A&T Seminar, CERN, Geneva, Switzerland, June 26, 2016

Particle-based Methods for Multiphase Systems and Applications to High Power Targets

Roman Samulyak

Department of Applied Mathematics and Statistics,
Stony Brook University, Stony Brook, NY USA

Computational Science Initiative, Brookhaven National Laboratory, USA

roman.samulyak@stonybrook.edu *rosamu@bnl.gov*

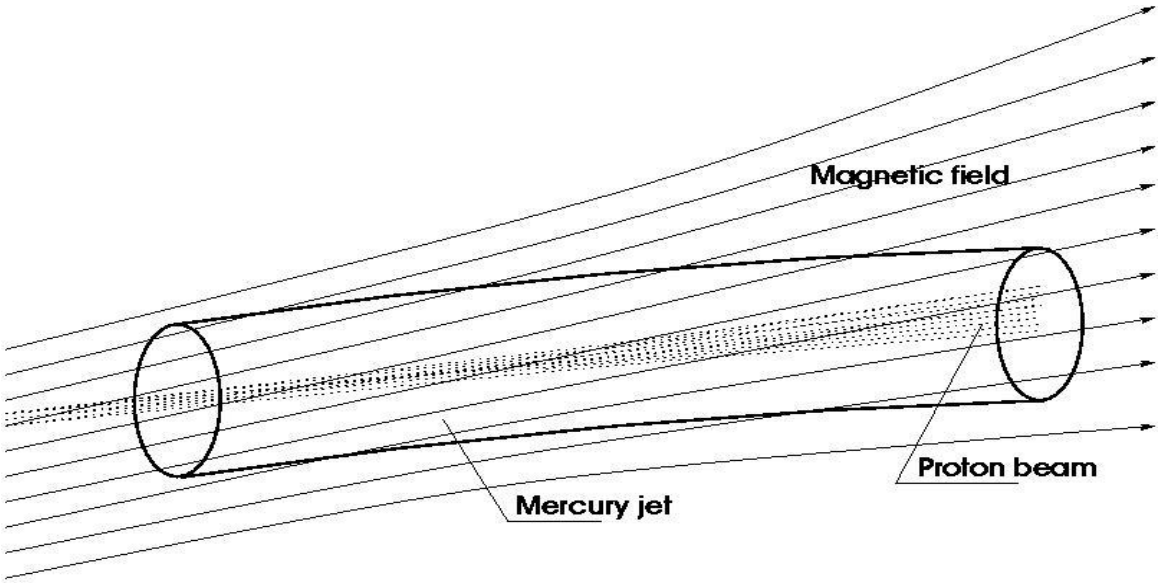
Previous work: simulation high power mercury targets within Muon Collider – Neutrino Factory Collaboration / MAP project

Collaboration with Kirk McDonald (Princeton University), Harold Kirk (BNL), Adrian Fabich (CERN) and other members of the targetry group

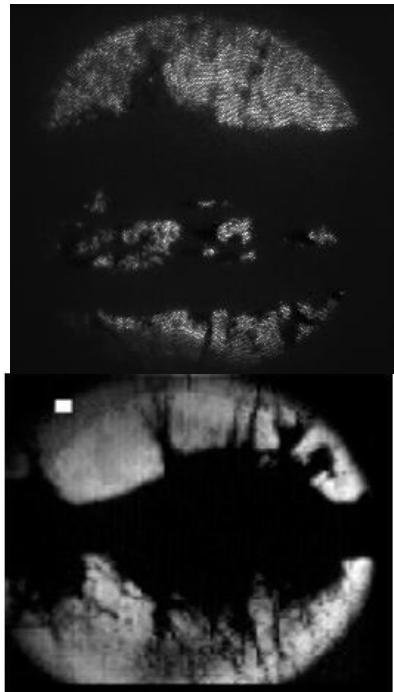
Mercury Jet Target for Neutrino Factory / Muon Collider

- Simulation goal: understanding of hydrodynamic response of targets

- Simulation problems
  - Entrance of the jet into magnetic field
  - Studies of surface instabilities, jet breakup, and cavitation
  - Critical component: accurate resolution of material interfaces



