitialization test

Taylor-Green vortex test setup:

$$
\begin{cases}
u(x,y) = & U_0 \sin\left(2\pi \frac{x}{L}\right) \cos\left(2\pi \frac{y}{L}\right) \\
v(x,y) = & -U_0 \cos\left(2\pi \frac{x}{L}\right) \sin\left(2\pi \frac{y}{L}\right) \\
p(x,y) = & \frac{\rho U_0^2}{4} \left[\cos\left(4\pi \frac{x}{L}\right) + \cos\left(4\pi \frac{y}{L}\right)\right]
\end{cases}
$$

Results
PETSc Reinitialization test





Results
Poisson equation convergence test

The accuracy of the operators used to solve the pressure Poisson equations are tested by initializing the velocity field:

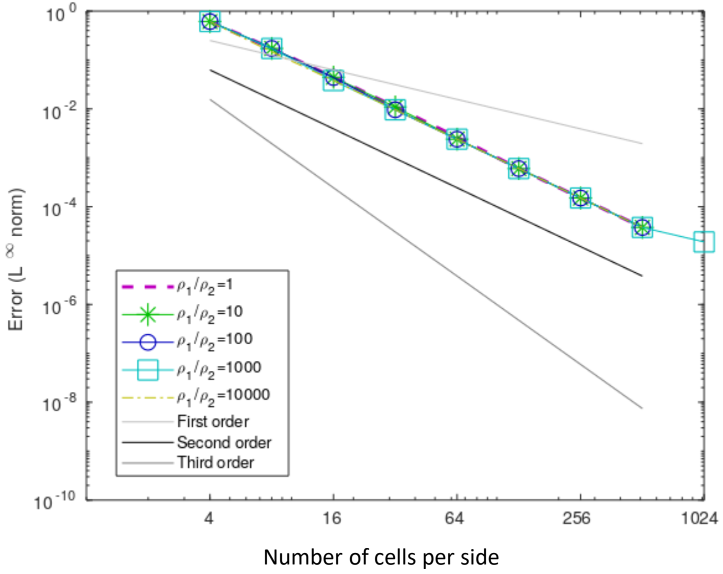
$$\begin{cases} u = \dfrac{-2\Delta t\pi \sin(2\pi x)\cos(2\pi y)}{\rho(x,\, y)} \\ v = \dfrac{-2\Delta t\pi \cos(2\pi x)\sin(2\pi y)}{\rho(x,\, y)} \end{cases}$$

A singe time step is performed, having disabled advection and the source term. The pressure is compared with the analytical solution:

