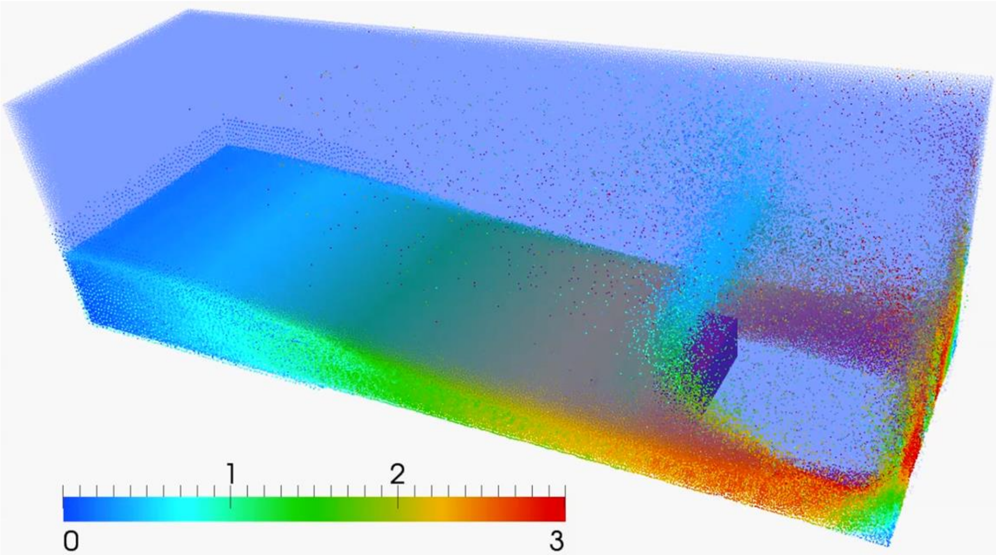
Multi-phase Modelling of violent air-water flows using DualSPHysics

Athanasios Mokos, Benedict D. Rogers, Peter K. Stansby

School of Mechanical, Aeronautical and Civil Engineering
University of Manchester
UK

Outline of Presentation

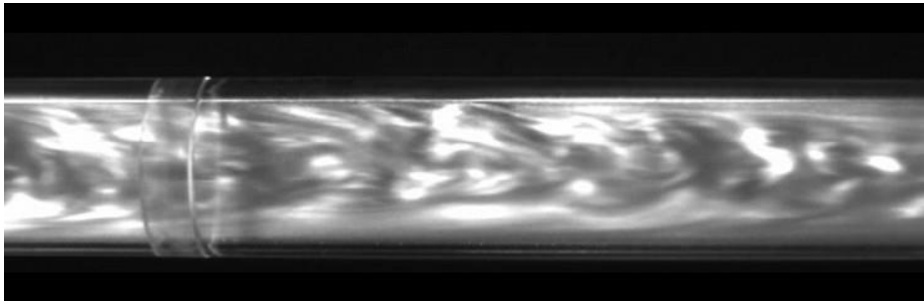
- Motivation for Research
- Multi-Phase Model
- Particle Shifting Algorithm
  - Issues on higher resolutions
  - Modelling of the air phase
- GPU Optimization
  - Demands of GPU programming
  - Optimization of SPH
- Results
  - Runtime Results
  - Sloshing Tank
  - 3D Obstacle Impact Case
- Conclusions & Future Developments



Motivation for Research

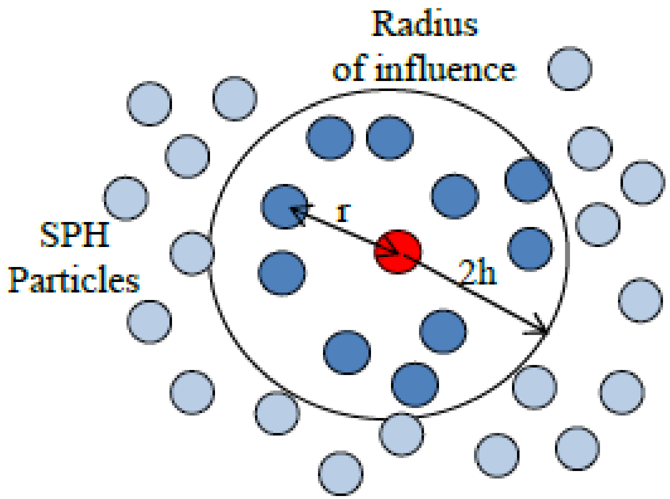
- Primary focus on violent water flows with breaking free surface, e.g. wave impact/slamming or potentially explosive pipe flows

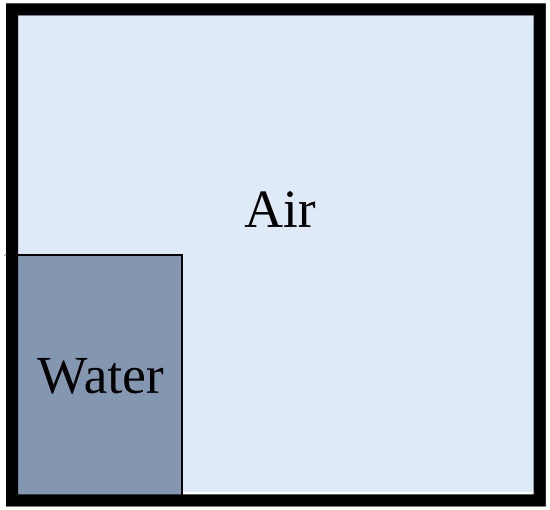
- Presence of air is important for violent flows: large changes in pressure and velocities, e.g. flip-through effect



Motivation for Research

- Lagrangian methods such as SPH are ideal as they can capture the fractured surface to minute detail
- Increased domain size due to the presence of air and the high-order phenomena necessitate high resolutions and large numbers of particles
- Develop an **efficient SPH Methodology for simulating air-water mixtures** using multi-phase model to improve on single-phase computations





Challenges of Multi-Phase SPH

- **Challenge 1**: Interaction of a gas and a liquid phase
  - Large density ratio (~1000)
  - Large pressure gradients in the interface
  - Treatment of the gas phase

- **SOLUTION**: Use of the Colagrossi & Landrini (2003) multi-phase model

SPH Formulation for Multi-Phase Flows

Equation of State: $$P(\rho) = P_0\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] - a\rho^2 + \chi$$

- Last two terms are only used for air particles
- α is a cohesion coefficient given by: $$a = 1.5g\frac{\rho_w}{\rho_a^2}L$$
- χ is the constant background pressure

SPH formulation for Navier-Stokes:

- Momentum
  - Water: $$\left\langle \frac{d\mathbf{u}}{dt} \right\rangle = -\frac{1}{\rho_i}\sum_j \left(\frac{m_j}{\rho_j}\left[p_j + p_i + \Pi_{ij}\right]\nabla_i W_{ij} \right)$$
  - Air: $$\left\langle \frac{d\mathbf{u}}{dt} \right\rangle = -\frac{1}{\rho_i}\sum_j \left(\frac{m_j}{\rho_j}\left[p_j + p_i + \Pi_{ij}\right]\nabla_i W_{ij} - 2a\rho_a^2 \frac{m_j}{\rho_j}\nabla_i W_{ij} \right)$$
- Mass: $$\left\langle \frac{d\rho}{dt} \right\rangle = \rho_i \sum_j \frac{m_j}{\rho_j}\left(\mathbf{u}_i - \mathbf{u}_j\right)\cdot\nabla_i W_{ij}$$

Challenges of Multi-Phase SPH

- **Challenge 1**: Interaction of a gas and a liquid phase

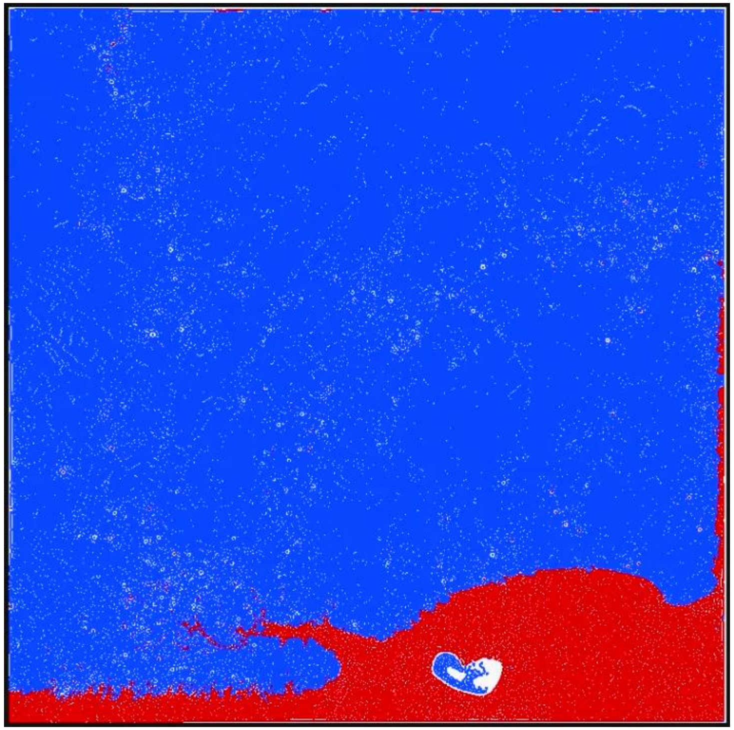
  - ~~Large density ratio (~1000)~~
  - ~~Large pressure gradients in the interface~~
  - Treatment of the gas phase