Target schematic

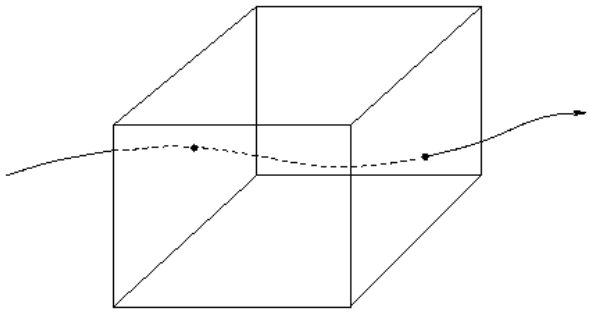


Jet disruptions: experiment

Eulerian and Lagrangian Approaches to Dynamics

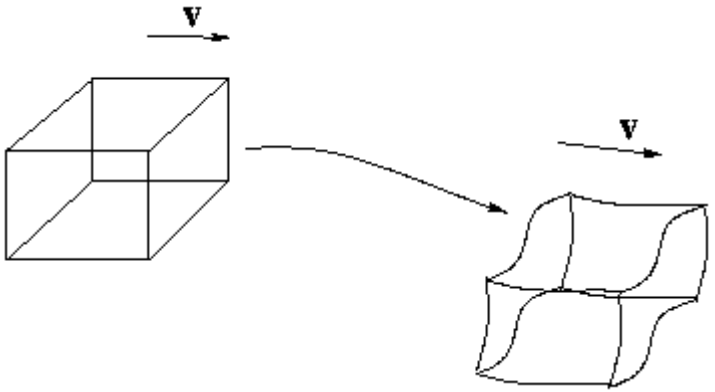
Eulerian method. Coordinate frame (or computational mesh) is fixed (laboratory frame) and the flow moves with respect to this frame

- Simple to use
- Difficult to capture dynamic interfaces



Lagrangian method. The moving substance is represented by a set of material elements that move together with the flow

- Natural resolution of material interfaces
- Major difficulty: the mesh is severely distorted by the flow
- Broadly used in 1D
- Ideal for solid dynamics (any dimension)
- Lagrangian method using particles:
  - Idea: replace Lagrangian fluid element with a particle

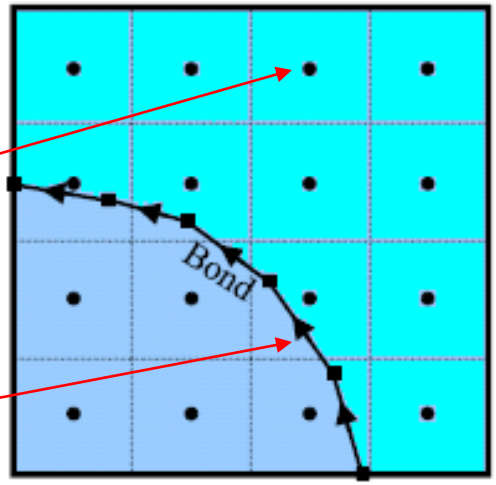


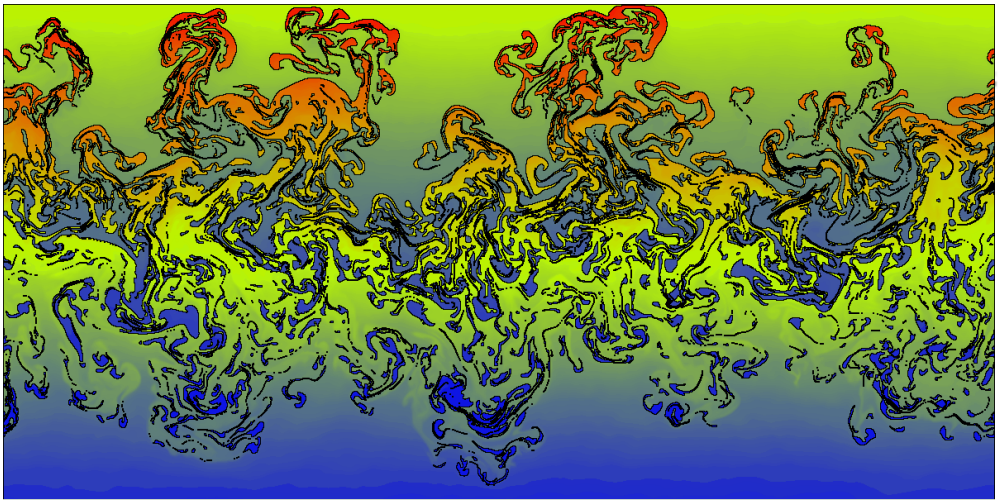
Front Tracking and FronTier Code

- Front tracking is a hybrid Lagrangian-Eulerian method for systems with sharp discontinuities in solutions or material properties

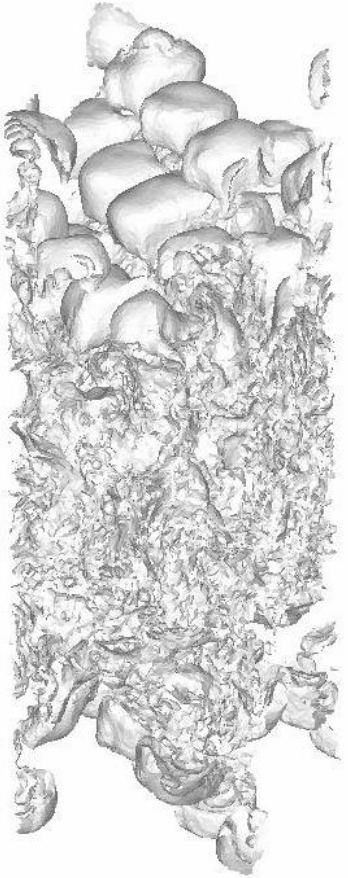
Volume filling rectangular mesh (Eulerian Coord.)

