IMPERIAL

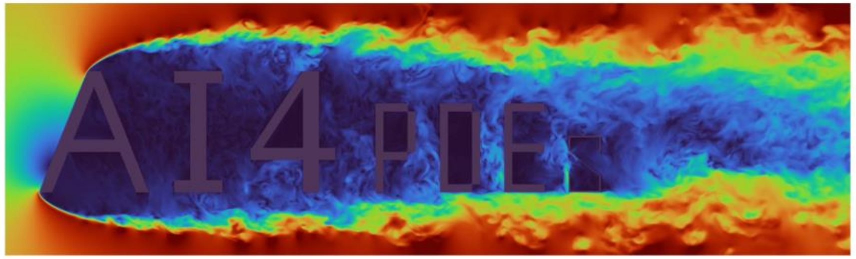

# AI for Scientific Discovery and Multiscale Modelling: - Demonstrating AI4PDEs and AI4Particles

Boyang Chen, Yueyan Li, Aniket Joshi, Nathalie Pinheiro,
Donghu Guo, Fangxin Fang, Claire Heaney, Christopher Pain, Ahmed Elsheikh

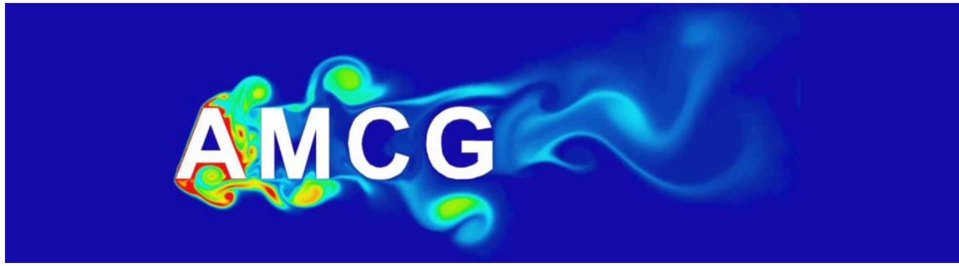
Applied Modelling and Computation Group

Department of Earth Science and Engineering

Imperial College London




25 Mar 2025
ECO-AI Workshop
Edinburgh, UK



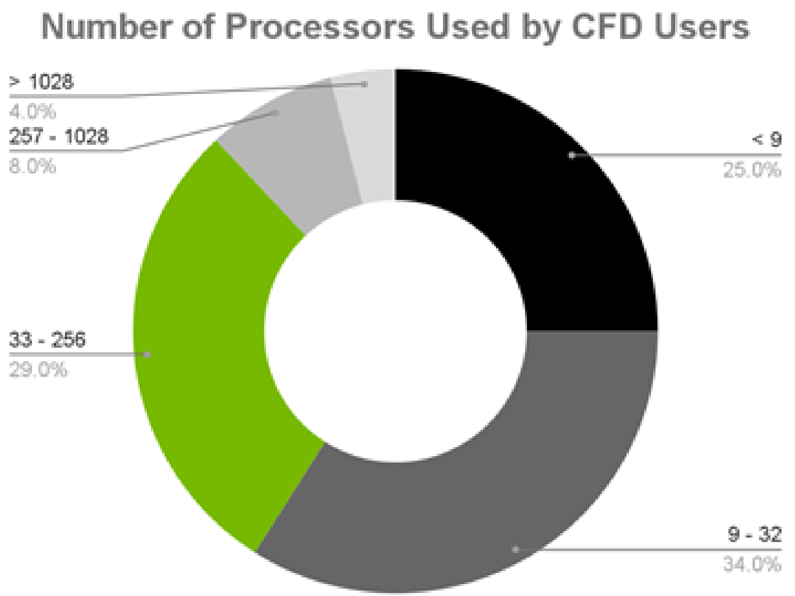
# Background and Motivation

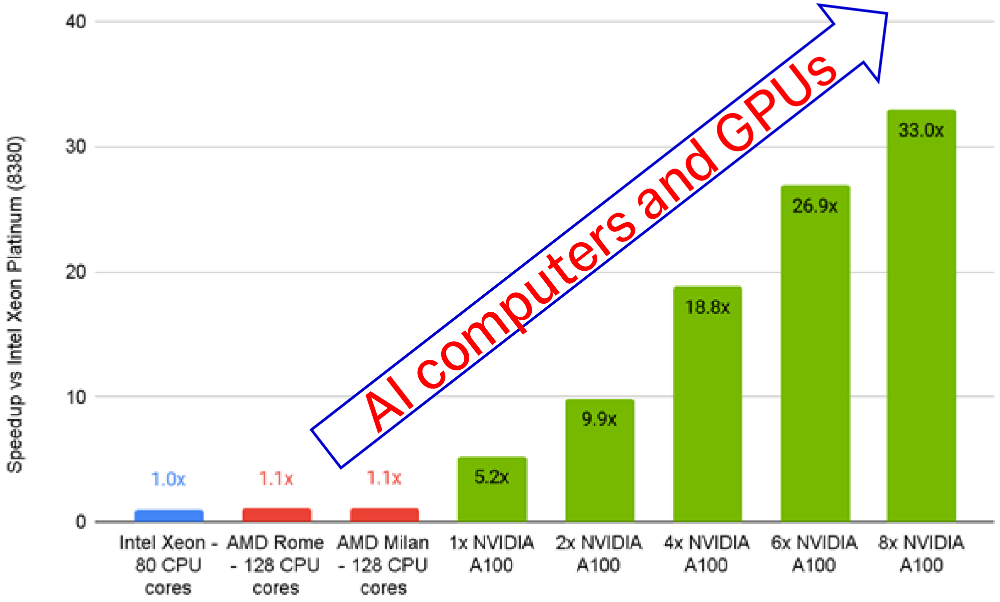
- AI & ML technology provide novel solution of complex problem
- Large-scale CFD simulation (exascale computing)
- Numerical modelers intend to develop GPU-based solver

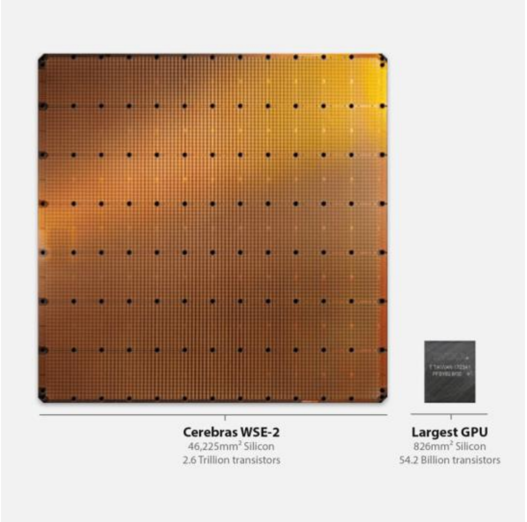
Relax the limitation of computational cost

- New AI computers
- Summary speed - total $10^6$~$10^9$ faster

AI models $10^3$~$10^6$ faster; AI computer $10^3$ faster




Cerebras 1M node chip


GRAPHCORE IPU

# Methodology – AI4PDEs

Artificial Intelligence for Partial Differential Equations (AI4PDEs)

Neural Networks for Partial Differential Equations (NN4PDEs)

- Design the values of kernels in ANNs without data training
- Represent the discretization of PDEs on structured mesh
- Produce identical solution to classical approaches

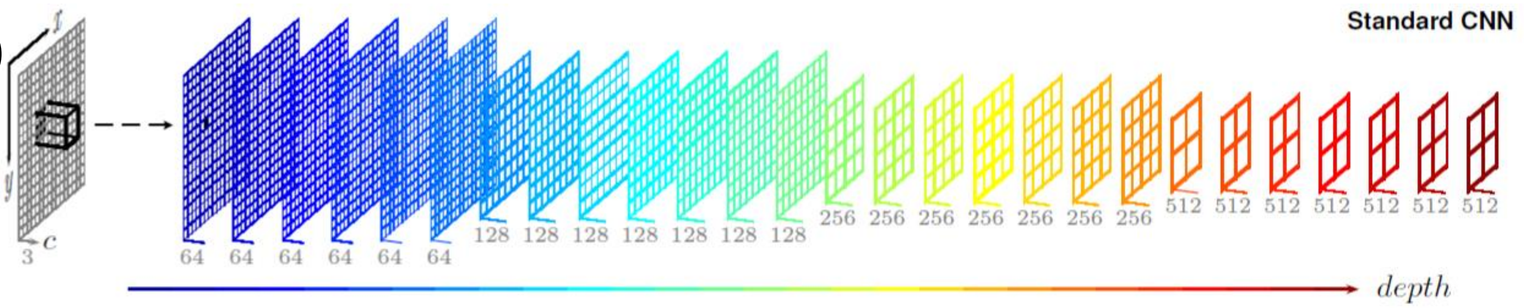

**Advantage**

✓ Easy implementation
✓ Less quantities of code
✓ More computational efficient than conventional CFD solver (~100 times)
✓ More accessible to optimize by GPU and AI computer
✓ Digital twins assimilating data and performing uncertainty quantification
✓ Long term model/code supported by community and AI software