- **Issues**:

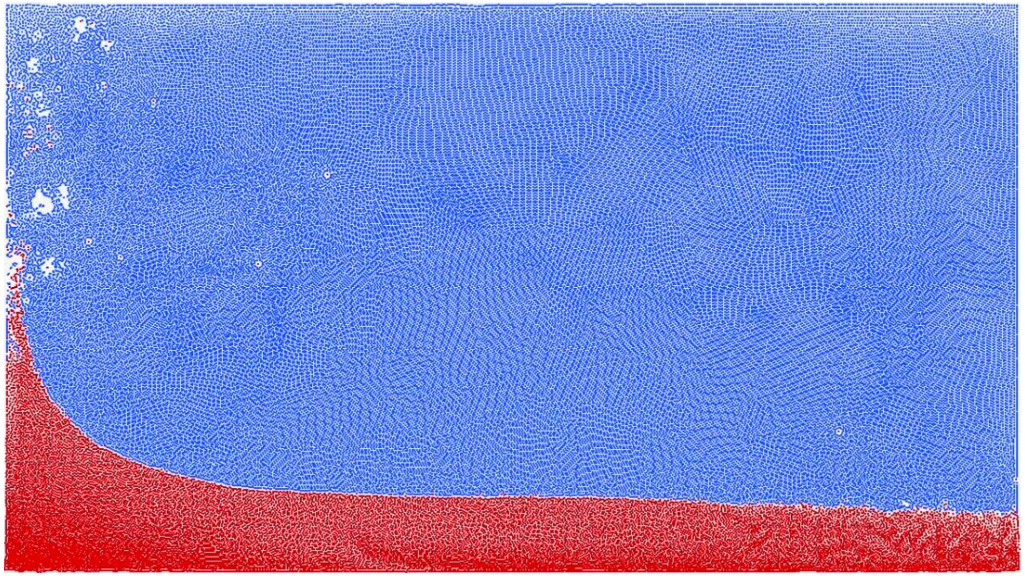
  - Gas phase treated as compressible liquid due to the equation of state
  - Does not expand to areas with lower concentration

Issues on Higher Resolutions

Dry Dam Break



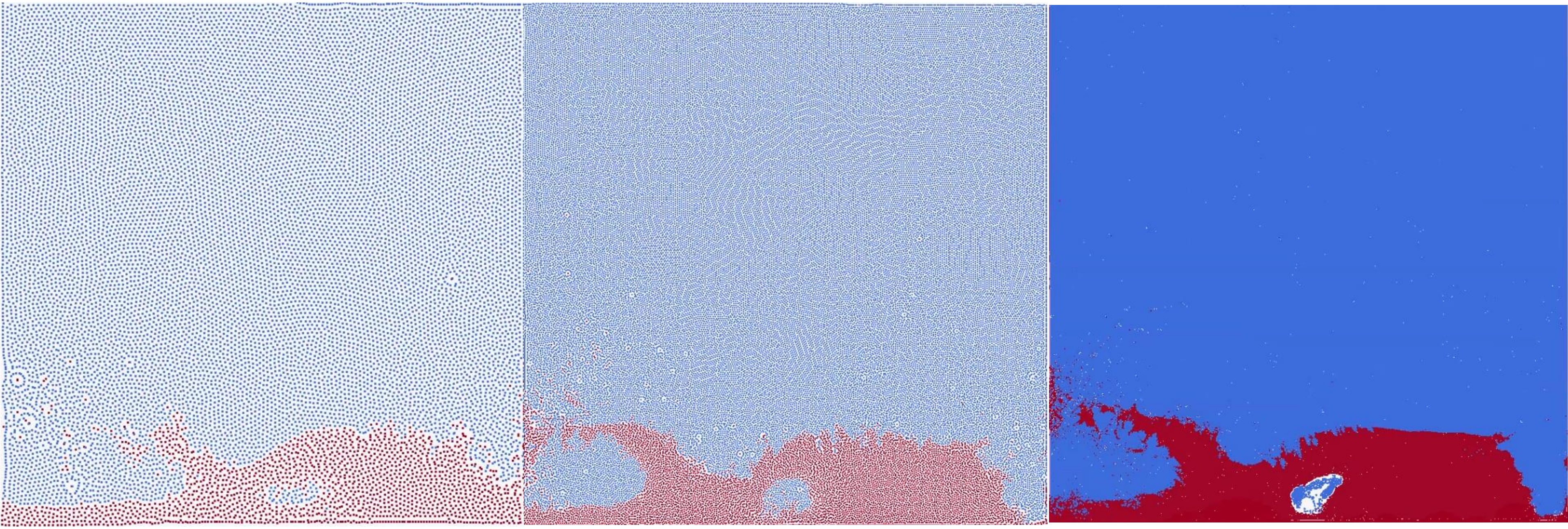
Sloshing Tank



- Voids created in the air phase
- Located in entrained flow and in the particle interface for more sensitive cases
- Issues with isolated water particles inside the air phase

Particle shifting

Issue: Voids appear only in high resolutions



Particle shifting

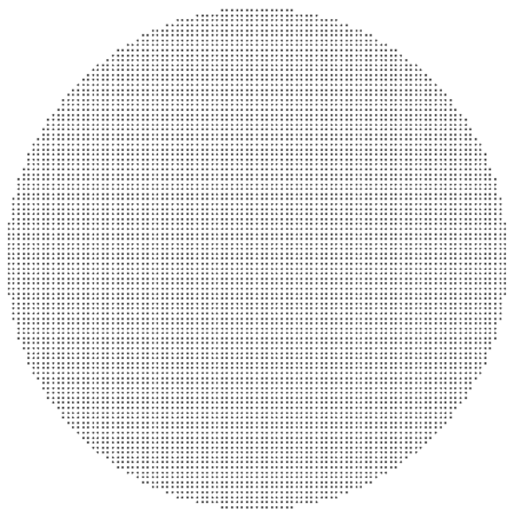
Issue: Voids appear only in high resolutions

Solution: Fickian-based approach by Lind et al. (2012)

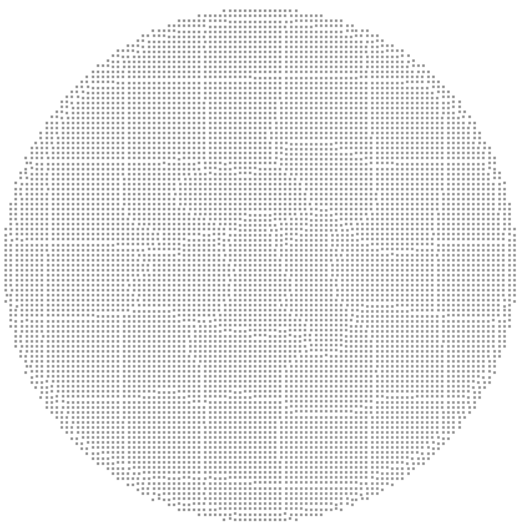
- Numerical treatment based on Fick's law: $$\delta \mathbf{r}_s = -D\nabla C_i$$
- Shifting dependent on particle concentration: $$\nabla C_i = \sum_j C_{ij} \frac{m_j}{\rho_j} \nabla W_{ij}$$
- Diffusion based on particle velocity (Skillen et al. 2013): $$D = -A_s h \|u\|_i \Delta t$$
- Free-surface correction term: $$\delta \mathbf{r}_s = -D\left(\frac{\partial C_i}{\partial s}\mathbf{s} + \frac{\partial C_i}{\partial b}\mathbf{b} + \alpha\left(\frac{\partial C_i}{\partial n} - \beta \right)\mathbf{n} \right)$$
    - Used only for the liquid phase