(N-1) dimensional Lagrangian mesh (interface)





Turbulent fluid mixing.
Left: 2D
Right: 3D (fragment of the interface)



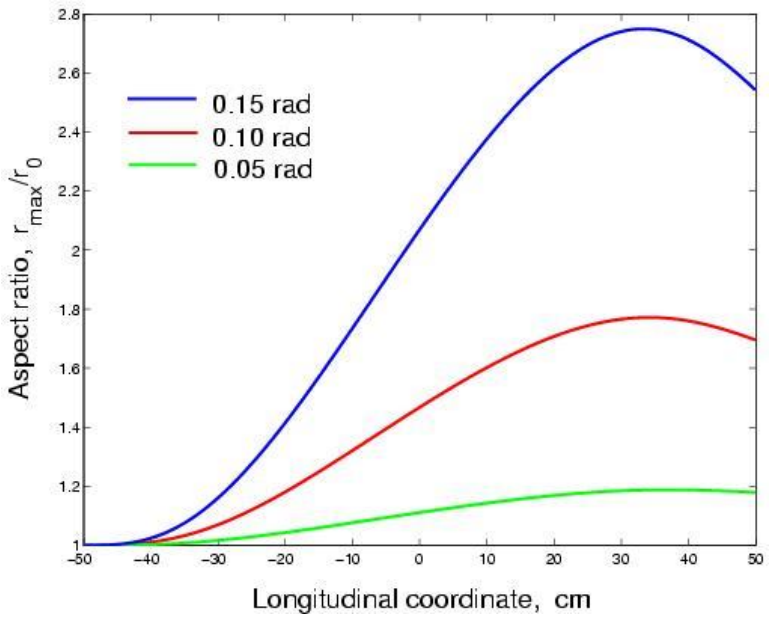
The *FronTier* Code

FronTier is a parallel 3D multiphysics code based on front tracking

- Negligible numerical diffusion across interfaces
- Physics models include
  - Compressible and incompressible fluid dynamics
  - MHD in low magnetic Reynolds number approximation
  - Realistic EOS models, phase transition models

Mercury Jet Entrance into 20 Tesla Transverse Magnetic Field

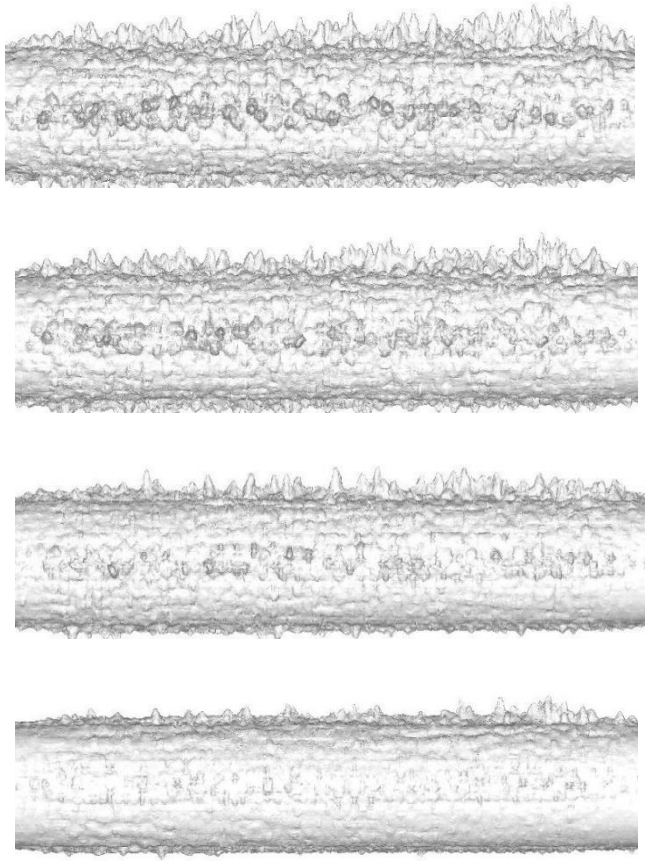
Distortion of the mercury jet in magnetic solenoid due to transverse component of the magnetic field



- Simulations showed large distortion (flattening) of the mercury jet entering magnetic solenoid
  - Under original design parameters of the MERIT experiment at CERN, the mercury jet would be transformed into a thin sheet
  - Cross section with proton beams and pion production would be significantly reduced
- To reduce this effect, MERIT experiment design parameters were changed