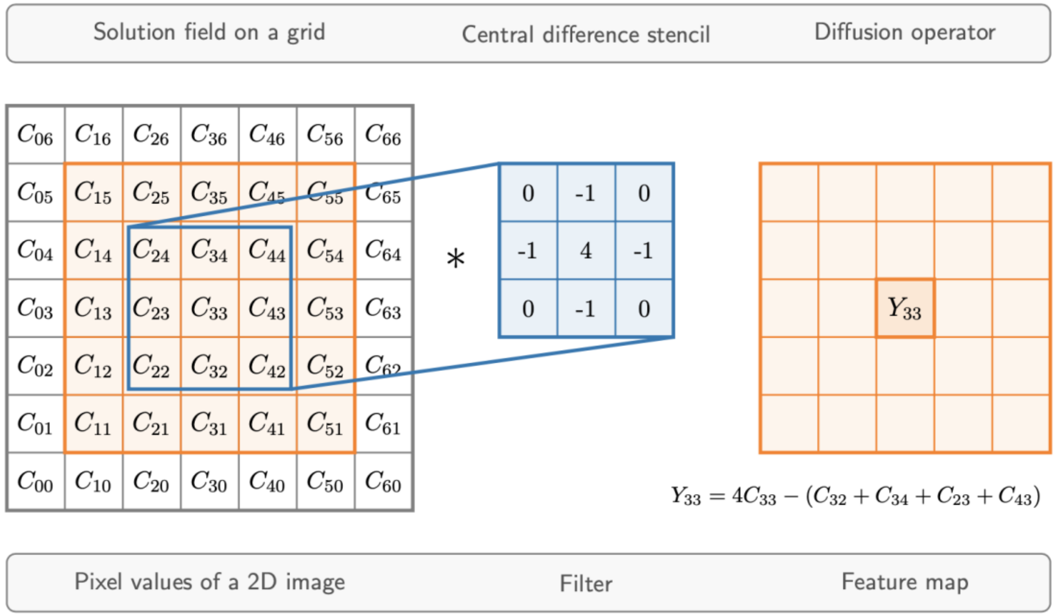
# Methodology - Convolutional neural networks

- Convolution neural networks (CNN)
- ✓ Matrix-free discretization of PDEs
- ✓ No training of the network
- ✓ Known weights in the filters
- Convolutional FEM (ConvFEM)
- ✓ Linear, quadratic and cubic finite element
- Rapid multigrid algorithms
- ✓ U-net structure of F-cycle and V-cycle

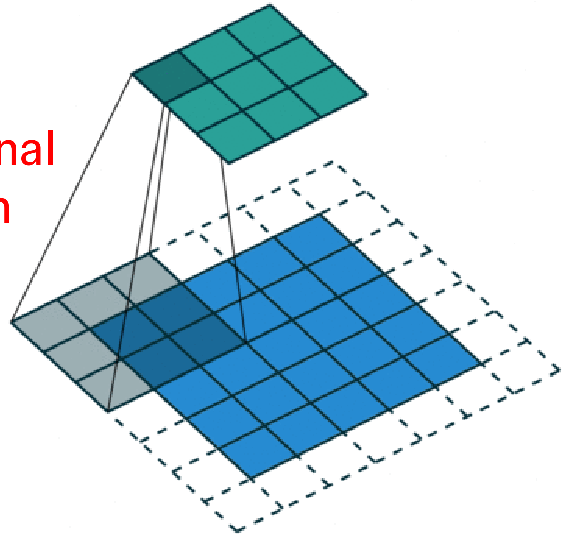
Restriction in multigrid



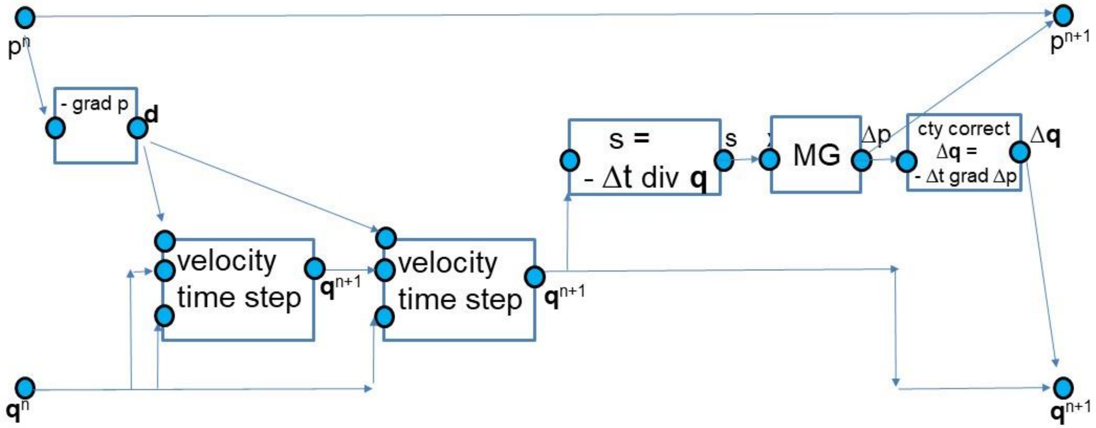

Weights in CNN filters




Convolutional operation



# Methodology - Solving CFD with AI Libraries



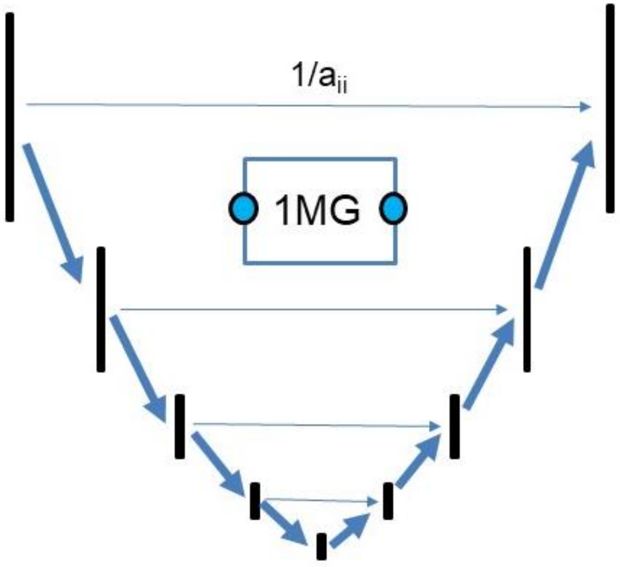

Pressure-velocity coupling

- Two-step time scheme
- Finite differencing method (FDM)
- Finite element method (FEM)
- F-cycle multi-grid for solving pressure
- Projection method
- Petrov-Galerkin stabilisation

$\nabla^2 \phi \longrightarrow$ Central differencing

FDM $\begin{pmatrix} 0.0 & 1.0 & 0.0 \\ 1.0 & -4.0 & 1.0 \\ 0.0 & 1.0 & 0.0 \end{pmatrix}$

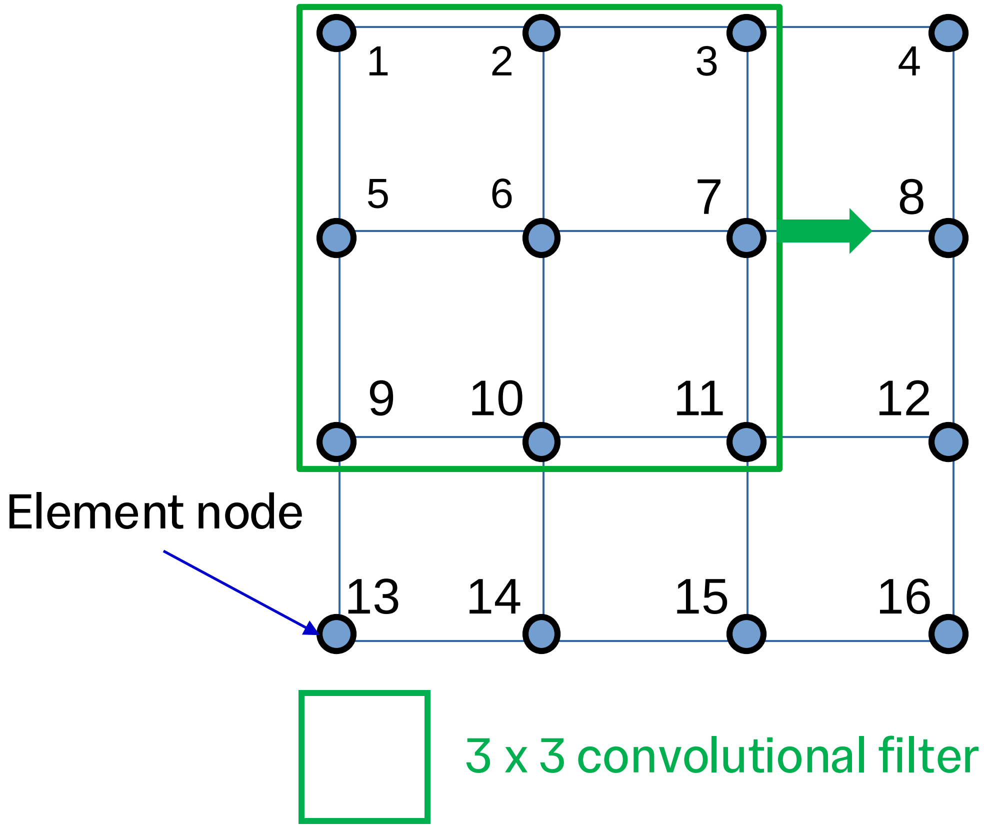
FEM $\begin{pmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & -3.0 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{pmatrix}$

