1st DualSPHysics users workshop

Current Developments:
Variable Resolution - Boundary Conditions

R. Vacondio, A.J.C. Crespo

UNIVERSITÀ DEGLI STUDI DI PARMA

Universidade de Vigo

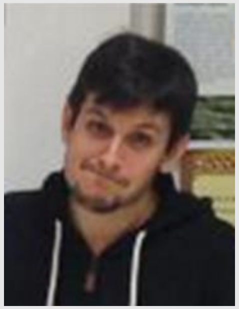
Acknowledgements

Ben Rogers, George Fourtakas, Peter K. Stansby, Paolo Mignosa, Jose M. Domínguez, M. Gómez-Gesteira, Jose L. Cercós-Pita





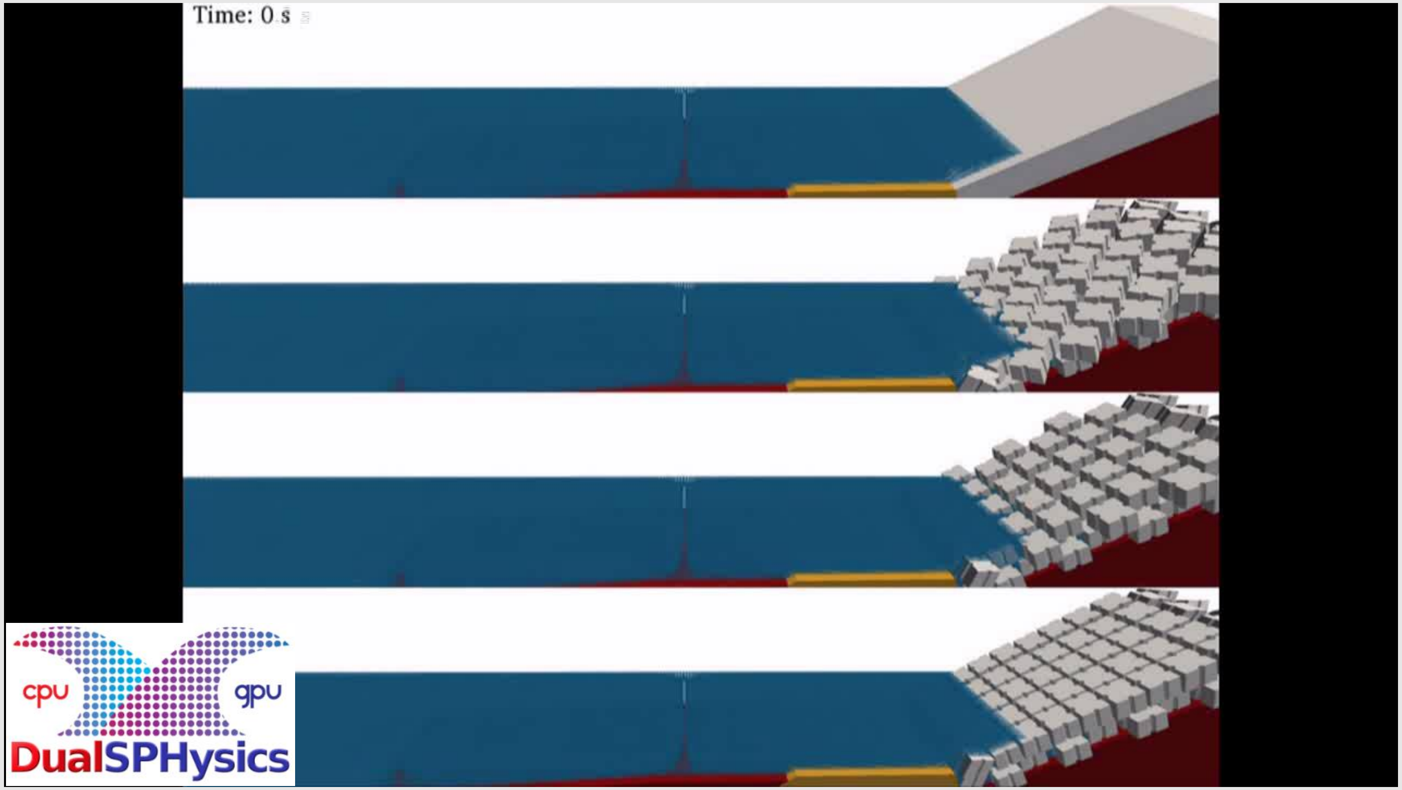




Part 1: Variable resolution

(SPHERIC Grand Challenge #4)

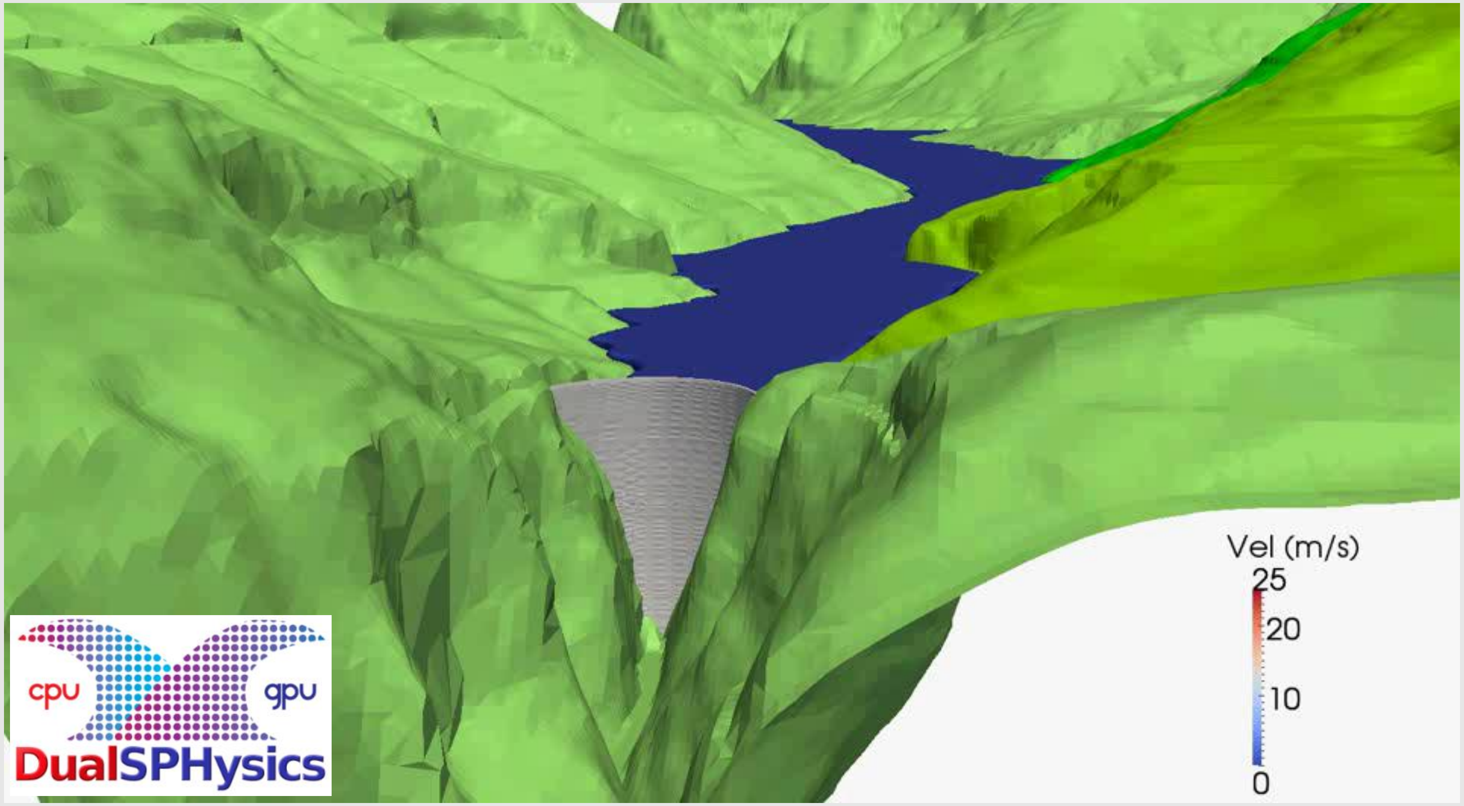
Computational time for SPH with uniform resolution (1)



Rubble mound Breakwater: On a GPU GTX 480 the simulation with 2 x 10^6 particles (Δx=0.15 m) requires 21 hours of computational time for 55 seconds of physical time

Altomare et al. Computers & Structures (2014)

Computational time for SPH with uniform resolution (2)



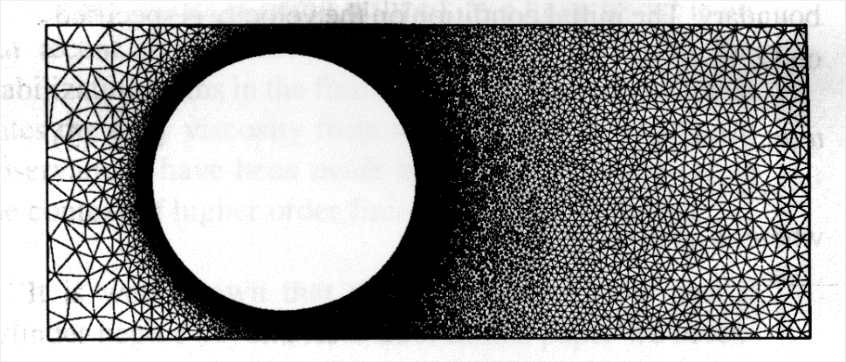
Vajont rockslide: On a GPU GTX 580, 4 x 10^6 particles (Δx=5 m) requires 62 hours of computational time for 21 min of physical time

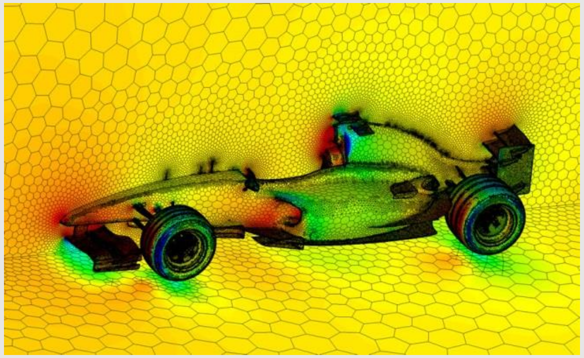
Vacondio et al. Advances in Water Resources (2013)

Motivation

- Despite the huge effort in parallelizing SPH codes (MPI+CUDA) long runtimes are still an issue.

- In Eulerian models variable resolution achieved using (dynamically adaptive) unstructured meshes:





Due to its Lagrangian nature, this is more challenging for SPH:

SPHERIC Grand Challenge #4: Can we achieve the same efficiency in SPH ?

Previous works about variable resolution in SPH :

Remeshing:

- *Koumoutsakos Ann. Rev. Fluid Mech. (2005)*: applied remeshing idea to SPH
- Multiblock space discretization: *Børve et al. JCP (2005)*

Static refinement:

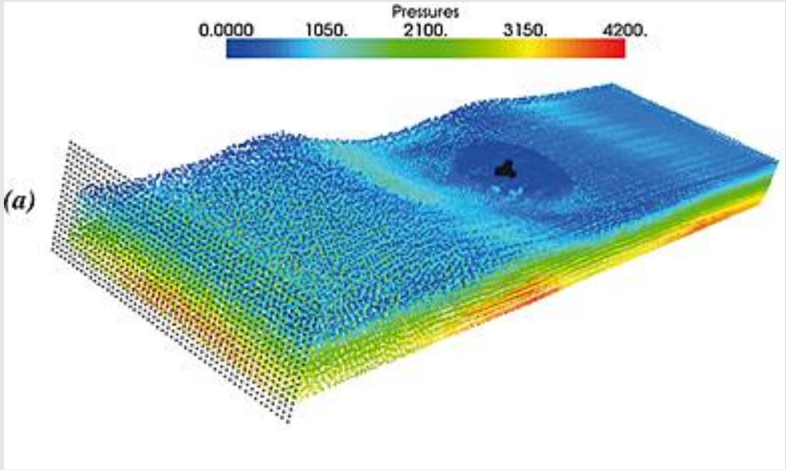
- Different initial resolution zones, no splitting: *Oger et al. JCP (2006)* and *Omidvar et al. IJNMF (2012).*

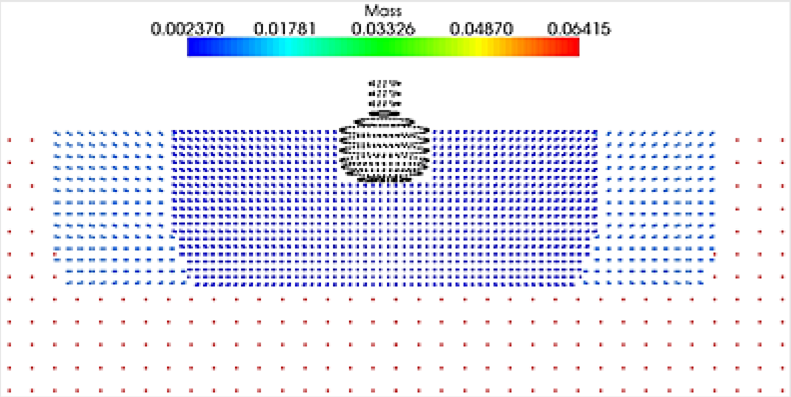
Dynamic refinement:

- Particle insertion and removal in 1D: *Lastiwka et al. IJNMF (2005)*
- Particle splitting: *Feldman and Bonet IJNME (2007), Lopez et al. Comput Mech (2013),*
- Splitting and coalescing: *Barcarolo et al. JCP (2014), Spreng et al. Comp. Part. Mech. (2014)*

Static particle distribution with different mass

- 3D simulations of energy device under extreme wave conditions





Numerical model	*Uniform particle distrib*	*Variable mass ratio 1:8*
# of particles	**918’000**	**139’000**
Computational time	7 days	1.5 days
Δx max (m)	0.02	0.04
Δx min (m)	0.02	0.02

Omidvar et al. IJNMF (2012)

WC-SPH variationally consistent scheme with variable *h*

WC-SPH formulation

Vacondio et al. 2013 CMAME:

- It is variationally derived
- It conserves both mass and momentum

$$\frac{d\rho_i}{dt} = \sum_j m_j \left(\mathbf{u}_i - \mathbf{u}_j\right) \cdot \nabla W_j(\mathbf{x}_i, h_j) + 2\delta h_i \sum_j m_j \bar{c}_{ij} \left(\frac{\rho_i}{\rho_j} - 1\right) \frac{\mathbf{r}_{ij}}{\mathbf{r}_{ij}^2 + \eta^2} \cdot \nabla W_j\left(\mathbf{x}_i, h_j\right)$$

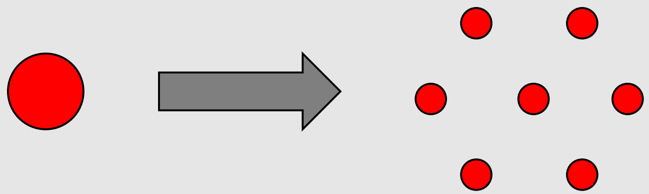
$$\frac{d\mathbf{v}_i}{dt} = \sum_j \frac{m_j}{\rho_j \rho_i}\left[p_i \nabla W_j\left(x_i, h_j\right) - p_j \nabla W_i\left(x_j, h_i\right)\right] + \sum_j \frac{m_j}{\rho_j \rho_i}\left(\Pi_{ij} \cdot \nabla W_j(\mathbf{x}_i, h_j)\right) + \mathbf{g}$$

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{v}_i$$

$$p = B\left[\left(\frac{\rho}{\rho_o}\right)^{\gamma} + 1\right]$$

Time integration with Simpletic scheme, Wendland kernel, δ – SPH

Increasing the resolution: particle splitting



Splitting procedure (1)

Key idea: split one particle into M daughter particles.