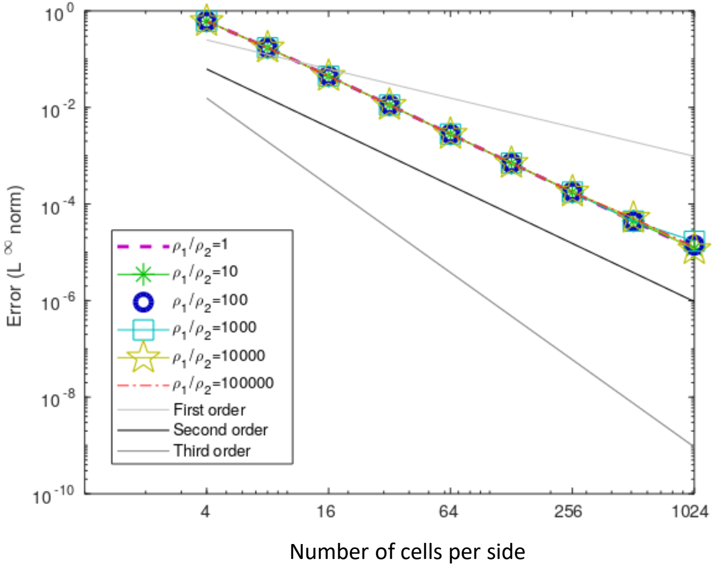
$$p = \cos(2\pi x)\cos(2\pi y)$$

Results
Poisson equation convergence test

2019 code

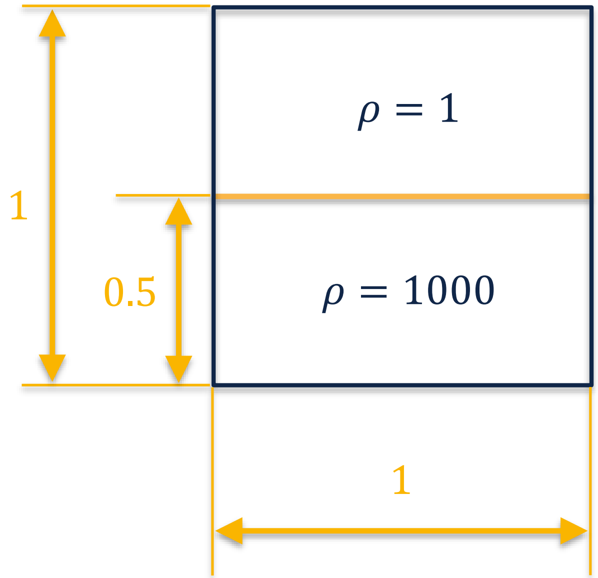


2020 code



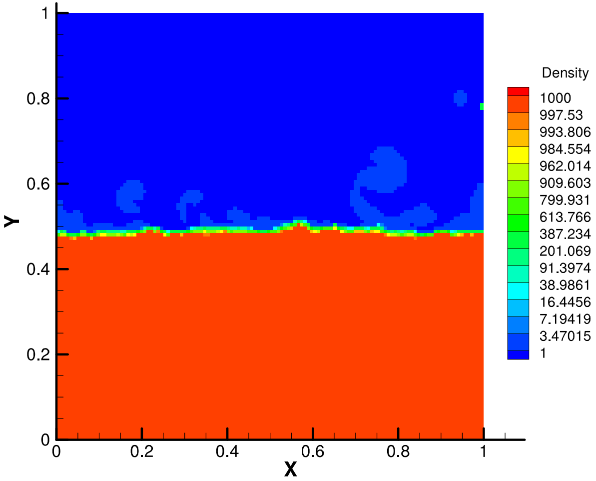
2020 Results

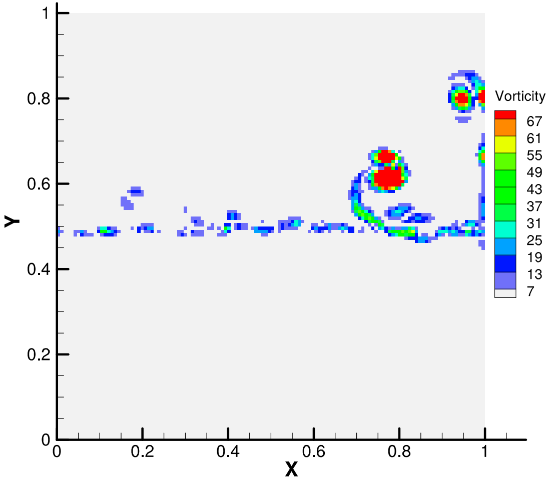
Hydrostatic test



A $[0,1] \times [0,1]$ domain surrounded by solid walls is filled with two immiscible fluids with different densities, subjected to the force of gravity. Both fluids are initially still.

