上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

CMHL COMPUTATIONAL MARINE HYDRODYNAMICS LAB
SHANGHAI JIAO TONG UNIVERSITY

The Simulation of Three-Dimensional Flow by Using GPU-based MPS Method

Xiang Chen, Decheng Wan

State Key Laboratory of Ocean Engineering,
School of Naval Architecture, Ocean and Civil Engineering,
Shanghai Jiao Tong University

spheric
ERCOFTAC

2017 SPHERIC International Workshop, Beijing, China
Beijing, China, Oct. 18-20, 2017: http://spheric-sph.org/index

Contents

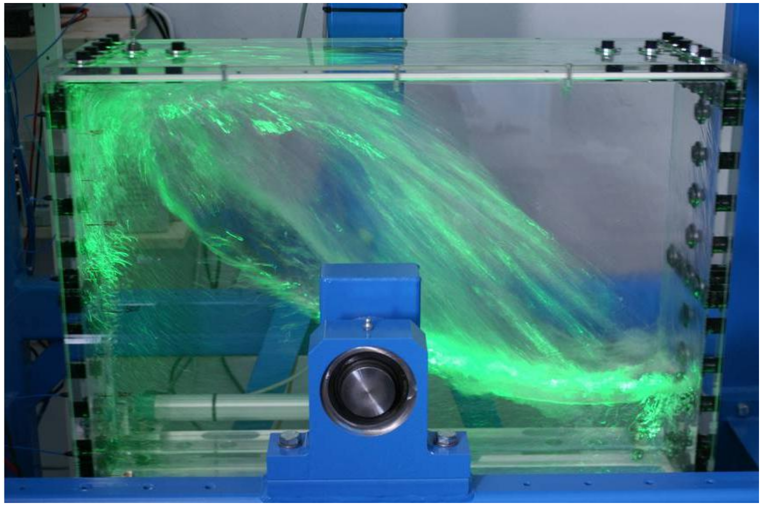
- Backgrounds
- Numerical methods
  - ✓ Modified MPS
  - ✓ MPSGPU-SJTU Solver
- Numerical Simulation
  - ✓ Dam Break Flow
  - ✓ Liquid Sloshing
- Conclusions and Ongoing Work

Backgrounds

Difficulty of violent flow simulation:

- Dealing with large deformations and nonlinear fragmentation of free surface

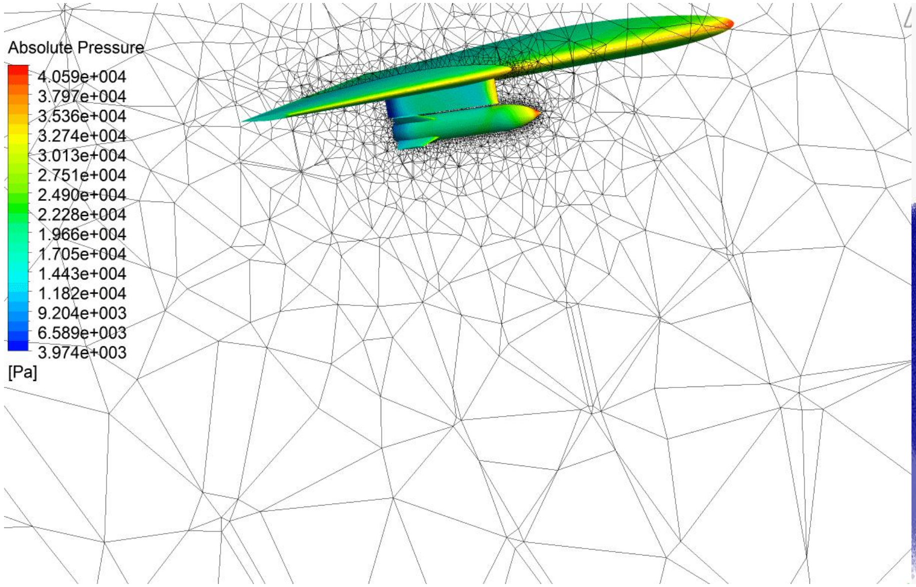
Dam Break



Liquid Sloshing

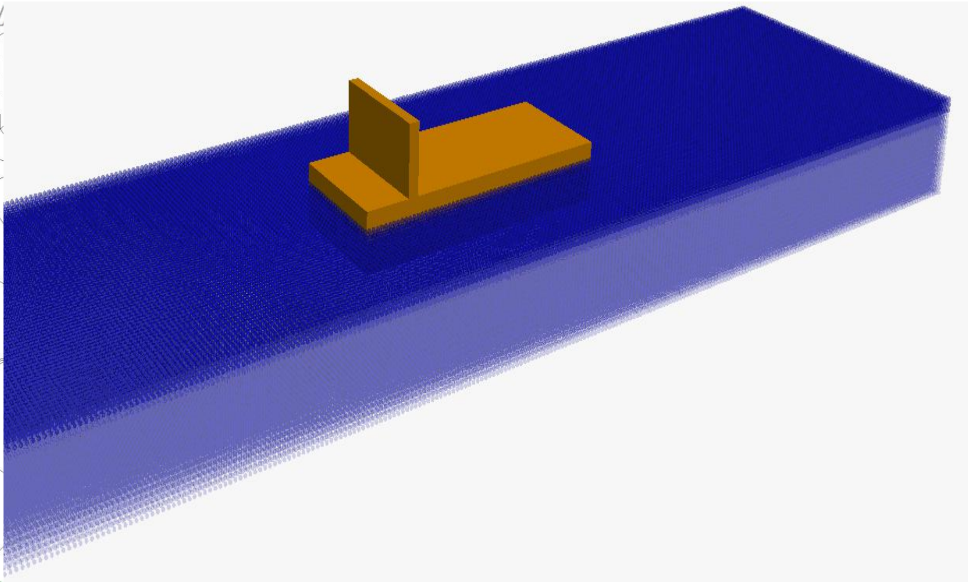
Backgrounds

Mesh Method



- Mesh distortion because of large deformation.
- Difficult to simulate nonlinear free surface.

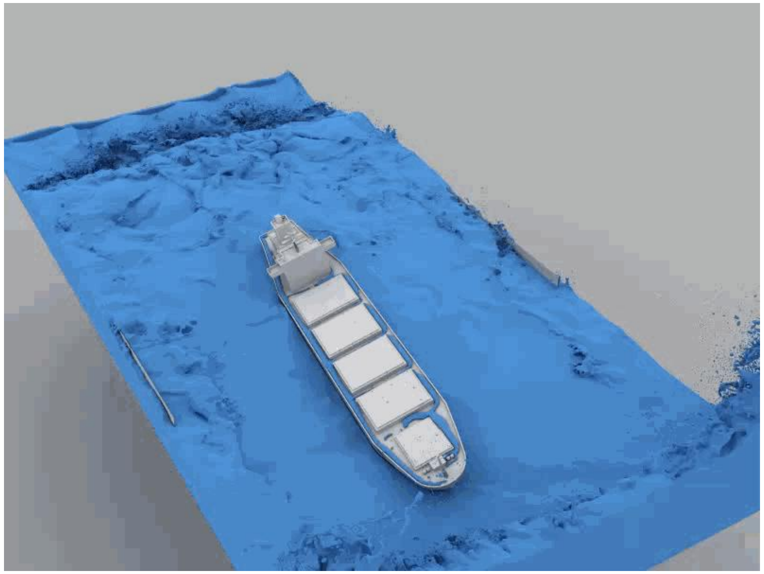
Meshfree Method



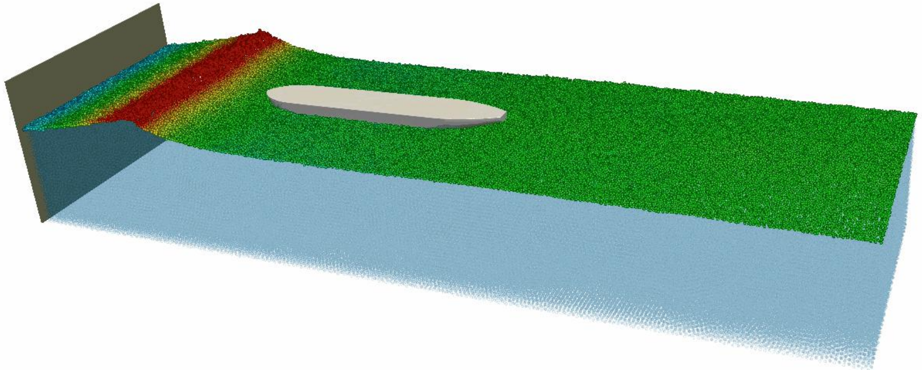
- Easily track free surfaces and moving boundaries.
- Removing the difficulties of re-meshing.