Modelling the Air Phase: Shifting with surface correction

Initial Particle Position



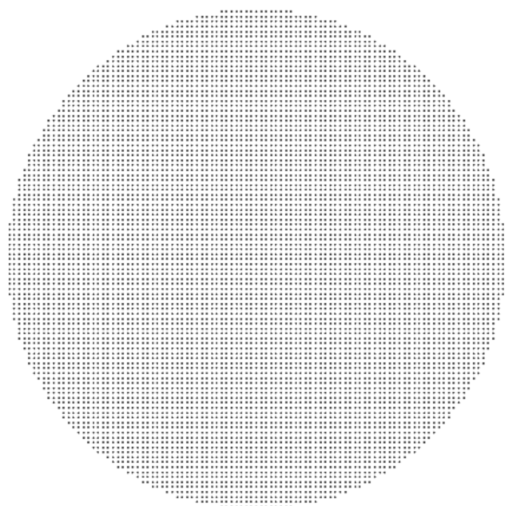
Position after 0.2s



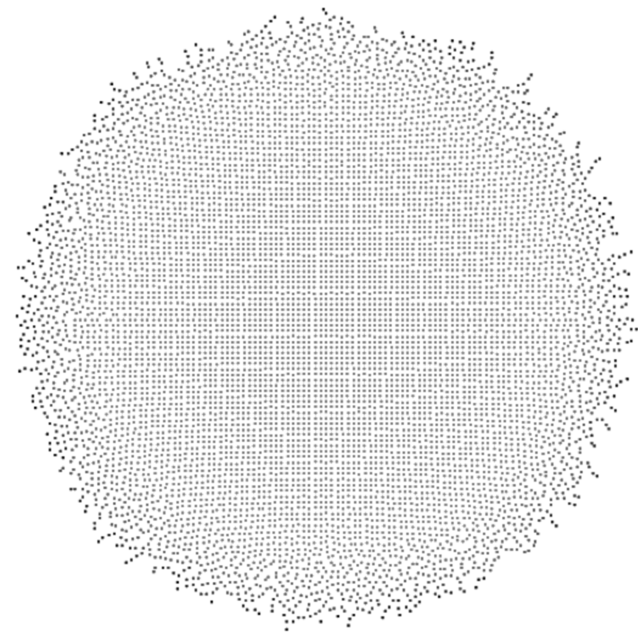
- Volume of air expanded by applying a constant pressure
- Air volume only slightly expands
- Minor repositioning of the particles inside the volume
- Increased number of particles close to the free surface

Modelling the Air Phase: Shifting without surface correction

Initial Particle Position



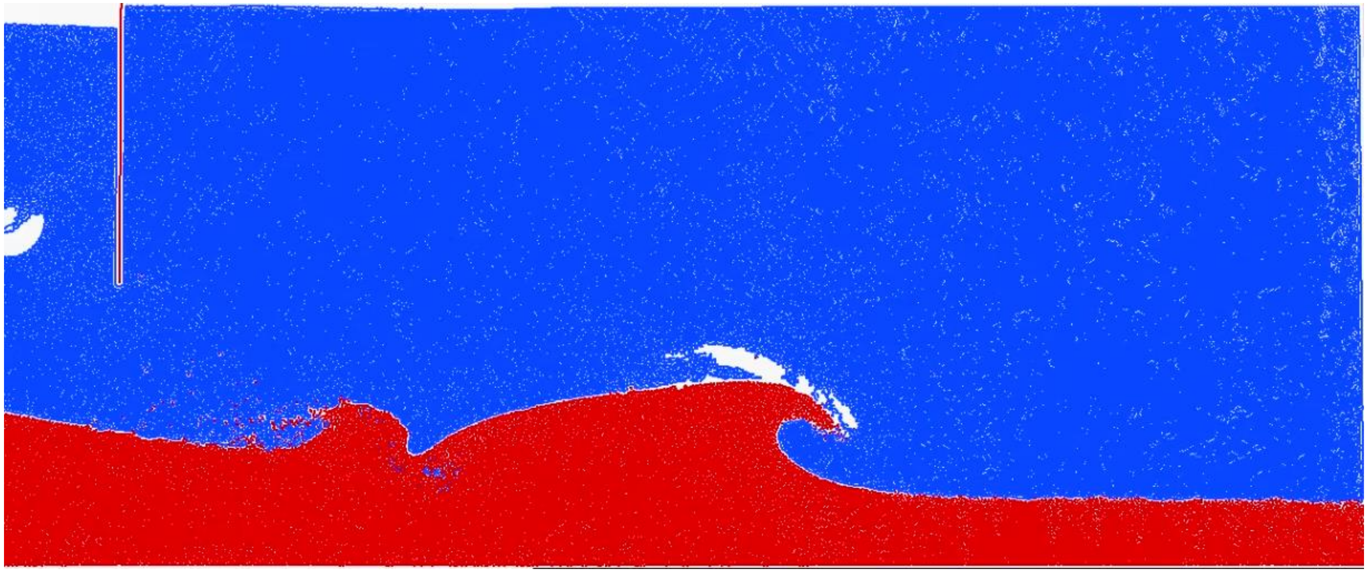
Position after 0.2s



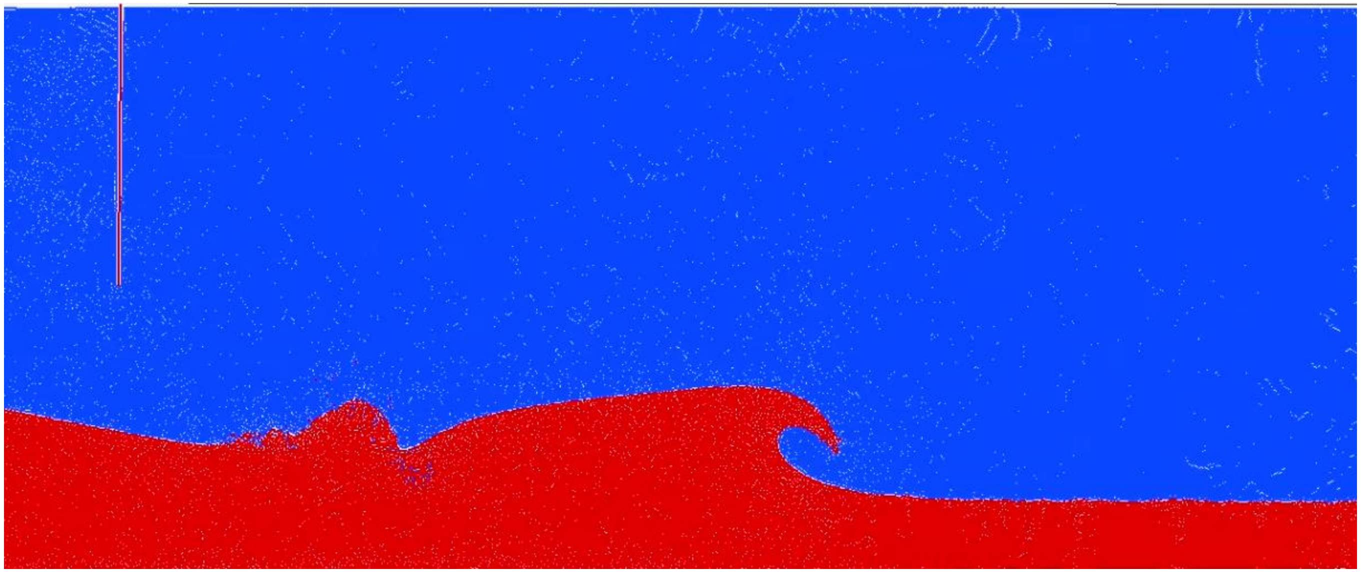
- Volume of air expanded by applying a constant pressure
- Air expands uniformly
- Concentration gradient is consistent with a full kernel
- Inconsistencies at the edge due to the single precision and the small number of particles

Wet Dam Break

- **Original Result**



- **Particle Shifting**



Challenges of Multi-Phase SPH

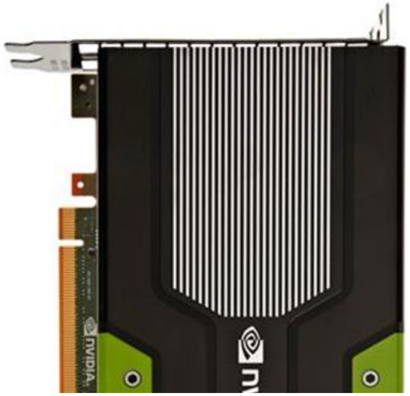
- ~~**Challenge 1**: Interaction of a gas and a liquid phase~~
  - ~~Large density ratio (~1000)~~
  - ~~Large pressure gradients in the interface~~
  - ~~Treatment of the gas phase~~

