GPU-based SPH modeling of flood with floating bodies in urban underground spaces

Jiansong Wu[1], Na Li[1], Hui Zhang[2], Robert A. Dalrymple[3]

[1]*Dept. of Safety Engineering, China University of Mining and Technology(Beijing)*
[2]*Institute of Public Safety Research, Tsinghua University, China*
[3]*Dept. of Civil Engineering, Johns Hopkins University*

SPHERIC Workshop，Beijing Oct. 17-20，2017

Outline

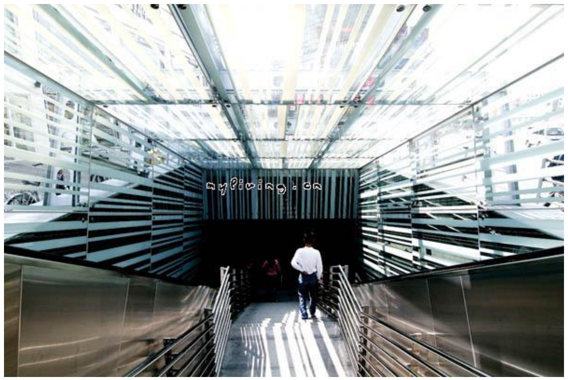
- Introduction
- Methodology
  - SPH fundamentals
  - GPUSPH model
- Results and discussions
- Conclusions

Outline

- **Introduction**
- Methodology
  - SPH fundamentals
  - GPUSPH model
- Results and discussions
- Conclusions

Background: catastrophic urban underground floods

- **Underground exploitation:** *malls, subways, car parking lots, etc.*



- **Flooding-prone:** *"innate" low-lying weakness*

New York, USA (July, 2005)

Beijing, China (July, 2012)

New York, USA (October, 2012)

Background: urban flood with floating bodies

Quebec, Canada(May, 2011)

Bangkok, Thailand (October, 2011)

Beijing, China (July 21, 2012)

Floating objects

Miyagi, Japan(March, 2011)

New York, USA (October, 2012)

Challenges: grid-based numerical modeling

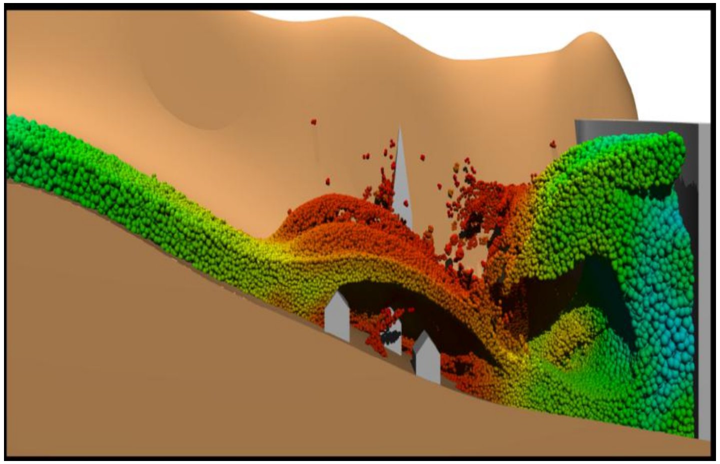
- Simplified 2-D models: depth-averaged Shallow-Water equations
  - Underlying assumption — hydrostatic pressure distribution
  - Water depth *h* — explicit
  - *Zoppou et al(1999), Mignot(2006), Soares et al(2008), Jeong et al(2012), Ishigaki(2013)*

- Fully 3-D models: Navier-Stokes equations
  - Water depth *h* — implicit —> need to track free surface:
    - VOF(Volume of Fluid): *Hatice et al (2010), Nam et al.(2010)*
    - LSM(Level Set method): *Nagata et al(2005), Gu et al(2009)*

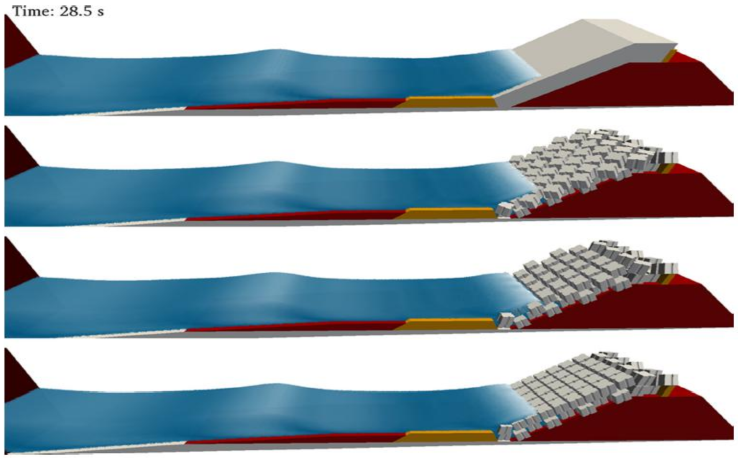
- **Drawbacks of grid-based models for simulating urban underground flooding**
  - Depth-averaged assumption limitation(2-D);
  - Difficult to track free surface for violent underground flooding flow(3-D);
  - **Difficult to handle floating bodies(2-D&3-D).**

Advantages: lagragian meshfree method(SPH)

- <u>Smoothed Particle Hydrodynamics(SPH)</u>
  - Lagragian meshfree method;
  - **<u>Attractive feature</u>**: the ability to handle very complex surface evolution, without any special surface tracking treatment
  - Fairly well deal with fluid-structure interactions: flows with fixed walls, floating bodies