Backgrounds

	SPH	MPS
Fluid	Compressible	Incompressible
Pressure	Equation of State	Pressure Poisson Equation
Time iteration	Explicit	Semi-implicit



SPH

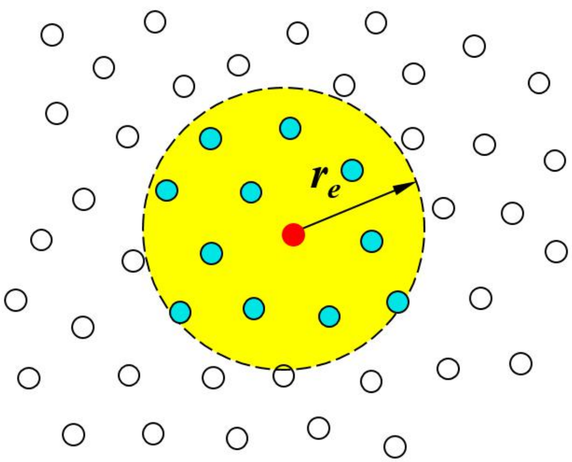


MPS

Backgrounds

Disadvantage of MPS Method

- **High computational cost (Searching neighbor particles, Solving Pressure Poisson Equation).**



Neighbor Particles

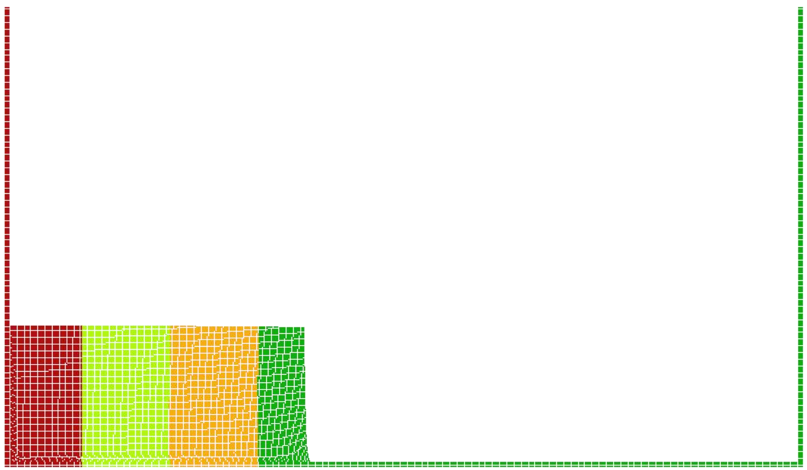
$$<\nabla^2 P^{k+1}>_i = (1-\gamma)\frac{\rho}{\Delta t}\nabla\cdot\vec{V}_i^* - \gamma\frac{\rho}{\Delta t^2}\frac{<n^*>_i - n^0}{n^0}$$

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,i} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i,1} & \cdots & a_{i,i} & \cdots & a_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,i} & \cdots & a_{n,n} \end{pmatrix}\begin{pmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b \\ \vdots \\ b_i \\ \vdots \\ b_n \end{pmatrix}$$

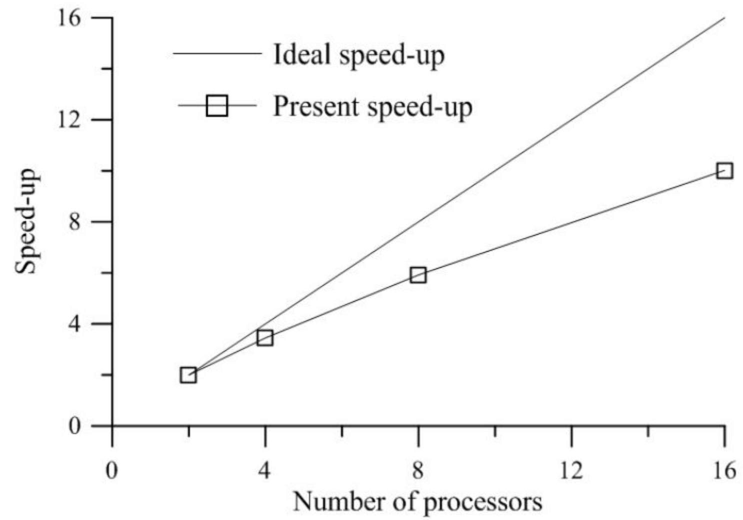
PPE

Backgrounds

- **CPU Parallel**



Dynamic load balancing (Zhang, 2014)

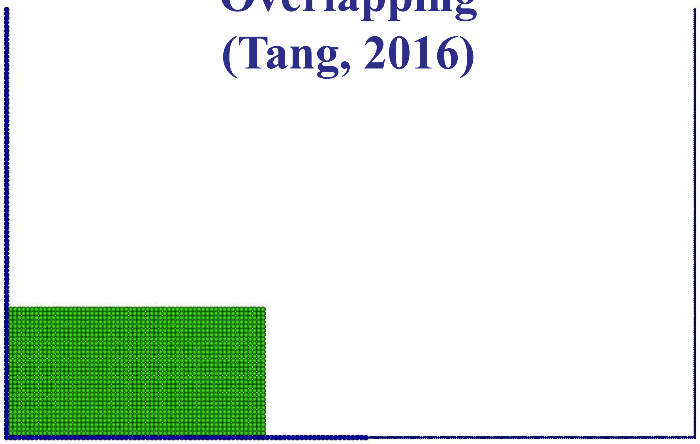


Speed-up of different processor numbers

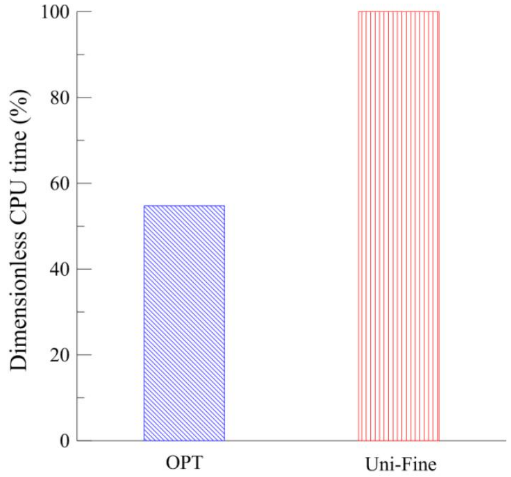
Backgrounds

Refined Particles

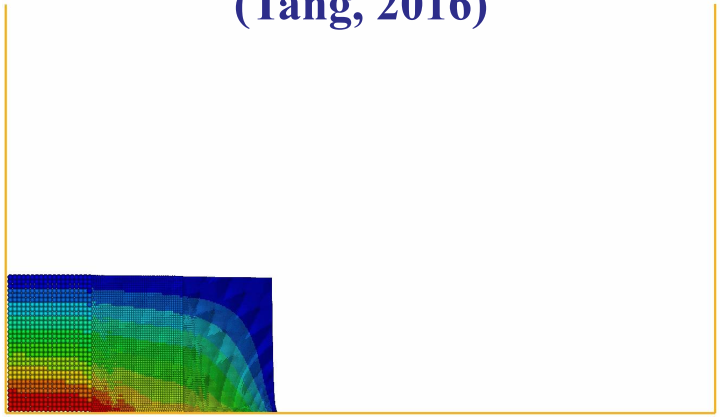
Overlapping
(Tang, 2016)



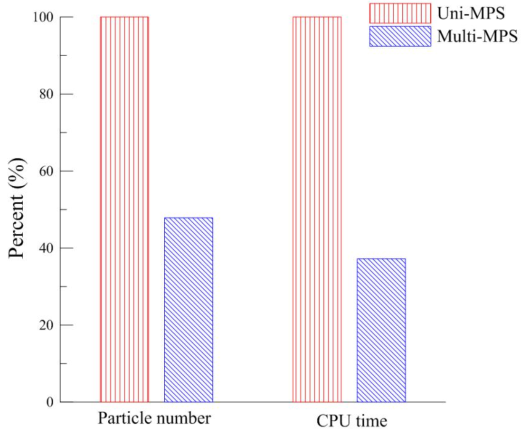
CPU time between OPT and Uni-Fine



Multi-resolution
(Tang, 2016)



CPU time between Multi-MPS and Uni-MPS



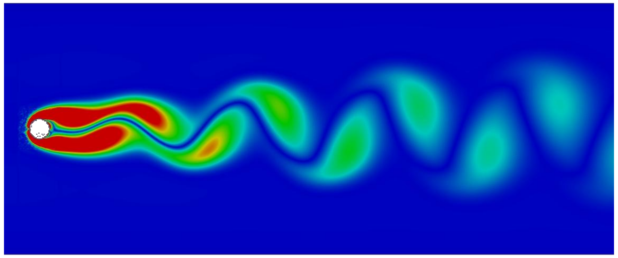
Backgrounds

COMPUTATIONAL FLUID DYNAMICS

APPLICATION	DESCRIPTION	SUPPORTED FEATURES	MULTI-GPU SUPPORT
Altair - AcuSolve	General purpose CFD software	Linear equation solver	Yes
ANSYS - Fluent	General purpose CFD software	Radiation heat transfer model, linear equation solver	Yes
Autodesk - Moldflow	Plastic mold injection software	Linear equation solver	Single only
CPFD Barracuda - VR and Barracuda	Fluidized bed modeling software	Linear equation solver, particle calculations	Single only
DHI MIKE 21	2D hydrological modelling of coast and sea	Hydrodynamics; Advection-dispersion; Sand and mud transport; coupled modelling; particle tracking; oil spill; ecological modelling; agent based modelling; various wave models	Yes
DHI MIKE FLOOD	1D & 2D urban, coastal, and riverine flood modelling	Hydrodynamics	Yes
FluiDyna aeroFluidX	Incompressible single-phase CFD software	Finite volume solver	Yes
FluiDyna - Culises for OpenFOAM	Solver library for general purpose CFD software	Linear equation solvers	Yes
FluiDyna nanoFluidX	General purpose CFD software	SPH solver	Yes
FluiDyna ultraFluidX	General purpose CFD software	Lattice-Boltzmann solver	Yes
midas NFX (CFD)	General purpose CFD software based on FEM	Linear equation solver(Iterative Solver and AMG Preconditioner)	Single only
Prometech - Particleworks	Particle-based CFD software	Implicit and explicit solvers	Yes
Turbostream Ltd.	CFD software for turbomachinery flows	Explicit solver	Yes
Vratis Speed IT FLOW	Incompressible single-phase CFD software	Finite volume solver	Single only
Vratis SpeedIT for OpenFOAM	Solver library for general purpose CFD software	Linear equation solvers	Yes