- **Mass, position, velocity, density and smoothing length** must be defined for each daughter particle
- Mass, momentum and energy conservation should be enforced

⇩

- Number of daughter particles: ideal numbers is 4 in 2D (it doubles the resolution) but it is not very convenient (see later)
- to reduce the degrees of freedom: we defined a priori the stencil and the smoothing length of the daughter particles
- the mass distribution of the daughter particles is obtained by minimizing the density error

Feldman and Bonet IJNME (2007), Vacondio et al. IJNMF (2012)

Splitting procedure (2)

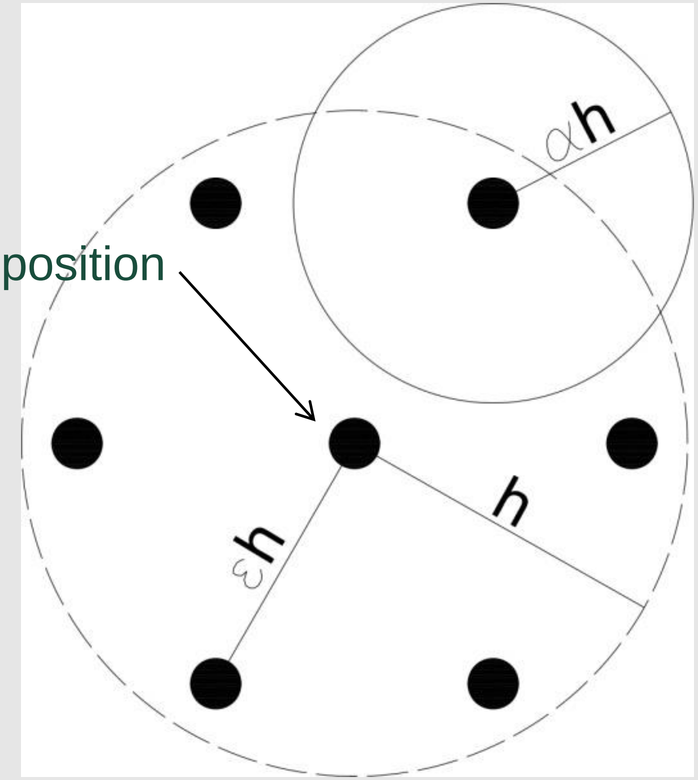
Particle positions and smoothing length are defined as.

$$h_k = \alpha h$$

$$d_k = \varepsilon h$$

$$m_k = \lambda_k m$$

Original particle position



Where α and ε are parameters define Particle position and smoothing length

Feldman and Bonet IJNME (2007), Vacondio et al. IJNMF (2011)

Density error minimization

Local density error: $$e(\mathbf{x}) = \rho(\mathbf{x}) - \rho^*(\mathbf{x}) = m_N W_N(x, h_N) - \sum_{k=1}^{M} m_k^* W_k(x, h_k)$$

Global density error: $$\mathrm{E} = \int_\Omega e(\mathbf{x})^2 \, d\mathbf{x}$$

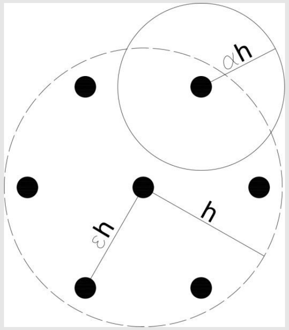
After some algebra … the best mass distribution is calculated as follows:

$$\mathrm{E}^* = \min_{\boldsymbol{\lambda}} \left\{ \overline{C} - 2\boldsymbol{\lambda}^T \overline{\mathbf{b}} + \boldsymbol{\lambda}^T \overline{\mathbf{Q}} \boldsymbol{\lambda} \right\} \qquad \lambda_k = \frac{m_k}{m_N}$$

With constraint for mass conservation: $$\sum_{j=1}^{M} \lambda_j = 1$$

State of the art of splitting in SPH

To dynamically vary the resolution in 2D: splitting and coalescing procedures are available



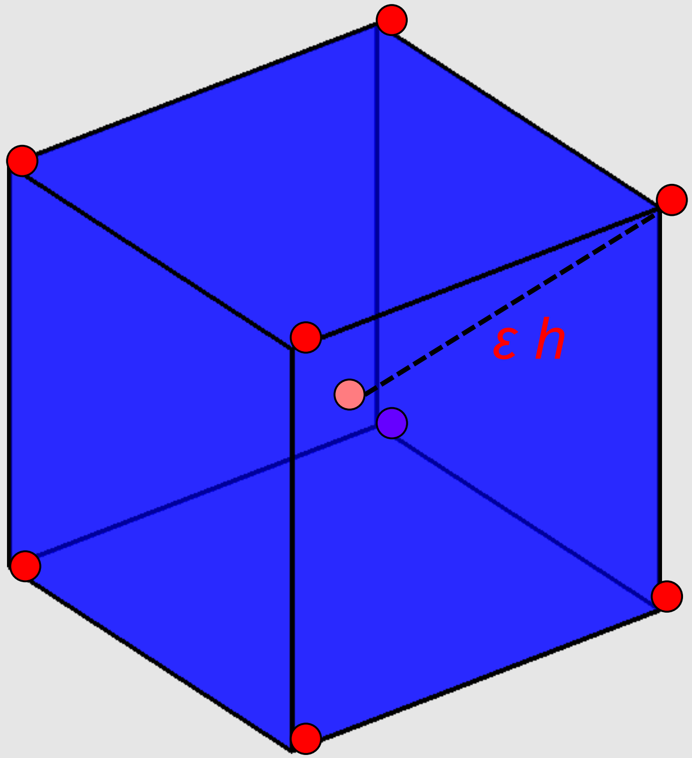
in 3D no literature available on splitting

?

Naïve approach: the Cube

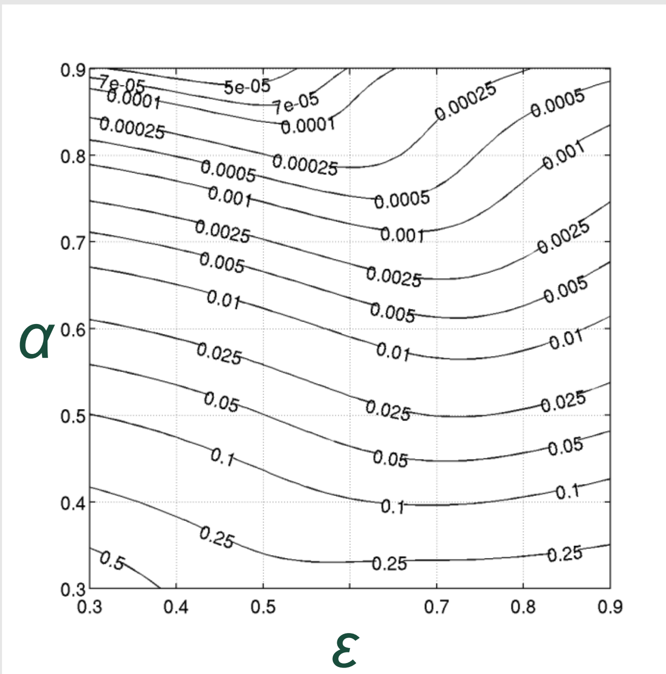
- One particle split in 9 Daughter Particles
- First DP in the cube centre (position of the original particle) and 8 particles in the cube vertices
- Wendland kernel
- *ε* and *α* parameters are varied between 0.3 and 0.9 to obtain a global density error matrix:

$$E^{*} = E^{*}(\varepsilon, \alpha)$$

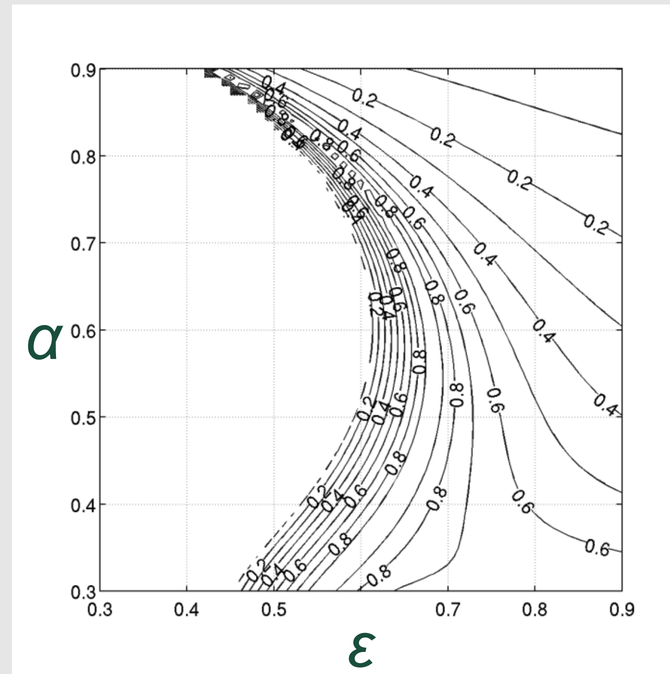


Naïve approach: the Cube (2)

Non-dimensional global density error



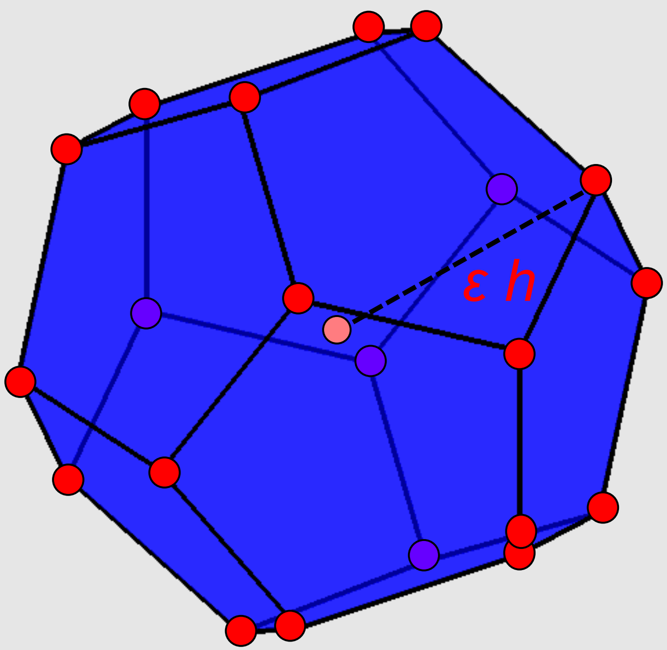
m_{min}/m_{max} ratio



Min. global density error is small, but just for α≈0.9

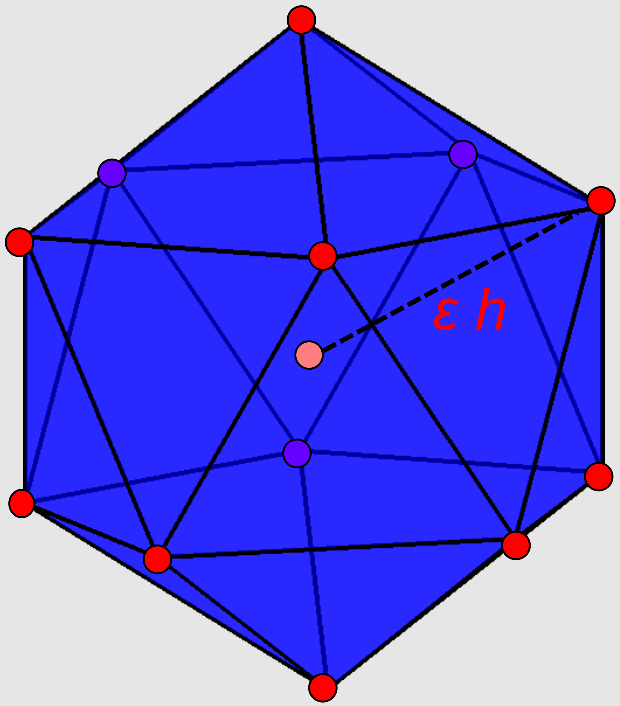
Platonic solids (spherical symmetry):

Dodecahedron



20 vertices

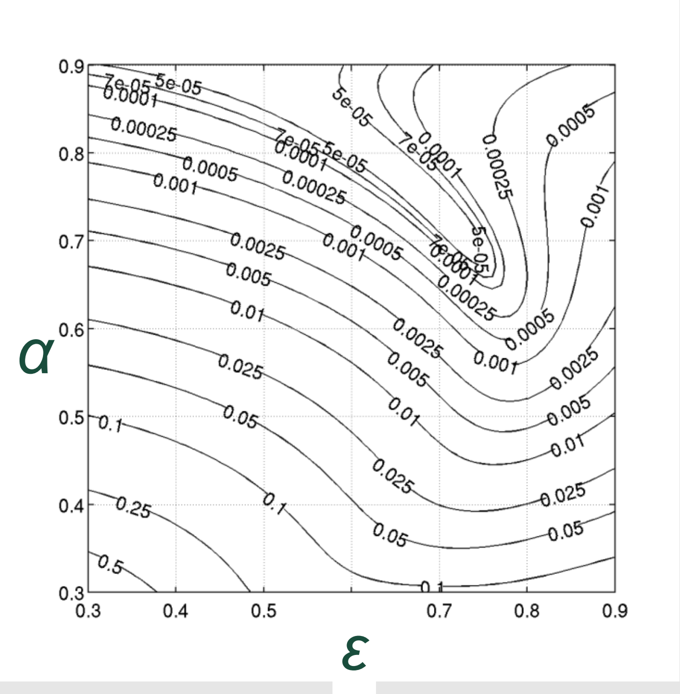
Icosahedron



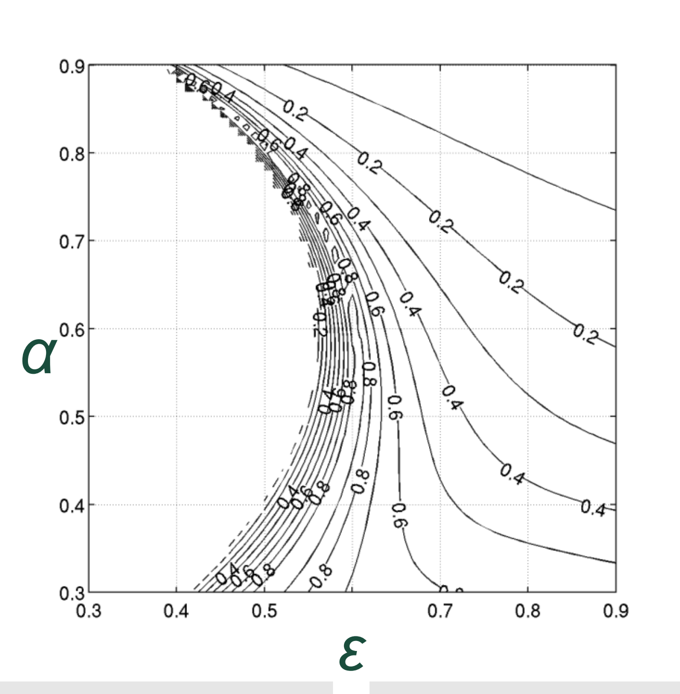
12 vertices

Dodecahedron (20 vertices)

Global density error



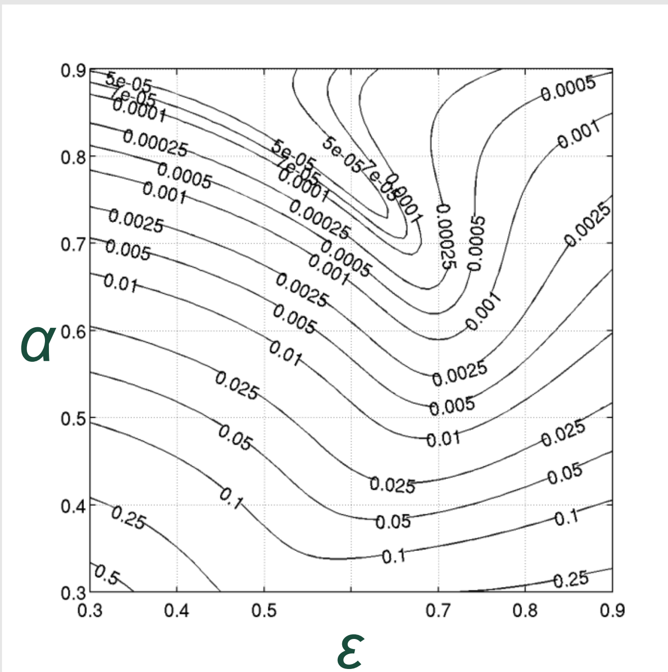
m_{min}/m_{max} ratio



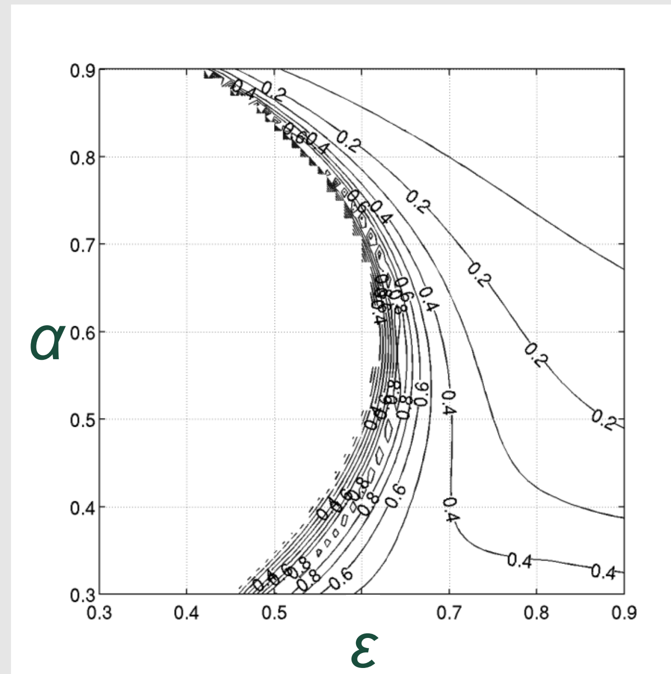
Global density error for given α, ε is smaller than in the cube pattern

Icosahedron (12 vertices)

Global density error

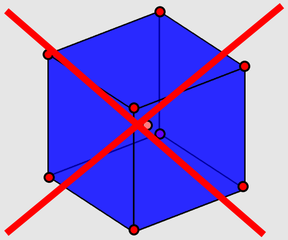


m_{min}/m_{max} ratio



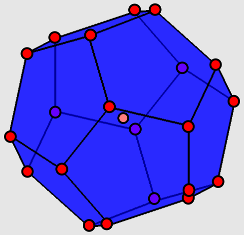
Similar to the Dodecahedron, but with less particles

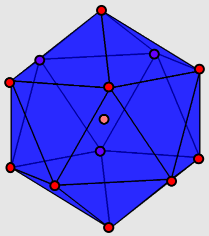
Which is the best stencil?