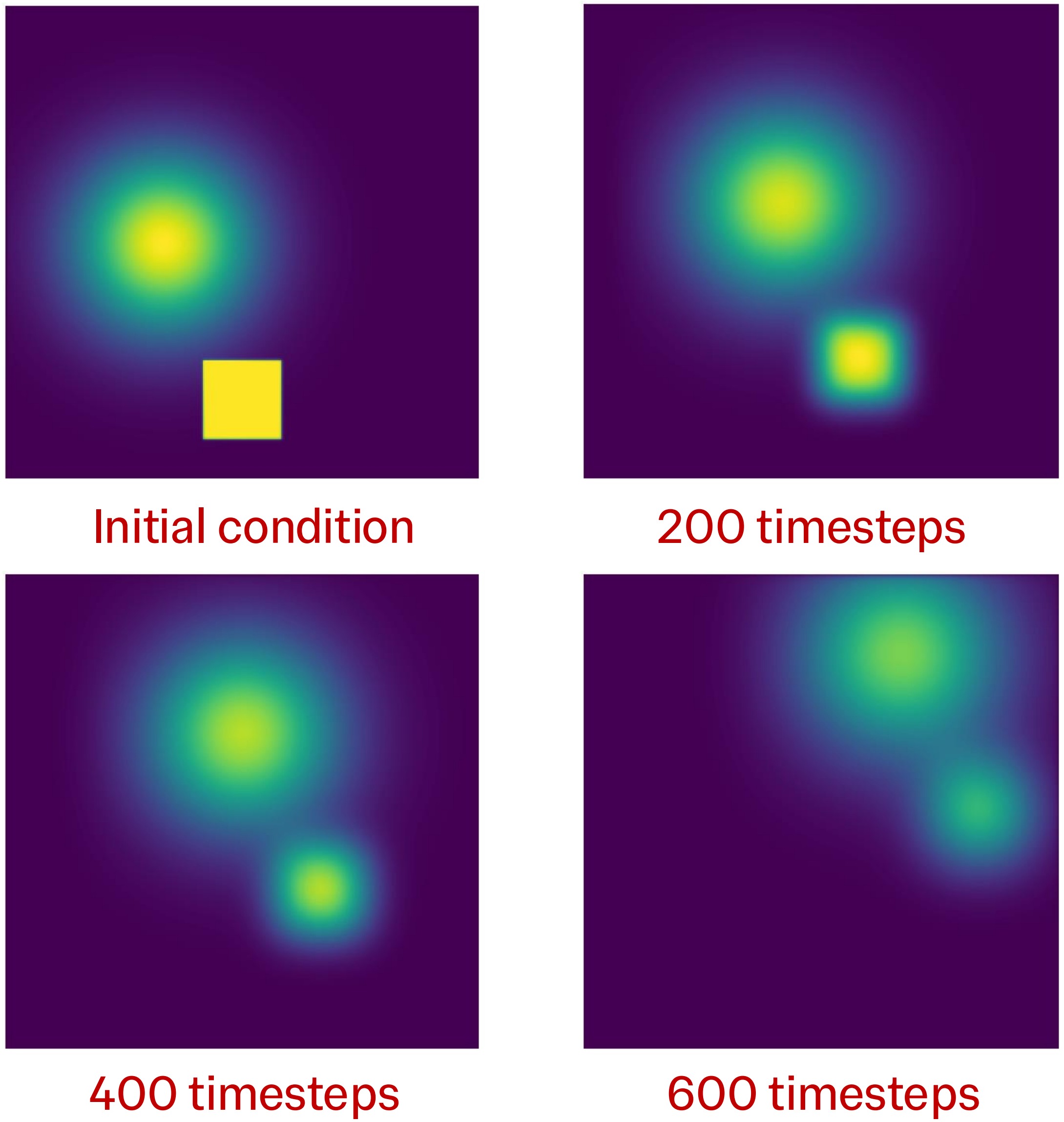



Multigrid with U-net structure

# Methodology - example of solving PDEs using AI libraries

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} - \nu\nabla^2 T = s$$

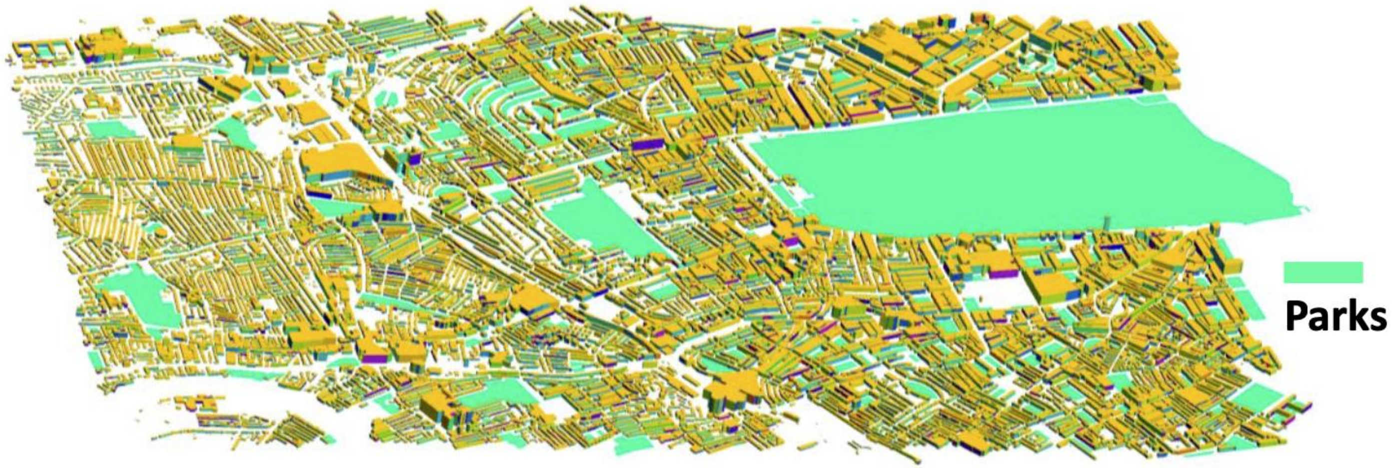


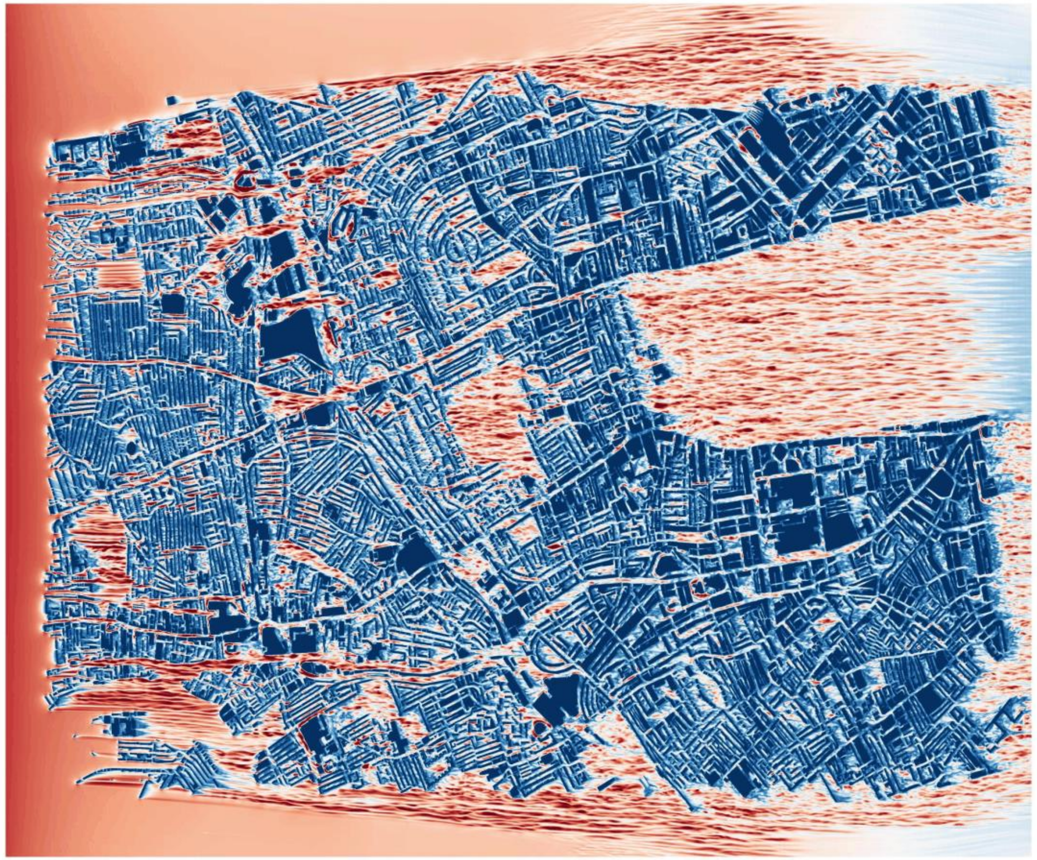
# Airflow modelling using AI4PDEs: South Kensington area

$$\frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} + v\frac{\partial q}{\partial y} + \sigma q - \nu\nabla\cdot\nabla q = -\nabla p \qquad \nabla\cdot q = 0$$

- 3D South Kensington area (5km x 4km)
- One-hour computational time → 5 hours
- Uniform inflow speed (from left to right) – 1 m/s
- 2 Billion nodes – 4 A100 GPUs




Schematic diagram of the area



Airflow speed (m/s)

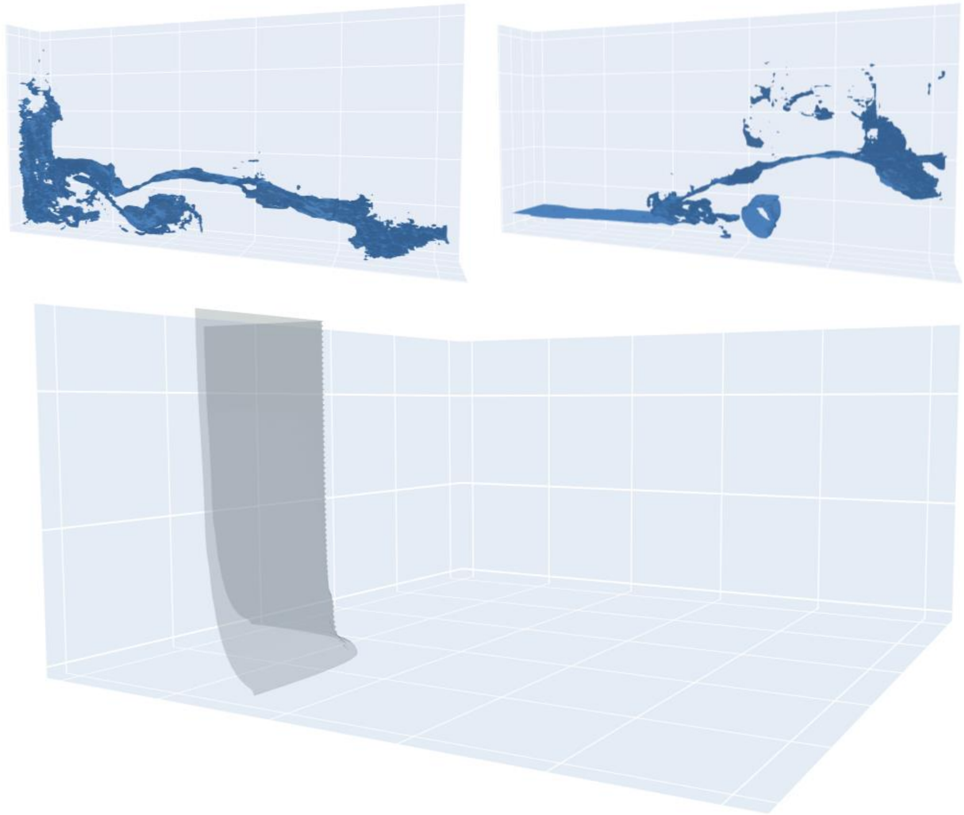
# Multiphase flow using AI4PDEs : Collapsing water column