GPU applications in fluid dynamics

OpenFOAM



ANSYS Fluent

Contents

- Backgrounds
- **Numerical methods**
  - ✓ **Modified MPS**
  - ✓ **MPSGPU-SJTU Solver**
- Numerical Simulation
  - ✓ Dam Break Flow
  - ✓ Liquid Sloshing
- Conclusions and Ongoing Work

Modified MPS

	Original	Modified
Kernel Function	$W(r)=\begin{cases}\dfrac{r_e}{r}-1 & 0\le r<r_e\\ 0 & r_e\le r\end{cases}$	$W(r)=\begin{cases}\dfrac{r_e}{0.85r+0.15r_e}-1 & 0\le r<r_e\\ 0 & r_e\le r\end{cases}$
Gradient Model	$<\nabla P>_i=\dfrac{D}{n^0}\sum_{j\neq i}\dfrac{P_j-P_i}{\lvert \boldsymbol{r}_j-\boldsymbol{r}_i\rvert^2}(\boldsymbol{r}_j-\boldsymbol{r}_i)\cdot W(\lvert \boldsymbol{r}_j-\boldsymbol{r}_i\rvert)$	$<\nabla P>_i=\dfrac{D}{n^0}\sum_{j\neq i}\dfrac{P_j+P_i}{\lvert \boldsymbol{r}_j-\boldsymbol{r}_i\rvert^2}(\boldsymbol{r}_j-\boldsymbol{r}_i)\cdot W(\lvert \boldsymbol{r}_j-\boldsymbol{r}_i\rvert)$
PPE	$<\nabla^2P^{k+1}>_i=-\dfrac{\rho}{\Delta t^2}\dfrac{<n^*>_i-n^0}{n^0}$	$<\nabla^2P^{k+1}>_i=(1-\gamma)\dfrac{\rho}{\Delta t}\nabla\cdot V_i^* -\gamma\dfrac{\rho}{\Delta t^2}\dfrac{<n^*>_i-n^0}{n^0}$
Free Surface Detection	$<n>_i^* \;<\; \beta\cdot n^0$	$<\boldsymbol{F}>_i=\dfrac{D}{n^0}\sum_{j\neq i}\dfrac{1}{\lvert \boldsymbol{r}_i-\boldsymbol{r}_j\rvert}(\boldsymbol{r}_i-\boldsymbol{r}_j)W(\boldsymbol{r}_{ij})$ $<\lvert \boldsymbol{F}\rvert>_i \;>\; 0.9\lvert F\rvert^0$

MPSGPU-SJTU Solver

Advantages of GPU

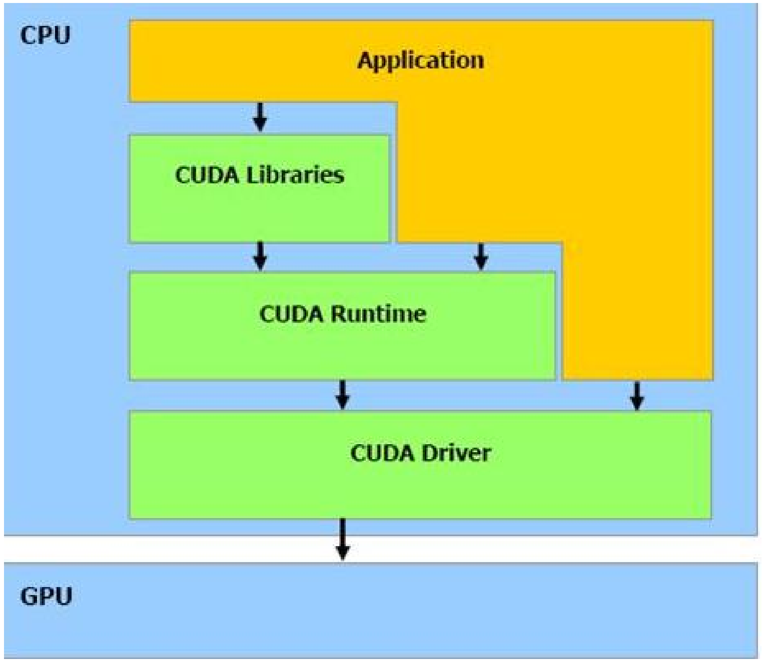
- **Possess more calculation threads to process data simultaneously.**
- **High FLOPS.**
- **Available on personal computers.**

MPSGPU-SJTU Solver

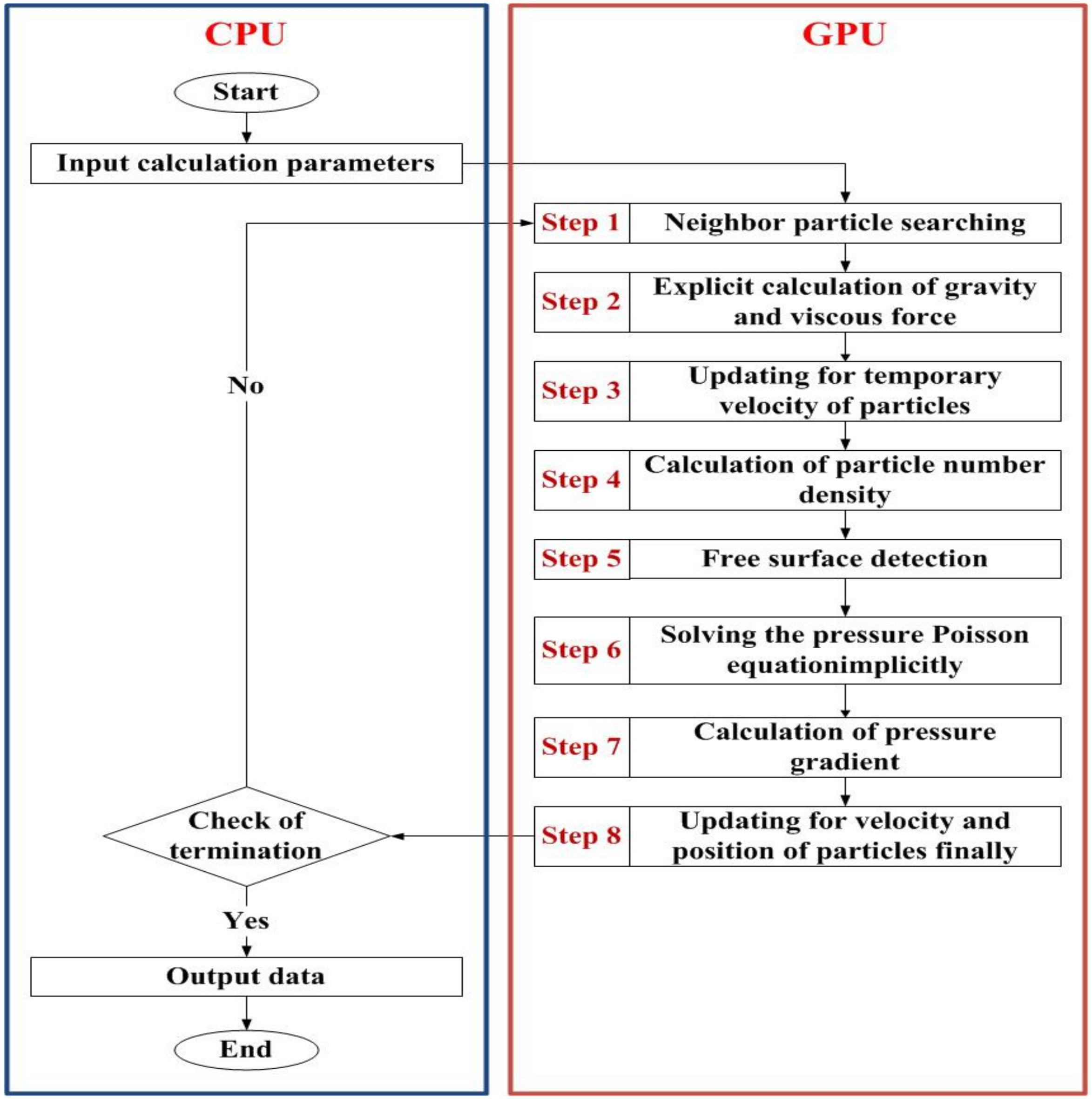
Features of CUDA:

- ✓ Applicable to NVIDIA gpus
- ✓ Based on C Language
- ✓ Easy to learn
- ✓ Abundant libraries
- ✓ Improved constantly

Framework of CUDA



MPSGPU-SJTU Solver



Contents

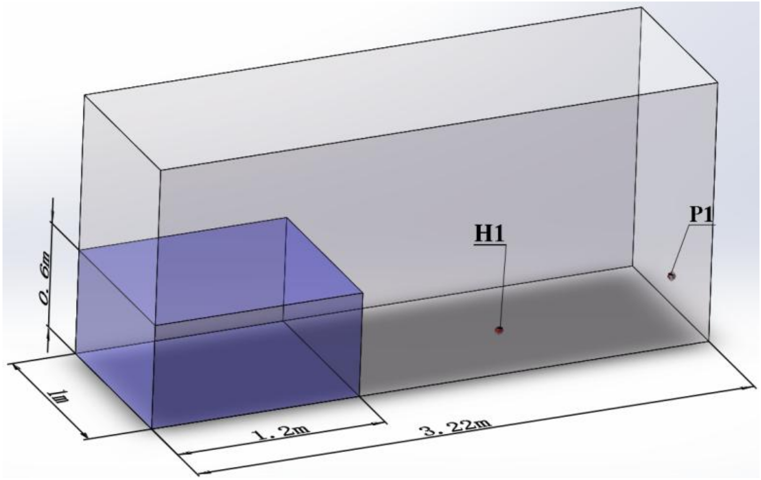
- Backgrounds
- Numerical methods
  - ✓ Modified MPS
  - ✓ MPSGPU-SJTU Solver
- **Numerical Simulation**
  - ✓ **Dam Break Flow**
  - ✓ **Liquid Sloshing**
- Conclusions and Ongoing Work

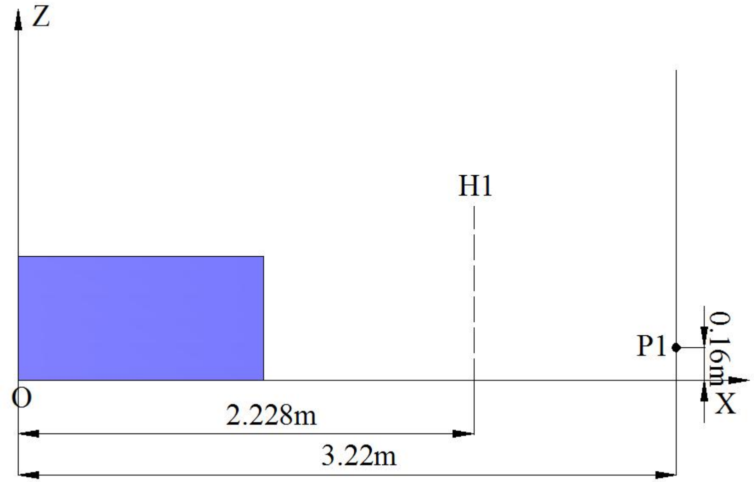
Numerical Simulation

Computational environment of CPU and CPU

	HPC	GPU
Card	Intel(R) Xeon(R) E5-2680 v2, 2.80 GHz	Tesla K40M
Memory	DDR3 1600, 16GB	12GB
Max Core	10	2880
Programming Language	C++	CUDA C/C++
Compiler	gcc, MVAPICH	CUDA 7.0 Cusp v0.5.1

Dam Break Flow





G. Colicchio, A. Colagrossi, M. Greco, and M. Landrini. "Free surface flow after a dam break a comparative study," 4th Numerical Towing Tank Symposium, Germany, September 2001.

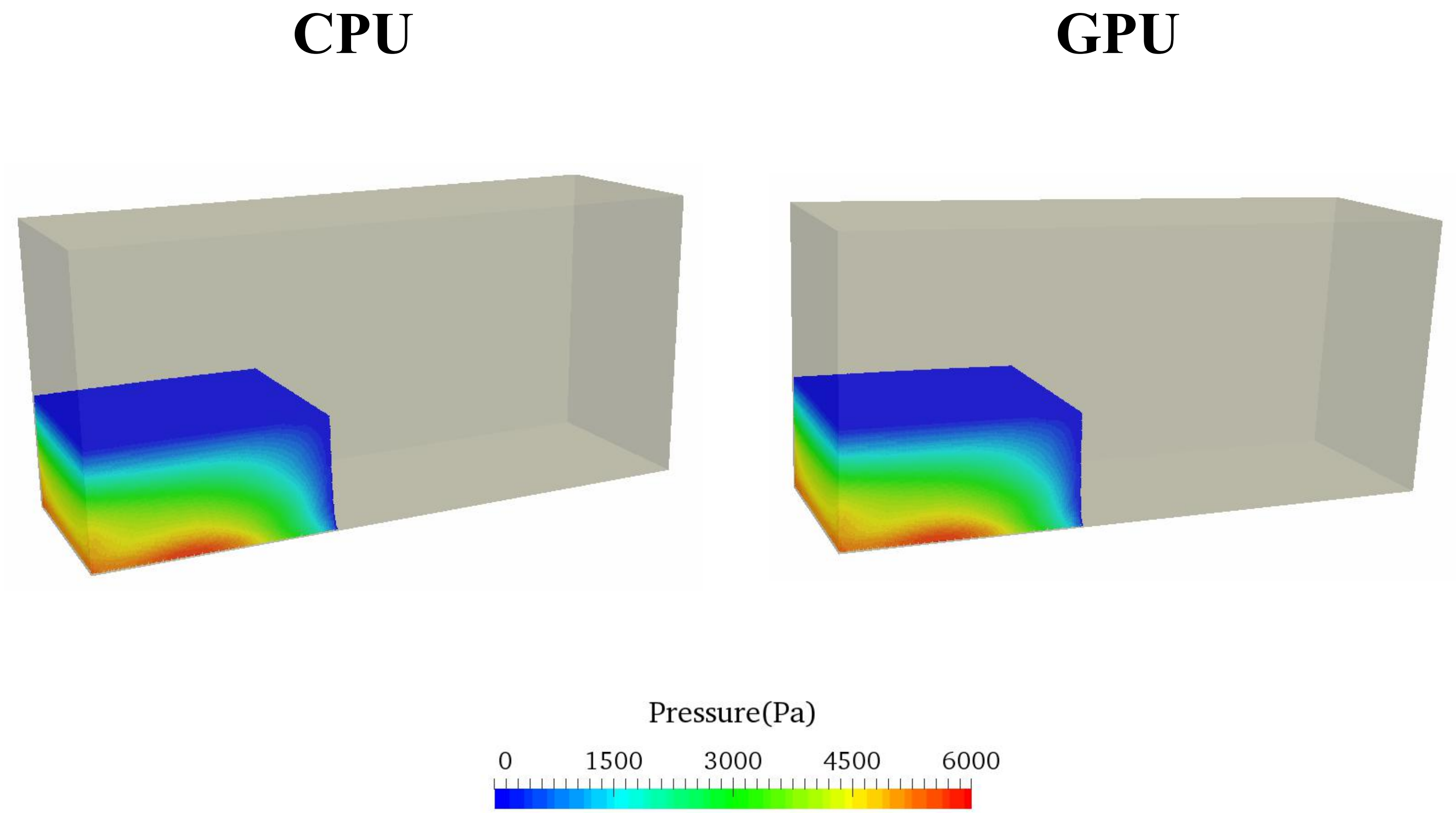
Parameters	Values
Fluid density	1000(kg/m^3)
Kinematic viscosity	1×10^{-6}(m^2/s)
Gravitational acceleration	9.81(m/s^2)
Particle spacing	0.01(m)
Fluid number	712800
Total number	1199205
Time Step	0.00025(s)

- **Computational Parameters**

	H1	P1
X(m)	2.228	3.22
Y(m)	0	0.5
Z(m)	0	0.16

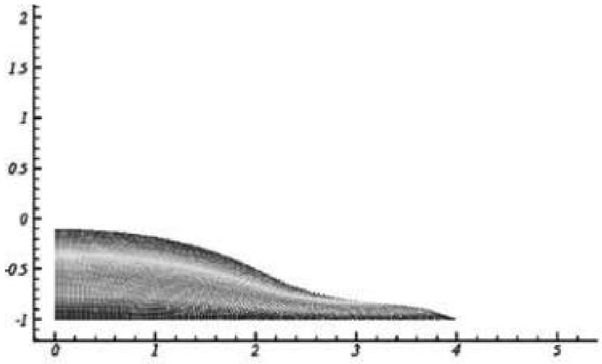
- **Arrangements of Probes**

Dam Break Flow

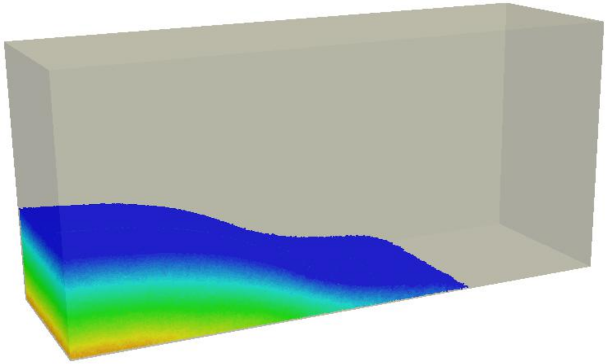


Dam Break Flow

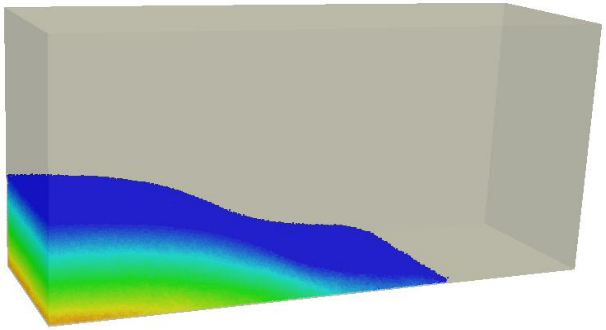
Exp.