MHD Simulation of the mercury jet interaction with proton pulses

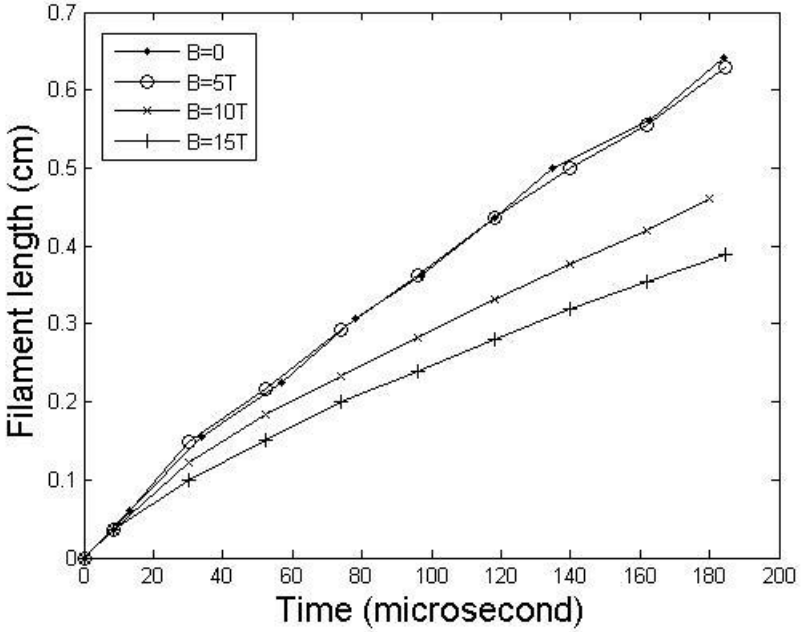
- Simulations predicted cavitation and surface filamentation
  - Cavitation is critical for the explanation of target behavior
- Demonstrated stabilizing effect of the magnetic field
  - Magnetic field reduces the amount of cavitation and velocity of filaments
- Agreement with MERIT experiments on the range of disruption velocities (10 – 50 m/s)



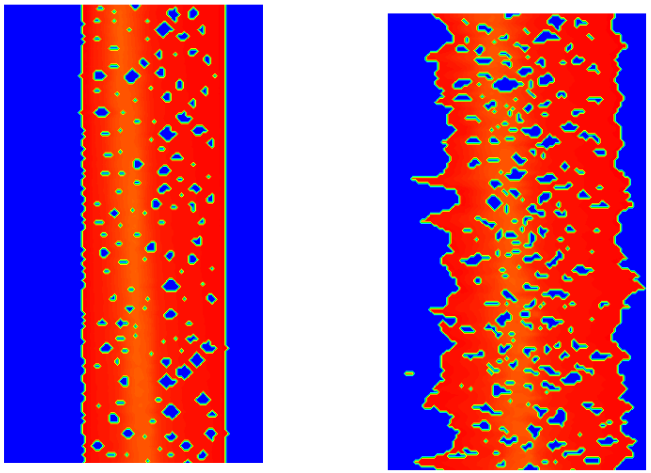
Mercury jet surface at 150 microseconds after the interaction with 12 teraproton pulse

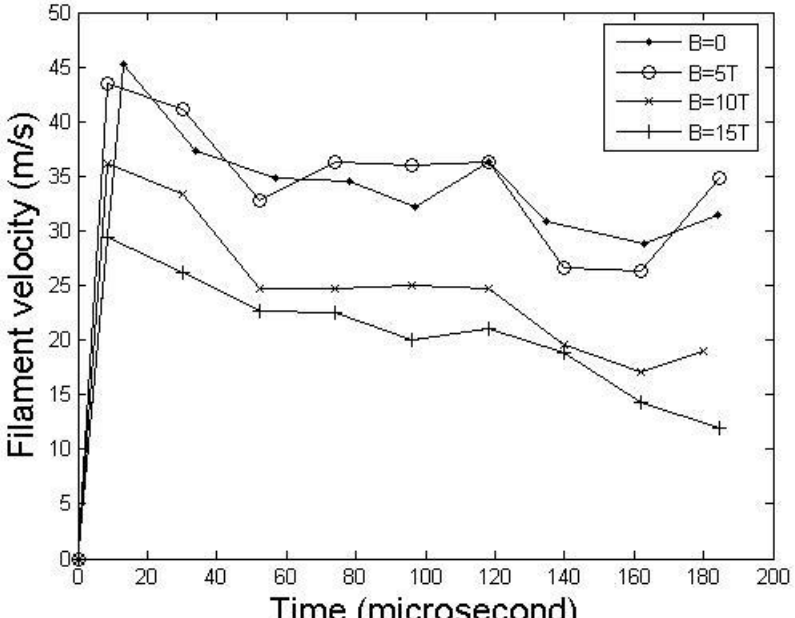
Growth of Filaments and Cavitaiton

The length and velocity of the fastest growing filaments and cavitation



Cavitation in the mercury jet: density distribution at 20 and 250 microseconts





Particle-based Methods for Multiphase Problems

Grid-based vs. Particle-based

We believe **Particle-based (meshless) methods will have much greater role in future simulations of multiphase systems**