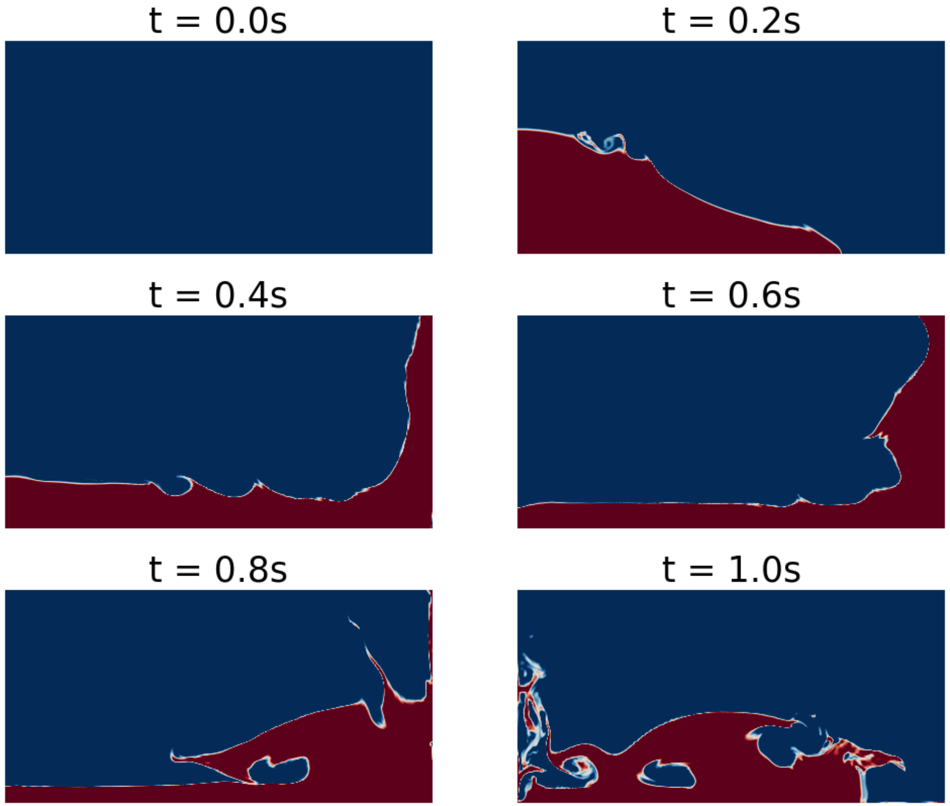
- Cubic domain size: 0.5 (m) x 0.5 (m) x 0.5 (m)
- Grid point: 512 x 512 x 512 (0.256 billion nodes)
- One single GPU

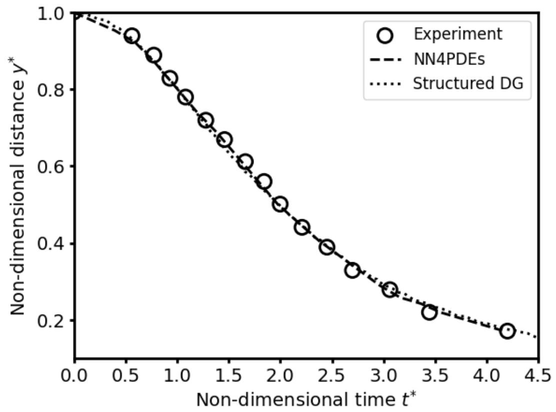
$$\rho\left(\frac{\partial q}{\partial t} + q \cdot \nabla q\right) + \sigma q - \nabla \cdot (\mu \nabla q) = -\nabla p + s_q + s_t$$

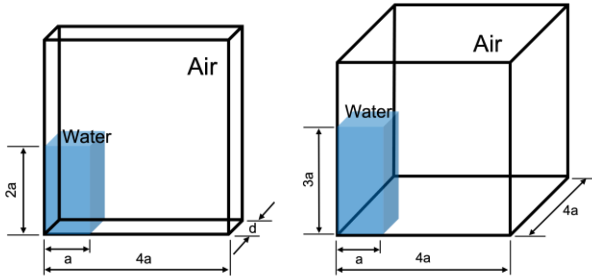
$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho q) = 0$$

$$\frac{\partial C}{\partial t} + q \cdot \nabla C = 0$$



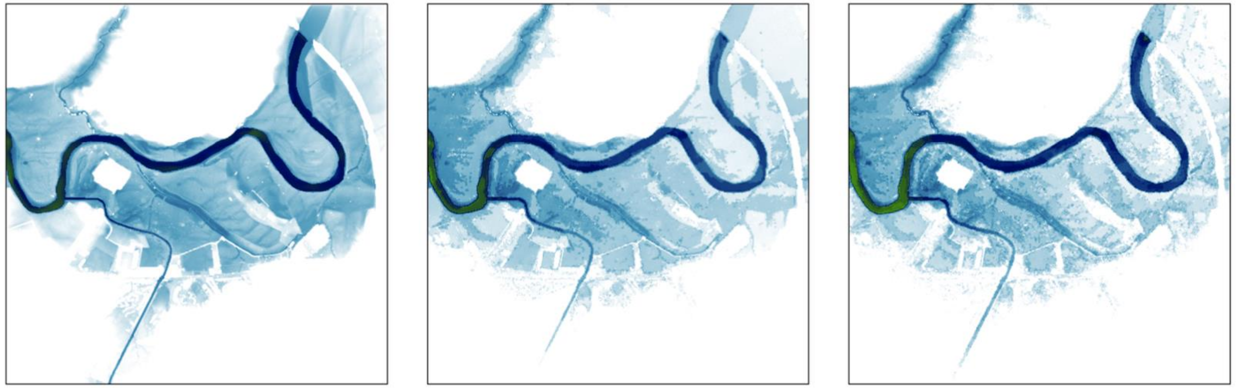


# Multiphase flow using AI4PDEs : Carlisle 2005 flooding

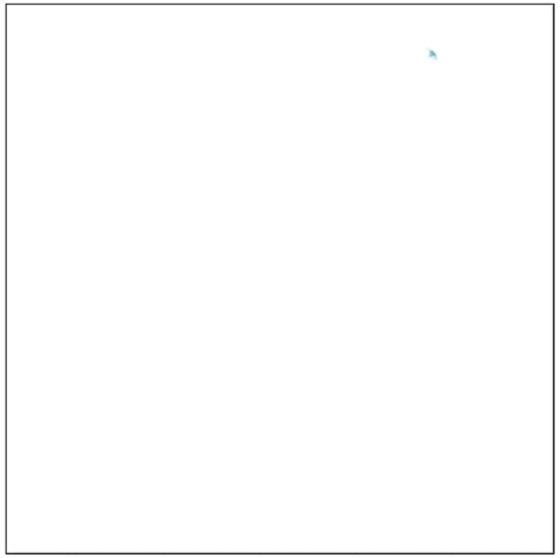
3D flooding using AI4Multiphase and comparison with 2D flooding model



AI4SWE - 951 × 611 AI4Multi - 512 × 512 × 128 AI4Multi - 512 × 512 × 256

Spatial variation of water depth in the flooded area after 10 hours as predict

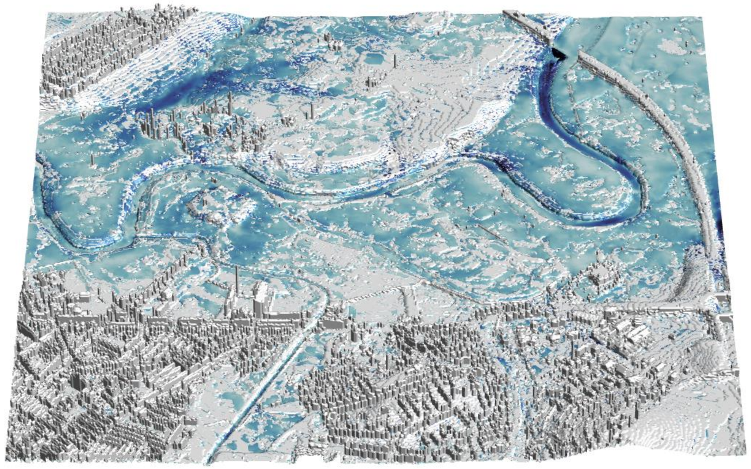
Water depth from 0 to 10 hours



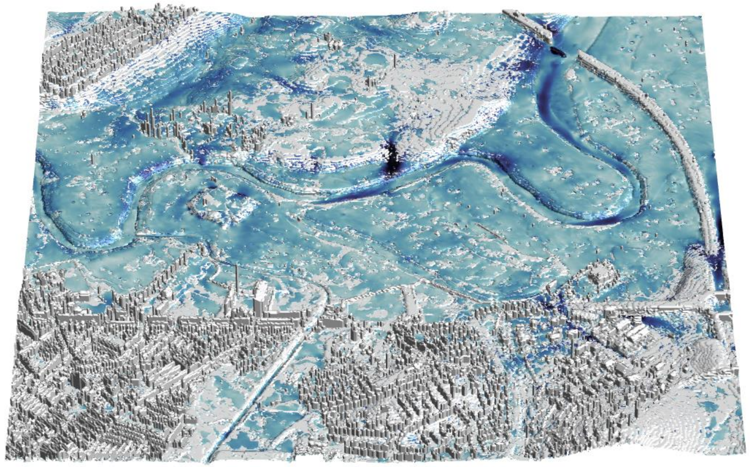
Google Map

Iso-surface colored by water speed (m/s) at two different time levels

T1



T2



# Multiphase flow using AI4PDEs : porous media flow

- Domain size: 0.05 (m) x 0.05 (m) x 0.05 (m)
- Grid spacing: 0.1mm x 0.1mm x 0.1mm
- Number of pellets: ~2000
- Application: $CO_2$ storage
- Surface tension model
- Pressure drop comparing with Ergun equations