Cubic is not the best stencil: error small only for h_k=0.9 h_M

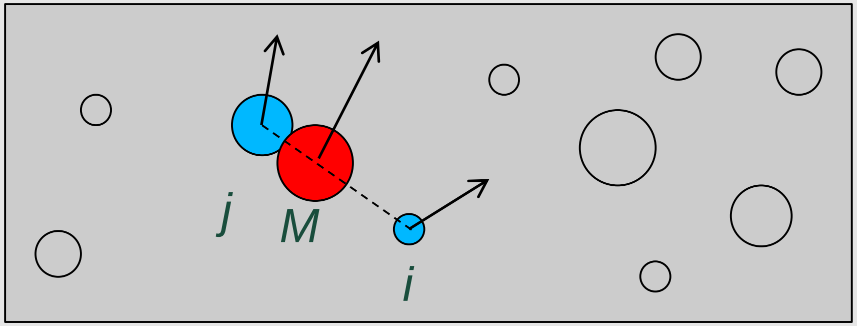
This means a lot of neighbors in the high resolution zone.





The global density error matrix obtained for Icosahedron and Dodecahedron are similar, but the **Icosahedron is more efficient** because it creates less daughter particles (12 vertices instead of 20)

To Reduce the resolution: Particle Coalescing (merging)

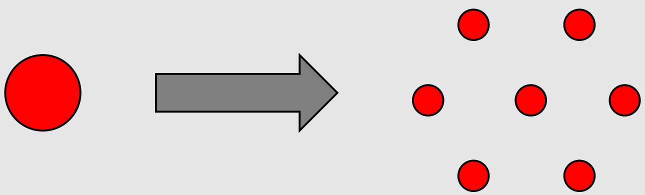


The same algorithm used in 2D and 3D (*Vacondio et al. 2013 CMAME*):

- Particles are coalesced in pairs
- mass and momentum conservation gives mass position and velocity of the new particle *M*
- The smoothing length h_M is obtained by enforcing zero density error.
- No further coalescing is possible for particle M in the same time iteration.

Parallel implementation (CPU & GPU)

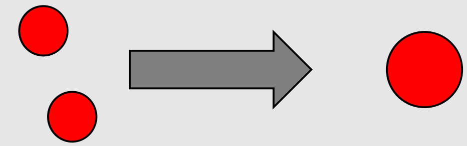
Splitting



list of particles to split

Create daughter particles

Coalescing



list of particles to coalesce

New “bigger” particles are created

Check to delete particles with the same “mother” **(sequential)**

Variable res. formulation overheads: *h* and *m* different for each particle, more memory access, and more floating point operation

Variable resolution
Test cases

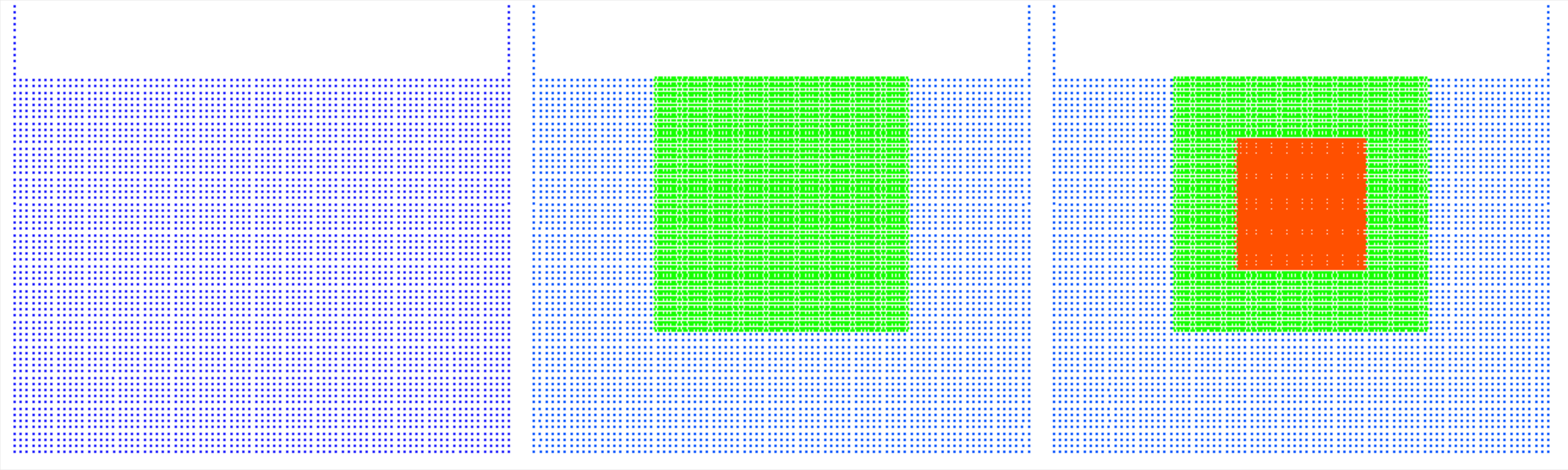
2-D still water tank

Δx_0=0.025 m, (Np=4800)

size of the box 2x1.5 m

Low artificial viscosity: α=0.01

uniform resolution | one level of splitting | two levels of splitting

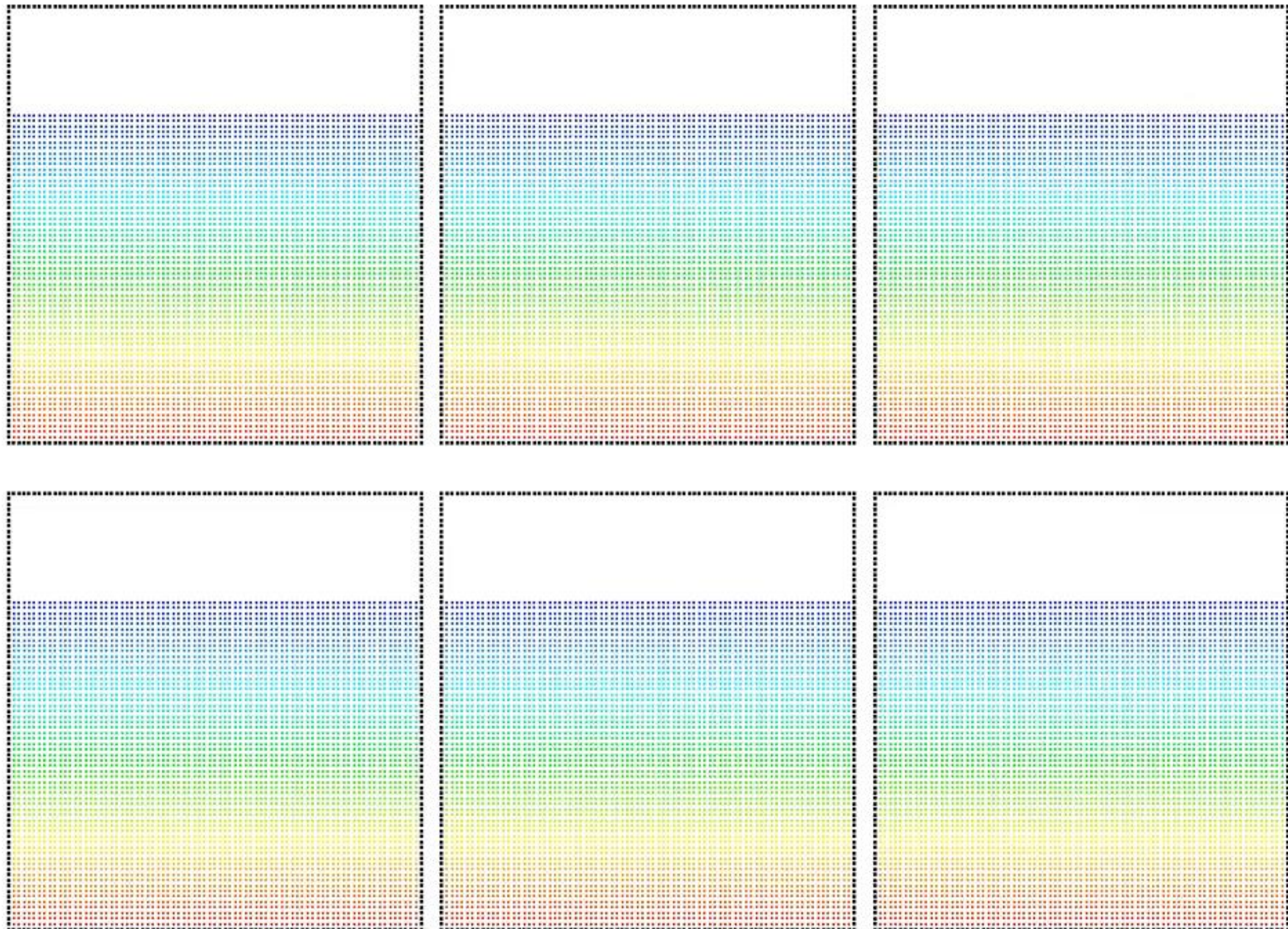


Pressure field

uniform resolution | one level of splitting | two levels of splitting

without δ–SPH

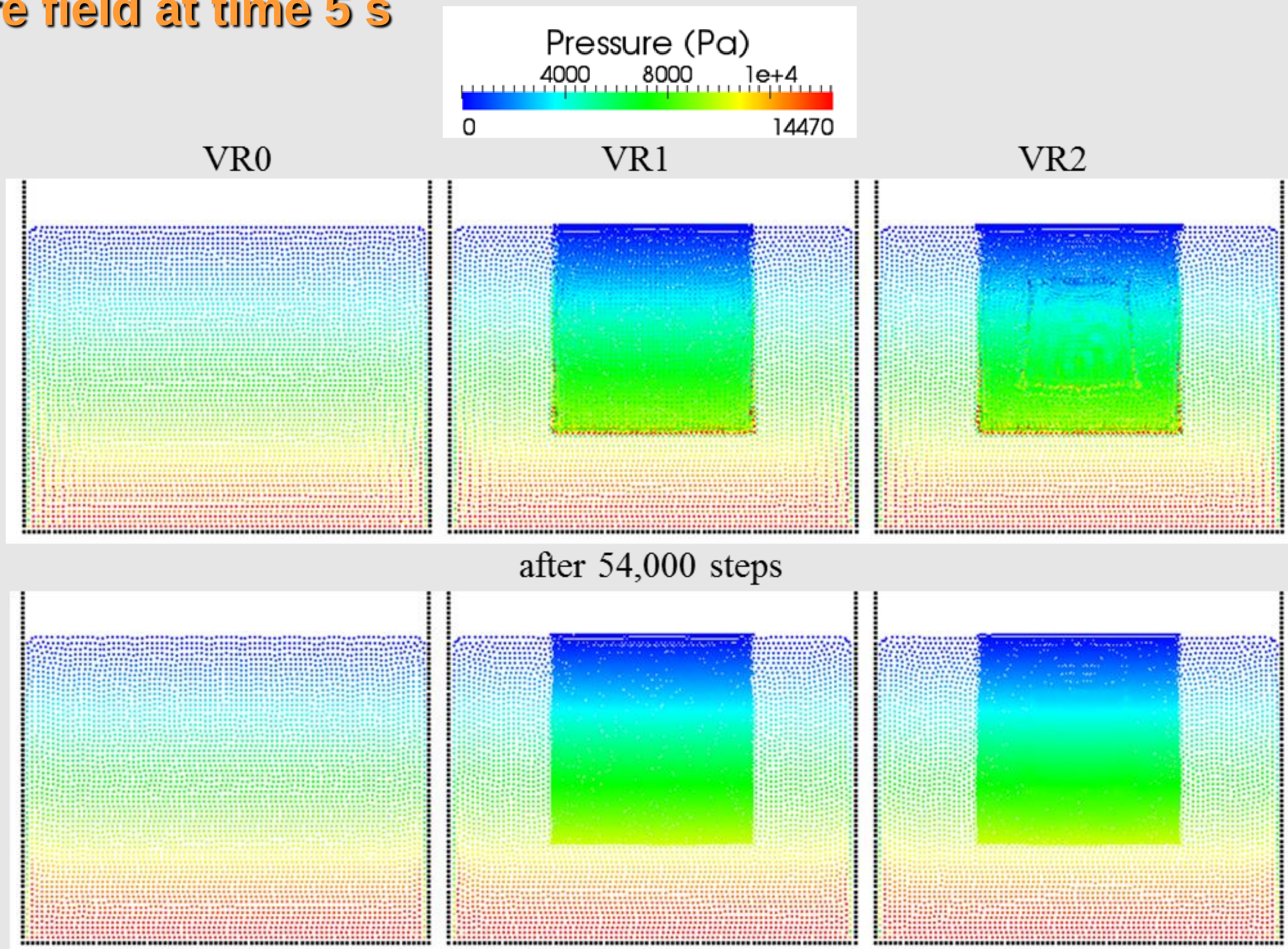
with δ–SPH



Pressure field at time 5 s

without δ–SPH

with δ–SPH



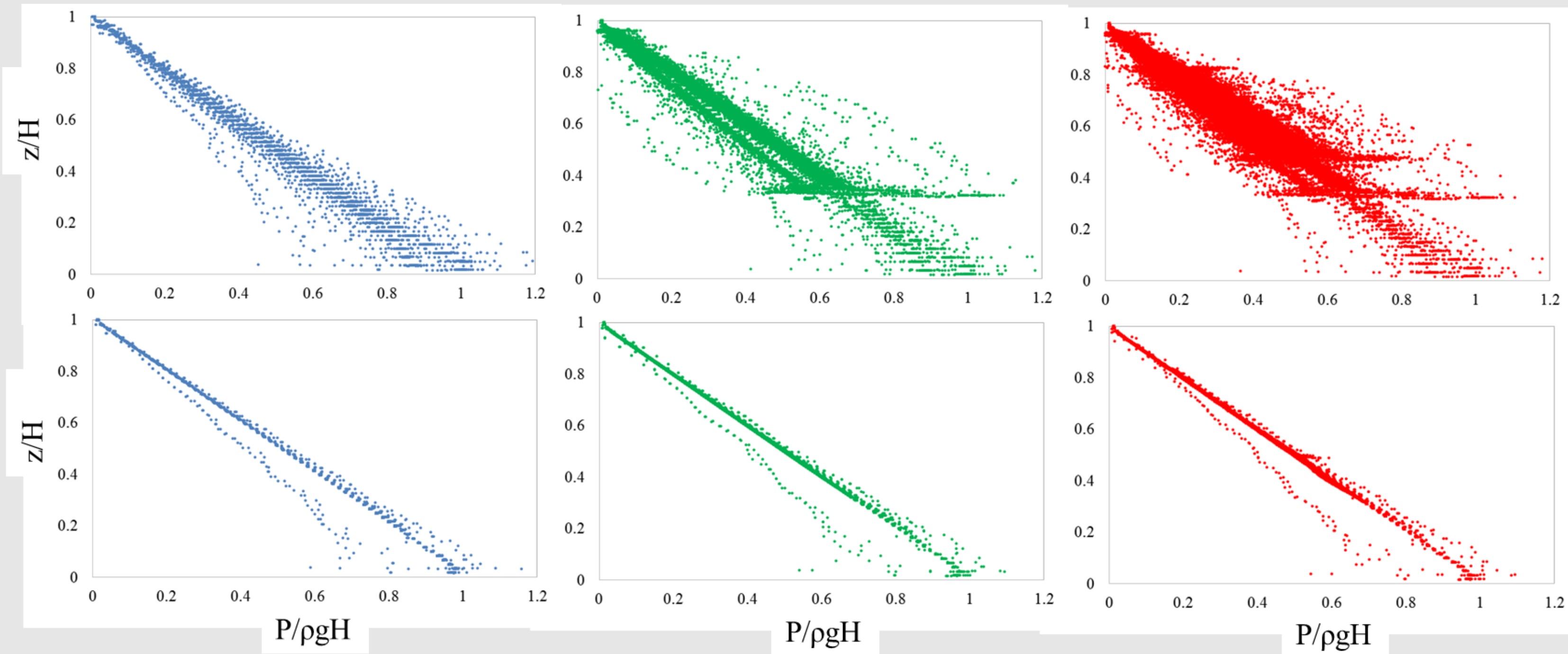
after 54,000 steps WITH deltaSPH

Vertical distribution of pressure at last instant
(t=5s, after 54k steps)