Adam et al. (2007)



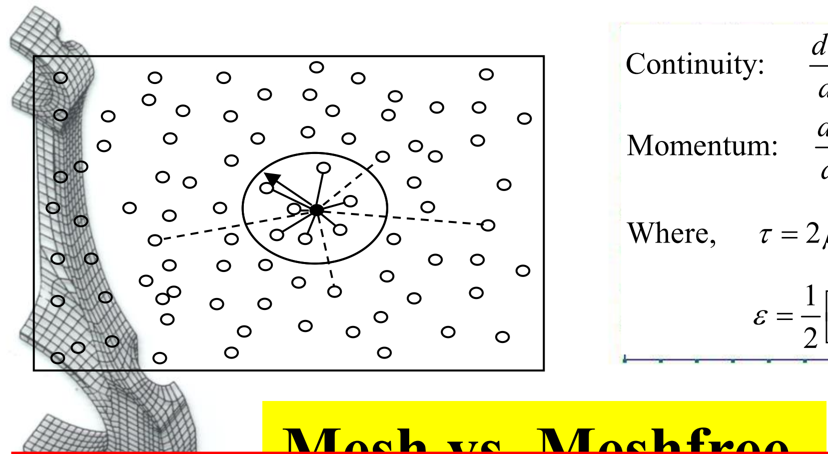
Altomare et al. (2014)

Outline

- Introduction
- **Methodology**
  - SPH fundamentals
  - GPUSPH model
- Results and discussions
- Conclusions

SPH fundamentals

- A **fully Lagrangian meshfree** CFD technique
- Fluid is represented by a set of particles each **with a given mass and velocity,** Navier-Stokes equations in SPH form are derived based on **integral and interpolant theory:**



Continuity: $$\frac{d\rho}{dt} = -\rho \nabla \cdot \vec{v}$$

Momentum: $$\frac{d\vec{v}}{dt} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \tau + \vec{f}$$

Where, $$\tau = 2\mu\varepsilon - \frac{2}{3}\mu\left(\nabla \cdot \vec{v}\right) I$$

$$\varepsilon = \frac{1}{2}\left[\nabla \vec{v} + \left(\nabla \vec{v}\right)^T\right]$$

Continuity: $$\frac{d\rho_i}{dt} = \sum_{j=1}^{N} m_j \left(\vec{v}_i - \vec{v}_j\right) \cdot \nabla_i W_{ij}$$

Momentum: $$\frac{d\vec{v}_i}{dt} = -\sum_{j=1}^{N} m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right) \nabla_i W_{ij} + \sum_{j=1}^{N} m_j \left(\frac{\mu_i \varepsilon_i}{\rho_i^2} + \frac{\mu_j \varepsilon_j}{\rho_j^2}\right) \cdot \nabla_i W_{ij} + \vec{f}$$

Where $$\varepsilon_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} \vec{v}_{ji} \nabla_i W_{ij} + \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left(\nabla_i W_{ij}\right) \vec{v}_{ji} - \left(\frac{2}{3} \sum_{j=1}^{N} \frac{m_j}{\rho_j} \vec{v}_{ji} \cdot \nabla_i W_{ij}\right) I$$

GPUSPH model

- SPH is **computationally expensive**, but computationally intensive
  - Particularly appropriate for employing GPU parallel computing
- GPUSPH/GPUSPHysics is programmed by CUDA, C++, OpenGL:

Home News Download/Install Developers Applications Test Cases References

GPUSPH: SPH free surface flow model for GPUs

Table 2 Speed-ups by GPU type and code procedure for GPU-SPHysics code for 677,360 particles

Procedure	GPU (# of processors)		
	8600 m (32)	8800 GT (110)	GTX 280 (240)
Neighbor list	4.4	11.8	$\boxed{15.1}$
Force calculation	28	120	$\boxed{207}$
Euler step	3.2	13.7	$\boxed{23.8}$

Hérault, A., G. Bi
Hydraulic Resea

Herault and Dalrymple et al.(2010)

GPUSPH model

- **Updated to Version** 4.1
  - Multi-GPUs, Multi-nodes/GPU clusters

GPUSPH

le cnam · Johns Hopkins Whiting School of Engineering · eDF · Laboratoire d'Hydraulique Saint-Venant · INGV

About

Features

Downloads

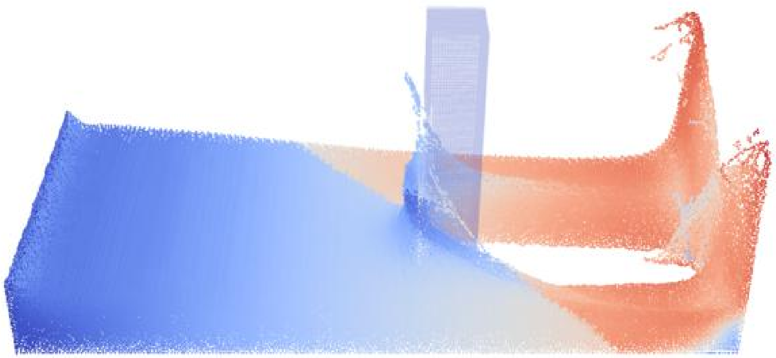
Documentation

Gallery

References

Contact

GPUSPH v4.1



The Smoothed Particle Hydrodynamics (SPH) method is a Lagrangian method for fluid flow simulation. In SPH the continuous medium is discretised into a set of particles that interact with each other and move at the fluid's velocity.