- **Challenge 2**: Computational treatment
  - SPH is computationally expensive
  - Increased number of particles due to the second phase

- **SOLUTION**: Use Graphics Processing Units (GPUs) for the simulation

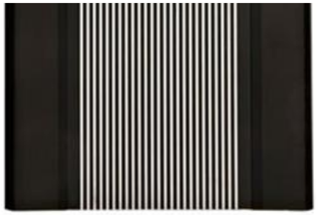
Challenges of Multi-Phase SPH

- ~~**Challenge 1**: Interaction of a gas and a liquid phase~~
  - ~~Large density ratio (~1000)~~
  - ~~Large pressure gradients in the interface~~
  - ~~Treatment of the gas phase~~



New Challenge: Optimise the multi-phase code for GPU

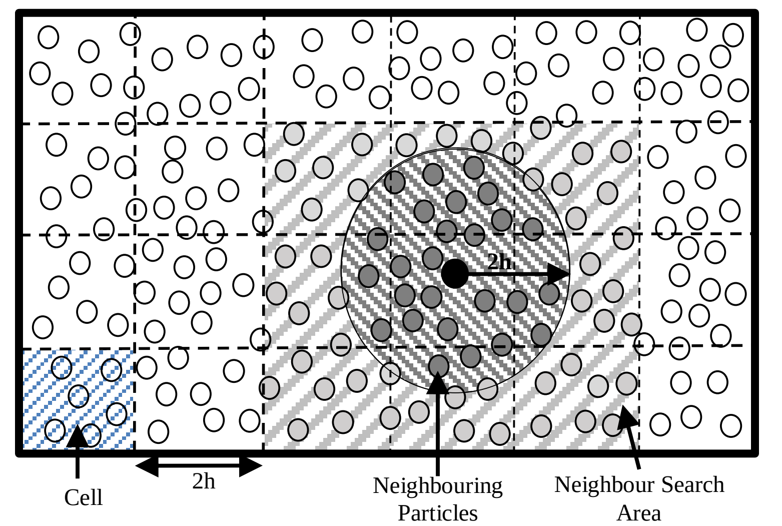
- Increased number of particles due to the second phase



- **SOLUTION**: Use Graphics Processing Units (GPUs) for the simulation

Demands of a multi-phase GPU code

- Distinguish particles belonging to different phases
  - Load different initial data for each phase
  - ID-system recognising phase of each particle (use of *mk* values)



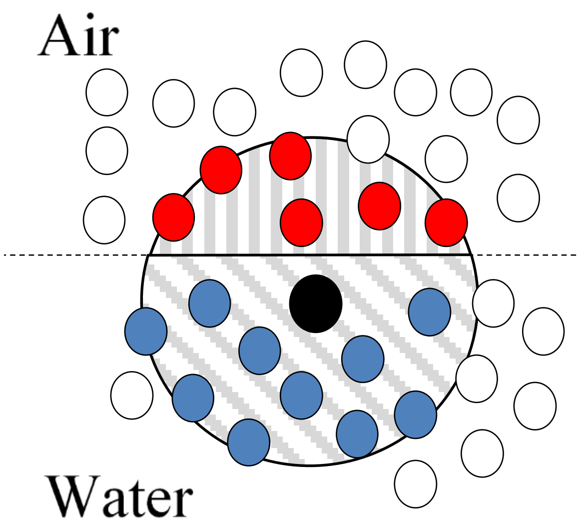
- Optimise the multi-phase model
  - Different equations used for each phase

- Maintain the existing structure of DualSPHysics
  - Integration with other capabilities of the code, such as motion
  - Maintain the efficient cell-linked-list structure

Optimisation of SPH on GPUs

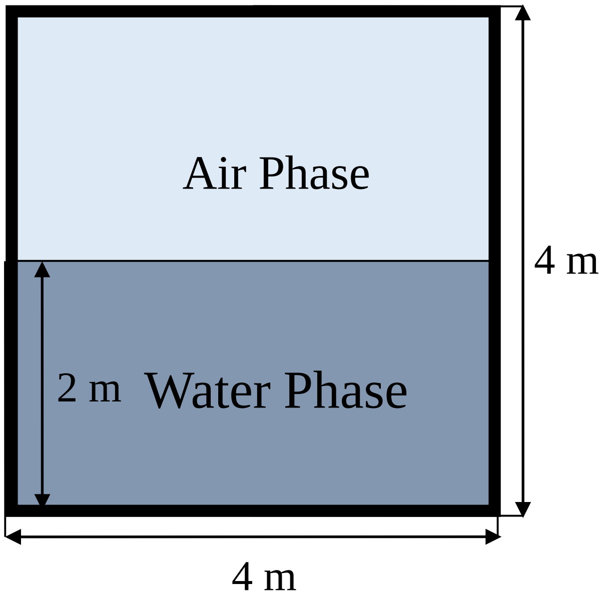
- Calculating inter-particle forces is the most demanding part of SPH

- Research has shown the best practices for optimising:
  - Eliminate conditional *if* statements
  - Reduction of logical operations
  - Minimise CPU-GPU interaction
  - Minimise memory (local and global) transfers
  - Balance computational load on the GPU
  - Separate particle and neighbour lists for each phase is beneficial for large particle numbers

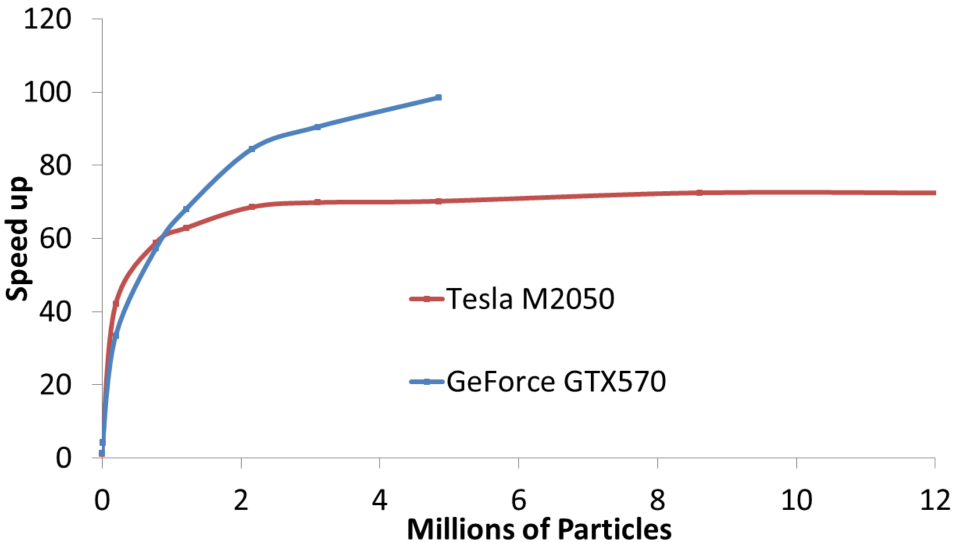


(Mokos et al. 2015)