uniform resolution | one level of splitting | two levels of splitting

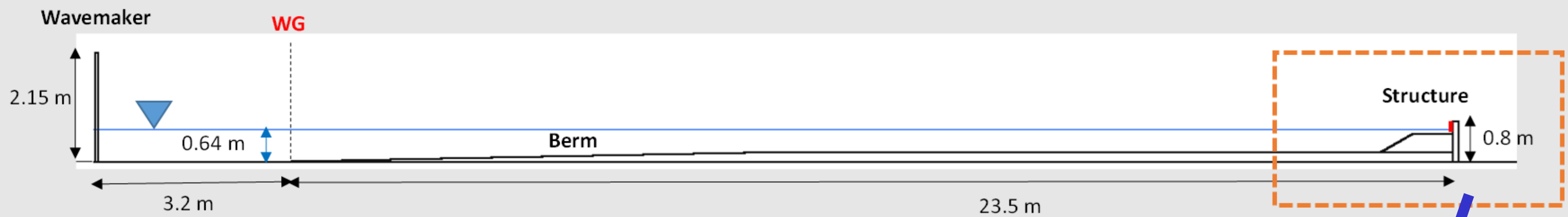
without δ–SPH

with δ–SPH

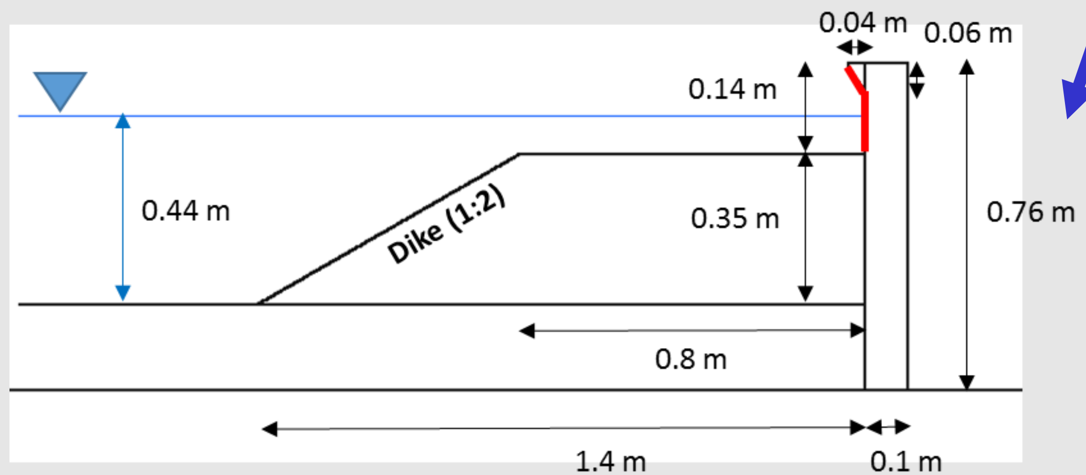


2-D wave experiment

Experimental campaign in Blankenberge Marina, Altomare et al. 2015(+), model scale 1:5



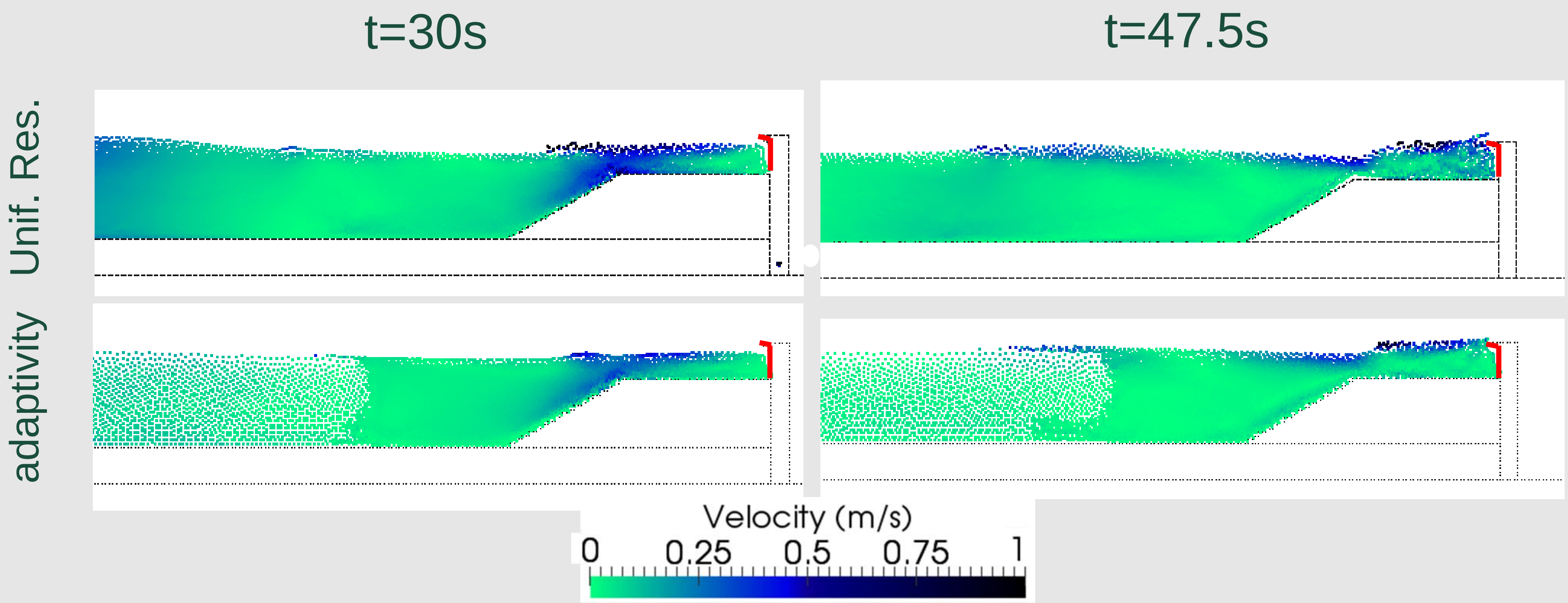
Wave height: 0.101m
Peak Period: 2.683s



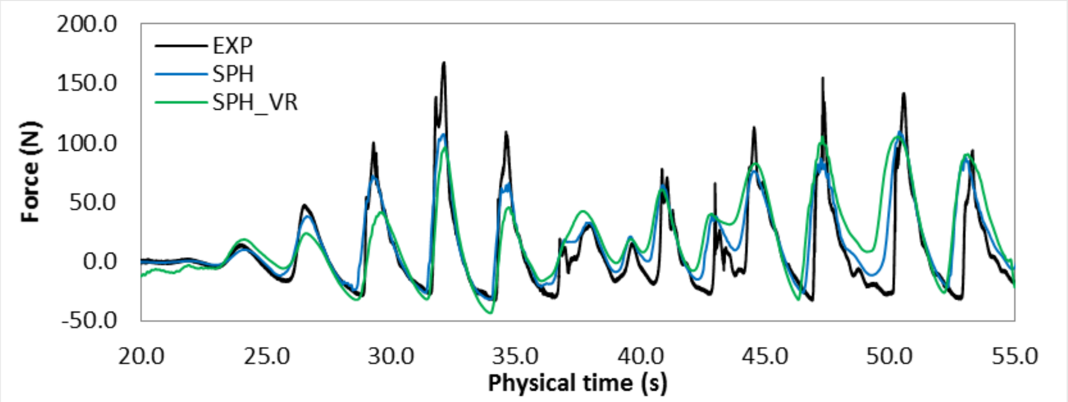
Two simulations:

- high resolution, no adaptivity (with Δx_0= 0.01 m)
- Initial coarse resolution (Δx_0=0.02 m), splitting activated close to the wall.

(+) Altomare et al. Coastal Engineering, 2015



Forces obtained with experiments, SPH and SPH-adaptive



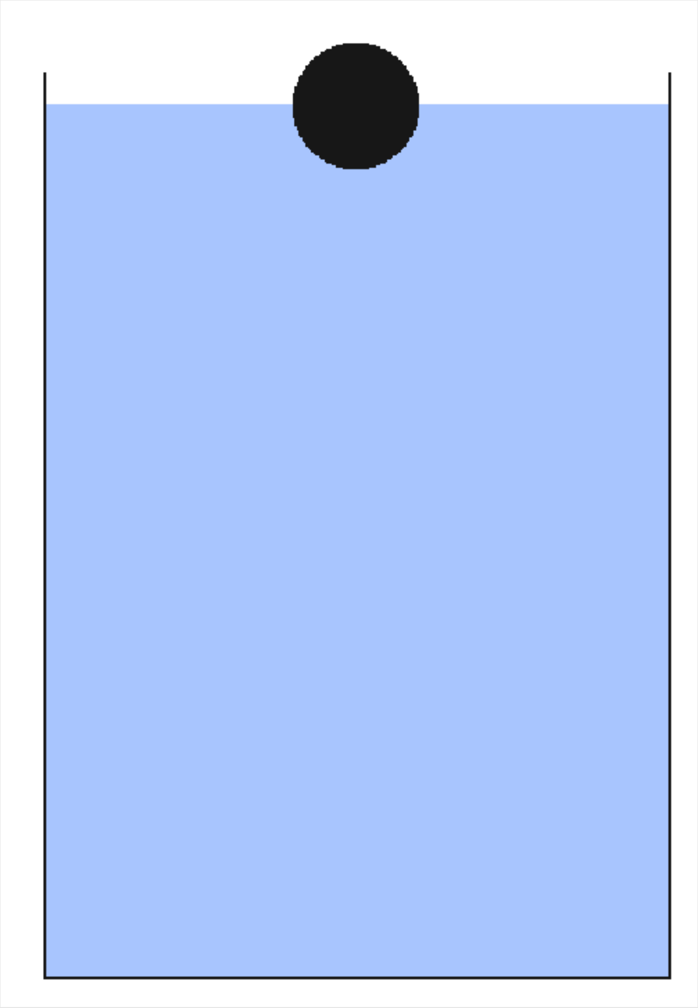
Runtimes

	SPH	SPH-adaptivity
Δx_0	0.01m	0.02m
Initial number of particles	135,883	35,670
Split particles	-	8,670
CPU runtime	*29.82h*	*5.81h*
GPU runtime	*42.40 min*	*18.77 min*

CPU speedup: 5.13 – GPU speedup: 2.26

Adaptivity overheads are more relevant for the GPU code (more registers, non coalesced memory access etc.)

2D-Falling sphere



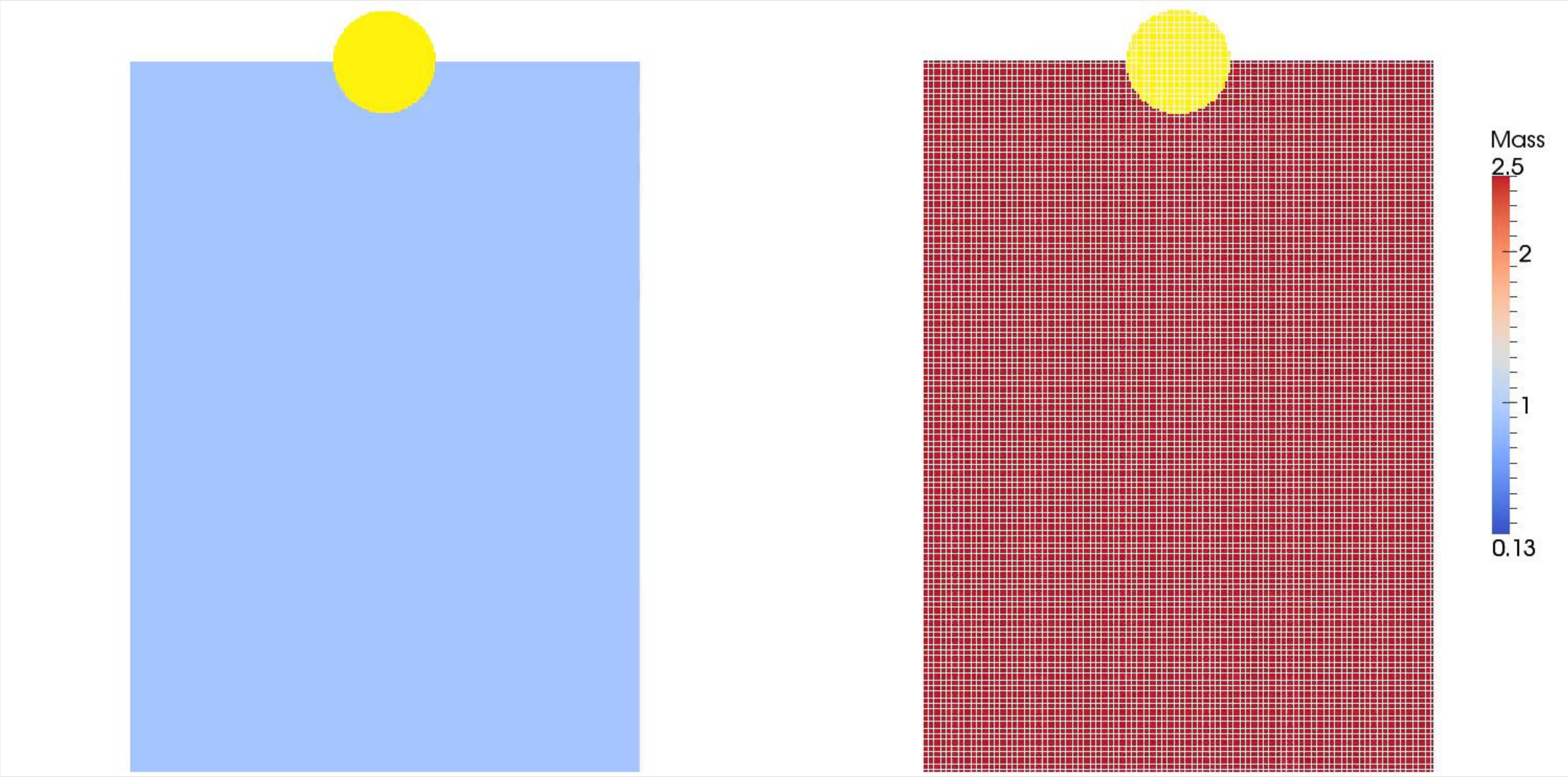
- 2-D sphere with radius=1 m
- density=1,200 kg/m^3
- be compared against VOF: Fekken (2004)
- SPH with Δx_0=0.03 m and no adaptivity
- SPH with Δx_0=0.05 m and **dynamic** adaptivity