Posts

Jul 19, 2017
GPUSPH v4.1 released
GPUSPH version 4.1 is out! Head over to the release tag on GitHub.

Oct 19, 2016
GPUSPH v4 released
GPUSPH version 4.0 is out! Head over to the release tag on GitHub.

Jun 3, 2014
GPUSPH v3 released
GPUSPH version 3.0 is out! Head over to the release tag on GitHub.

May 31, 2014

info@gpusph.org

GPUSPH

GPUSPH

GPUSPH was the first implementation of SPH to run entirely on GPU with CUDA and aims to provide a basis for cutting edge SPH simulations

GPUSPH model

- **More features implemented:**

SPH formulations
- WCSPH single-fluid
- WCSPH multi-fluid
- WCSPH multi-fluid based on Grenier's formulation

Time schemes
- Symplectic second order Euler scheme

Kernels
- Cubic spline
- Quadratic
- Wendland
- Gaussian

Time step
- Fixed
- Adaptive

Parallelism
- MultiGPU
- Multinode
- Arbitrary domain split
- Load balancing
- Asynchronous computations

Open boundaries (only with SA boundaries)
- User defined velocity inlet/outlet conditions
- User defined pressure inlet/outlet conditions

Fluid-solid interactions
- Floating objects
- Objects with joints
- Moving objects (imposed motion)
- Collision detection
- Planes

Viscosity formulations
- Arithmetic mean (in the viscous operator)
- Harmonic mean (in the viscous operator)

Post-processing
- Vorticity computation
- Probes (TESTPOINTS)
- Surface detection
- Flux computation
- Energy computation
- User defined (CALC_PRIVATE)

Stabilisation procedures
- Artificial viscosity
- XSPH
- Ferrari correction
- Moving least-squares
- Shepard smoothing

Turbulence models
- k-epsilon (only with SA boundaries)
- LES (only with dynamic boundaries)

Time step
- Fixed
- Adaptive

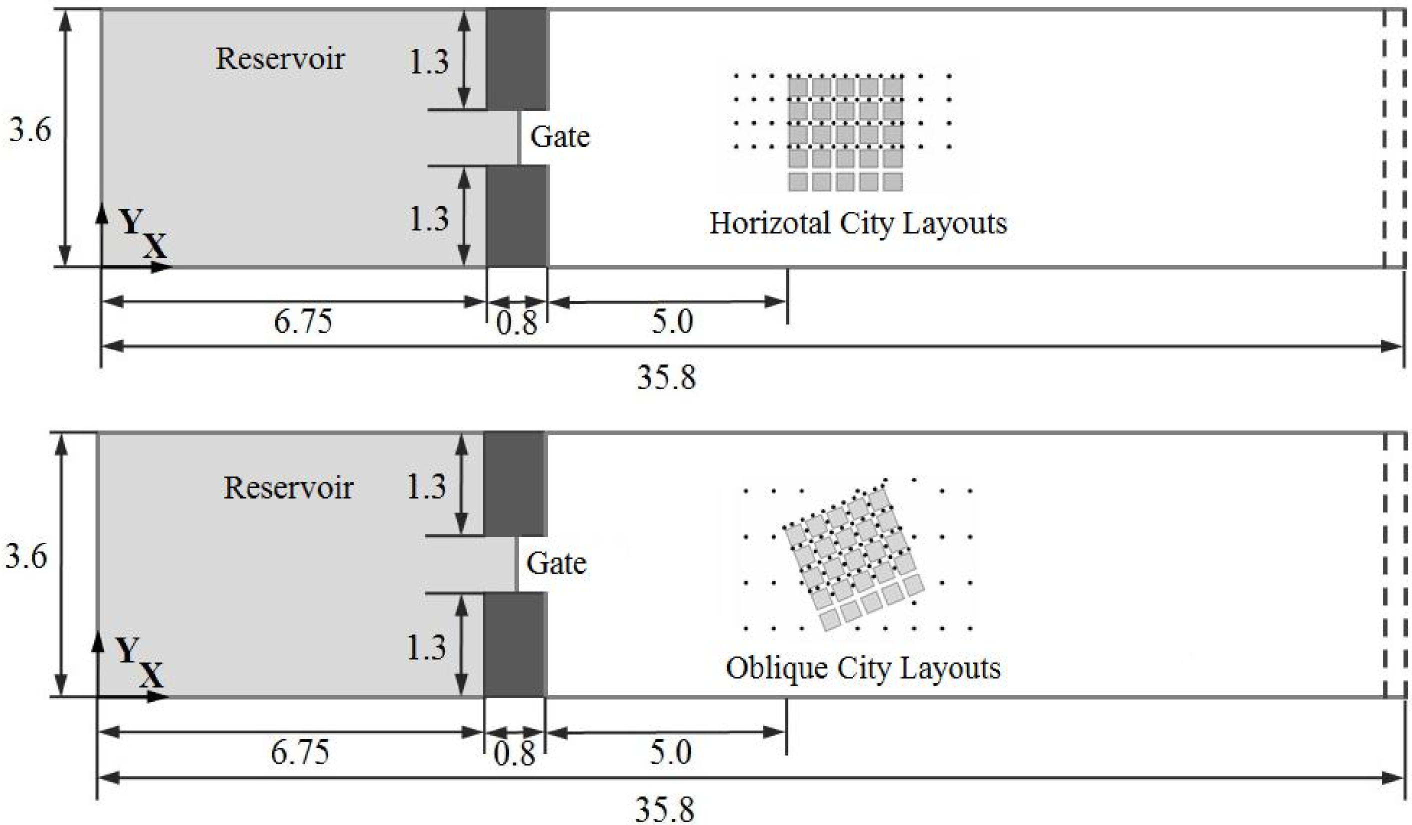
- **GPUSPH website:** **www. gpusph.org**