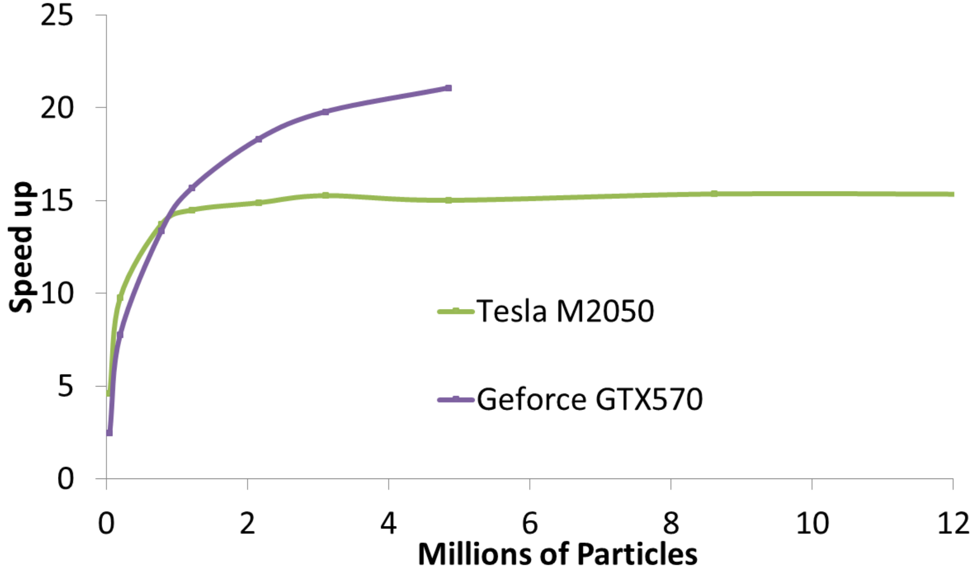
Runtime Results – 2D



Single Thread

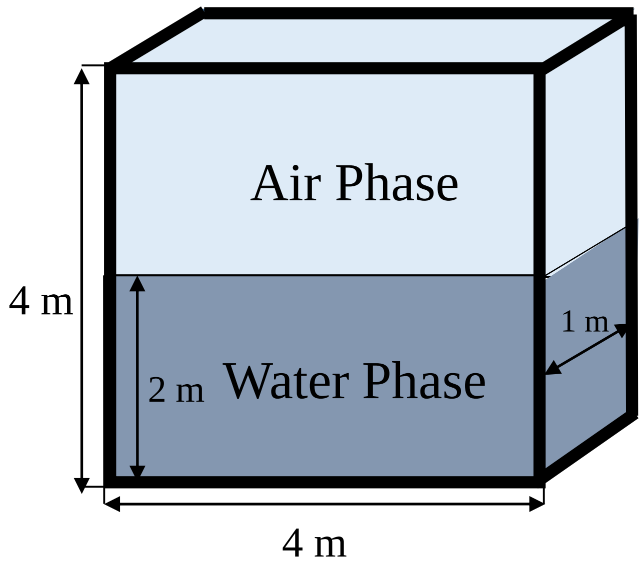


OpenMP

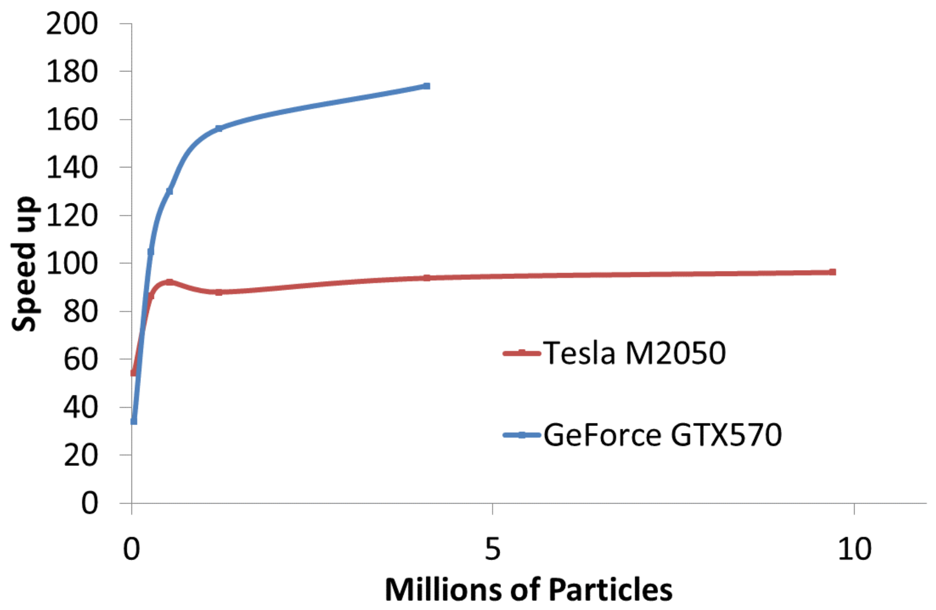


- Results for version 2
- Speed-up is GPU-dependent
- Speedup up to **100** compared to single CPU core
- Speedup up to **21** compared to an 8-thread OpenMP computation

Runtime Results – 3D

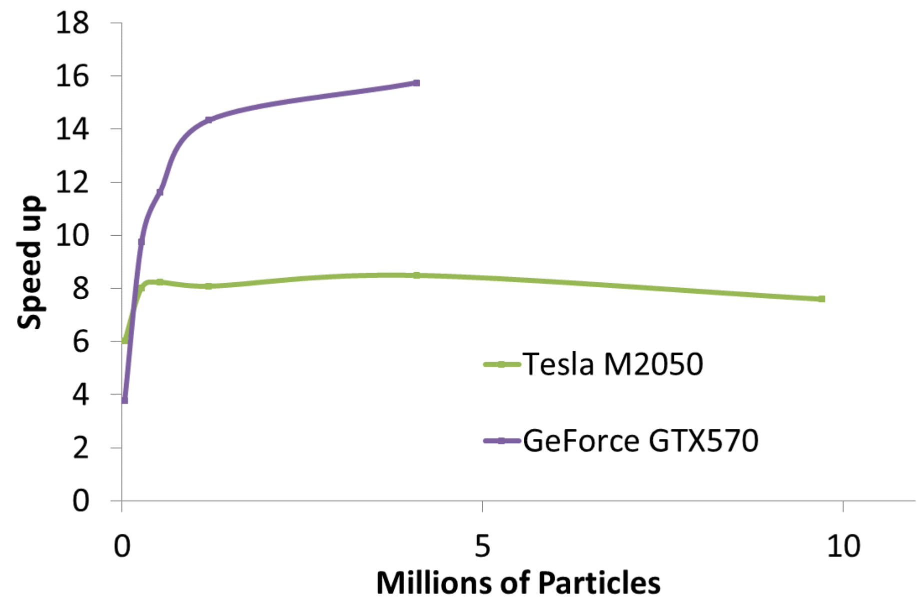


Single Thread

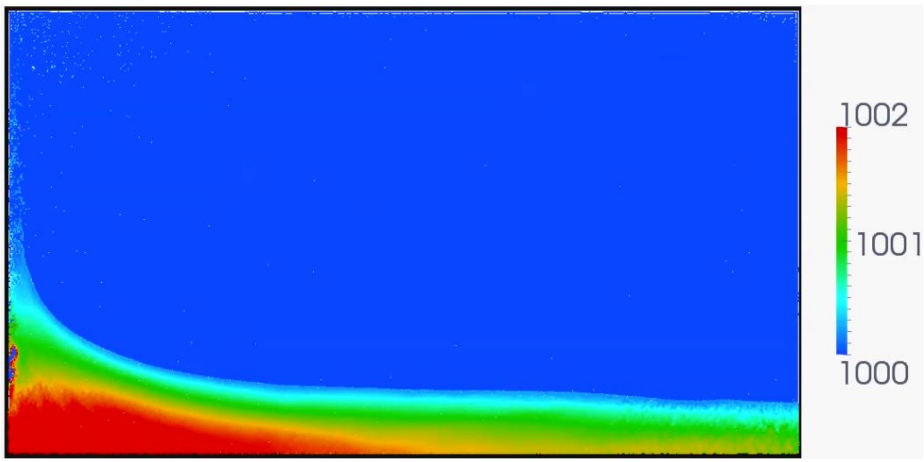


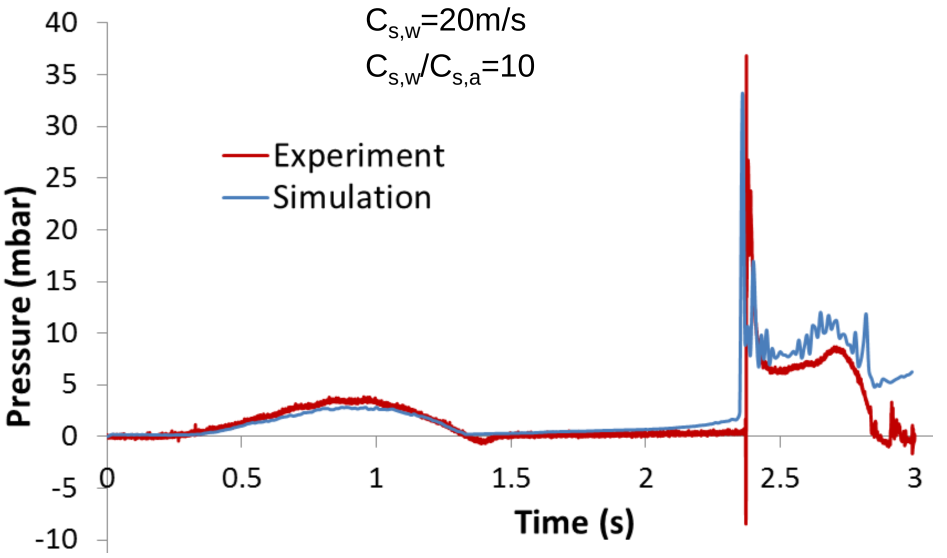
- Results for version 2
- Speed-up is GPU-dependent
- Speedup up to 170 compared to single CPU core
- Speedup up to 16 compared to an 8-thread OpenMP computation