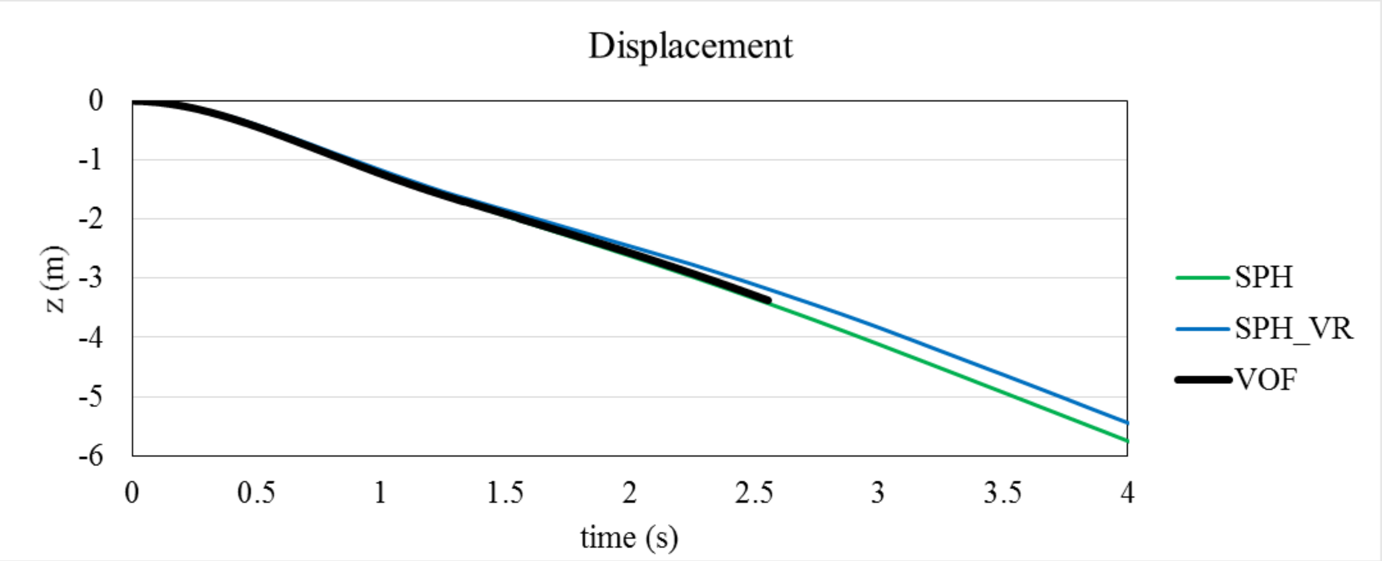
dynamic adaptive region
is used

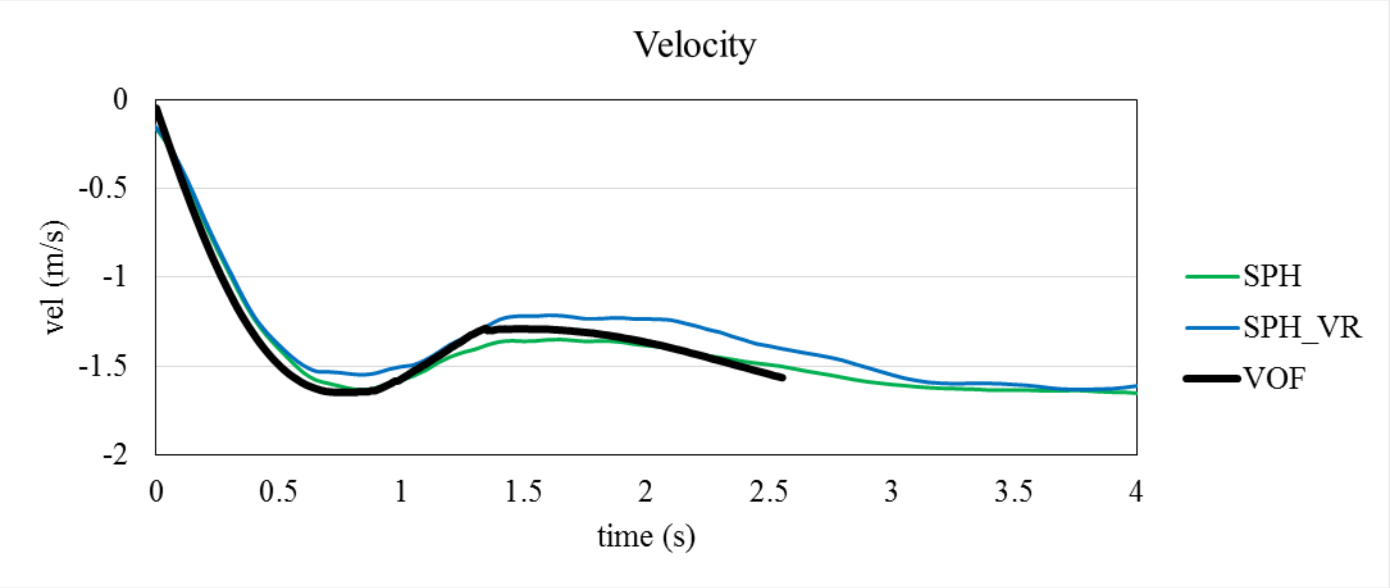
Fekken G. Numerical simulation of free surface flow with moving rigid bodies, Ph.D. Thesis, University of Groningen, 2004

High resolution Δx_0=0.03 m
no adaptivity

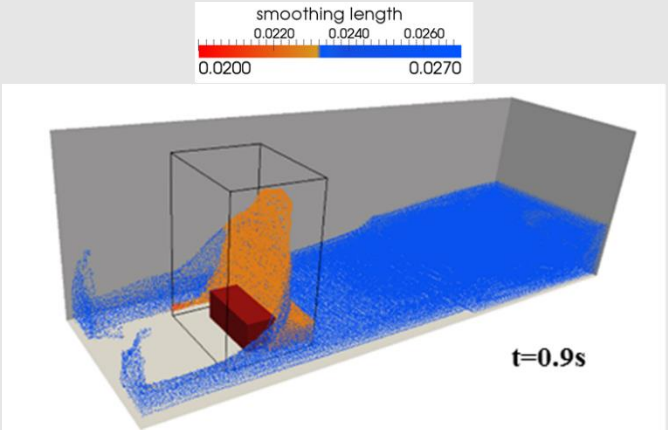
Low Res Δx_0=0.05 m
dynamic adaptivity




Displacement
0
-1
-2
-3
-4
-5
-6
z (m)
0
0.5
1
1.5
2
2.5
3
3.5
4
time (s)
SPH
SPH_VR
VOF


Velocity
0
-0.5
-1
-1.5
-2
vel (m/s)
0
0.5
1
1.5
2
2.5
3
3.5
4
time (s)
SPH
SPH_VR
VOF

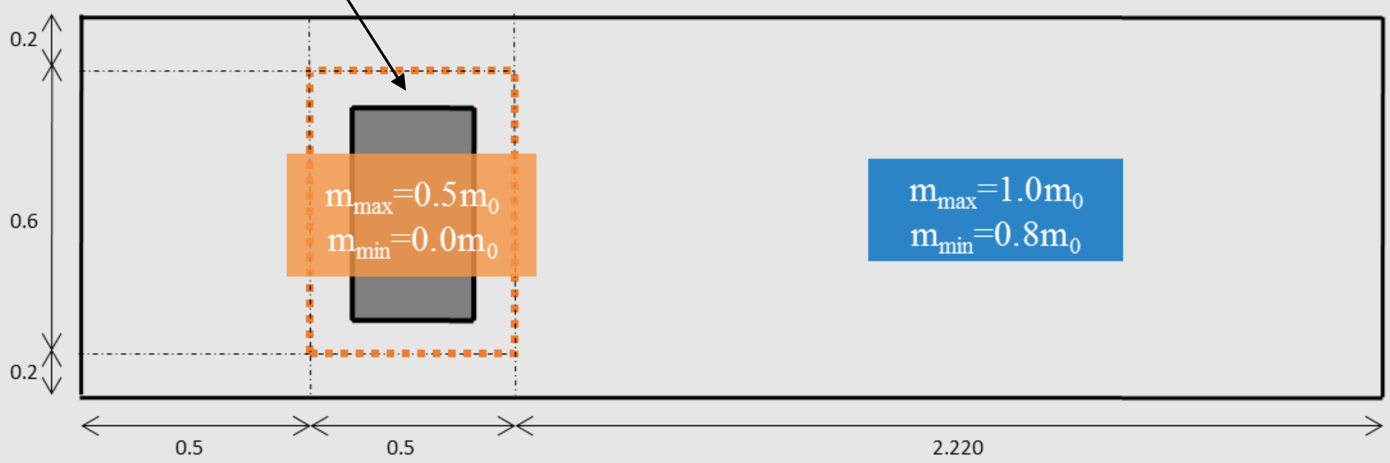
SPHERIC Benchmark Case #2.

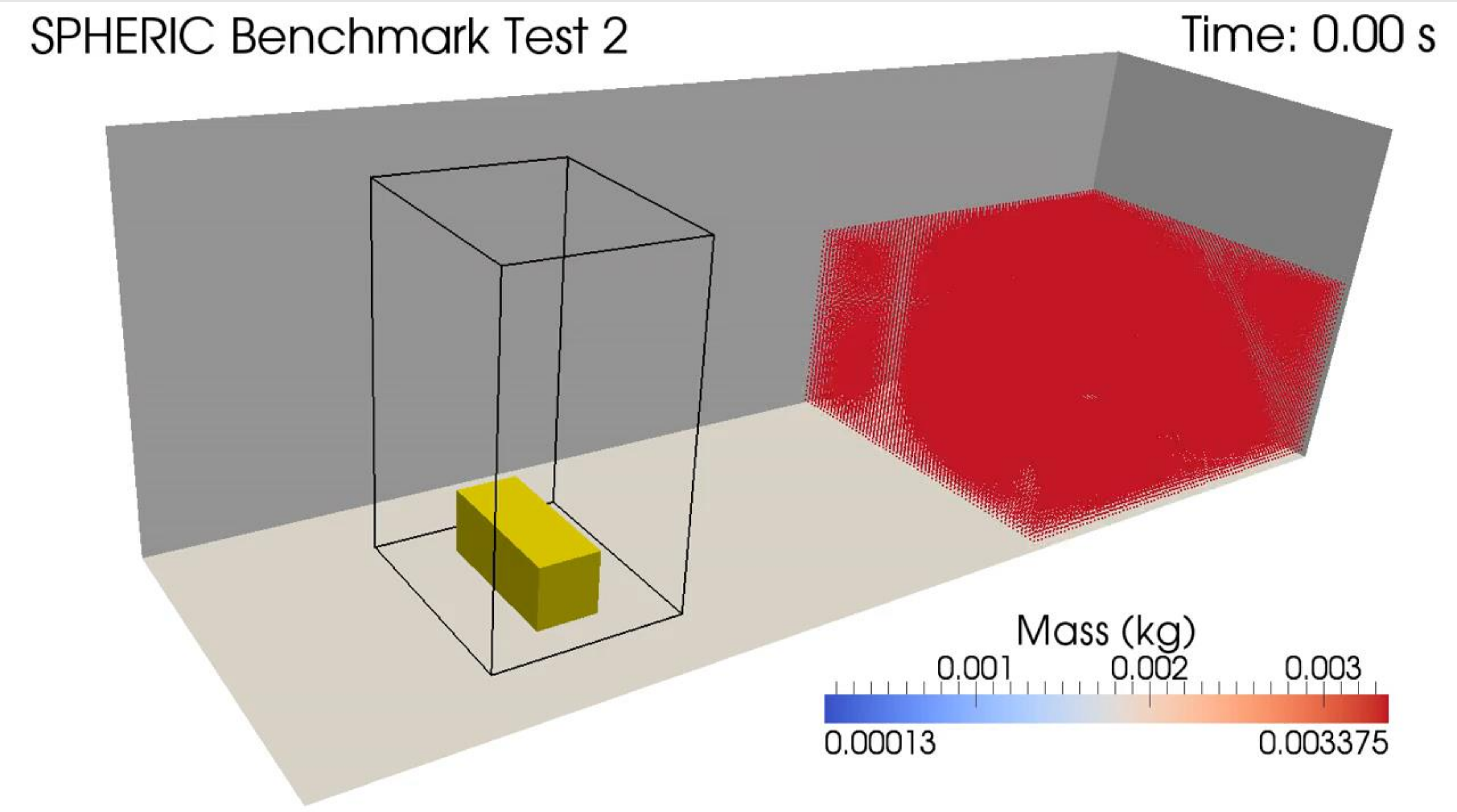


Two simulations:

- No adaptivity, Δx_0=0.008m
- Adaptivity (splitting and coalescing) Δx_0 =0.015m

High resolution region

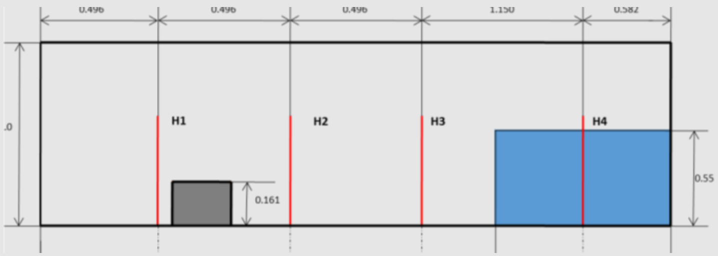



SPHERIC Benchmark Test 2
Time: 0.00 s
Mass (kg)
0.001
0.002
0.003
0.00013
0.003375

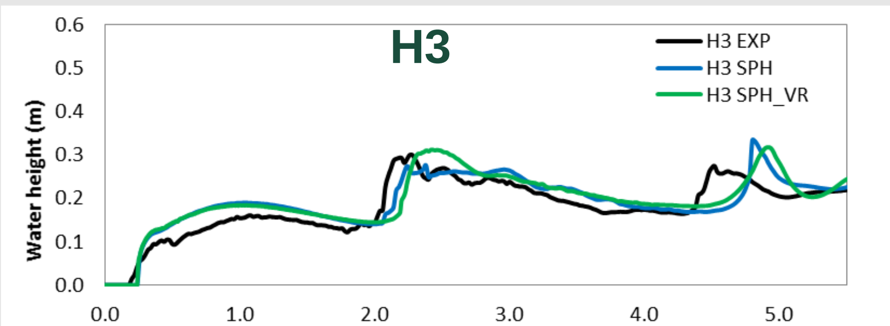
SPHERIC Benchmark Case #2.

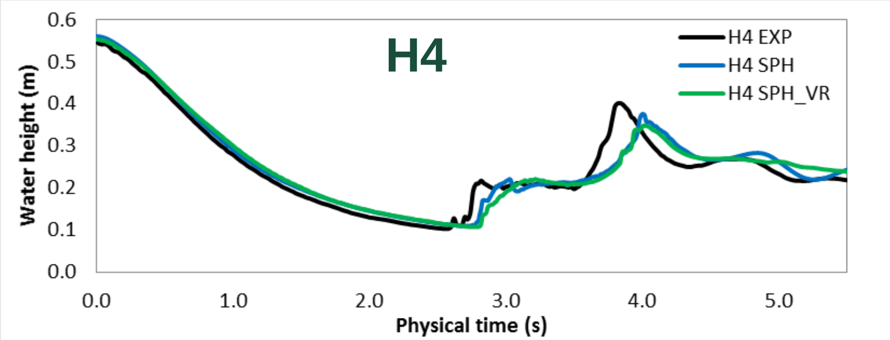
Two simulations:

- no adaptivity: **blue**
- Adaptivity: **green**

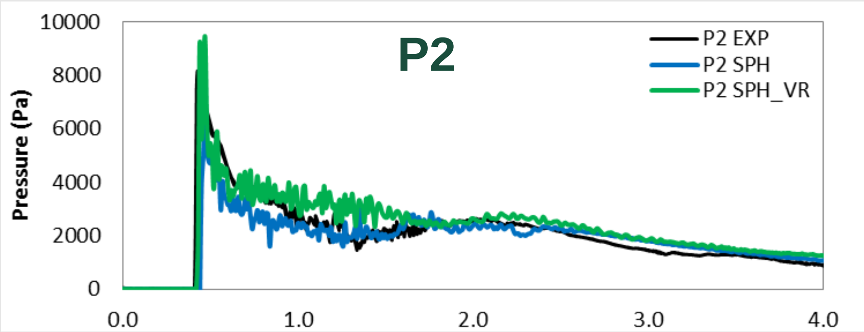


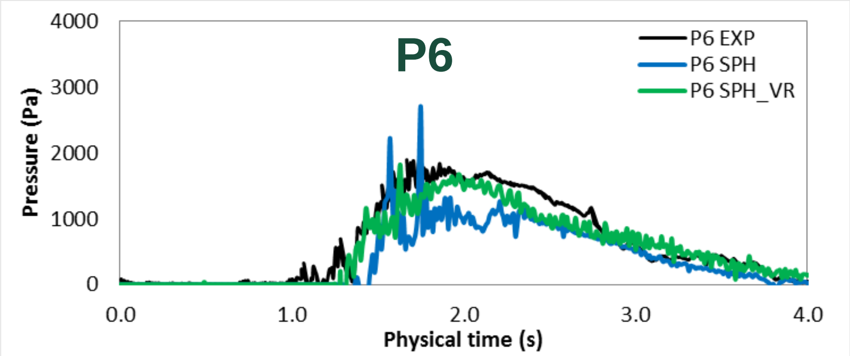
Water height





Pressure

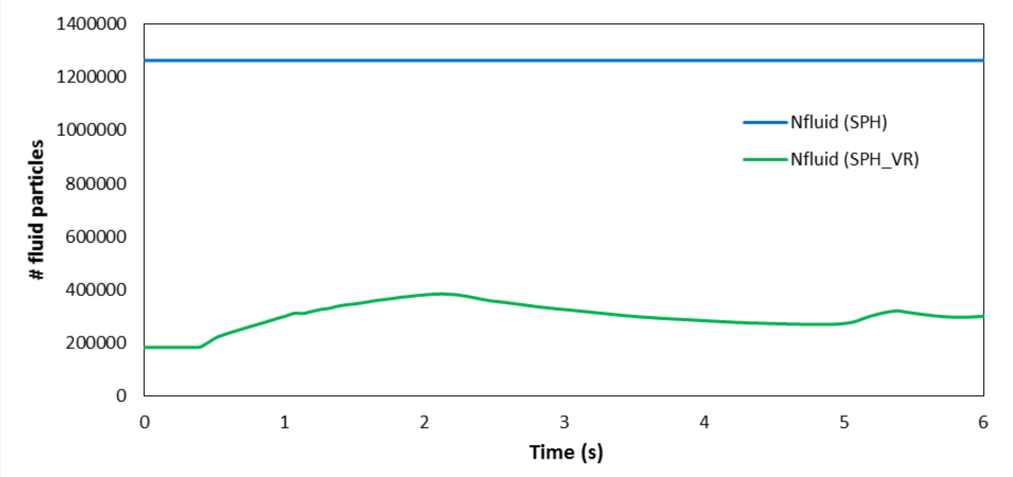




SPHERIC Benchmark Case #2.