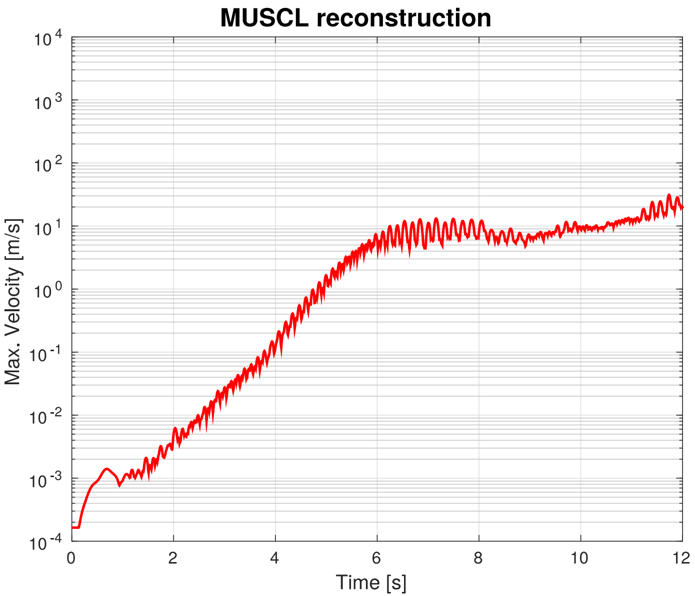
2020 Results
Hydrostatic test

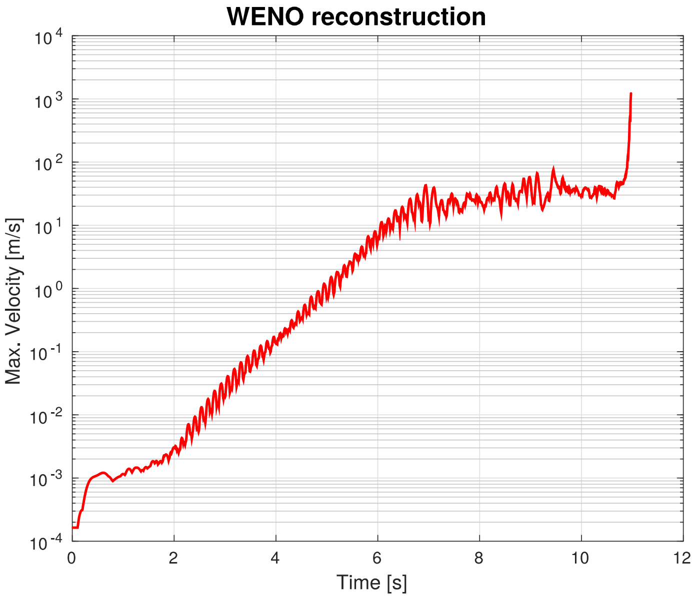




Spurious oscillations ripple the interface between the two fluids, shedding vortices in the lighter one.

2020 Results
Hydrostatic test





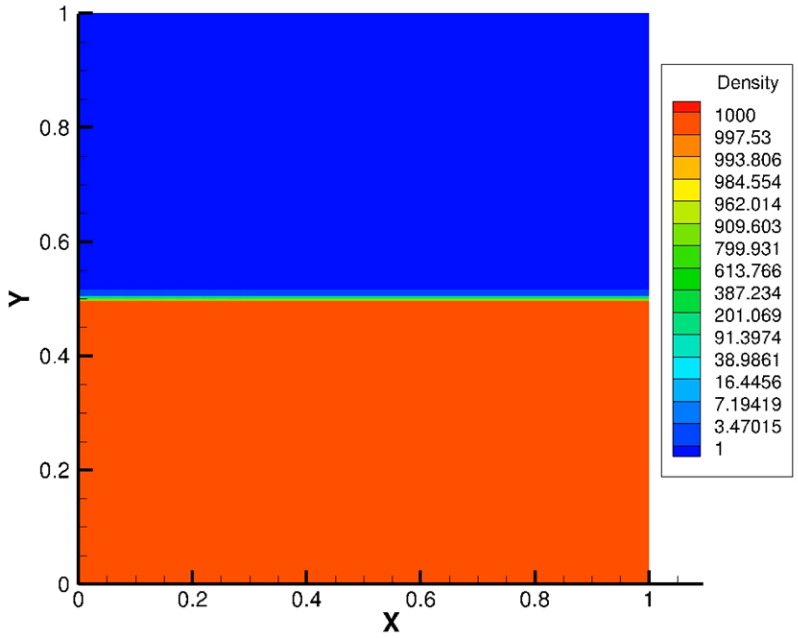
Growth of the oscillations amplitude for a 32x32 grid

2020 Results
Hydrostatic test

- The oscillations amplitude grows exponentially at the beginning, then stabilizes.
- The simulation using WENO reconstruction crashes, however further tests ran smoothly.
- These simulations use forward Euler for time integration, further tests with the fourth order Runge-Kutta scheme showed a lower asymptotical oscillation amplitude.

2020 Results
Linear sloshing

- The same domain as in the hydrostatic test case was used, but with the addition of a horizontal acceleration equal to 0.5 m/s^2.
- The interface does not start from an equilibrium position, hence it starts to oscillate.
- A 256x256 grid was used, the simulation ran for 4 seconds.