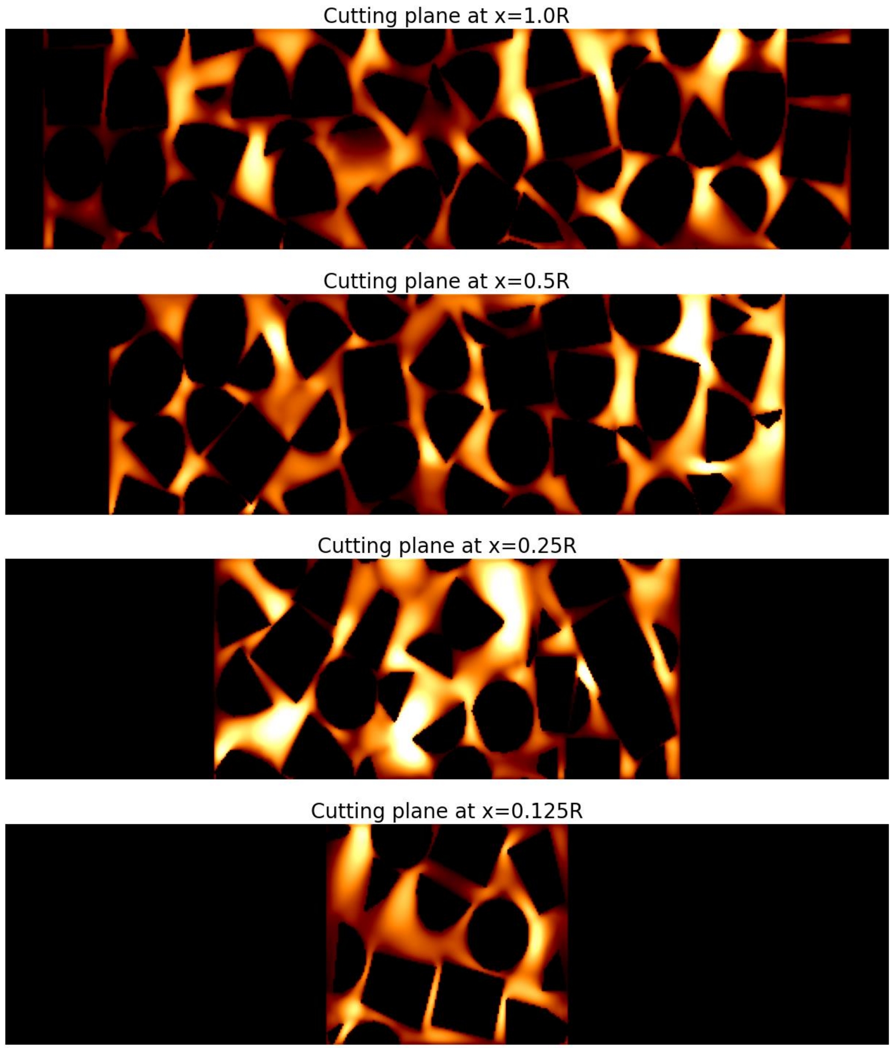
Two scenarios:

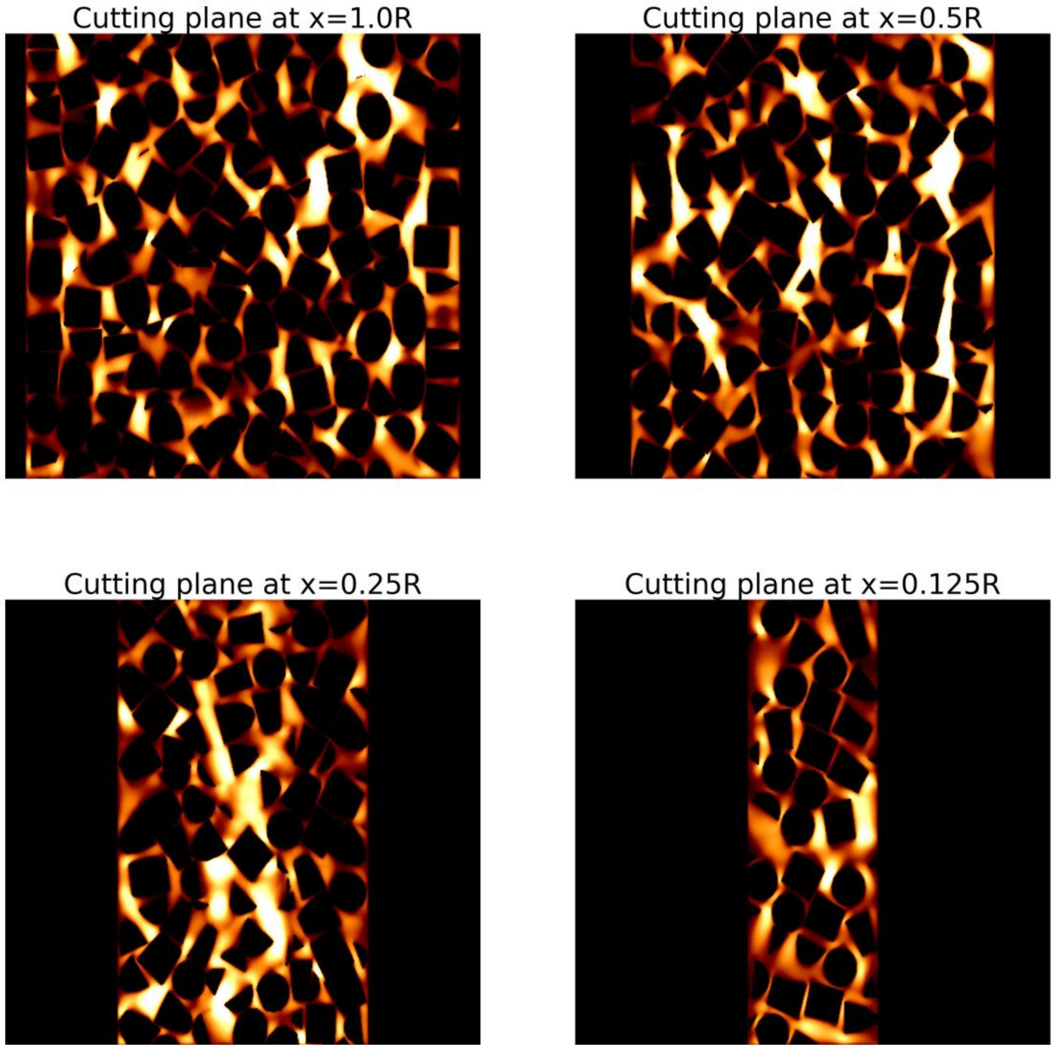
❖ Air is injected from bottom to top (single-phase)

❖ Air is injected into water from bottom to top (multi-phase)