	SPH	SPH-adaptivity
Δx_0	0.008m	0.015m
Initial number of particles	1,262,816	184,275
New daughters by splitting	-	$2.2\ 10^6$
Coalesced particles	-	$3{,}9\ 10^6$
CPU runtime	173 h	85 h
GPU runtime	3.60h	1.99h

# of fluid particles



CPU speedup: 2.04

GPU speedup: 1.80

This looks bad but:

$$\frac{(\#\text{ part SPH})}{(\#\text{ part SPH-adapt})} = 4.2$$

Conclusion on variable resolution

- Open Source Parallel SPH code with variable resolution and adaptivity has been presented
- Both 2D and 3D
- OpenMP and CUDA versions of the code have been developed, Speedup / overheads have been discussed
- Code validated in 2D and 3D against experiments and numerical simulation
- Formulation is adapted for particles with different size with negligible errors at interface between different resolution


fresh
POP CORN
DELICIOUS
CRISP
NUTRITIOUS

Part 2: Boundary Conditions
(SPHERIC Grand Challenge #3)

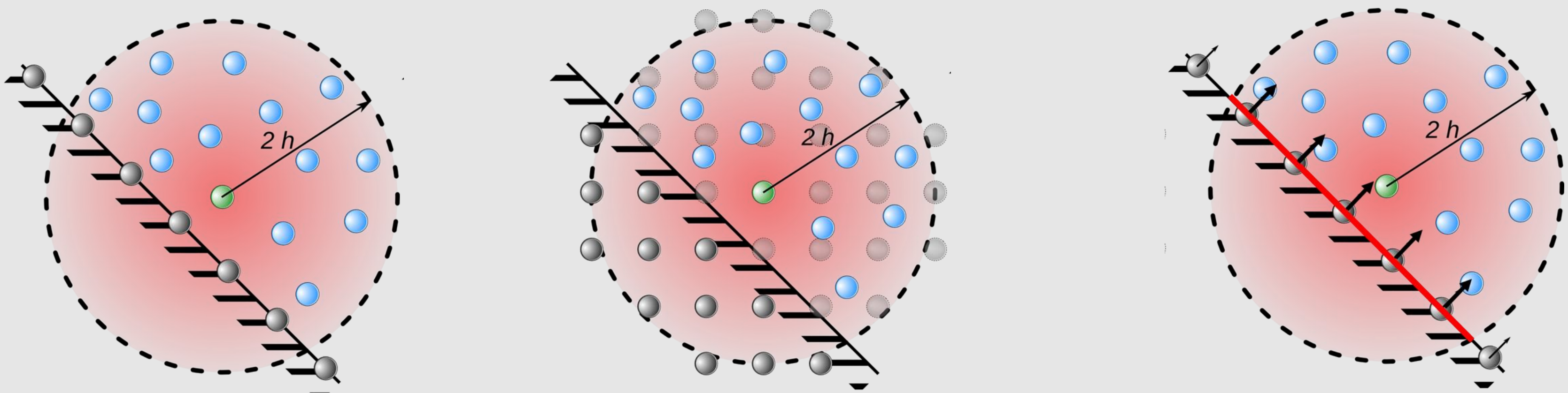
Different type of Boundary Conditions

(i) Boundary repulsive force

(ii) Fluid extensions to the solid boundary

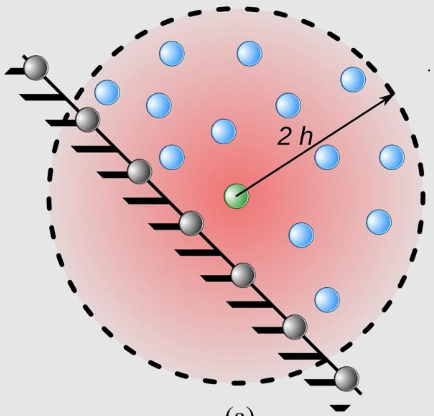
(iii) Boundary integral (analytical or semi-analytical)

Dynamic boundaries (DBC) from group (i-ii)	Local Uniform STencil (LUST) from group (ii)	Boundary Integral (INTEGRAL) from group (iii)



(a) (b) (c)

Dynamic boundaries (DBC)

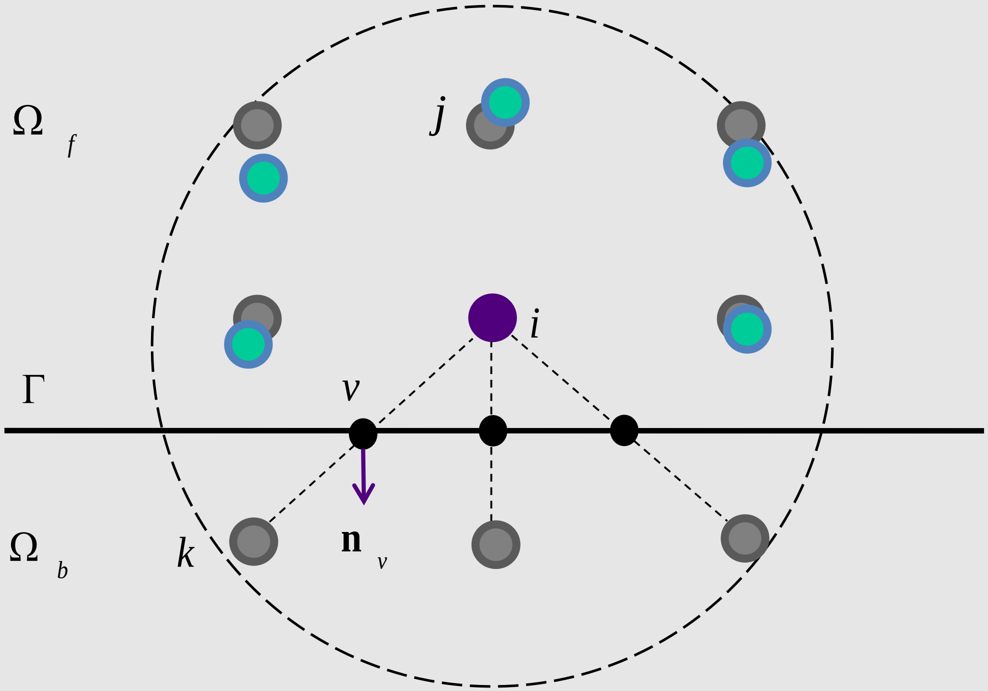


(a)

- same continuity equation as for the fluid particles
- computationally efficient
- Kernel truncation error which prevents convergence
- Over repulsion of fluid particles

$$\frac{d\rho_i}{dt} = \sum_j \rho_i \frac{m_j}{\rho_j} \boldsymbol{v}_{ij} \cdot \nabla_i W_{ij}$$

Local Uniform Stencil (LUST) Concept



- Regular stencil of fictitious particles is centered around fluid particles

- Fictitious particles in the fluid domain are deleted.

- The remaining fictitious particles, are used to solve cont. and momentum equations

- No kernel truncation

- It can deal with complex boundary

- Approximately first order consistent

- Computationally more expensive than DB

Local Uniform Stencil (LUST)

The density of the fictitious particles is corrected hydrostatically based on the density of the fluid particle.

$$\rho_k = \rho_i + \left[\rho_0 \sqrt[7]{\frac{\rho_0 g_z \mathbf{x}_{ik} \cdot \mathbf{n}_v}{B} + 1} - \rho_0 \right],$$

The pressure is then evaluated through the EOS.

$$P_k = B\left(\left(\frac{\rho_k}{\rho_0} \right)^7 - 1 \right).$$

The velocities of the fictitious particles are assigned according to Takeda et al.'s anti-symmetric mirroring formulation.

$$\mathbf{u}_k = (\mathbf{u}_i - \mathbf{u}_v)\frac{\mathbf{x}_{vk} \cdot \mathbf{n}_v}{\mathbf{x}_{iv} \cdot \mathbf{n}_v} - \mathbf{u}_v .$$

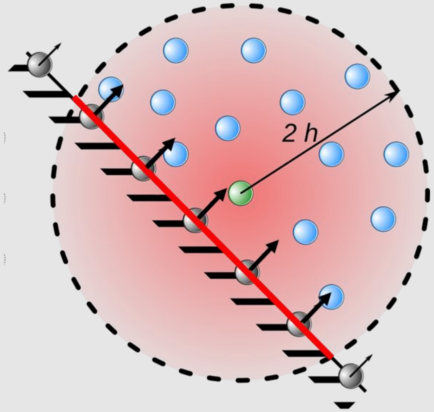
Momentum equation

$$\left\langle \frac{\mathrm{d}\,\mathbf{u}}{\mathrm{d}\,t} \right\rangle_i = -\sum_{j \in \Omega_f} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla W_{ij} - \sum_{k \in \Omega_b} m_k \left(\frac{P_i}{\rho_i^2} + \frac{P_k}{\rho_k^2} + \Pi_{ik} \right) \nabla W_{ik},$$

Continuity equation

$$\left\langle \frac{\mathbf{d}\,\rho}{\mathbf{d}\,t} \right\rangle_i = \sum_{j \in \Omega_f} m_j (\mathbf{u}_i - \mathbf{u}_j) \cdot \nabla W_{ij} + \sum_{k \in \Omega_b} m_k (\mathbf{u}_i - \mathbf{u}_k) \cdot \nabla W_{ik}.$$

Boundary Integral (INTEGRAL)



(c)

Numerical approximation of the integral

SPH interpolant

Continuous formulation

$$\langle f(\boldsymbol{x})\rangle = \frac{1}{\gamma_h(\boldsymbol{x})}\int_{\boldsymbol{y}\in\Omega} f(\boldsymbol{y})W_h(\boldsymbol{x}-\boldsymbol{y})d\boldsymbol{y}$$

$$\gamma_h(\boldsymbol{x}) := \int_{\boldsymbol{y}\in\Omega} W_h(\boldsymbol{x}-\boldsymbol{y})d\boldsymbol{y}.$$

Discrete formulation

$$\langle f\rangle_i = \frac{1}{\gamma_i}\left(\sum_{j\in\mathrm{Fluid}} \frac{f_j}{\rho_j} W_{ij}\, m_j\right)$$

$$\gamma_i = \sum_{j\in\mathrm{Fluid}} \frac{1}{\rho_j} W_{ij}\, m_j$$

Integrals along the boundary are replaced by a sum of area elements

Boundary Integral (INTEGRAL)

SPH differential operator

Continuous formulation

$$\langle \mathcal{D}f(\boldsymbol{x})\rangle = \frac{1}{\gamma_h(\boldsymbol{x})}\int_{\boldsymbol{y}\in\Omega} \mathcal{D}f(\boldsymbol{y})\, W_h(\boldsymbol{x}-\boldsymbol{y})d\boldsymbol{y},$$

$$\langle \mathcal{D}f(\boldsymbol{x})\rangle = \frac{1}{\gamma_h(\boldsymbol{x})}\left(\int_{\boldsymbol{y}\in\Omega} f(\boldsymbol{y})\cdot\nabla W_h(\boldsymbol{y}-\boldsymbol{x})d\boldsymbol{y} + \int_{\boldsymbol{y}\in\partial\Omega} f(\boldsymbol{y})\cdot \boldsymbol{n}(\boldsymbol{y})W_h(\boldsymbol{y}-\boldsymbol{x})d\boldsymbol{y}\right)$$

Discrete formulation

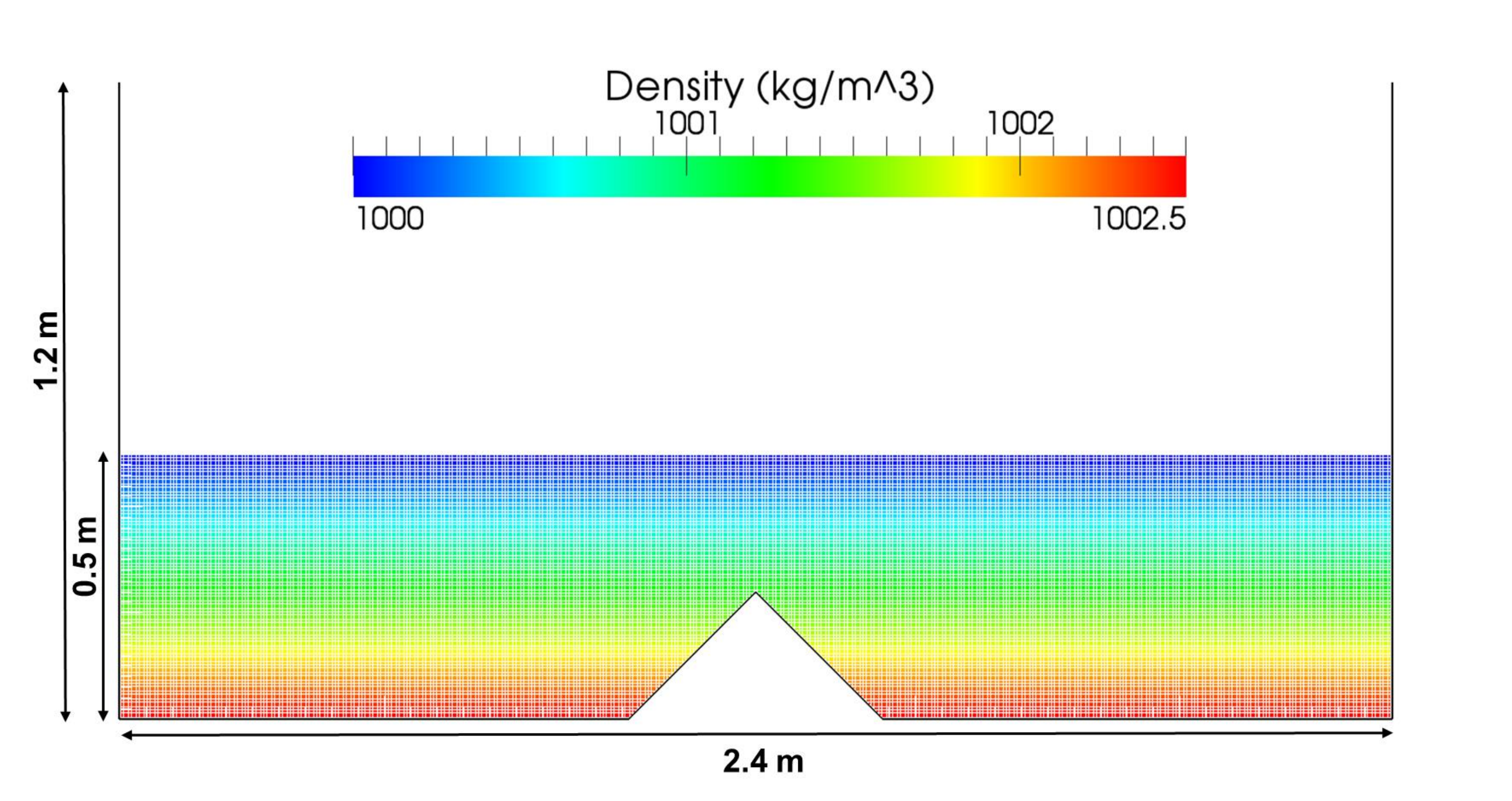
$$\langle \mathcal{D}f\rangle_i = \frac{1}{\gamma_i}\left(\sum_{j\in\mathrm{Fluid}} \frac{f_j}{\rho_j}\cdot\nabla W_{ij}\, m_j + \sum_{j\in\mathrm{Boundary}} f_j\cdot \boldsymbol{n}_j W_{ij}\, s_j\right)$$

Variable resolution

Test cases

Can you guess which is the first test case?