2020 Results
Linear sloshing

An analytic expression for the vertical position of the interface can be derived from linearized potential theory[12][13]:

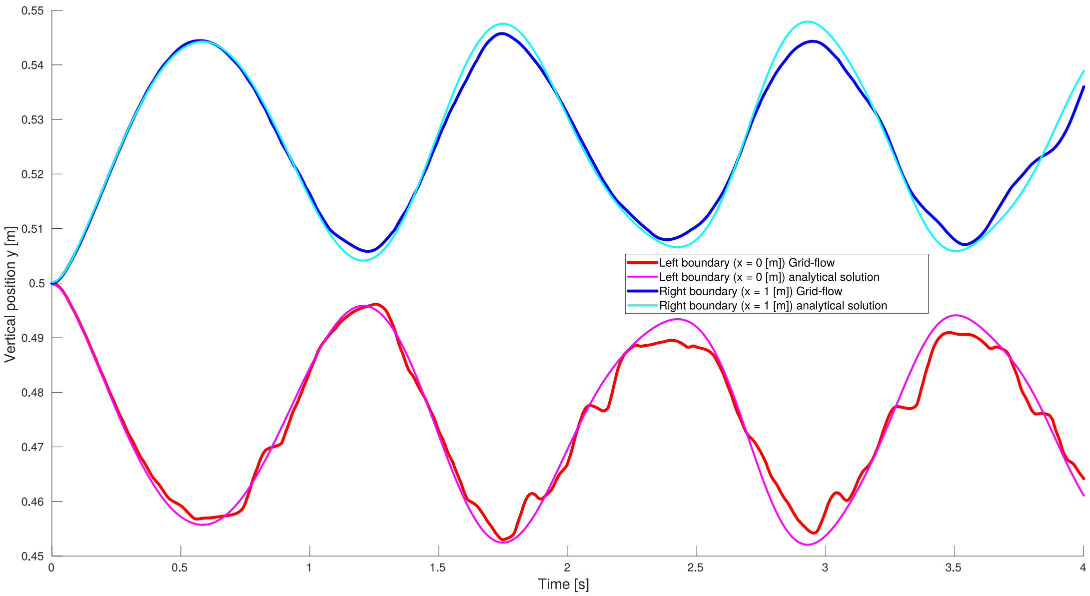
$$k_n = \frac{n\pi}{L}$$

$$\omega_n^2 = \frac{g_0 k_n \left(\rho_2 - \rho_1\right)}{\rho_1 \coth\left(\omega_{2n+1} t\right) + \rho_2 \coth\left(k_{2n+1} x\right)}$$

$$y\left(x,\, t\right) = h_2 + \frac{a}{g_0}\left[x - \frac{L}{2} + \sum_{n=0}^{\infty} \frac{4}{L k_{2n+1}^2} \cos\left(\omega_{2n+1} t\right) \cos\left(k_{2n+1} x\right)\right]$$

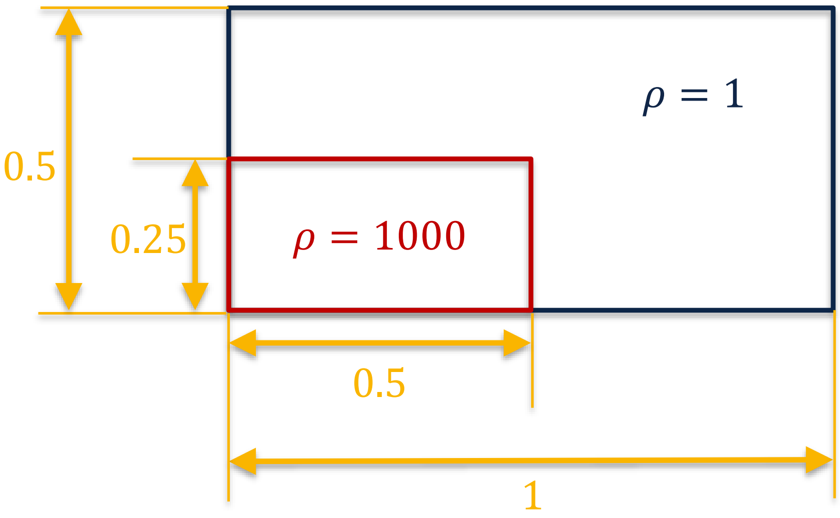
[12]: Amélie Bardin. Développement de méthodes d’interface dans un solveur hydrodynamique basé sur une résolution volumes finis sur maillage cartésien en hypothèse faiblement compressible. PhD thesis, 2015.
[13]: N Grenier, J.-P. Vila, and P Villedieu. An accurate low-Mach scheme for a compressible two-fluid model applied to free-surface flows. Journal of Computational Physics, 252:119, 2013.