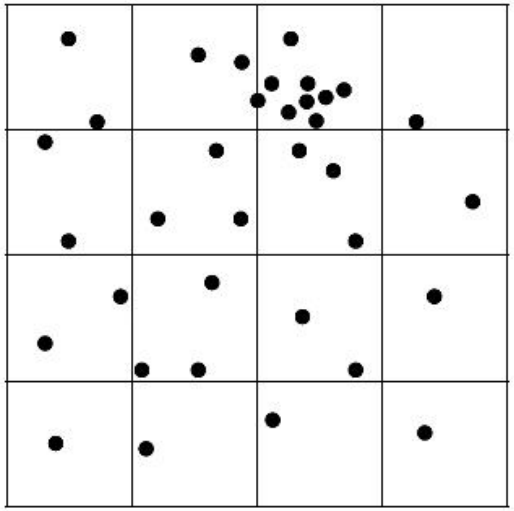
- **Traditional methods:** Eulerian mesh-based PDE discretization with special algorithms for resolving interface (Volume-of-fluid, Level Set, Front tracking etc.)
  - Enhancement by various adaptive features (adaptive mesh refinement, AMR)
  - Require very complex meshes, potential loading balancing problems
  - Complexity causes potential difficulties in porting to new supercomputer architectures (GPU's, Intel-MIC's)

Particle-based (meshless) methods:

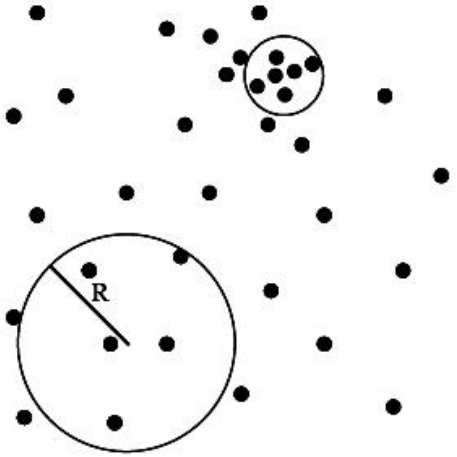
- Exact conservation (Lagrangian formalism)
- Capable of simulating extremely large non-uniform domains (natural adaptivity)
- Ability to robustly handle **material interfaces of any complexity**
- Scalability on modern multicore supercomputers
- **Simplicity:** 3D code is just slightly more complex compared to a 1D code
- **Bridge the gap between continuum and atomistic approaches**

Introduction to Smoothed Particle Hydrodynamics.

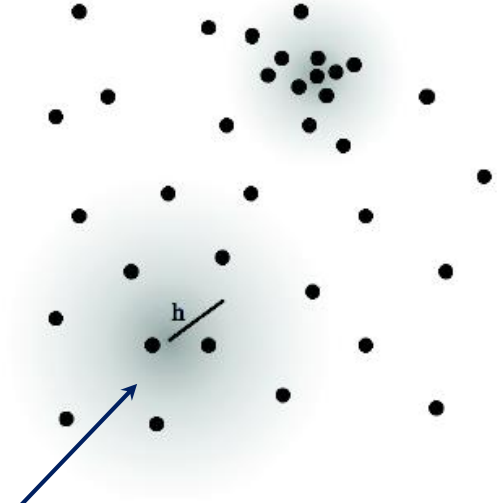
Computing density of continuum using particles



Particle-mesh methods



Sum of particles in disks



SPH:

- Density is weighted sum of particles
- Each particle represents a Lagrangian cell
- No particle connectivity

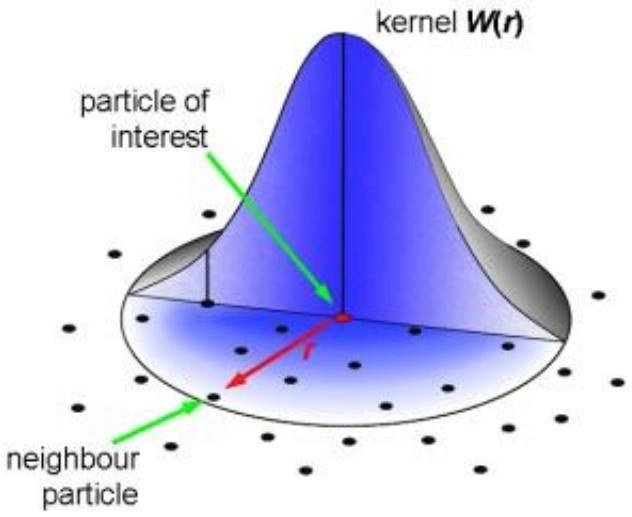
$$\rho_i = \sum_j m_j W_{ij}(h)$$

Main Approximations of SPH

• Dirac's delta-function removes the integration $$A(\vec{r}) = \int A(\vec{r}')\,\delta(\vec{r} - \vec{r}')d\vec{r}'$$

• Kernel approximation replaces the delta-function with a smooth kernel function $$A^W(\vec{r}) = \int A(\vec{r}')W(\vec{r} - \vec{r}', h)d\vec{r}'$$

• Approximation of this integral using particle distributions $$A_i^W = \sum_j \frac{m_j}{\rho_j} A_j W_{ij}(h)$$



For example, the density can be computed as

$$\rho(\mathbf{r}) = \sum_b m_b W(\mathbf{r} - \mathbf{r}_b, h)$$

Main Approximations of SPH (2) - Kernel

- The kernel should satisfy the properties to validate the integral approximation:

$$\int W\left(\mathbf{r}-\mathbf{r}',h\right)\mathbf{dr}'=1 \qquad \lim_{h\to 0} W\left(\mathbf{r}-\mathbf{r}',h\right)=\delta\left(\mathbf{r}-\mathbf{r}'\right)$$