# Multiphase flow using AI4PDEs : porous media flow

## Flow speed (m/s) in the tank – single phase flows



# Multiphase flow using AI4PDEs : porous media flow

## Flow speed (m/s) in the tank – multi phase flows

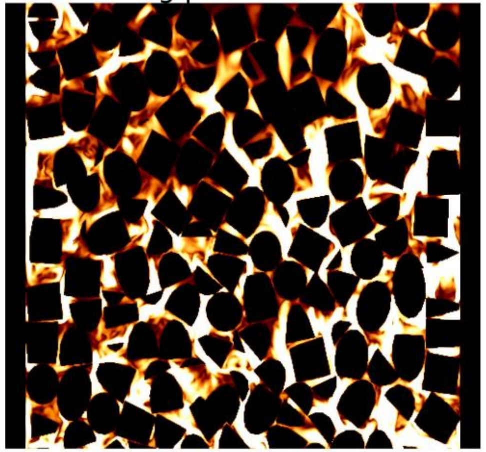
Cutting plane at x=1.0R

Cutting plane at x=0.5R

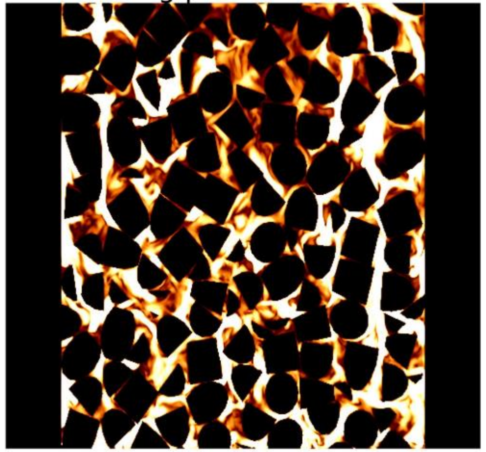
Cutting plane at x=0.25R

Cutting plane at x=0.125R

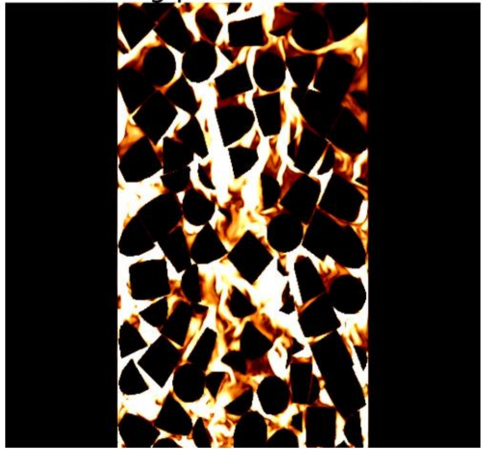
Cutting plane at x=1.0R



Cutting plane at x=0.5R



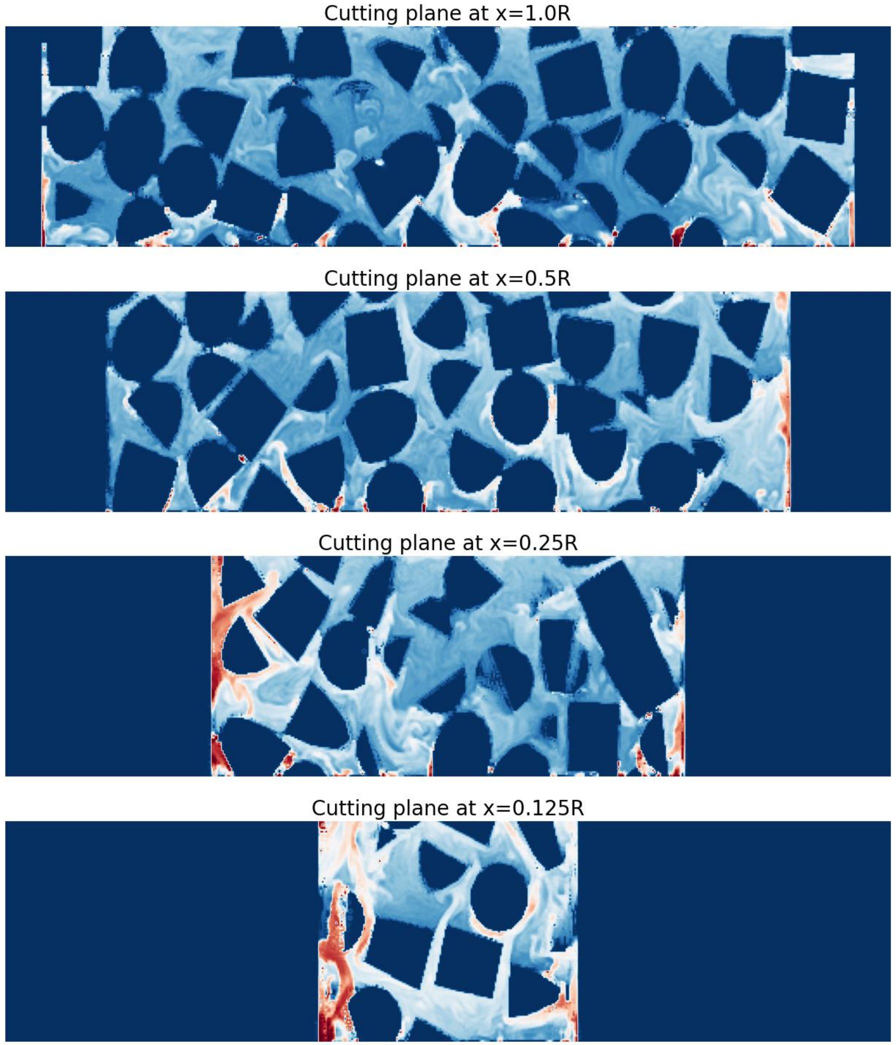
Cutting plane at x=0.25R

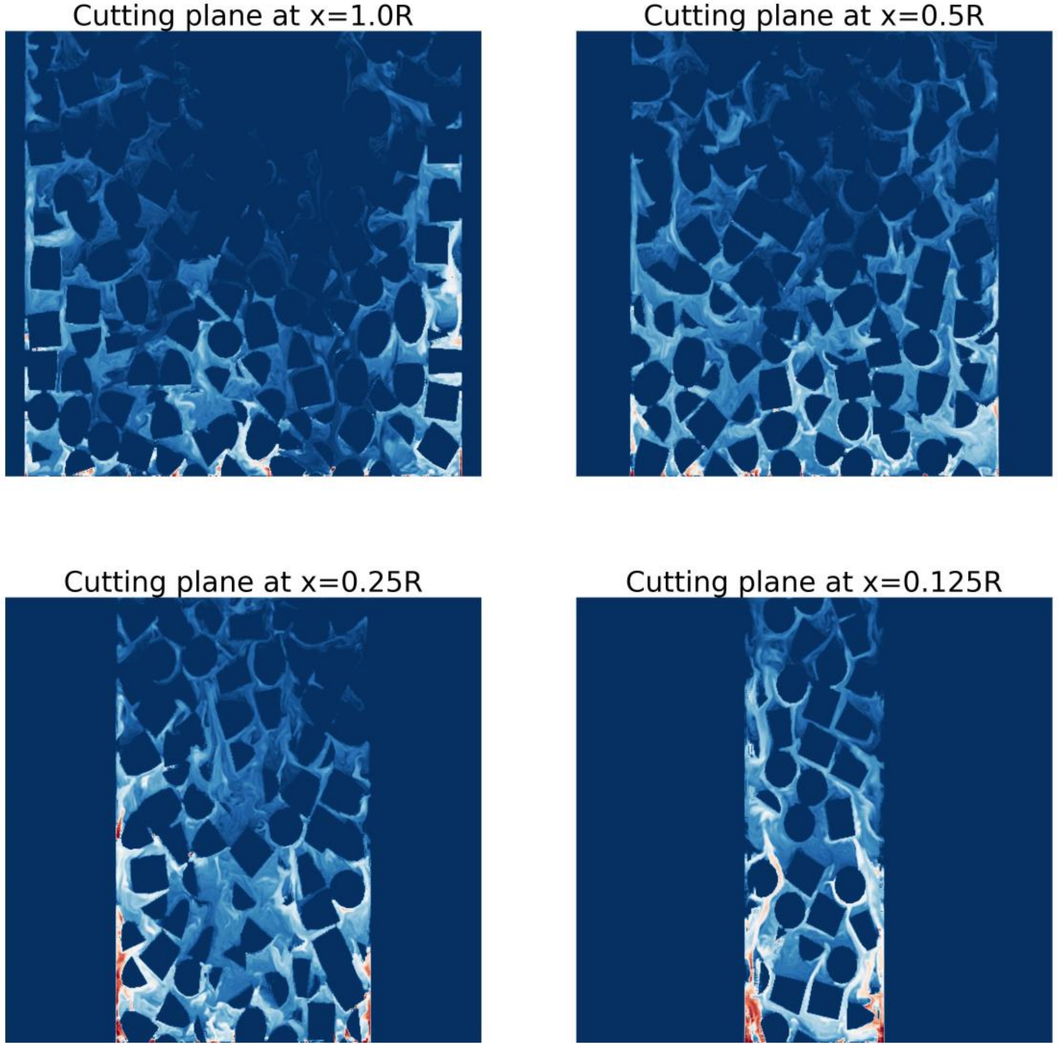


Cutting plane at x=0.125R

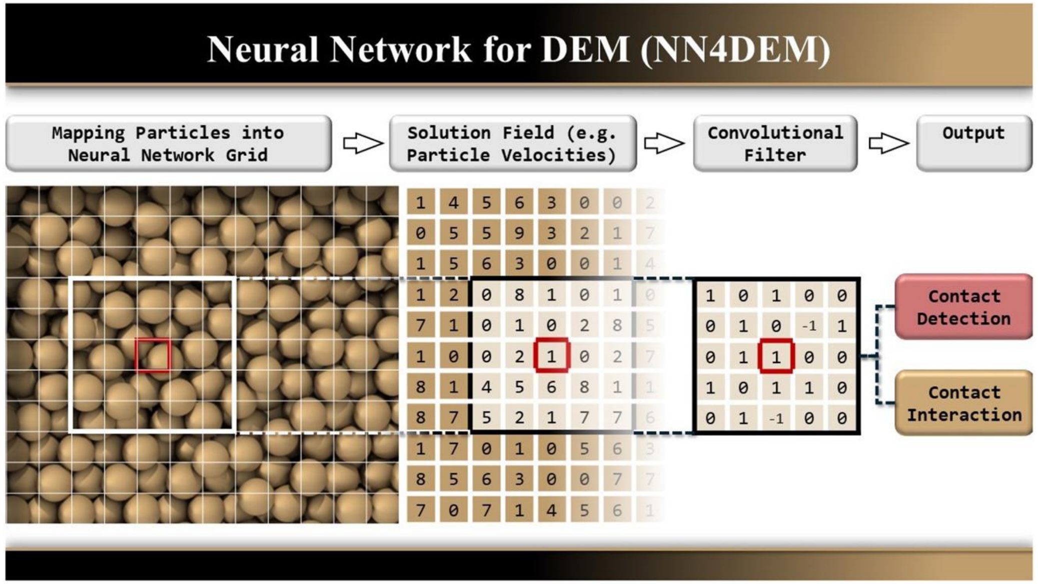
# Multiphase flow using AI4PDEs : porous media flow