2020 Results
Linear sloshing



Results

Dambreak



A pocket of denser fluid is initially contained in a rectangle on the lower left of the domain, gravity $(g_0 = -9.81)$ acts vertically on both fluids.

Results
Dambreak

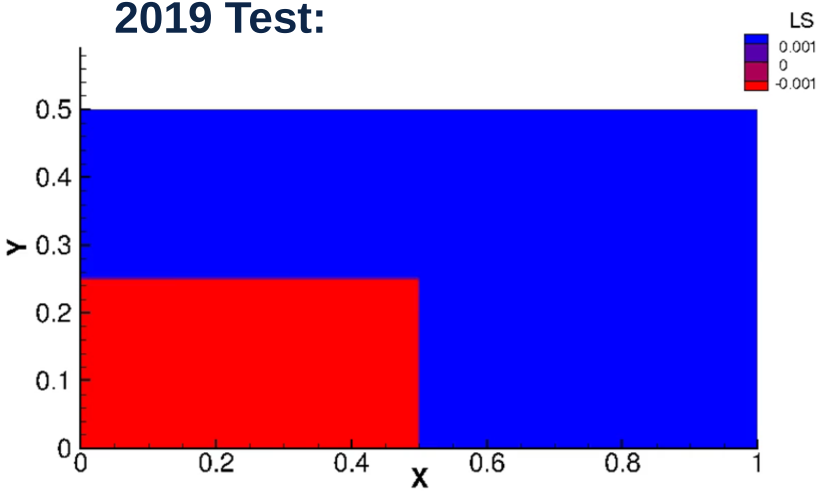
2019 Test:

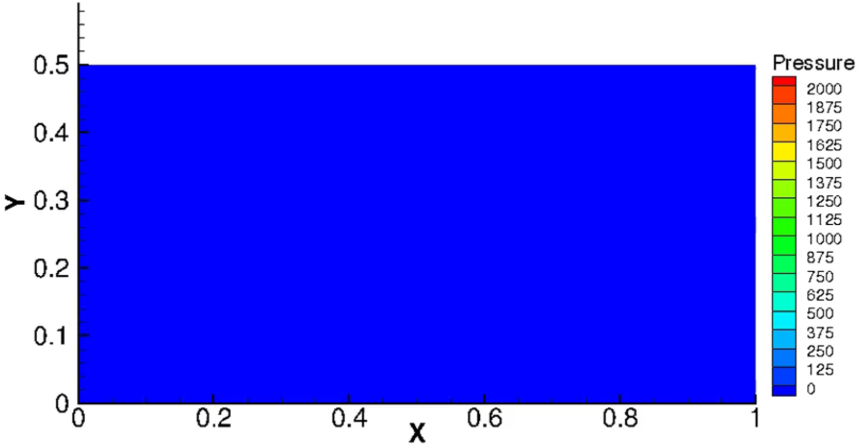
- The velocity is advected using WENO, the Level Set using a first order accurate reconstruction

- Dynamic AMR is used. The base grid is 128x64 cells, the finest cells have dimensions $\Delta x = 1.953125 \cdot 10^{-3}$.

Results
Dambreak

2019 Test:



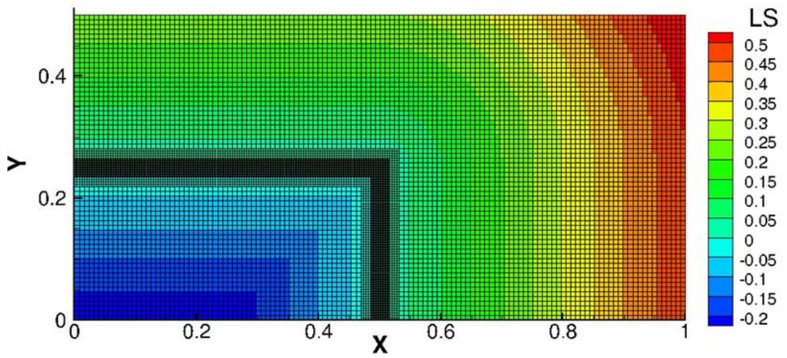


Results
Dambreak

2019 Test:

- The denser fluid disappears after 15 time units of simulation. This is likely due to the excessive diffusion of the Level-Set.