- For example, the popular cubic spline kernel:

$$W\left(q,h\right)=\frac{\sigma}{h^{v}}\begin{cases}1-\frac{3}{2}q^{2}+\frac{3}{4}q^{3} & \text{if } 0\leq q\leq 1\\ \frac{1}{4}\left(2-q\right)^{3} & \text{if } 1\leq q\leq 2\\ 0 & \text{if } q>2\end{cases}$$

$$q=\frac{|\mathbf{r}-\mathbf{r}'|}{h}$$

Main Approximations of SPH (3) – First Derivatives

- Approximation of the first derivative of a field function *A*:

$$\nabla A_I(\mathbf{r}) = \int \nabla A(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h)\, \mathbf{dr}'$$

- Using integration by parts and ignore the surface integrals, we have

$$\nabla A_I(\mathbf{r}) = \int A(\mathbf{r}') \nabla W(\mathbf{r} - \mathbf{r}', h)\, \mathbf{dr}'$$

- Basic SPH discretization is:

$$\nabla A_s(\mathbf{r}) = \sum_b m_b \frac{A_b}{\rho_b} \nabla W(\mathbf{r} - \mathbf{r}_b, h)$$

Main Approximations of SPH (4) – Non-Unique Forms

- Derivatives of quantities *A* are expressed in terms of derivatives of known kernels *W*
  - This allows us to discretize PDE's
  - Non-uniqueness of derivatives:

$$\nabla A = \frac{1}{g}\left(\nabla\left(gA\right) - A\nabla g\right)$$

$g \equiv 1$ **[1]**

$$\nabla A_a = \sum_b \frac{m_b}{\rho_b}\left(A_b - A_a\right)\nabla_a W_{ab}$$

$g \equiv \rho$ **[2]**

$$\nabla A_a = \frac{1}{\rho_a}\sum_b m_b\left(A_b - A_a\right)\nabla_a W_{ab}$$

[3]

$$\nabla A = g\left[\nabla\left(\frac{A}{g}\right) + \frac{A}{g^2}\nabla g\right] \xrightarrow[g \equiv \rho]{} \nabla A_a = \rho_a \sum_b m_b\left(\frac{A_b}{g_b^2} + \frac{A_a}{g_a^2}\right)\nabla_a W_{ab}$$

Discretization of Compressible Fluid Dynamics Equations

- Lagrangian fluid equations

$$\frac{d}{dt}\mathbf{v} = -\left(\frac{1}{r}\right)\nabla P$$

$$\frac{dr}{dt} = -(r)\nabla\cdot\mathbf{v}$$

$$\frac{de}{dt} = -\left(\frac{P}{r}\right)\nabla\cdot\mathbf{v}$$

- Momentum conservation equation

$$\frac{dv_i}{dt} = \sum_j m_j\left(\frac{P_j - P_i}{r_i r_j}\right)\nabla_i W_{ij}(h) \quad \text{or} \quad \frac{dv_i}{dt} = -\sum_j m_j\left(\frac{P_j}{r_j^2} + \frac{P_i}{r_i^2}\right)\nabla_i W_{ij}(h)$$

using **[1]**

using **[3]**

Smoothed Particle Hydrodynamics (SPH) code

- Developed an advanced SPH code
  - Several smoothed particle kernels
  - Advanced time stepping methods
  - Parallelized for distributed memory supercomputers
  - GPU parallelization of most expensive processes

- Simulation experience:
  - obtained good numerical results results for some classes of problems
  - simulations dependent on accuracy of linear and nonlinear wave dynamics miserably failed

Inaccurate numerical derivatives

• The biggest SPH problem: very low accuracy of derivatives (zero-order, non-convergent), even for constant smoothing radius

- SPH derivative gives the similar accuracy to FD if particles are placed on rectangular mesh (due to cancellation of cross-terms)
- Accuracy rapidly decreases if particles even slightly deviate from the mesh
- The chain below is not based on rigorous approximation theory

$$A(\vec{r}) = \int A(\vec{r}')\,\delta(\vec{r}-\vec{r}')d\vec{r}' \longrightarrow A^W(\vec{r}) = \int A(\vec{r}')W(\vec{r}-\vec{r}',h)d\vec{r}'$$

$$A_i^W = \sum_j \frac{m_j}{\rho_j} A_j W_{ij}(h) \longrightarrow \nabla A_i^W$$

Summary of SPH difficulties [Hopkins, GIZMO: A New Class of Accurate, Mesh-Free Hydrodynamic Simulation Methods, Mon. Not. R. Astron. Soc, 2014]

Method	Order	Conservative	Long time stability	Number of neighbors	Difficulties
Traditional SPH (GADGET, GASOLINE, TSPH)	0	yes	yes	~32	noise, E0 errors
"Modern" SPH (P-SPH, SPHS,PHANTOM, SPH-Gal)	0	yes	yes	~128-442	Excess diffusion, E0 errors, expense
"Corrected" SPH Integral-SPH, Moving Least Squares, Morris-96	0-1	no	no	~32	Errors grow non-linearly, stability
Godunov-SPH	0	yes	yes	~300	Instability, expense, E0 errors

Why is SPH accurate (for some problems)?