## Indicator field (0 to 1) in the tank - multi phase flow



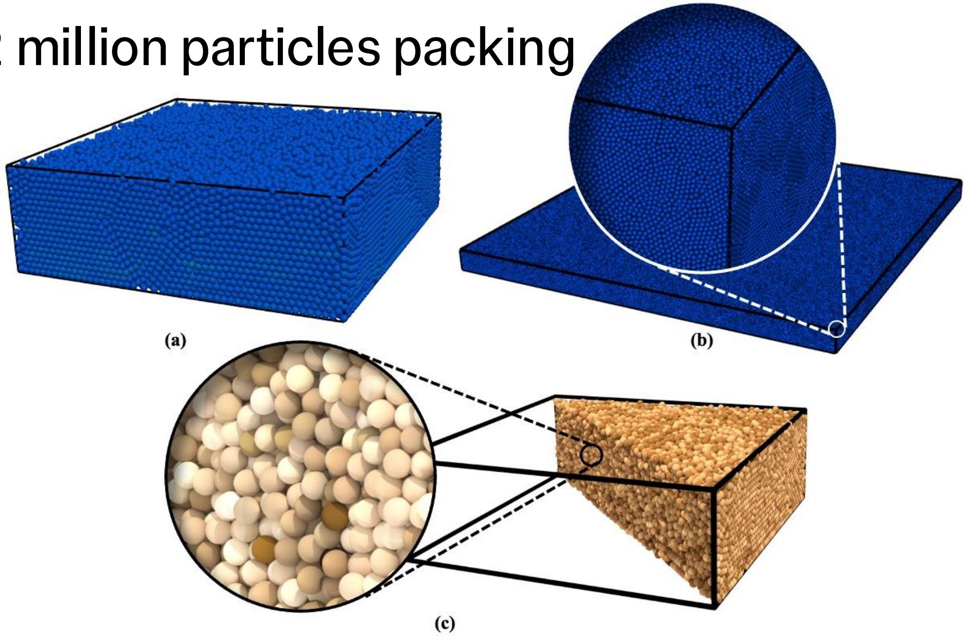



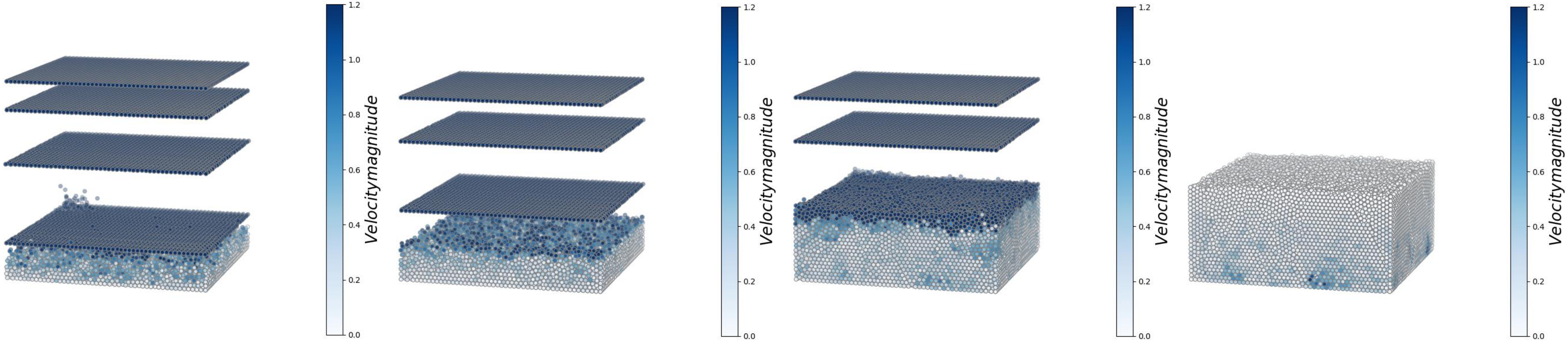
# Example of particle packing using AI4particle/systems




$$\dot{\vec{v}} = \frac{\vec{F}}{m} + \vec{g}\,,$$
$$\dot{\vec{\omega}} = \frac{\vec{T}}{I}\,.$$

32 million particles packing

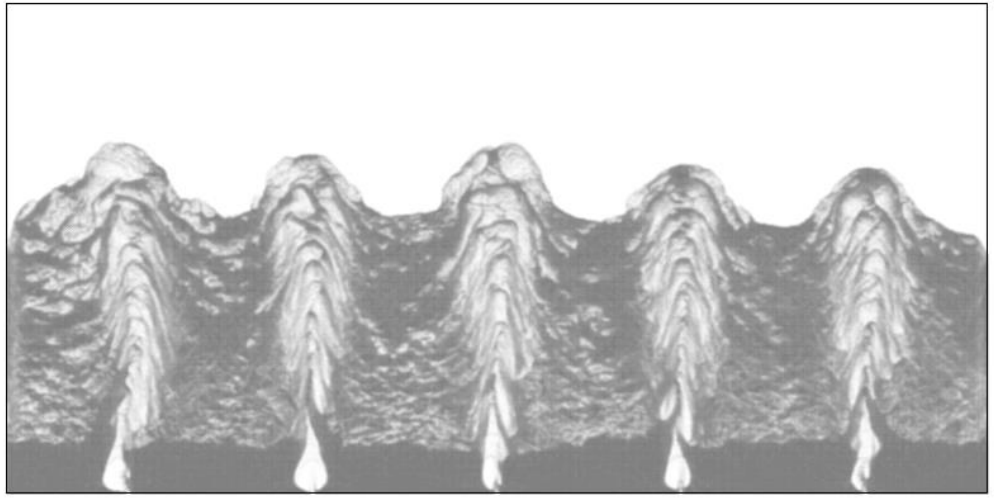


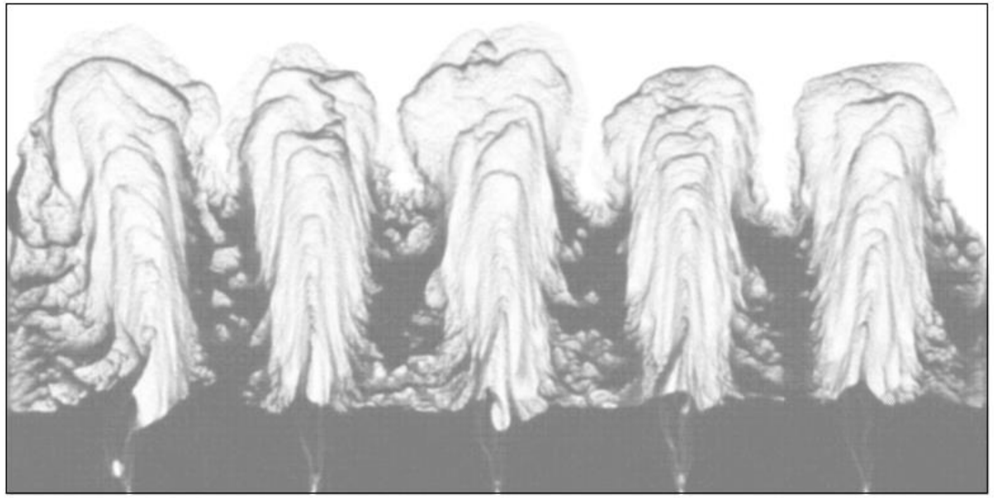
# Fluidised bed flow modelling – AI4Sytems (AI4PDEs + DEM)