Outline

- Introduction
- Methodology
  - SPH fundamentals
  - GPUSPH model
- **Results and discussions**
- Conclusions

1. Dam-break flow through idealized city layouts

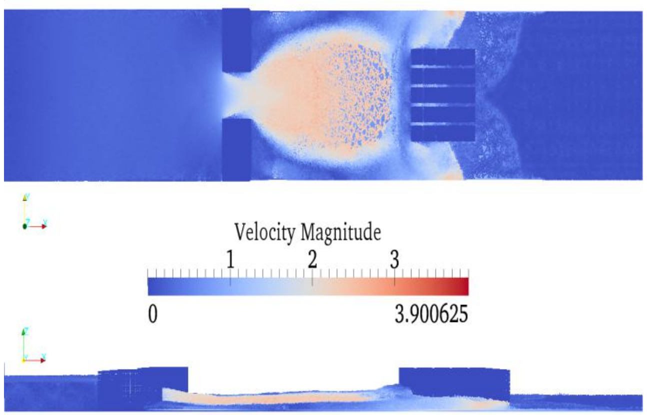
- Laboratory experiment set-ups:



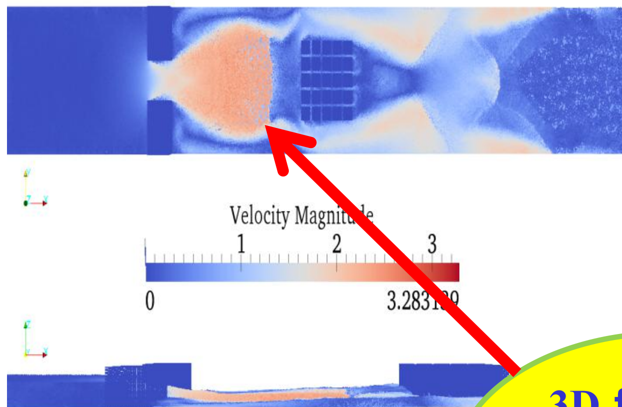
Soares-Frazao and Zech(2008)

1. Dam-break flow through idealized city layouts

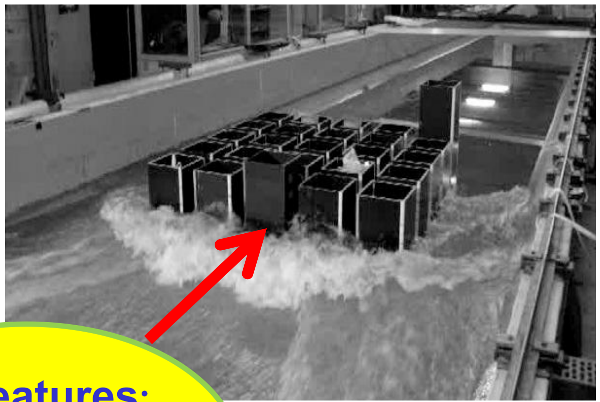
- Flow regime comparisons



calculated velocity field(t = 4.0 s)

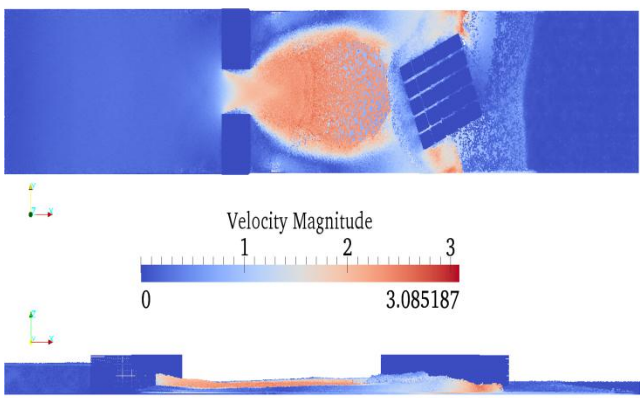


calculated velocity field(t

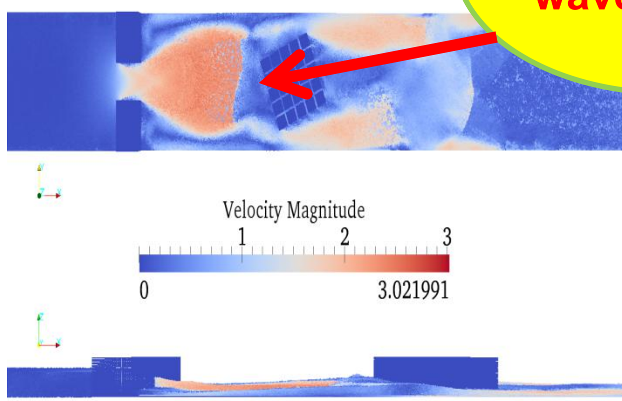


shot (horizontal cas

3D features:
hydraulic jumps,
wave zones



calculated velocity field(t = 4.0 s)