- Despite operating with inaccurate derivatives, SPH may produce accurate results
- Our own experience
  - Our simulation showed reasonable results for some problems
  - But we did not obtain accurate results with wave propagation problems
  - Especially inaccurate for coupled problems (hyperbolic + elliptic)

Why is SPH stable?

- SHP is an example of Forward-in-Time, Centered-in-Space discretization of a hyperbolic PDE; should't it be unstable?
- But SPH is remarkably stable. SPH code does not crash even if solutions develop into unphysical states
- Replacing SPH derivatives with very accurate GFD (generalized finite difference, future slides) derivatives produces an unconditionally unstable code!
- Why bad derivatives lead to a stable discretization and accurate derivatives lead to an unstable scheme?

Lagrangian and Hamiltonian Dynamics of Particles

Let's consider the Lagrangian and Hamiltonian functionals for the system of particle we intended to use for the SPH approximation of fluid

$$L = \sum_b m_b \left[\frac{1}{2} v_b^2 - u_b(r_b, S_b) \right] \quad H = \sum_a \mathbf{v}_a \times \frac{\partial L}{\partial \mathbf{v}_a} - L, \quad du = T\,dS + \frac{P}{r^2}\,dr$$

Let's express the density of the system via the smooth kernel and find

$$\frac{\partial r_b}{\partial \mathbf{r}_a} = \frac{1}{W_b} \sum_c m_c \frac{\partial W_{bc}(h_b)}{\partial \mathbf{r}_a} \left(\delta_{ba} - \delta_{ca} \right)$$

- Then the Lagrange equation $\frac{d}{dt}\left(\frac{\partial L}{\partial \mathbf{v}_a} \right) - \frac{\partial L}{\partial \mathbf{r}_a} = 0$

is precisely equivalent to the SPH momentum conservation equation,

- the time-derivative of the Hamiltonian is equivalent to the SPH energy conservation equation

Inconsistencies of SPH explained

- Inaccurate SPH discretization of Euler equations is identical to accurate Lagrange / Hamilton equations for the same particle system (interacting via isentropic potential energy)
- Hamiltonian formulation reveals non-linear properties of the system such as symmetries, invariants, behavior of the global system
- Hamiltonian structure is responsible for the long term stability
- Approximation of derivatives is related to linear errors

Daniel Price, SPH developer: **"linear errors do not lie"**, so it will always be true that … [other formulations] … will always be more accurate for linear or weakly nonlinear problems. …. Ideally, one would want both, exact derivatives and conservation … and to my knowledge this is yet to be convincingly demonstrated by any particle method …"

To overcome SPH deficiencies and preserve all its advantages, we are working on a Lagrangian particle technique based on generalized finite differences.

New Lagrangian Particle Method

- We keep only one idea of SPH: each particle represents a Lagrangian fluid cell
- Need to satisfy accuracy, stability, and efficiency on modern hardware

- Key novel features of our method:
  - The method is **free of artificial parameters**
  - A stable particle-based upwinding method was invented
  - Accuracy: derivatives based on generalized finite differences (optimal coefficients of a local stencil are found via least squares)
    - High order methods

- We work on both hyperbolic and elliptic solvers
- Parallelization for distributed memory supercomputers and GPU's

Governing Equations

- 1D Lagrangian formulation of the Euler equation in the conservative form:

$$U_t' + \left[F(U')\right]_x = 0, \qquad U' = \begin{pmatrix} V \\ u \\ E \end{pmatrix}, \; F(U') = V \begin{pmatrix} -u \\ P \\ Pu \end{pmatrix},$$

- Assume any EOS, $e = f(P, V)$, we can rewrite in the state vector $U = [V \; u \; P]^T$

$$U_t + A(U)U_x = 0, \qquad U = \begin{pmatrix} V \\ u \\ P \end{pmatrix}, \quad A(U) = V \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & K & 0 \end{pmatrix}, \qquad K = \left(P + \frac{\partial e}{\partial V}\right) \Big/ \frac{\partial e}{\partial P}$$

For example, using the polytropic gas EOS $e = \frac{PV}{\gamma - 1}$ $K = \left(\frac{c}{V}\right)^2$

Upwinding for Lagrangian Particles (1)

- Performing matrix diagonalization, we obtain three advection equations

$$U_t + R\Lambda R^{-1} U_x = 0,$$

$$R^{-1}U_t + \Lambda R^{-1} U_x = 0,$$

$$\Lambda = V \begin{pmatrix} 0 & & \\ & \sqrt{K} & \\ & & -\sqrt{K} \end{pmatrix} \quad R^{-1} = \begin{pmatrix} 1 & 0 & \frac{1}{K} \\ 0 & -\frac{1}{2\sqrt{K}} & -\frac{1}{2K} \\ 0 & \frac{1}{2\sqrt{K}} & -\frac{1}{2K} \end{pmatrix}$$