SPH

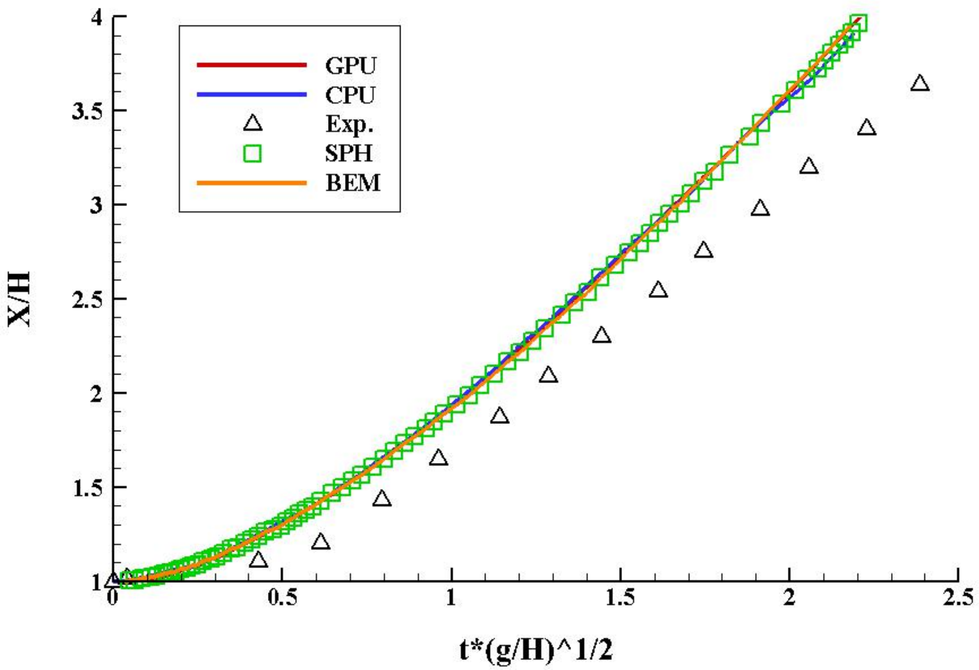


CPU



GPU

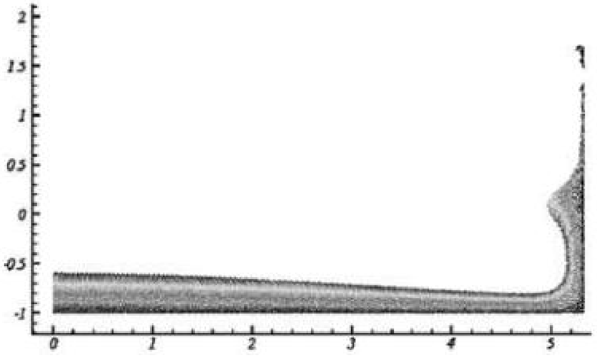
Dam Break Flow



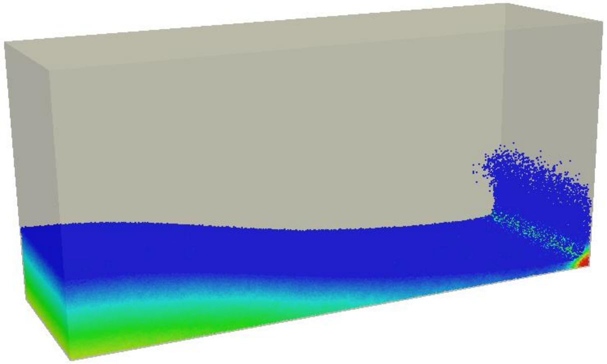
The water-front of GPU, CPU experiment, SPH and BEM

Dam Break Flow

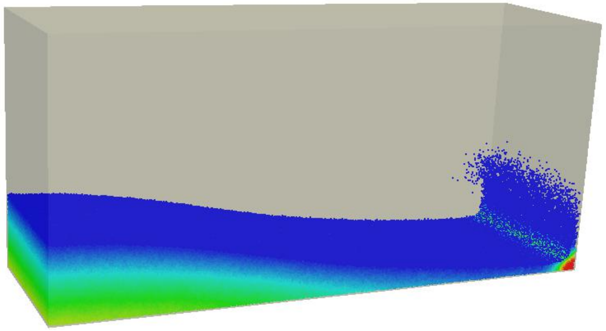
Exp.