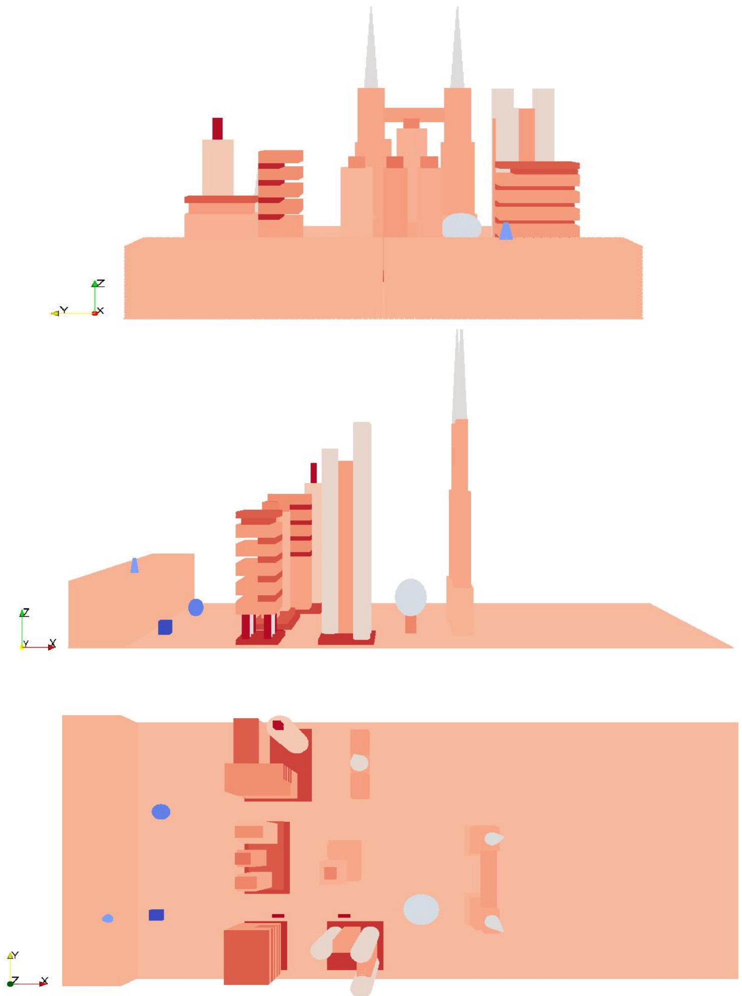


calculated velocity field (t = 8.0 s)

lab snapshot (oblique case)

2. Flooding with floating bodies through surface city layouts

- Simulation set-ups



Initial simulation conditions:

1 GPU(Tesla C2075);

3 floating objects;

Particle spacing=0.02m;

dt=0.00003s;

Total particles: 1,079,334.

- **Implementation of floating objects**:
  - moving rigid objects with translation and rotation(*Monaghan,2003,2005*)

The force adding on boundary particles: $\vec{f}_k = \sum_{a=1}^{N_{WP}} \vec{f}_{ka}$

Newton's third law: $m_k \vec{f}_{ka} = -m_a \vec{f}_{ak}$

Motion and rotation equations of boundary particles of floating bodies:

$$I\frac{d\vec{\Omega}}{dt} = \sum_{k=1}^{N_{BP}} m_k (\vec{r}_k - \vec{R}_0) \times \vec{f}_k$$

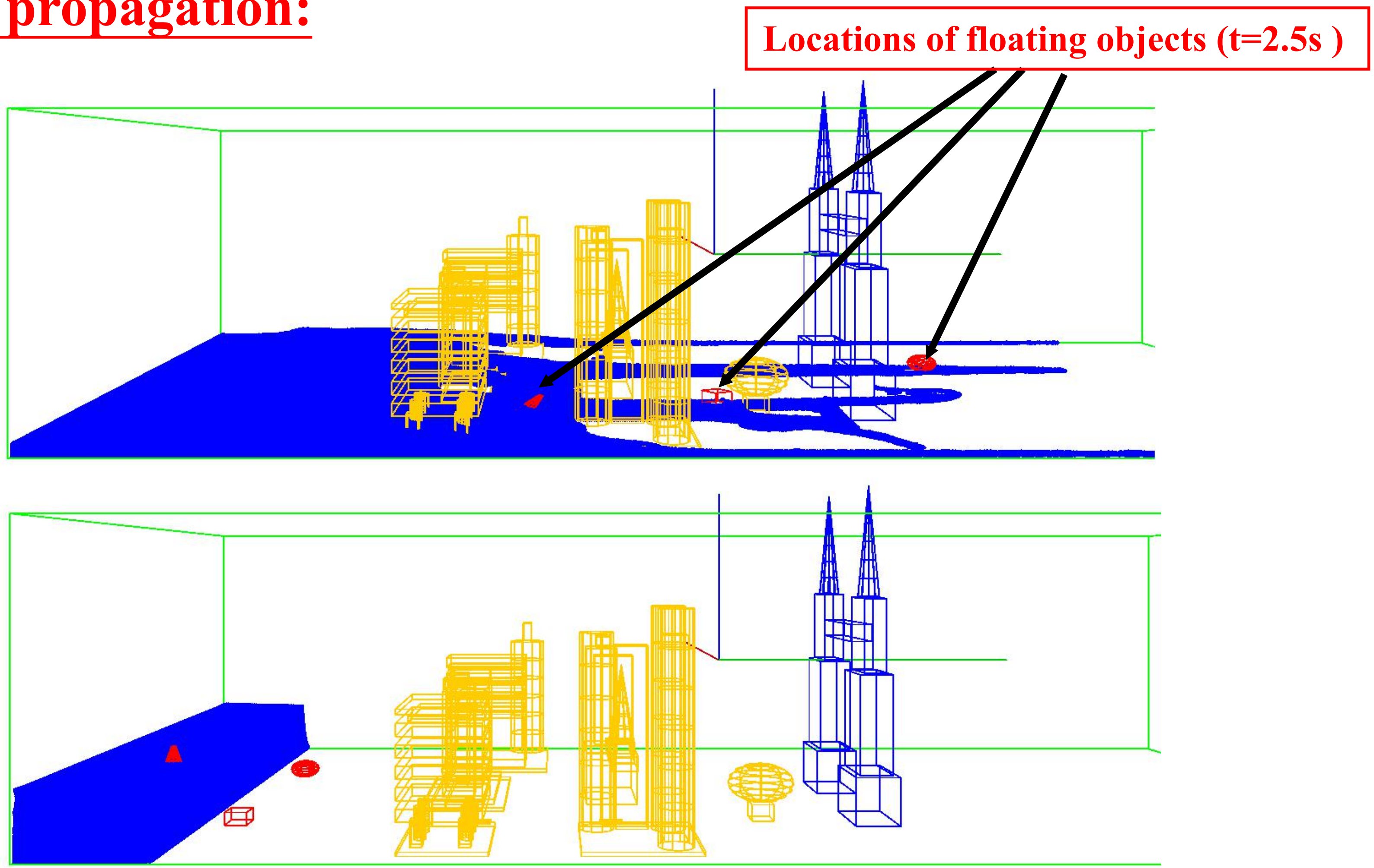
$$M\frac{d\vec{v}}{dt} = \sum_{k=1}^{N_{BP}} m_k \vec{f}_k$$