OpenMP

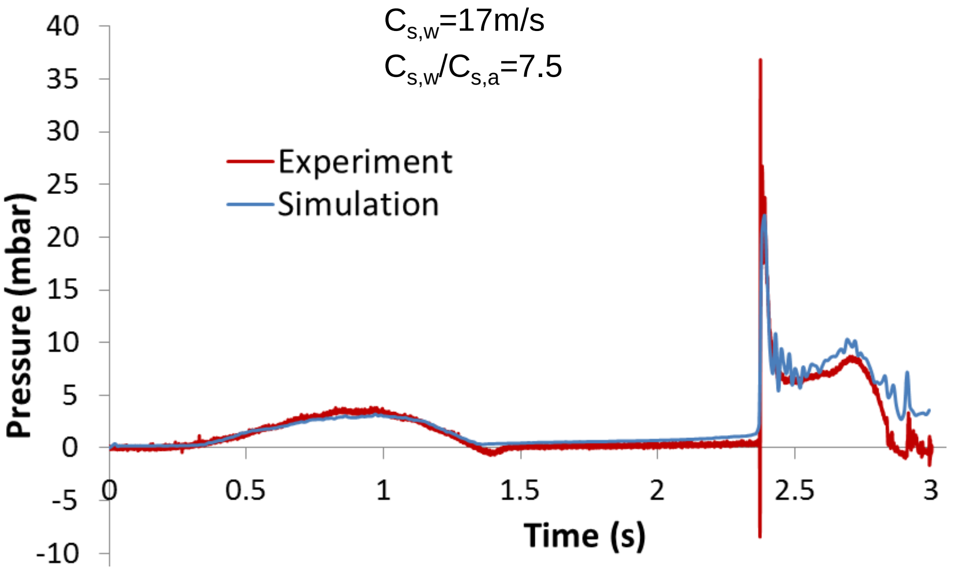


New Cases: Sloshing Tank

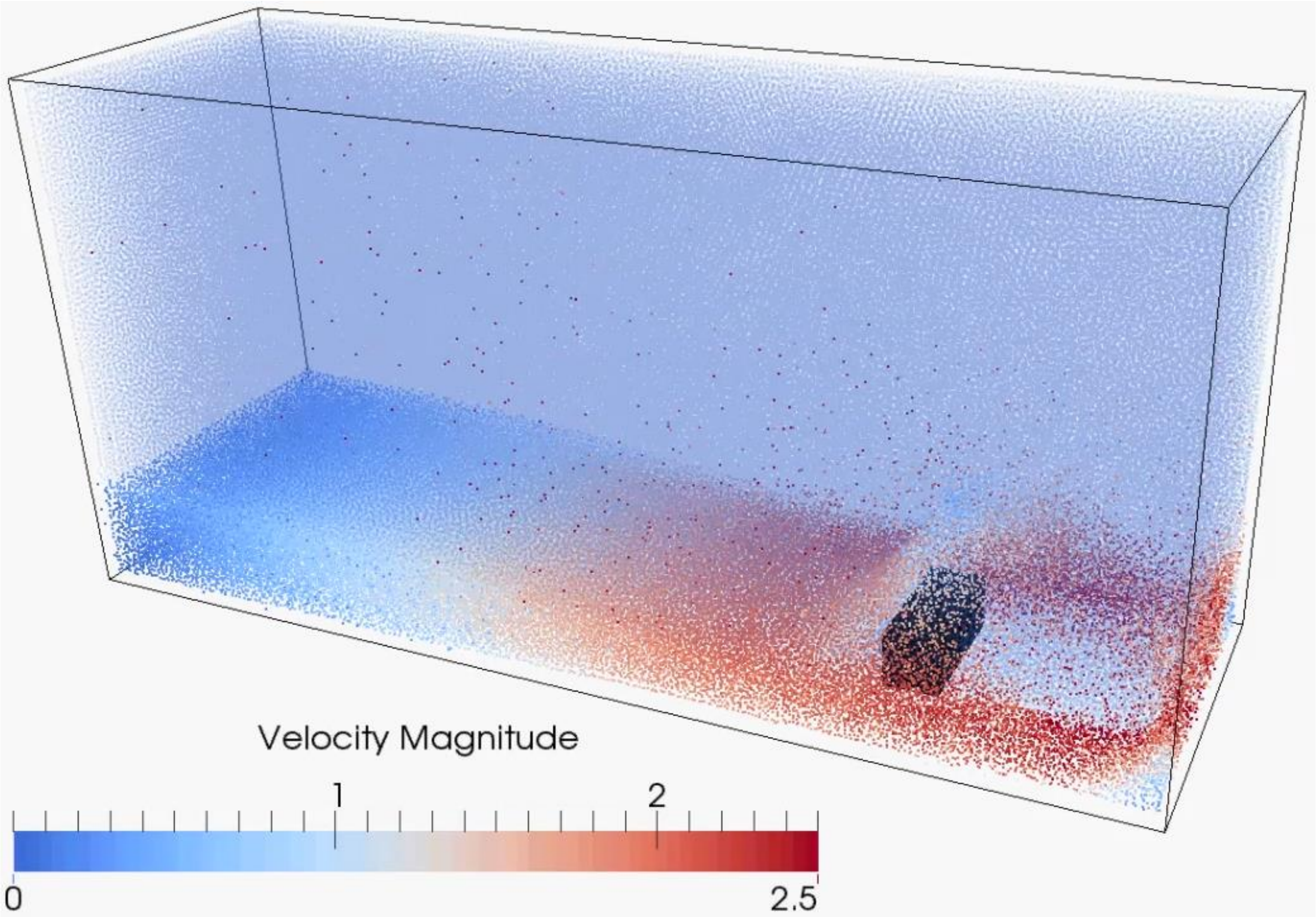




- Smooth interface maintained
- Good prediction of the pressure
- Value of the pressure peak depends:
    - Speed of sound for the water
    - Speed of sound ratio
    - Artificial viscosity coefficient