20M particles in 2D – running on one single GPU

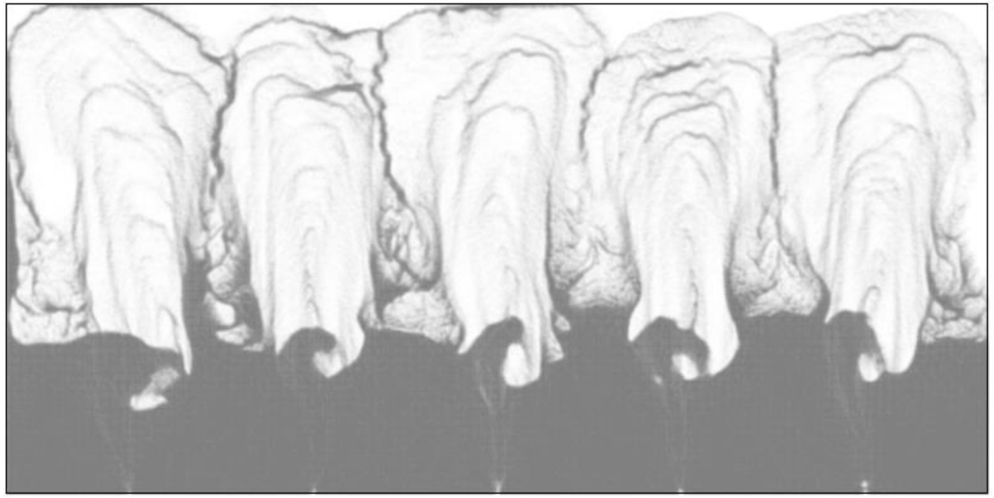
T1



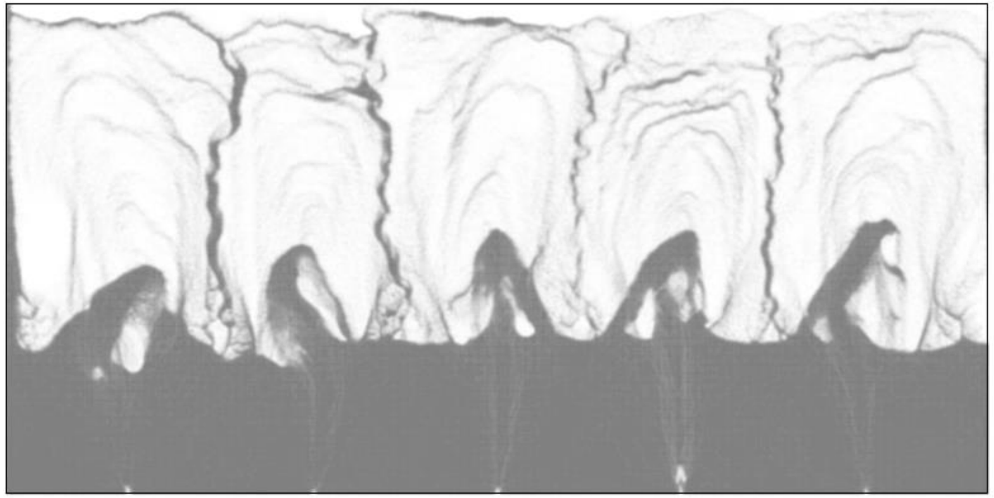
T2



T3



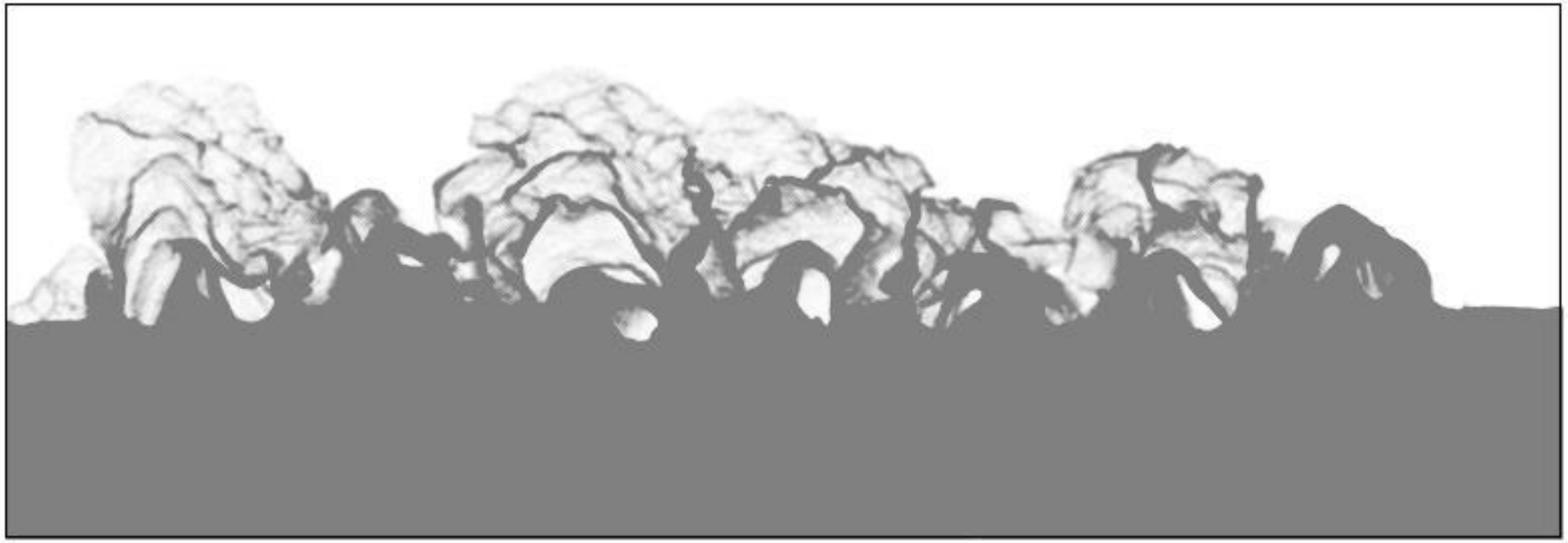
T4



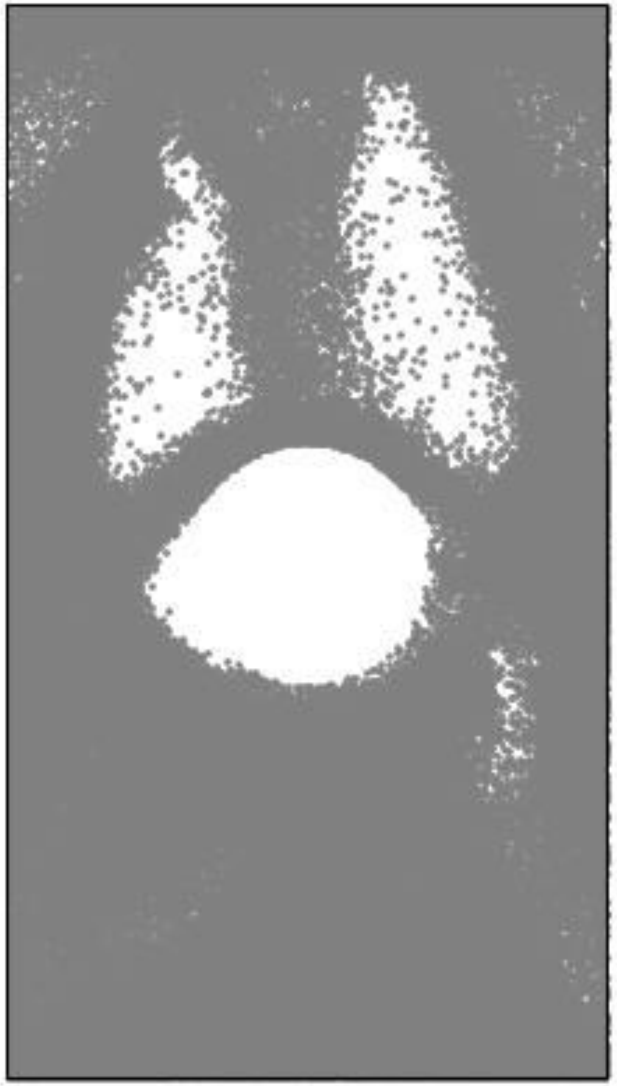
# Example of 2D fluidised bed coupling NN4Particle/System with fluid solver

- CFD-DEM simulation (NN4PDEs + NN4Particles)
- Molecular modelling application

Fluidised bed with an obstacle



40M particles in 2D – running on one single GPU



# Summary and conclusion (NN4PDEs)

- A new numerical solver using AI libraries is proposed, namely NN4PDEs or AI4PDEs. (NVIDIA Modulus)
- The new approach has been validated to predict physical mechanics.
- This approach has been demonstrated on a number of benchmark and realistic problems.

Single-phase flows, multi-phase flows, solid mechanics

- We found the advantage of this approach.

Flexible → CPUs, GPUs and new AI computers

Rapid Computing → digital twins, sensitivity and optimization

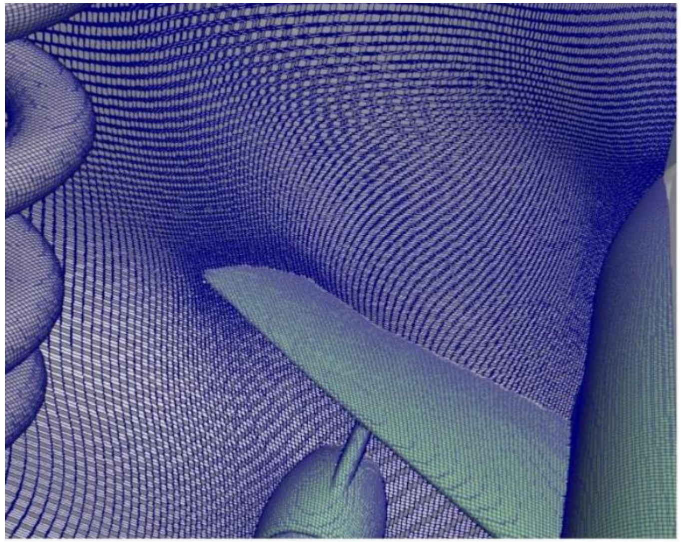
Portable → easy to maintain and develop further

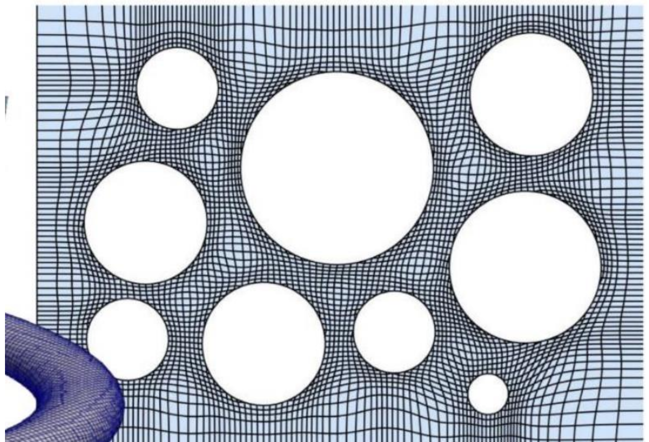
- Further work

Multi-physics solver (molecular dynamics, reservoir system)

Exascale computing (NVIDIA)

Complicated mesh system (distorted and unstructured mesh)





# References (NN4PDEs)


- Phillips TRF, Heaney CE, Chen B, Buchan AG, Pain CC. Solving the discretised neutron diffusion equations using neural networks. International Journal for Numerical Methods in Engineering. (2023) Nov 15;124(21):4659-86.
- Phillips TRF, Heaney CE, Chen B, Buchan AG, Pain CC. Solving the Discretised Boltzmann Transport Equations using Neural Networks: Applications in Neutron Transport. arXiv preprint arXiv:2301.09991. (2023).
- Chen B, Heaney CE, Pain CC. Using AI libraries for Incompressible Computational Fluid Dynamics. arXiv preprint arXiv:2402.17913. (2024).
- Chen B, Heaney CE, Gomes JL, Matar OK, Pain CC. Solving the Discretised Multiphase Flow Equations with Interface Capturing on Structured Grids Using Machine Learning Libraries. Computer Methods in Applied Mechanics and Engineering. (2024) 426:116974.
- Chen B, Nadimy A, Heaney CE, Sharifian MK, Estrem LV, Nicotina L, Hilberts A, Pain CC. Solving the Discretised Shallow Water Equations Using Neural Networks. Advances in Water Resource. (2024)
- Naderi S, Chen B, Yang T, Xiang J, Heaney CE, Latham JP, Wang Y, Pain CC. A discrete element solution method embedded within a Neural Network. Powder Technology. (2024) 448:120258.
- Li, Lin., Xiang, J., Chen, B., Heaney, C E., Dargaville, S., Christopher C. Pain. Implementing the Discontinuous-Galerkin Finite Element Method using Graph Neural Networks. Neural Networks. (2024)
- A. Nadimy, B. Chen, Z. Chen, C. E. Heaney, C. C. Pain Solving the Discretised Shallow Water Equations Using Non-Uniform Grids and Machine-Learning Libraries, SSRN preprint. (2025)

IMPERIAL

# Thank you