Velocity of boundary particles of floating bodies: $\vec{u}_k = \vec{v} + \vec{\Omega} \times (\vec{r}_k - \vec{R}_0)$

Boundary particles of floating bodies moves: $\frac{d\vec{x}_k}{dt} = \vec{u}_k$

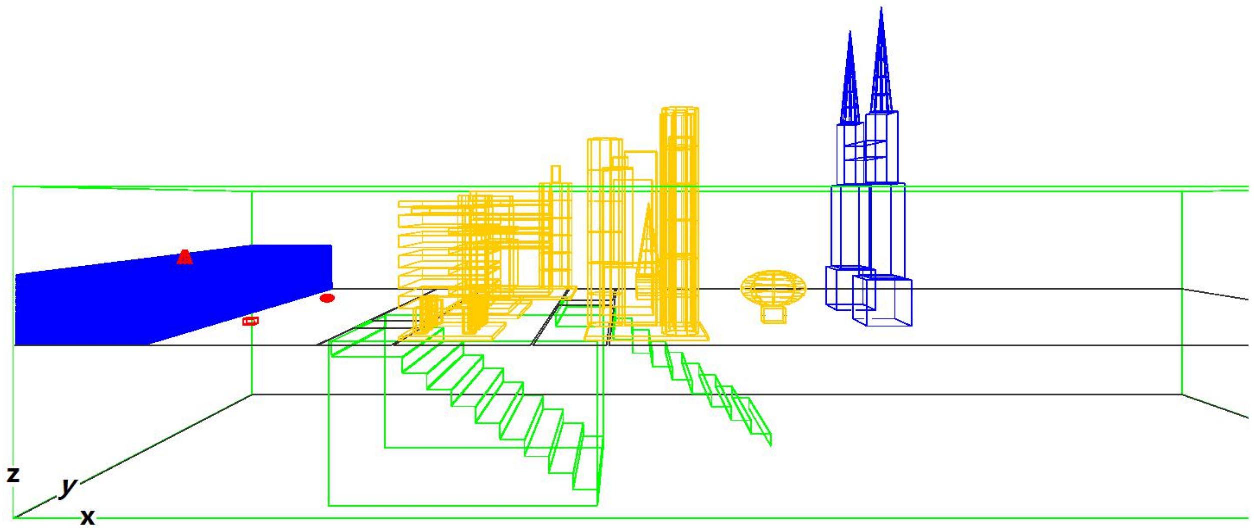
2. Flooding with floating bodies through surface city layouts

- **Flood propagation:**



3. Flooding with floating objects into building basements

- Simulation set-ups:
- **One-floor basement, two entrances each with a 10-step staircase**



Initial simulation conditions:

Computational domain: 12m×5.4m×4.0m;