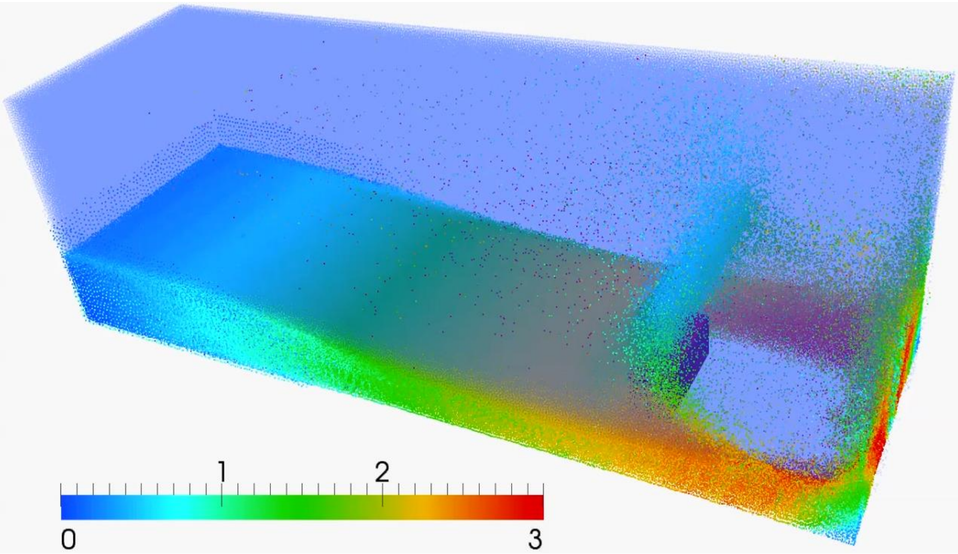
3D Dam break



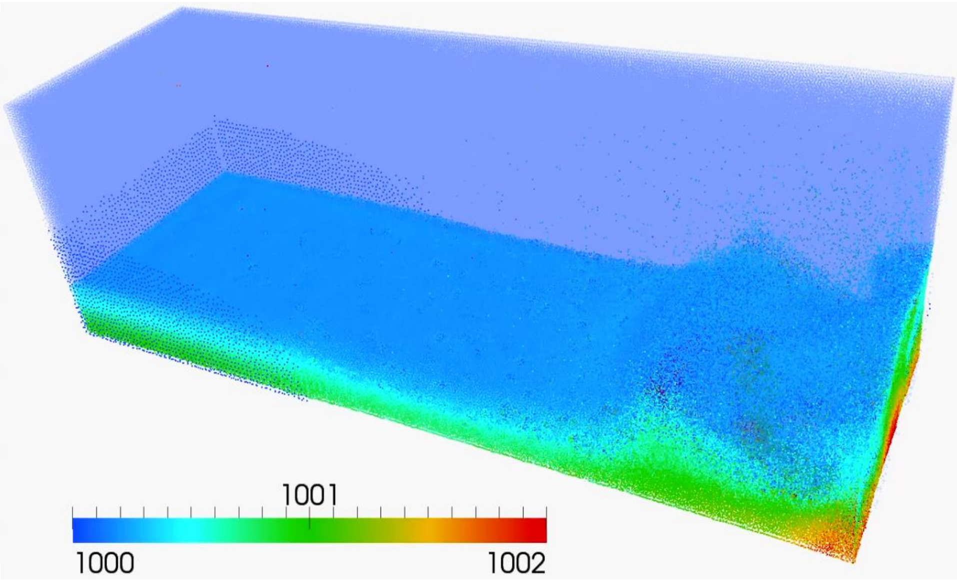
- 670,000 particles: 87,000 water particles / 535,000 air particles
- Simulation runtime: 35 hours on a 5-year old card for 8s

3D Dam break

Velocity



Water Density



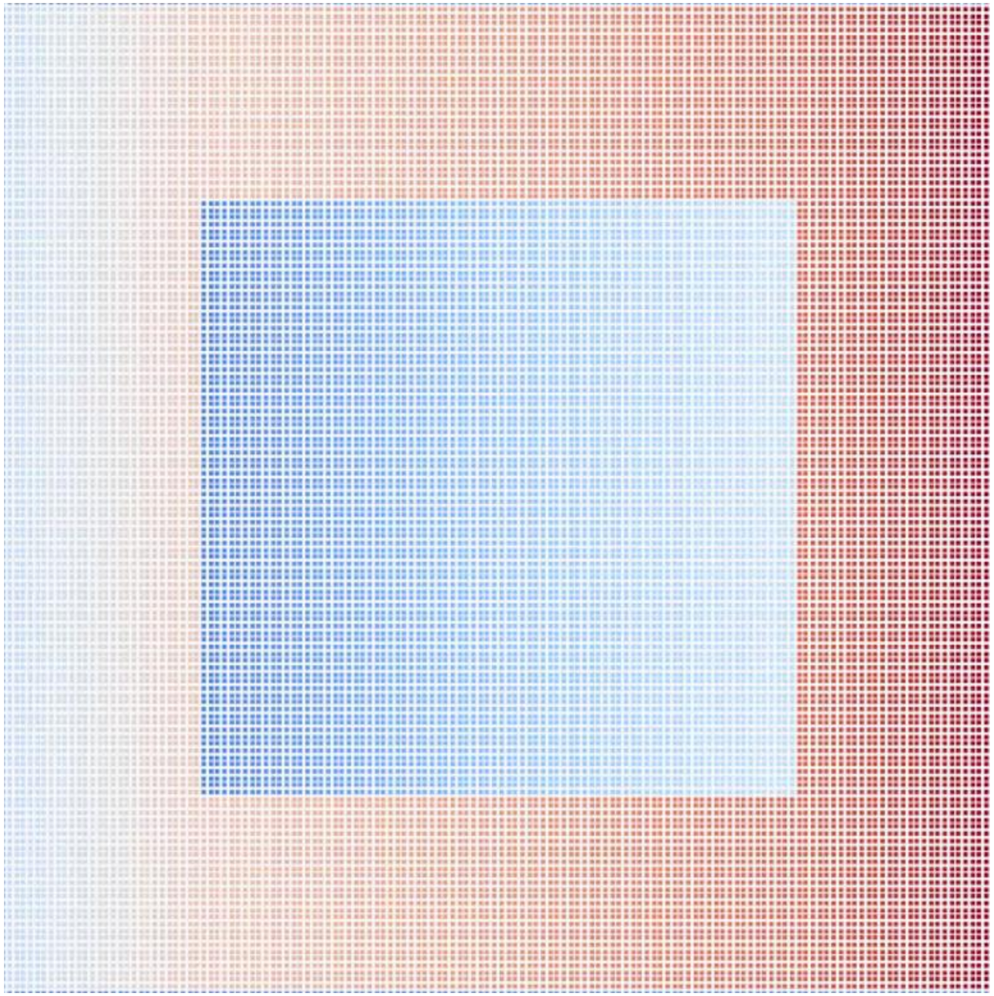
- Simulation of 5 million particles
- Simulation runtime: 140 hours for 3s
- Equivalent in OpenMP DualSPHysics: 1120 hours or 47 days

Surface Tension

- Design of DualSPHysics allows for easy modifications and additions to the code

- Surface tension based on the CSF model (Hu & Adams 2006)

- Uses a colour index function

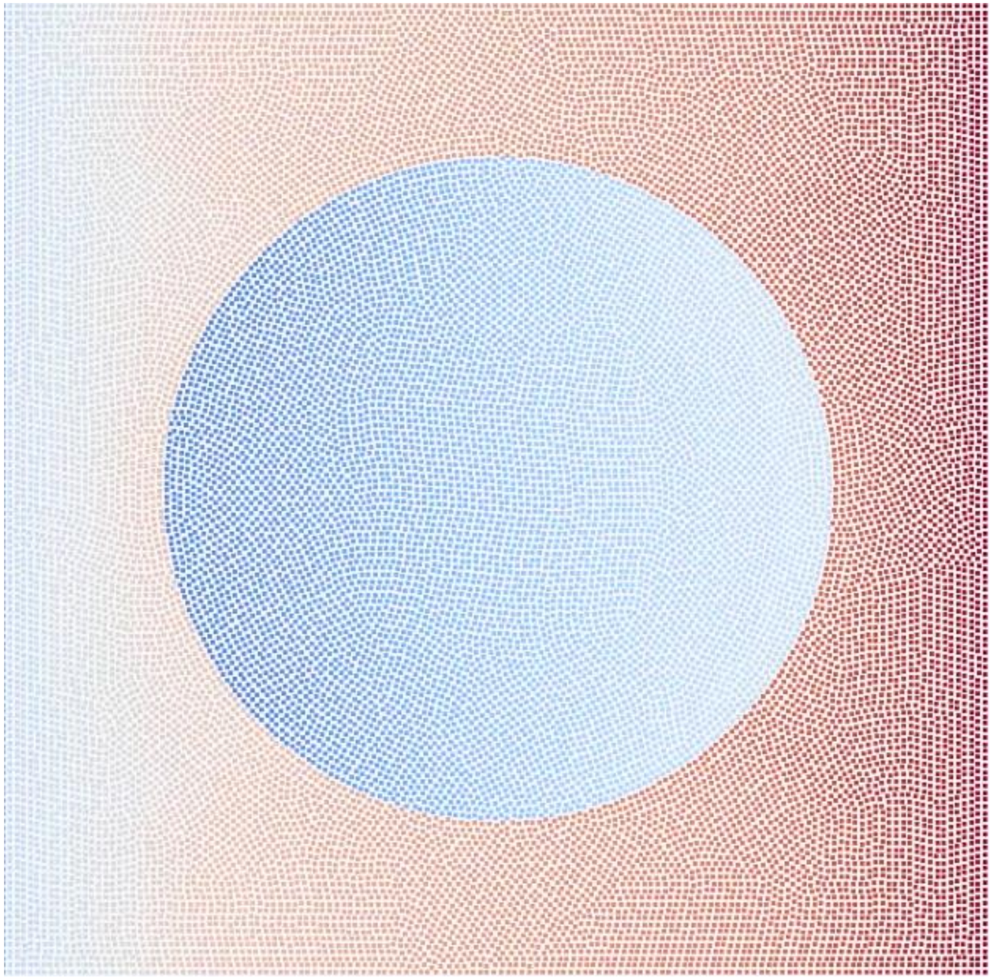


Surface Tension

- Design of DualSPHysics allows for easy modifications and additions to the code

- Surface tension based on the CSF model (Hu & Adams 2006)