SPH

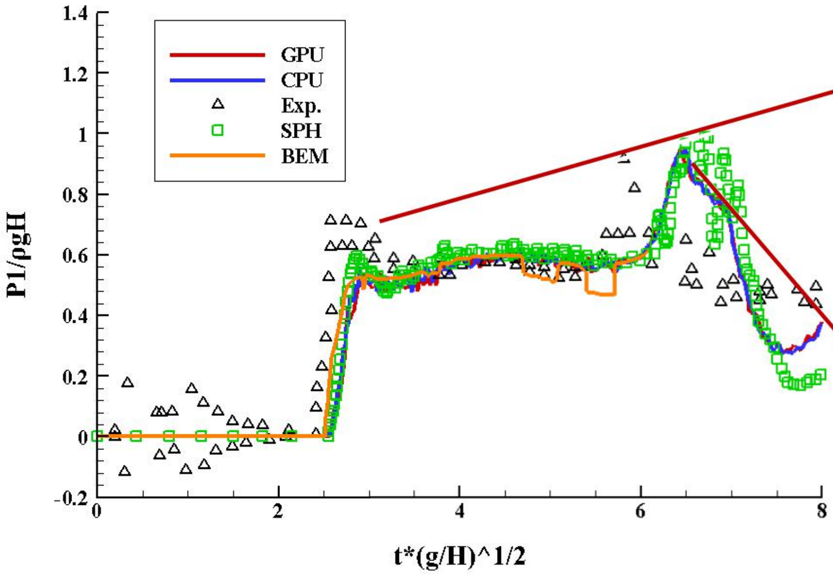


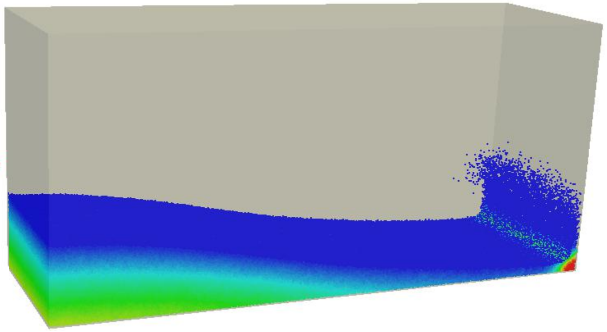
CPU

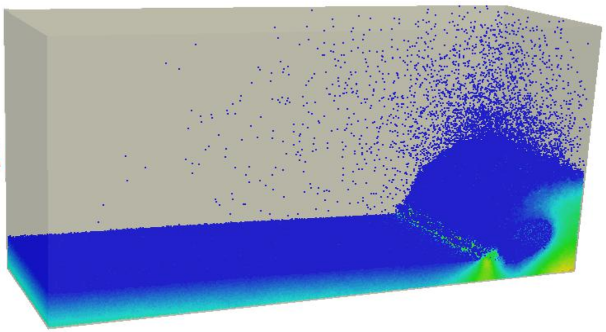


GPU

Dam Break Flow

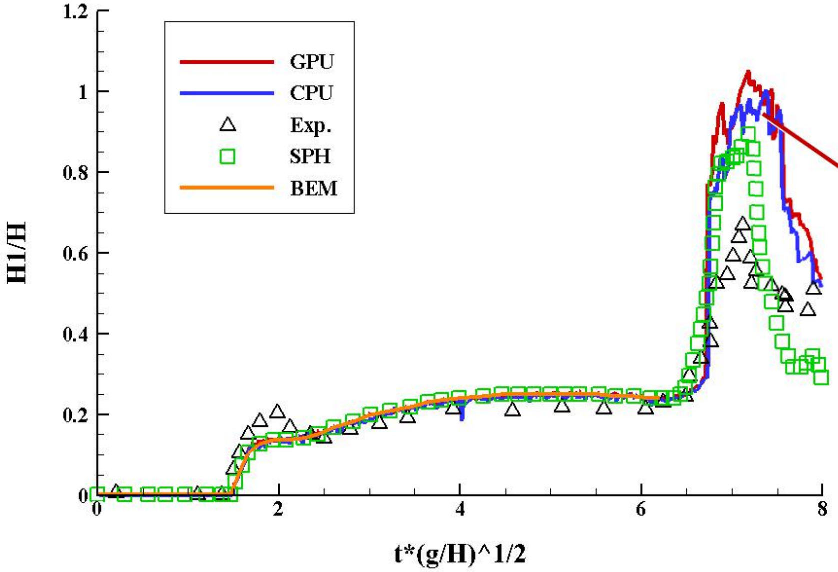


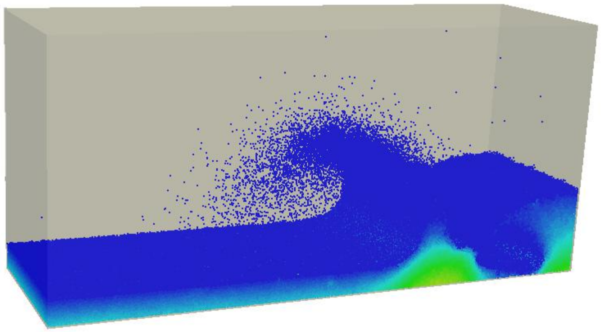




Impact pressure on wall

Dam Break Flow

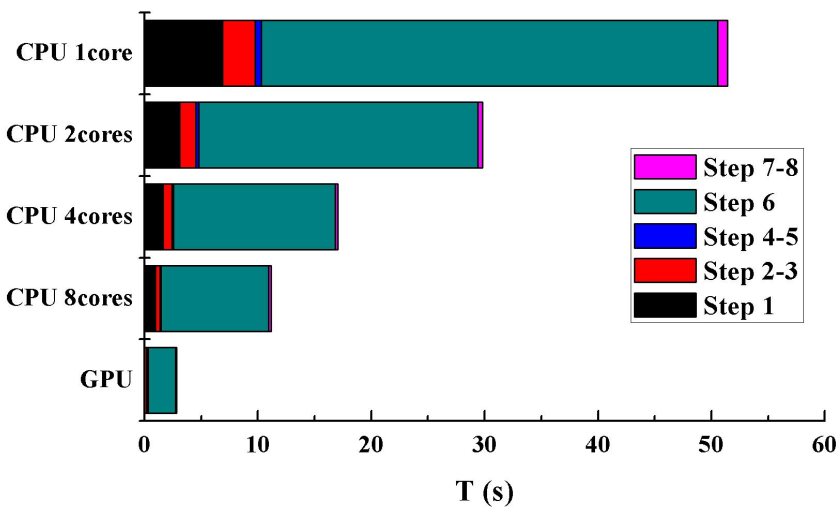


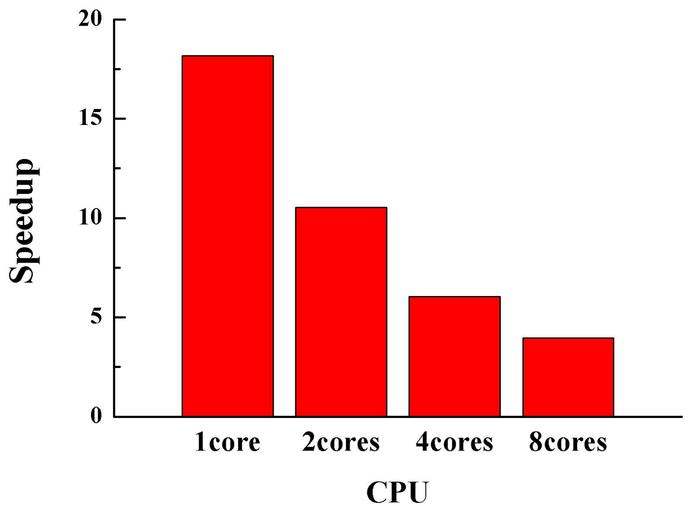


Wave height

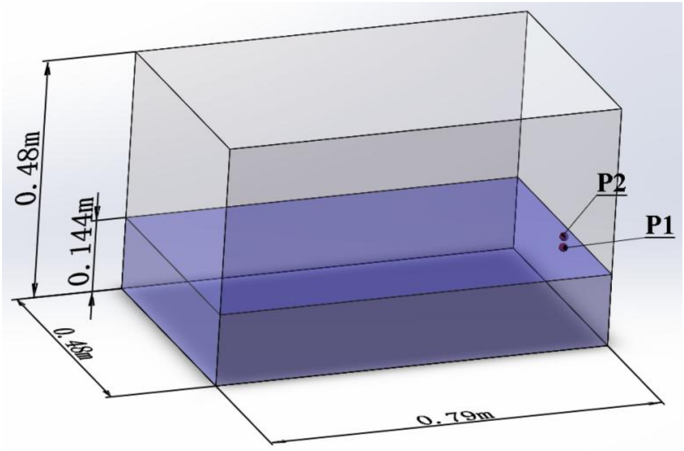
Dam Break Flow

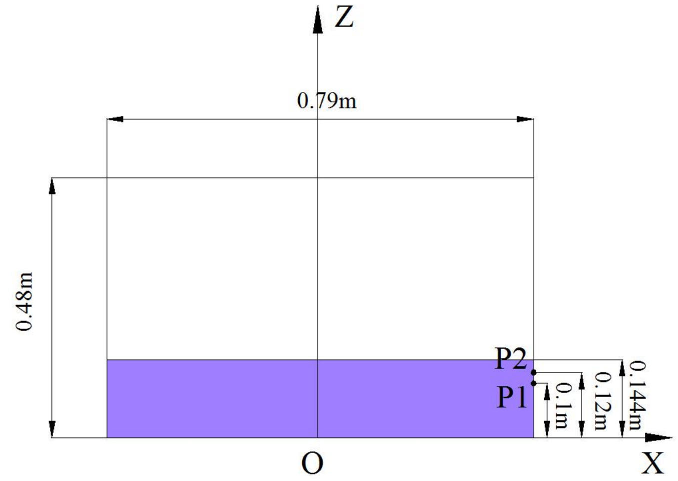
	Step 1 (s)	Step 2-3 (s)	Step 4-5 (s)	Step 6 (s)	Step 7-8 (s)
CPU 1core	6.886	2.870	0.562	40.258	0.825
CPU 2cores	3.096	1.437	0.245	24.609	0.423
CPU 4cores	1.641	0.754	0.140	14.277	0.257
CPU 8cores	0.944	0.436	0.087	9.494	0.213
GPU	0.004	0.202	0.101	2.433	0.089





Liquid Sloshing





Y. K. Song, K. A. Chang, Y. Ryu, and S. H. Kwon. "Experimental study on flow kinematics and impact pressure in liquid sloshing," Experiments in Fluids, vol. 54, pp.1–20, September 2013.

Parameters	Values
Surge Amplitude	0.0575(m)
Filling Ratio	30%
Excited Frequency	4.49(rad/s)
Particle spacing	0.005(m)
Fluid number	432535
Total number	678373
Time Step	0.0002(s)