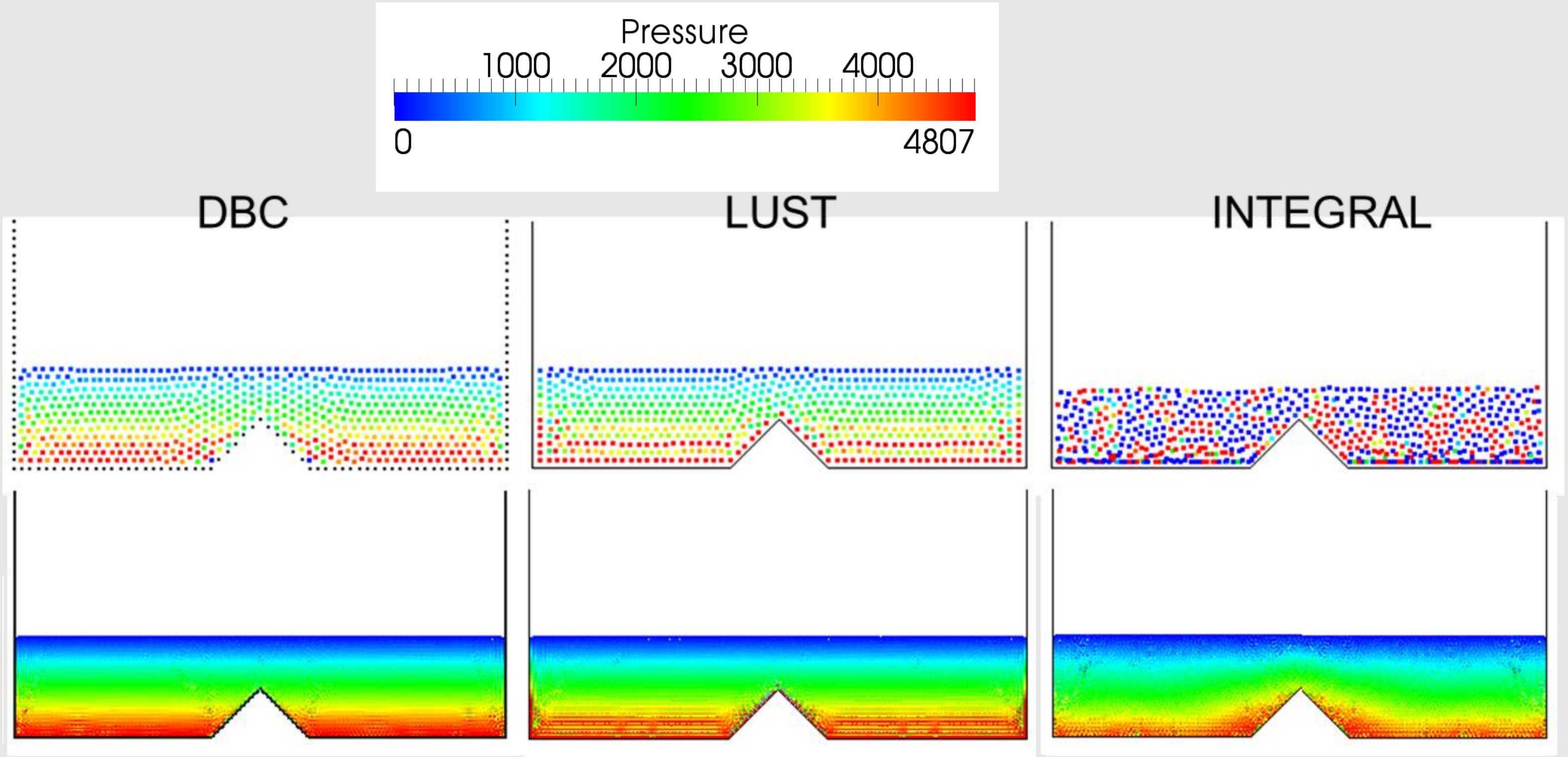
TEST 1: Still water with a wedge (2-D)



- A low value of viscosity is used (α_π=0.01)
- No density filter
- h/dp=1.3

TEST 1: Still water with a wedge (2-D)

Convergence test



dp = 0.04

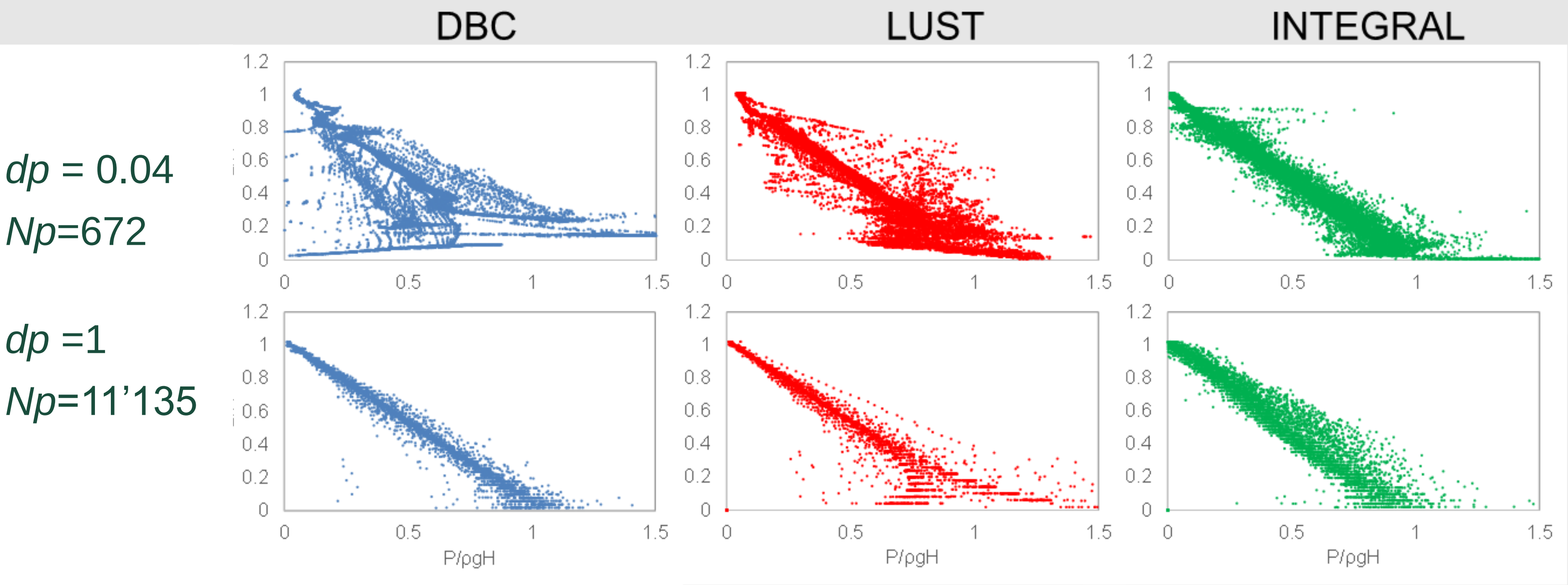
Np=672

dp =0.01

Np=11’135

Results are analysed after 20 seconds of physical time (85,000 steps)

TEST 1: Still water with a wedge (2-D)

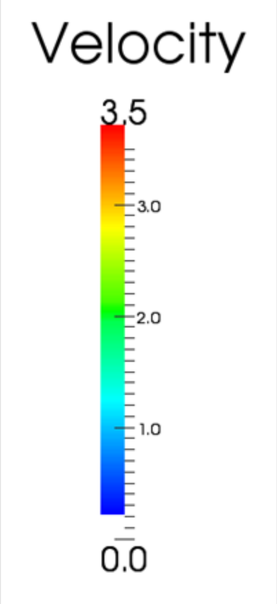


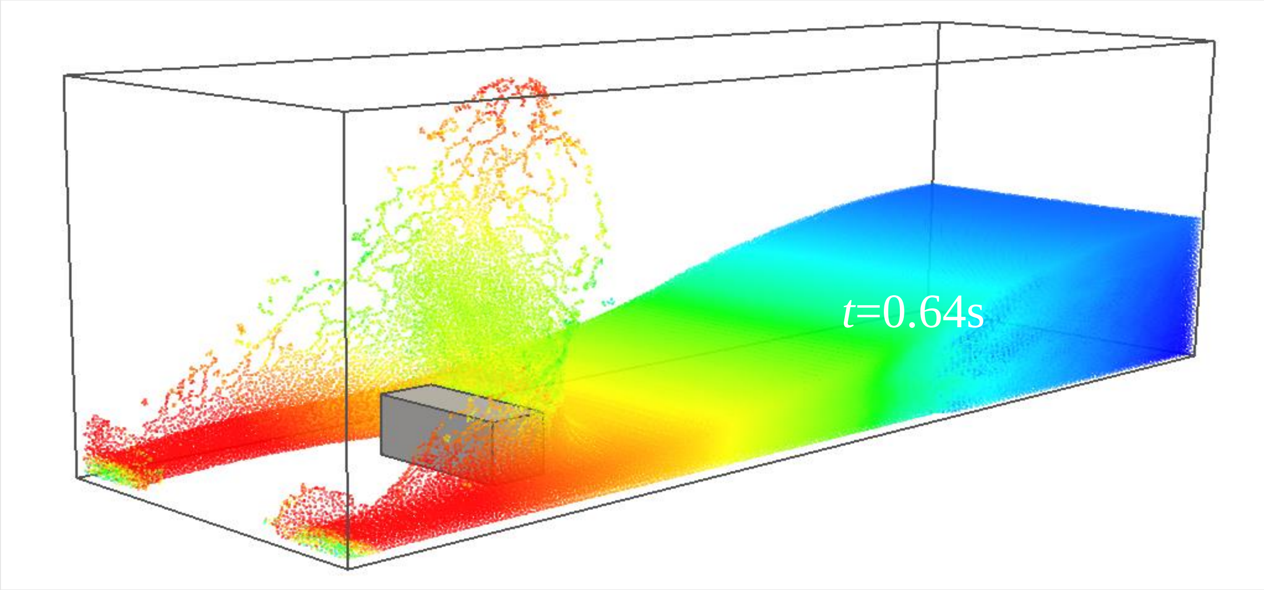
TEST 2: SPHERIC Benchmark Test Case #2

SPHERIC Benchmark Test Case #2

δ-SPH is now employed to obtain smoothed density distribution

Dp=0.01, Np=800k, Time= 6.0 s



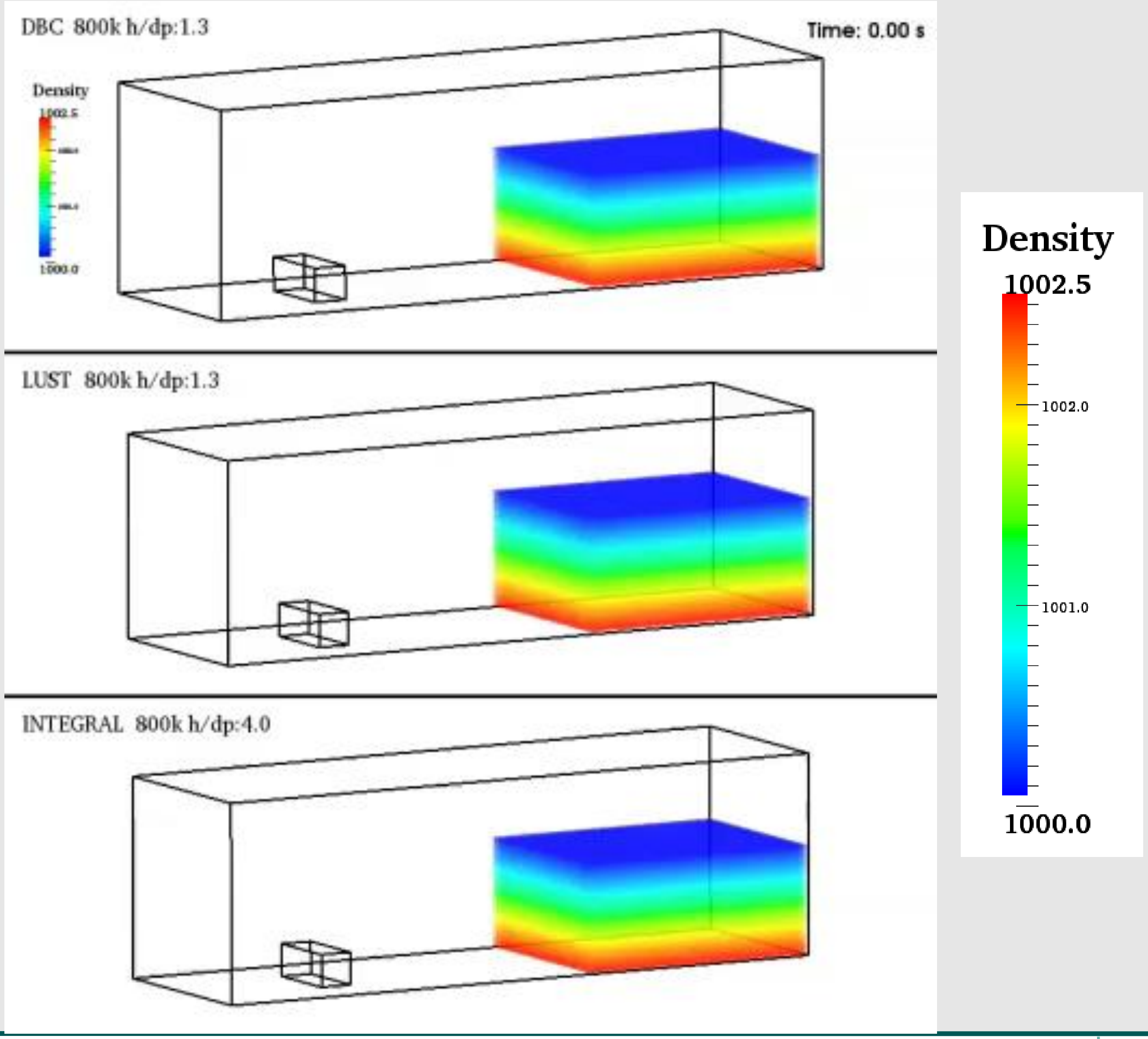


TEST 2: SPHERIC Benchmark Test Case #2

DBC

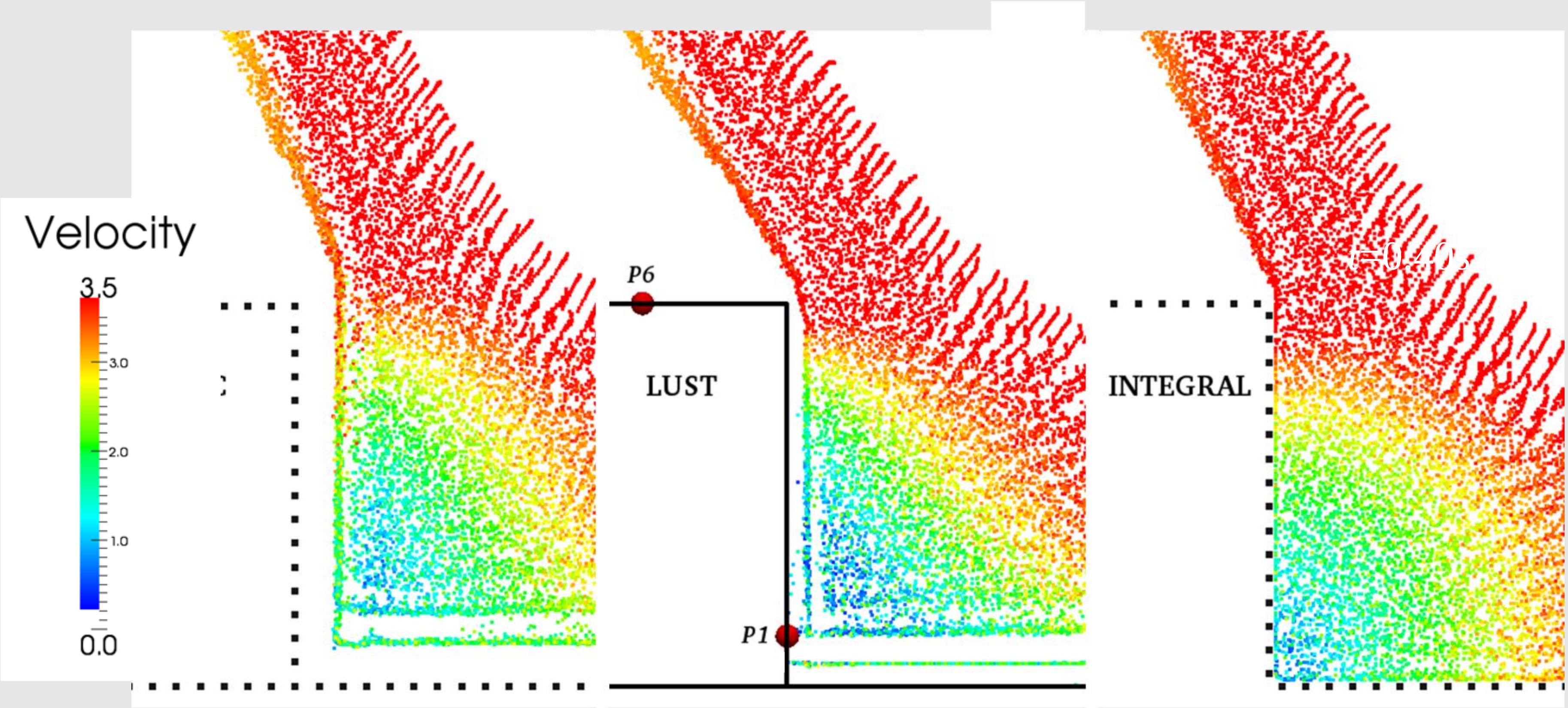
LUST

INTEGRAL



TEST 2: Dam break with obstacle (3-D)