- As seen on the right, the Level-Set becomes almost uniform by the end.

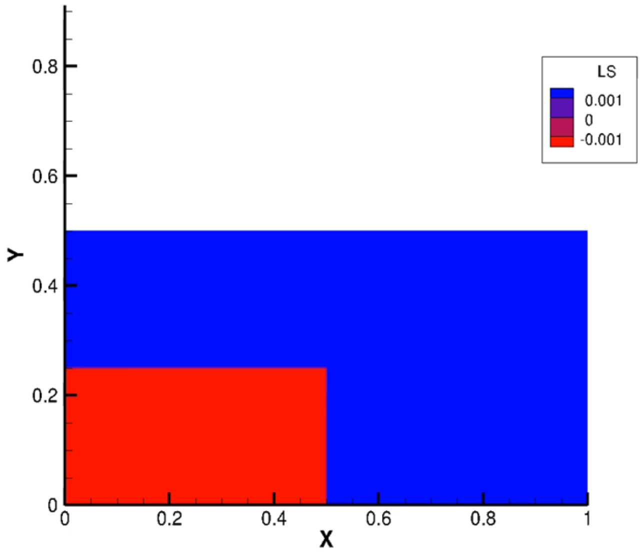


Results
Dambreak

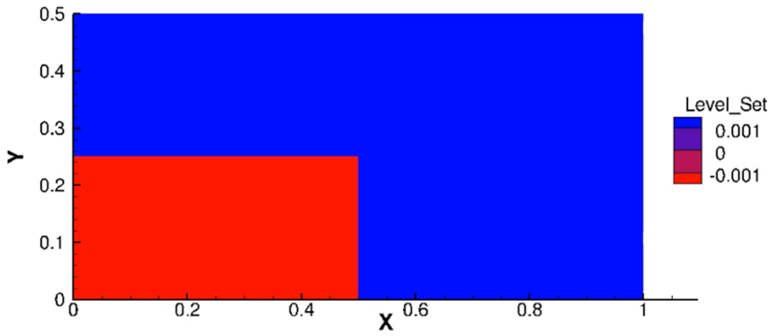
- The dambreak test was re-run with the 2019 code on a static 256x128 grid.

- This revealed stability problems.

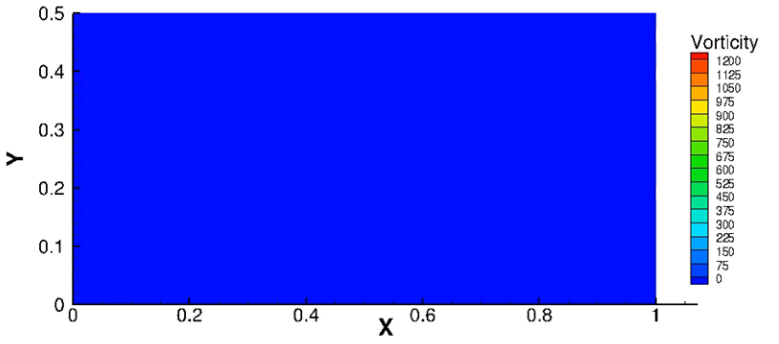
- A careful choice of parameters, however, allows the 2019 code to run until completion.



Results
Dambreak



t = 0



- The same setup was run on the 2020 code, using WENO for both the velocity and the Level Set
- Despite various attempts and variations, the use of WENO for both the velocity and the Level Set always causes divergence.