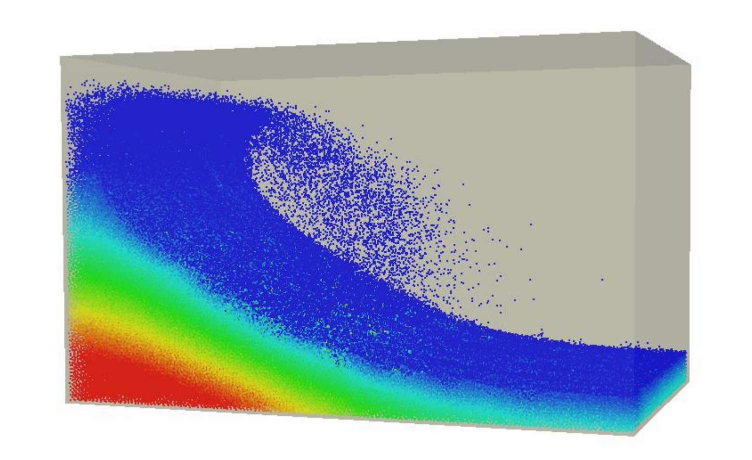
- **Computational Parameters**

	P1	P2
X(m)	0.395	0.395
Y(m)	0	0
Z(m)	0.1	0.12

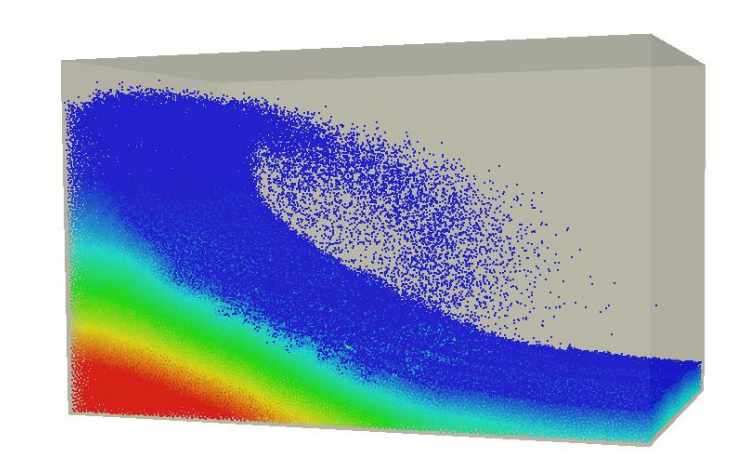
- **Arrangements of Probes**

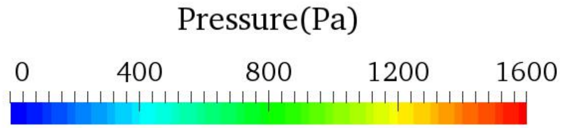
Liquid Sloshing

CPU



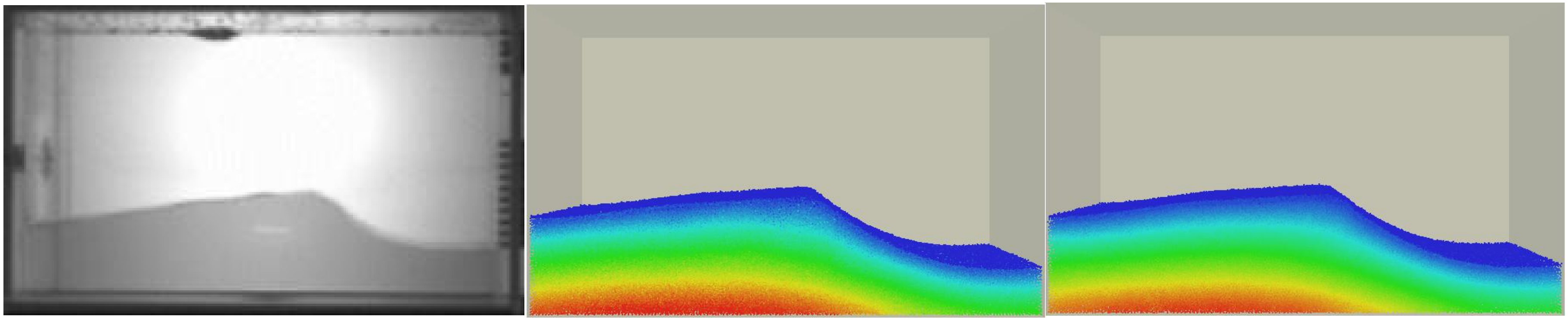
GPU



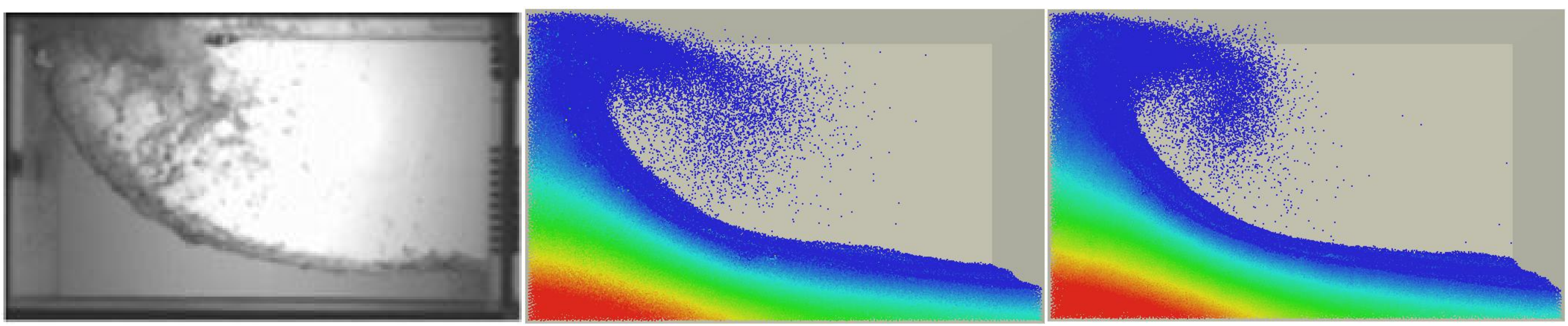


Liquid Sloshing

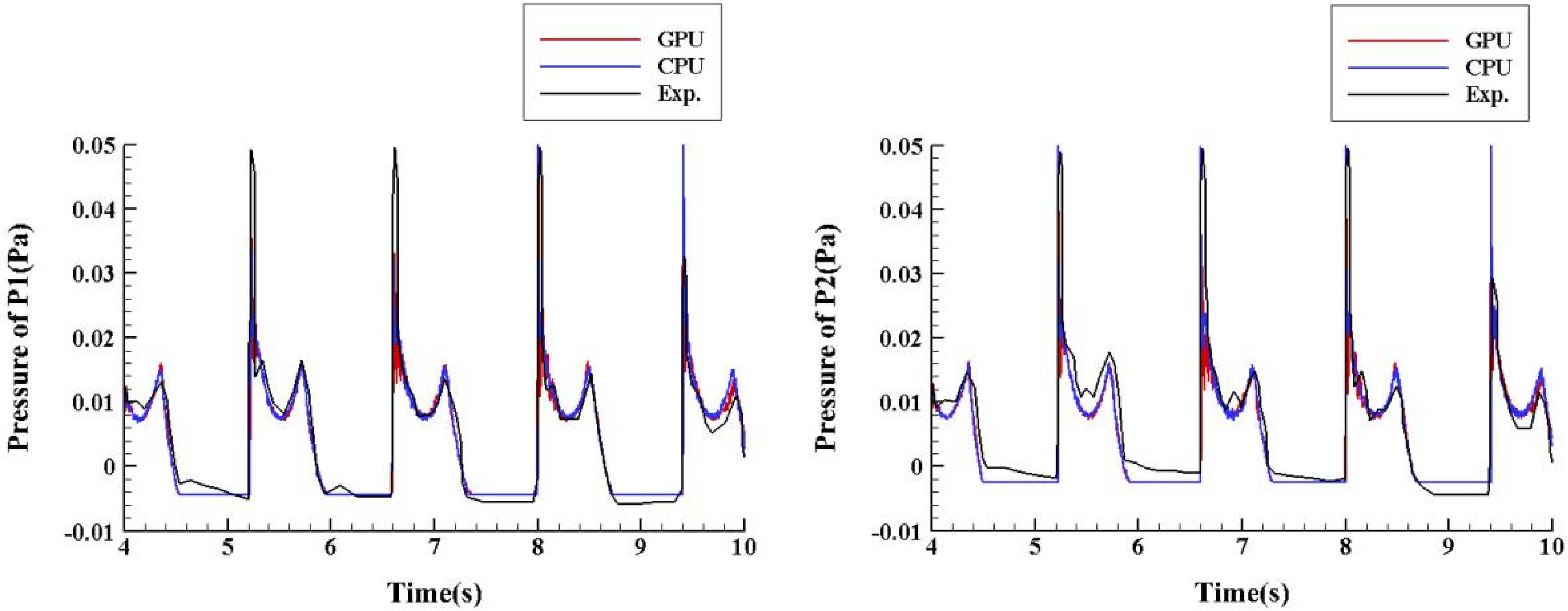
Exp.	CPU	GPU



Exp.	CPU	GPU

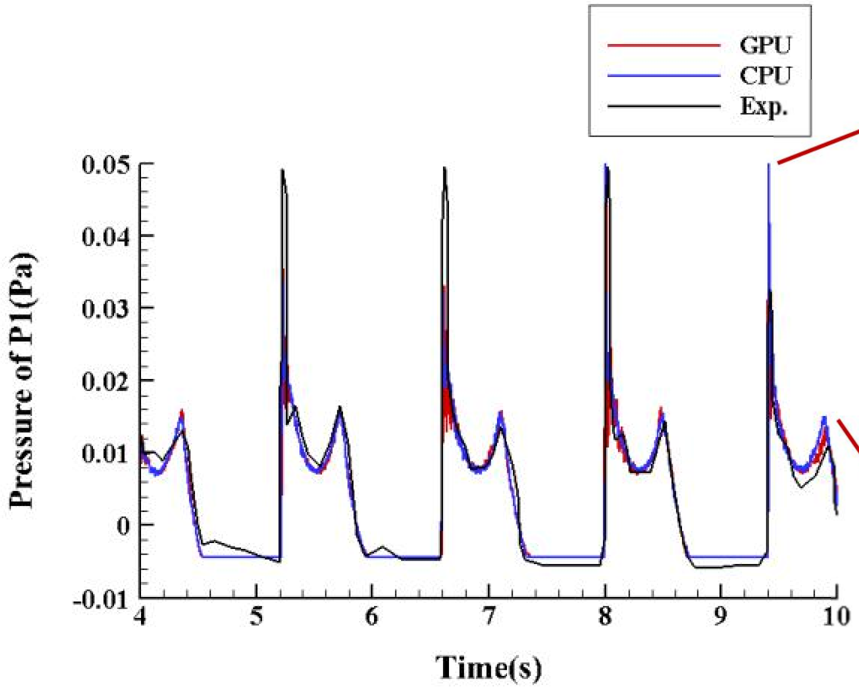


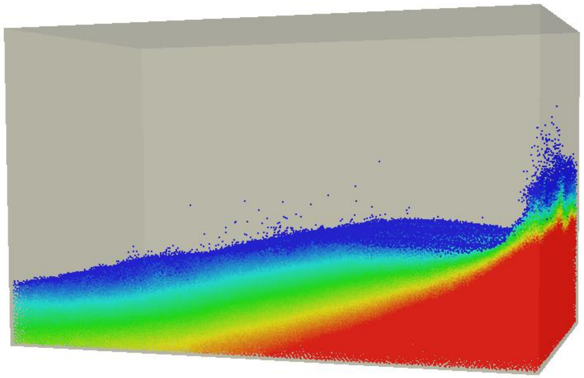
Liquid Sloshing

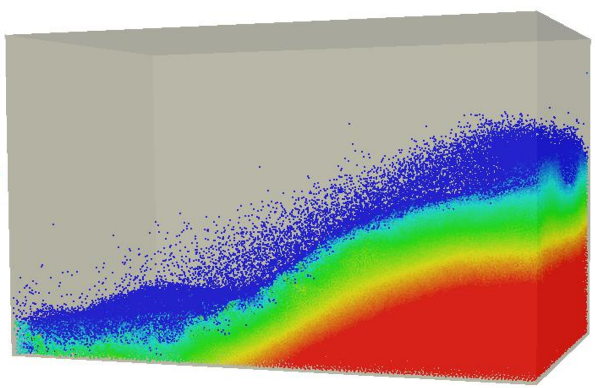


The Variation of Pressure on P1 and P2

Liquid Sloshing

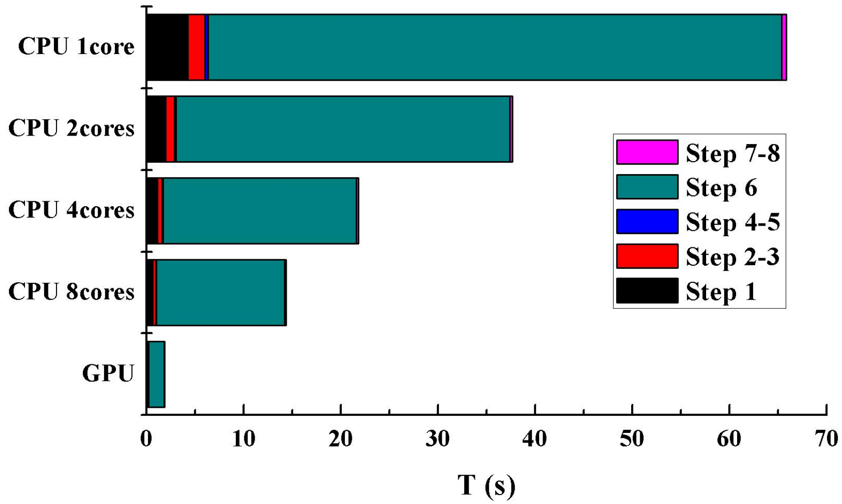


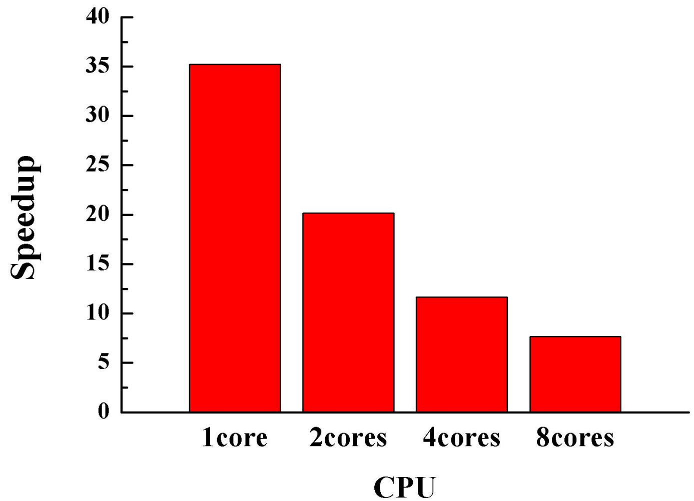




Liquid Sloshing

	Step 1 (s)	Step 2-3 (s)	Step 4-5 (s)	Step 6 (s)	Step 7-8 (s)
CPU 1core	4.257	1.735	0.341	59.037	0.498
CPU 2cores	1.990	0.880	0.162	34.365	0.276
CPU 4cores	1.108	0.513	0.094	19.890	0.179