Dp=0.01, Np=800k, Time= 0.40 s



k_{DBC}= 1.2 k_{LUST}= 0.5 $k_{INTEGRAL}$

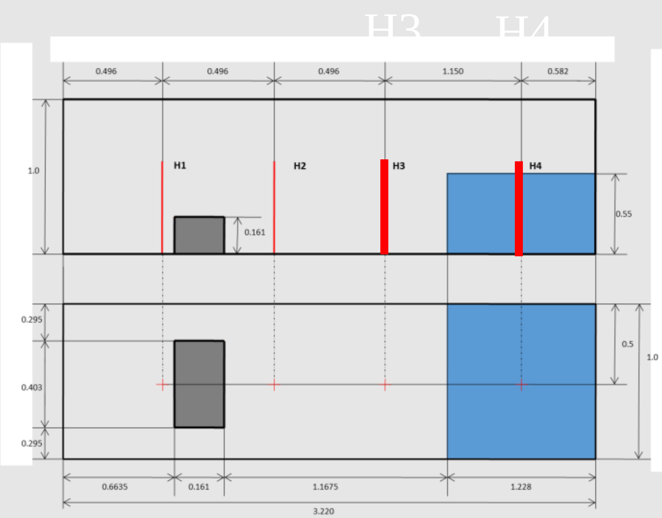
TEST 2: Dam break with obstacle (3-D)

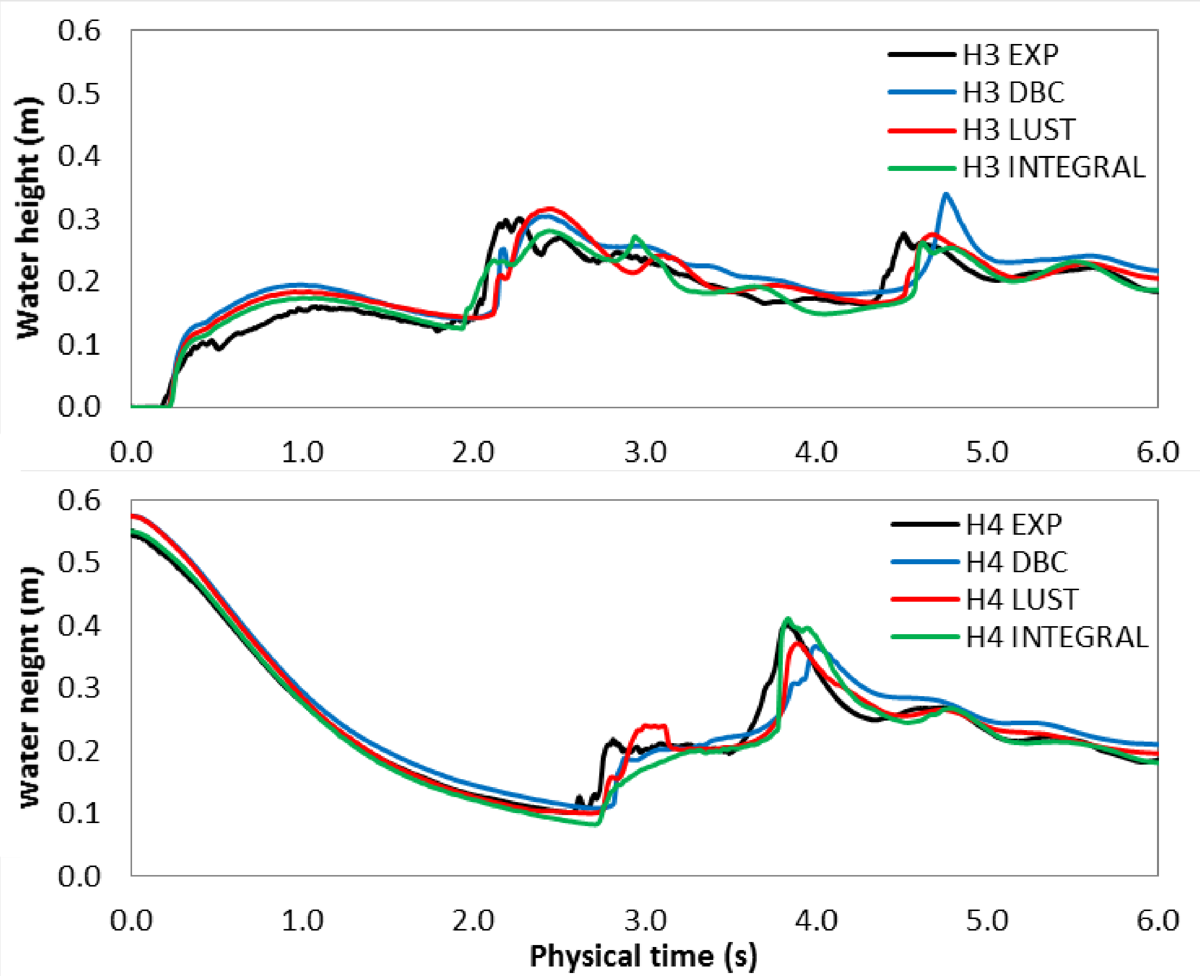
Experimental and numerical water heights measured at the probes H3 and H4 with dp=0.01m and h/dp=1.3

DBC

LUST

INTEGRAL





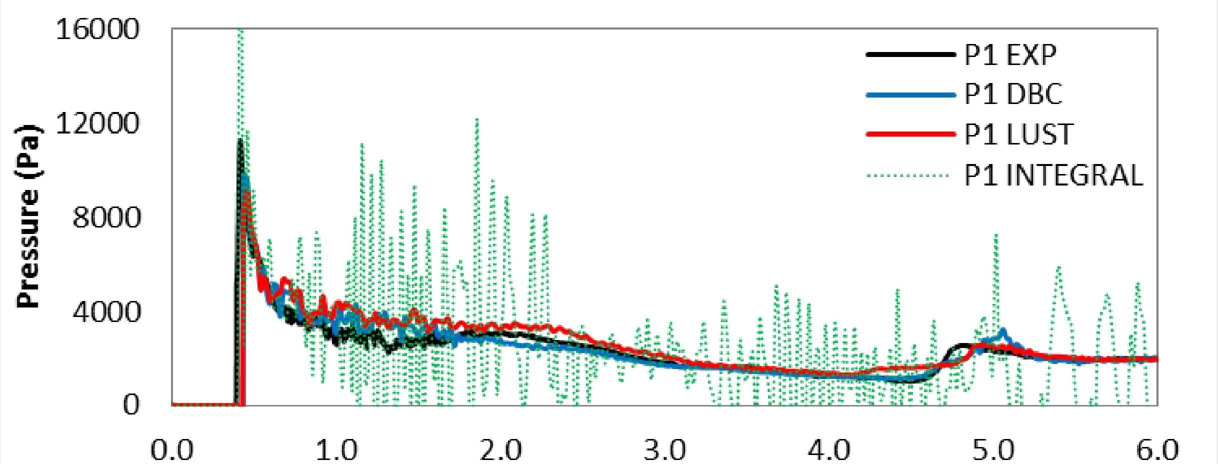
TEST 2: Dam break with obstacle (3-D)

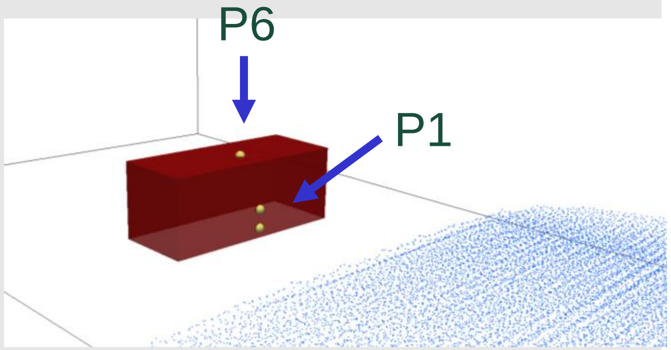
Experimental and numerical pressures measured at the sensors P1 and P6 with dp=0.01m and h/dp=1.3

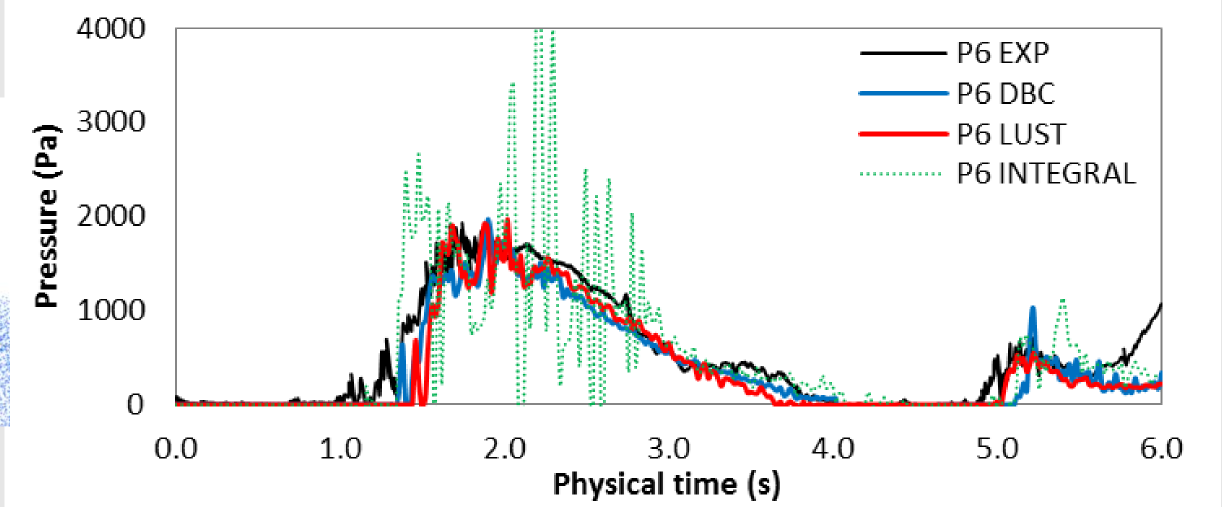
DBC

LUST

INTEGRAL





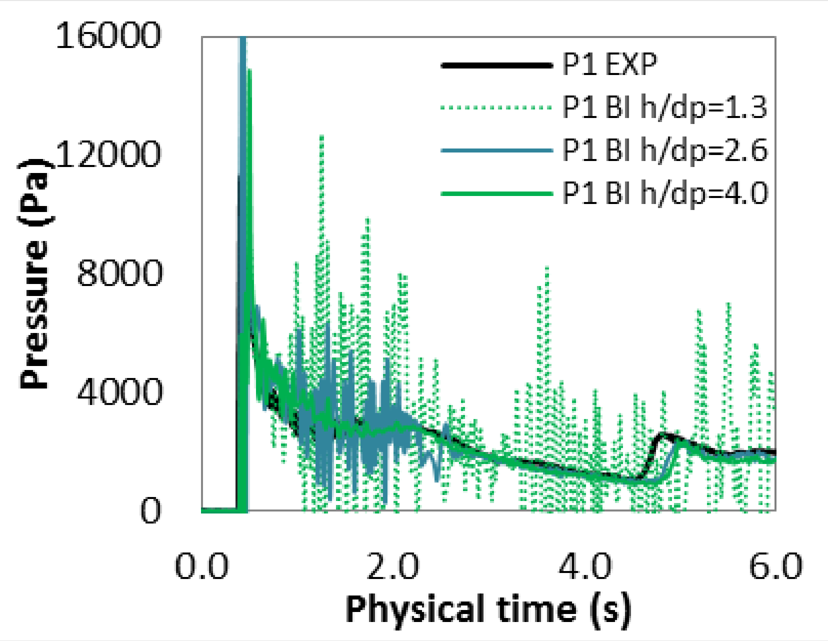


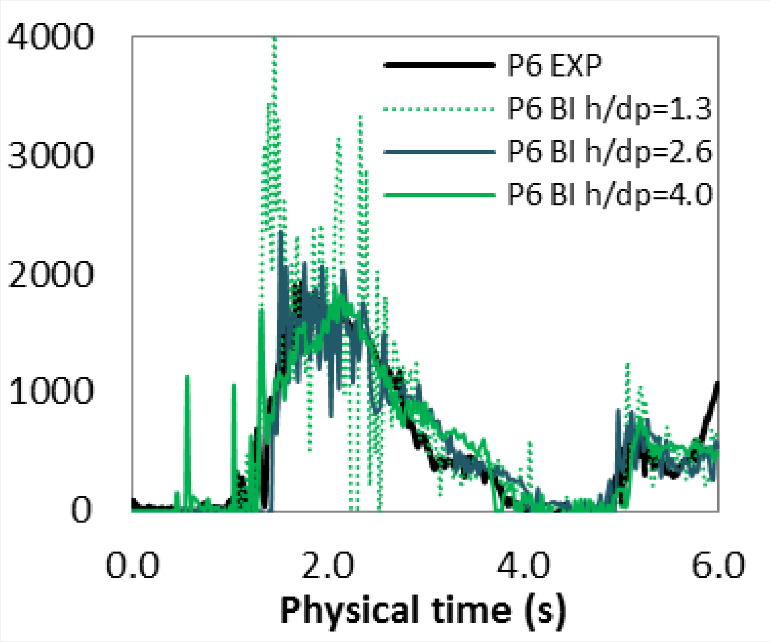
TEST 2: Dam break with obstacle (3-D)

Experimental and numerical pressures measured at the sensors P1 and P6 with h/dp=1.3, 2.6, 4.0

INTEGRAL

h/dp=4.0





TEST 2: Dam break with obstacle (3-D)

Performance Analysis

Test	BC type	Np	h/dp	Runtime (h)
TEST 2 Dam break 3-D	DBC	100k	1.3	0.13
		800k		**1.47**
	LUST	100k	1.3	0.18
		800k		**1.99**
	INTEGRAL	100k	1.3	0.19
		800k		**2.53**

The ratio h/dp needs to be increased till 4 to get good results for INTEGRAL.

Conclusions

A comparison of three different boundary conditions has been performed.

DBC can be applied to arbitrary 2-D and 3-D geometries, BUT a high repulsive force is generated acting on the fluid particles resulting in a separation distance.

LUST BC is more computationally expensive than DBC but more accurate and it addresses most of the issues of DBC.

INTEGRAL methodology requires large number of neighbours within the support (as discussed in the consistency notes and demonstrated in the test cases) to obtain good accuracy.