- Uses a colour index function

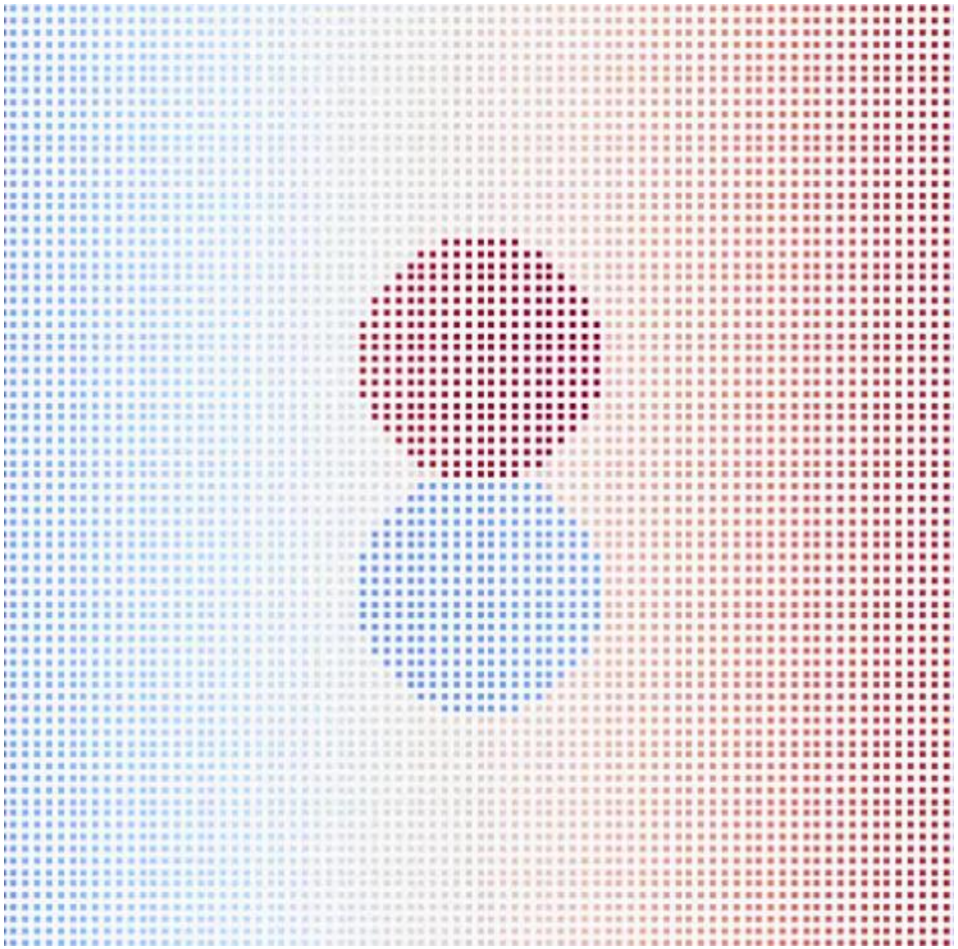


Surface Tension

- Design of DualSPHysics allows for easy modifications and additions to the code

- Surface tension based on the CSF model (Hu & Adams 2006)

- Uses a colour index function

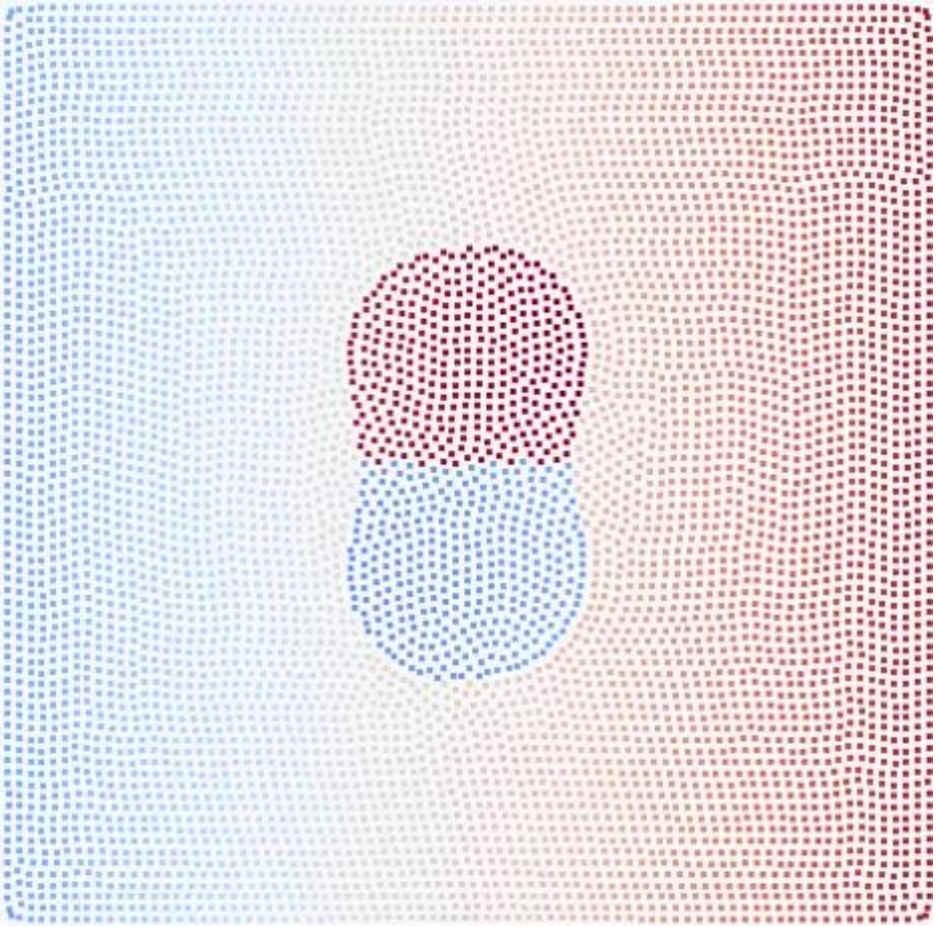


Surface Tension

- Design of DualSPHysics allows for easy modifications and additions to the code

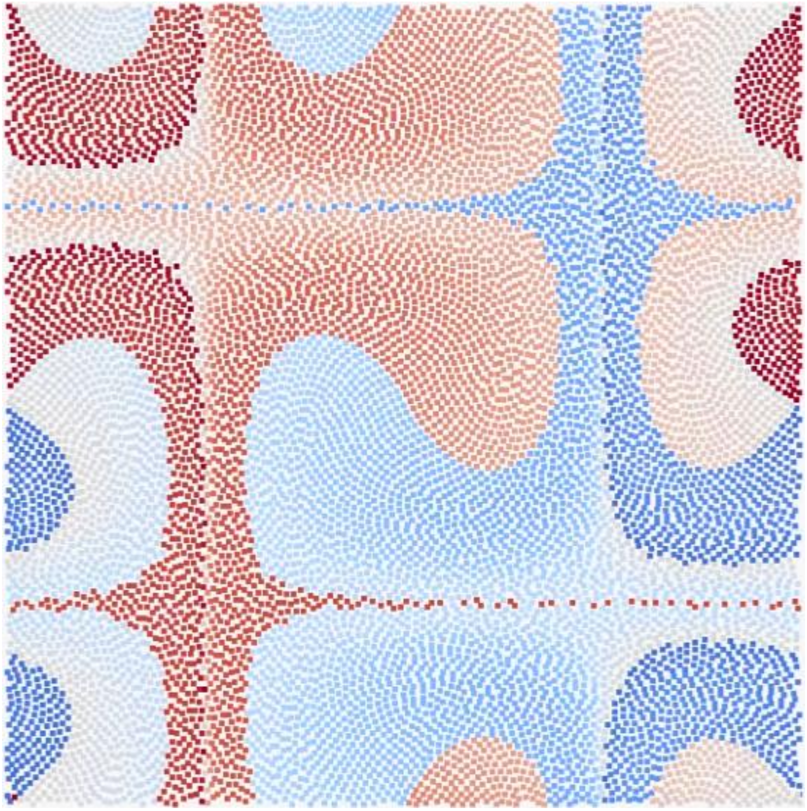
- Surface tension based on the CSF model (Hu & Adams 2006)

- Uses a colour index function



Future Work

- Use the code for less violent flows
- Implement variable particle resolution and multi-GPU support
- Comment the multi-phase code and document changes from original DualSPHysics code
- Release a validated open version of the code



Thank you

Acknowledgments

- U-Man: George Fourtakas, Peter Stansby, Steve Lind, Antonios Xenakis, Stephen Longshaw
- U-Vigo: Jose Dominguez, Alex Crespo, Moncho Gomez-Gesteira
- U-Parma: Renato Vacondio

Websites

- Free open-source **DualSPHysics**

code:

http://www.dual.sphysics.org