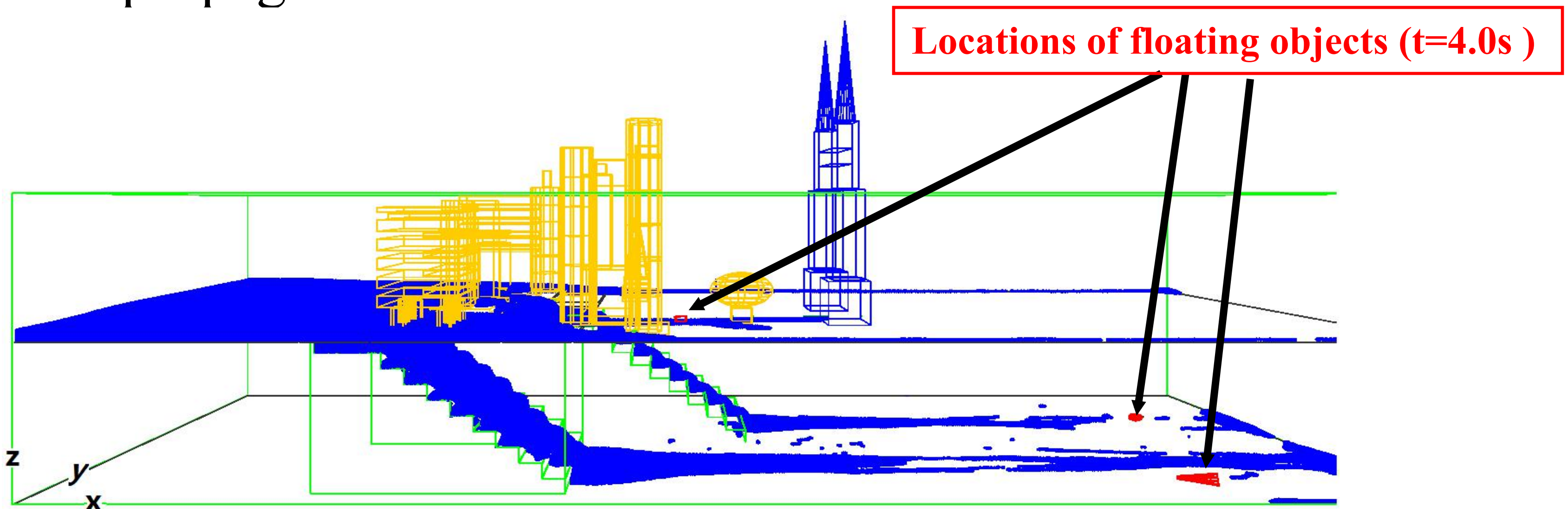
Particle spacing=0.02m； Total particles: 1,348,488;

dt = 0.00003s (variable time step); 1 GPU (Tesla C2075）;

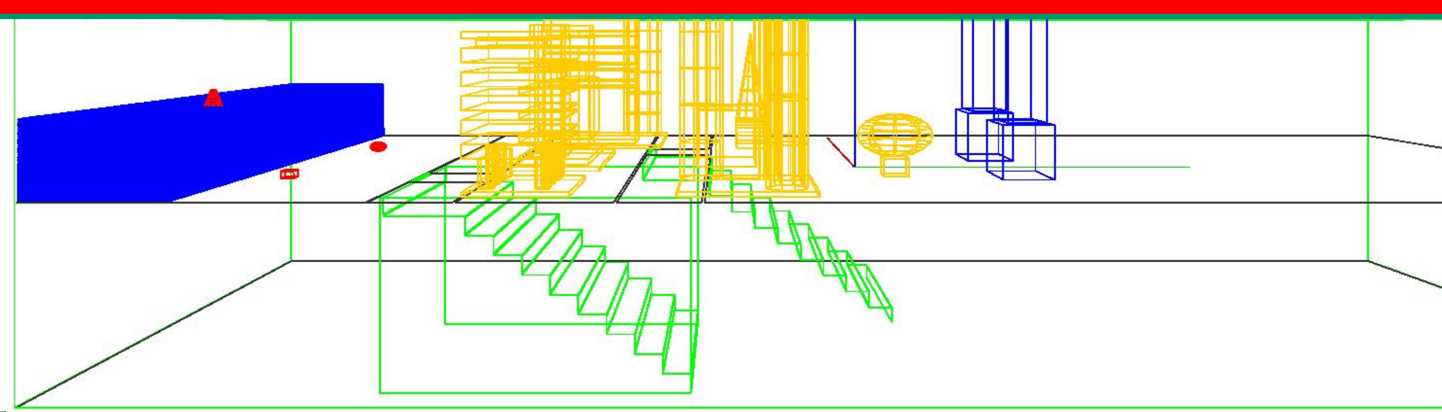
Physical time: 4 s; Computation time: 1047.43 s

3. Flooding with floating objects into building basements

- Flood propagation：

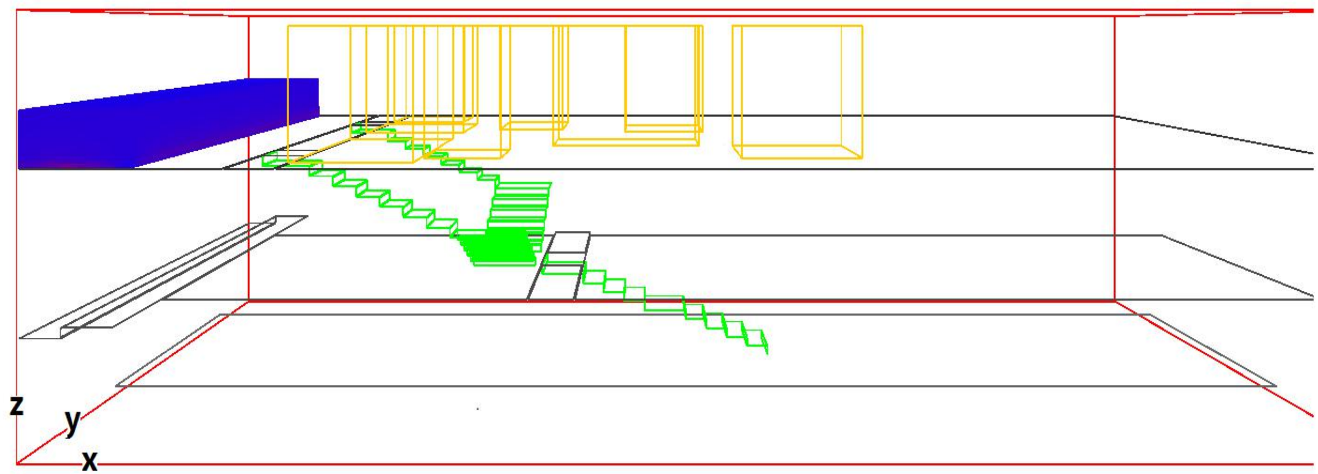


<u>Significance:</u> track the flooding path and analyze the complex flow features