CPU 8cores	0.637	0.328	0.058	13.166	0.185
GPU	0.003	0.126	0.062	1.622	0.058





Contents

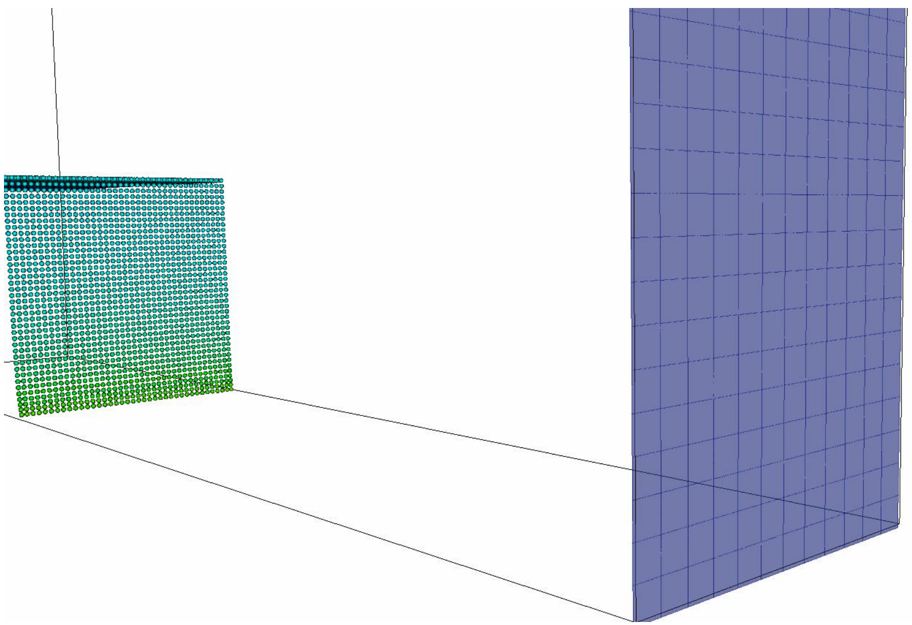
- Backgrounds
- Numerical methods
  - ✓ Modified MPS
  - ✓ MPSGPU-SJTU Solver
- Numerical Simulation
  - ✓ Dam Break Flow
  - ✓ Liquid Sloshing
- **Conclusions and Ongoing Work**

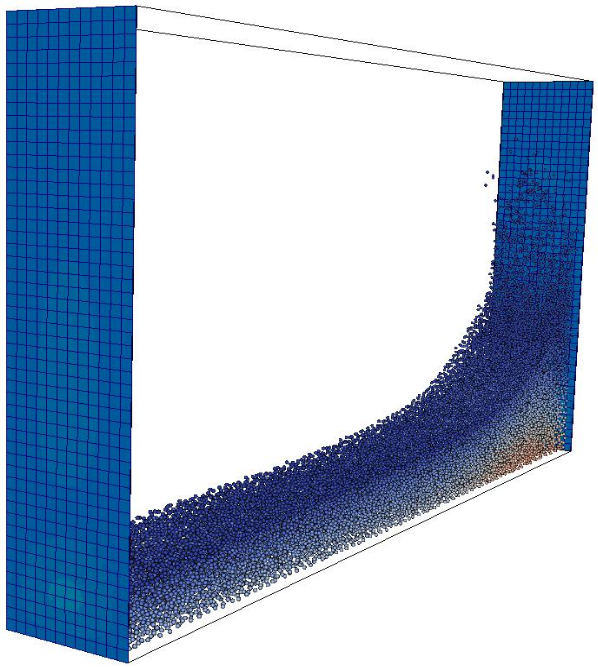
Conclusions

- MPSGPU-SJTU, an in-house solver based on GPU acceleration technique is developed to simulate the three-dimensional violent flows such as dam break and sloshing.
- The numerical results of GPU simulation shows a good agreement with CPU calculation, SPH, BEM and experiment.
- The speedup of every calculation step between GPU and CPU solvers is up to 35.

Ongoing Work

- Develop the module of fluid-structure interaction in GPU code.





Thank You！

CMHL Computational Marine Hydrodynamics Lab
Shanghai Jiao Tong University

http://dcwan.sjtu.edu.cn