CENTRALE NANTES

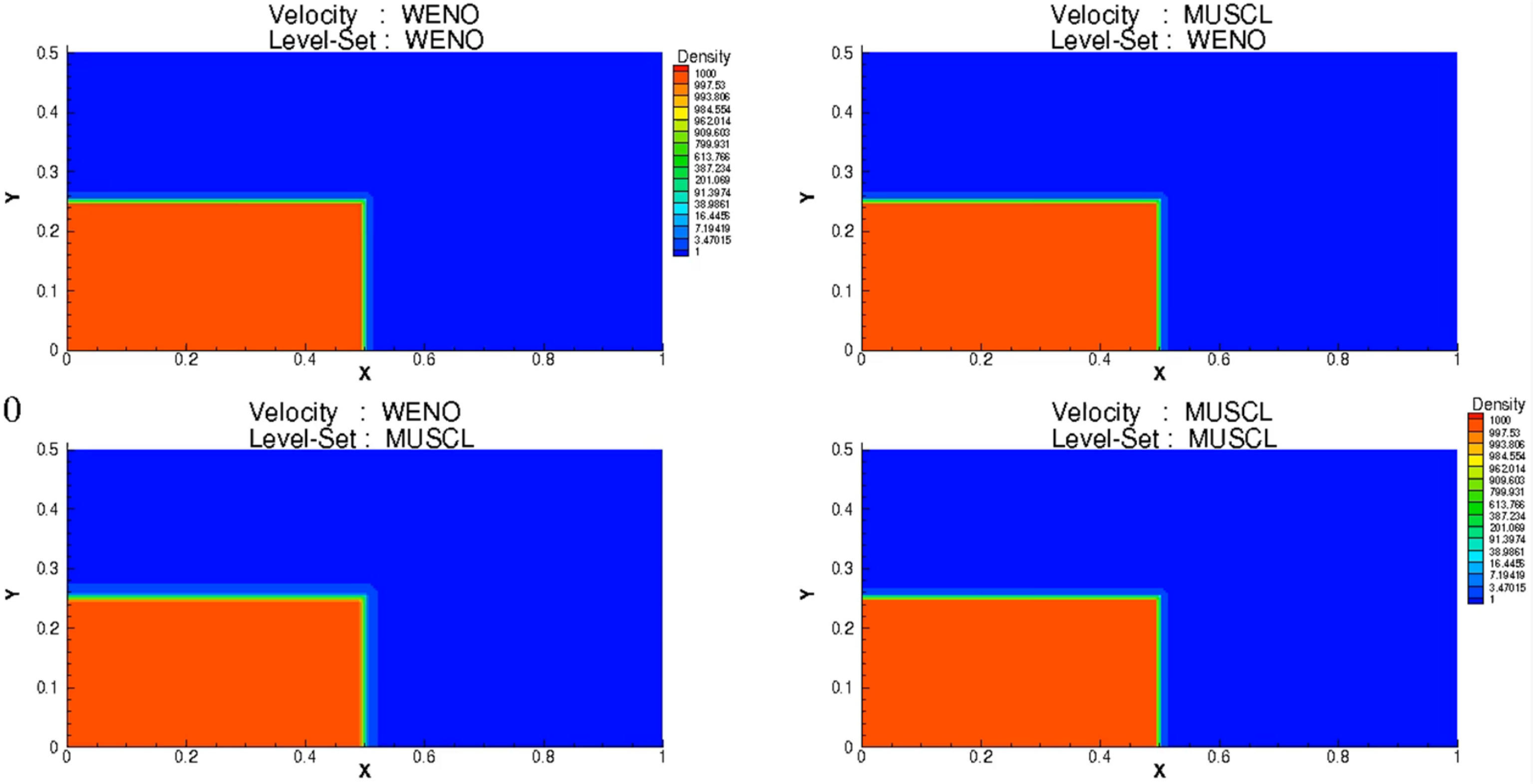
Results
Dambreak

Advection strategies tested on the 2020 code

1. Velocity reconstructed with WENO, Level-Set reconstructed with WENO.
2. Velocity reconstructed with WENO, Level-Set reconstructed with MUSCL.
3. Velocity reconstructed with MUSCL, Level-Set reconstructed with WENO.
4. Velocity reconstructed with MUSCL, Level-Set reconstructed with MUSCL.

Results
Dambreak

$t = 0$



Results
Dambreak

	WENO reconstruction for the velocity	MUSCL reconstruction for the velocity
WENO reconstruction for the Level-Set	Always unstable. Smoothness indicators can slow down instability.	Stable if smoothness indicators are active. Denser fluid does not disappear. Irregular density field.
MUSCL reconstruction for the Level-Set	Analogous to the 2019 code. Stable under certain conditions. Smoothness indicators play no role. Denser fluid disappears completely.	Stable and regular. Denser fluid does not disappear. Lower velocity.