4. Flooding in a two-storey underground space

- Simulation set-ups:
- **Two-storey basement, two entrances each with a 90-degree turning staircase**



Initial simulation conditions:

Computational domain: 16.0m×8.0m×6.0m;

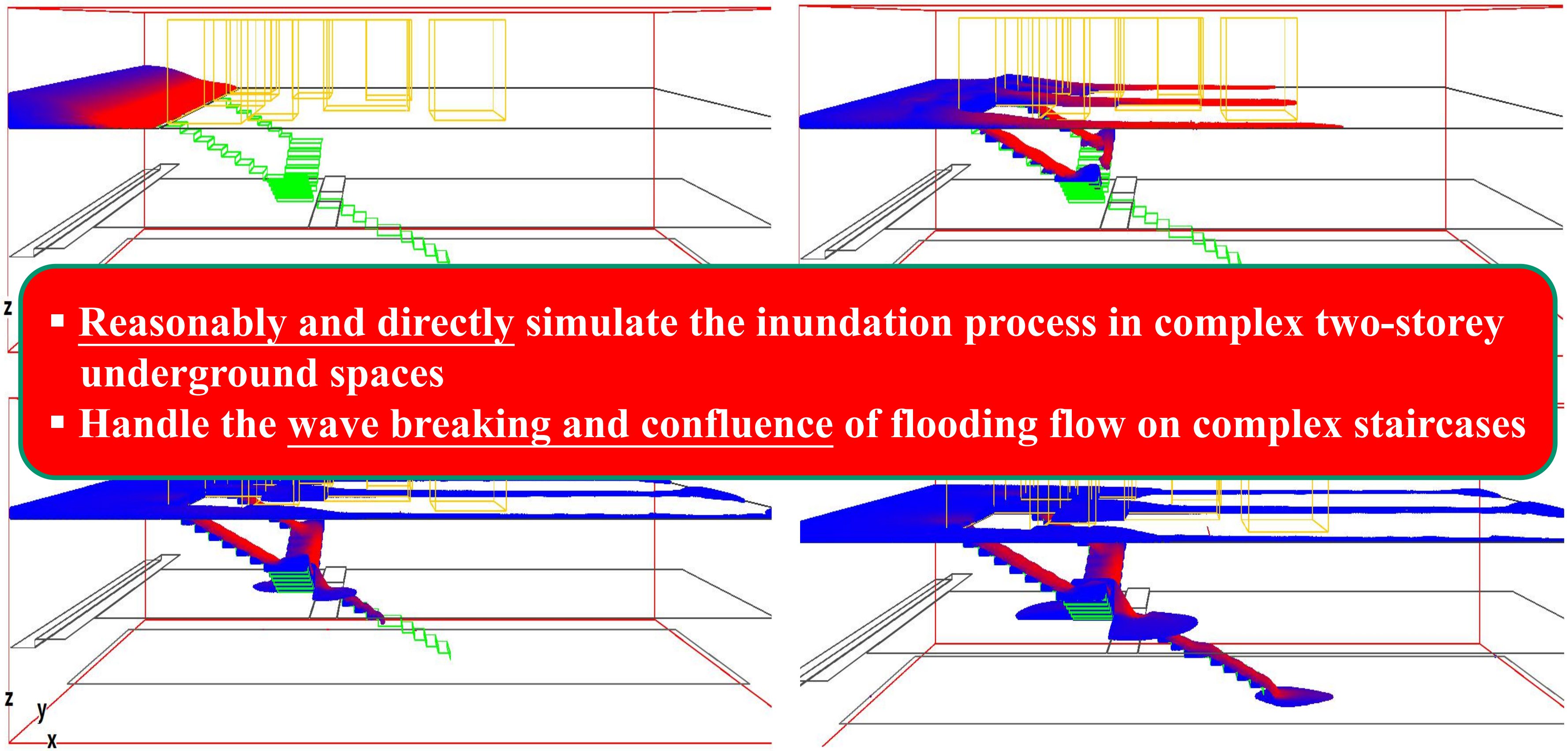
Particle spacing=0.02m； Total particles: 2,904,901;

dt = 0.00003s (variable time step); 1 GPUs (Tesla C2075）;

Physical time: 10 s; Computation time: 1635.94 s

4. Flooding in a two-storey underground space

- Flood propagation:



- **Reasonably and directly simulate the inundation process in complex two-storey underground spaces**
- **Handle the wave breaking and confluence of flooding flow on complex staircases**

Velocity field of flooding propagation（t=0.5 s, 2.0 s, 6.0 s,10.0 s ）

Outline

- Introduction
- Methodology
  - SPH fundamentals
  - GPUSPH model
- Results and discussions
- **Conclusions**

Conclusions and future work

- Reasonable results
  - Reasonably and directly simulate the inundation process in complex undergr ound spaces
  - Handle wave breaking and confluence of flooding flow on complex staircases
  - Incorporate floating objects in flooding flow: fairly well simulate interactions between moving objects and flooding flow, and also interactions between floating objects and buildings, infrastructures, fixed walls

- More validations and improvements
  - More quantitative comparisons with other numerical methods, experimental d ata, or underground inundation events will be conducted.
  - Handling flooding in LARGE-SCALE geometrically complex urban undergrou nd facilities will be further developed.

Thanks for your attention！