Represent the system in the component-wise form

$$V_t + \frac{1}{K} P_t = 0,$$

$K>0$

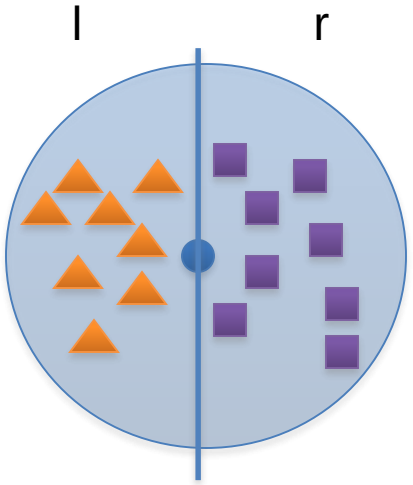
$$-\frac{1}{2\sqrt{K}} u_t - \frac{1}{2K} P_t = -V\sqrt{K}\left[-\frac{1}{2\sqrt{K}} u_x - \frac{1}{2K} P_x\right], \quad \longleftrightarrow \text{ Wave from left}$$

$$\frac{1}{2\sqrt{K}} u_t - \frac{1}{2K} P_t = V\sqrt{K}\left[\frac{1}{2\sqrt{K}} u_x - \frac{1}{2K} P_x\right]. \quad \longleftrightarrow \text{ Wave from right}$$

Upwinding for Lagrangian Particles: 1st order scheme

- Adding the subscripts l and r to the spatial derivatives

$$
\begin{aligned}
V_t &= \frac{V}{2}\left(u_{xr}+u_{xl}\right)-\frac{V}{2\sqrt{K}}\left(P_{xr}-P_{xl}\right), \\
u_t &= \frac{V\sqrt{K}}{2}\left(u_{xr}-u_{xl}\right)-\frac{V}{2}\left(P_{xr}+P_{xl}\right), \\
P_t &= -\frac{VK}{2}\left(u_{xr}+u_{xl}\right)+\frac{V\sqrt{K}}{2}\left(P_{xr}-P_{xl}\right).
\end{aligned}
$$



■ : RHS neighbors
▲ : LHS neighbors

- First order discretization of temporal derivatives of the state (V, u or P) at the location of particle j

$$\frac{\text{state}_j^{n+1}-\text{state}_j^{n}}{\Delta t}$$

- Moving the particles

$$\frac{x^{n+1}-x^{n}}{\Delta t} = \frac{1}{2}\left(u^{n}+u^{n+1}\right)$$

- The above yields a first order scheme

Computing Derivatives.

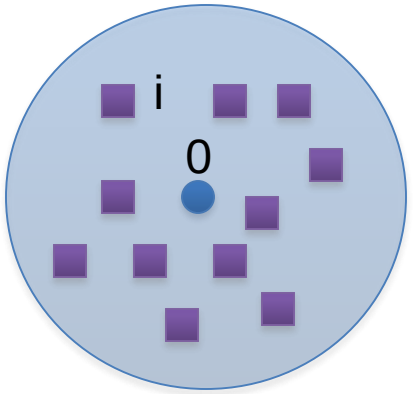
Local Polynomial Fitting (Generalized Finite Differences)

• In 2D at the vicinity of a point 0, the function value in the location of a point i can be expressed as

$$U_i = U_0 + h_i \left.\frac{\partial U}{\partial x}\right|_0 + k_i \left.\frac{\partial U}{\partial y}\right|_0 + \frac{1}{2}\left(h_i^2 \left.\frac{\partial^2 U}{\partial x^2}\right|_0 + k_i^2 \left.\frac{\partial^2 U}{\partial y^2}\right|_0 + 2h_i k_i \left.\frac{\partial^2 U}{\partial x \partial y}\right|_0 \right) + \dots$$

• Second order approximation

$$\tilde{U} = U_0 + h_i\theta_1 + k_i\theta_2 + \frac{1}{2}h_i^2\theta_3 + \frac{1}{2}k_i^2\theta_4 + h_i k_i \theta_5$$



• Using *n* neighbours:

$$\begin{bmatrix} h_1 & k_1 & \frac{1}{2}h_1^2 & \frac{1}{2}k_1^2 & h_1k_1 \\ h_2 & k_2 & \frac{1}{2}h_2^2 & \frac{1}{2}k_2^2 & h_2k_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_n & k_n & \frac{1}{2}h_n^2 & \frac{1}{2}k_n^2 & h_nk_n \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} = \begin{bmatrix} U_1 - U_0 \\ U_2 - U_0 \\ \vdots \\ U_n - U_0 \end{bmatrix}$$

Solve using QR to obtain derivatives **convergent to prescribed order**

Second-Order Accurate Lagrangian Particle Method

Generalization of a Beam-Warming Scheme $O(\Delta t^2, \Delta x^2, \Delta t \Delta x)$

$$U_t + A(U)U_x - \frac{\Delta t}{2} A^2(U) U_{xx} = 0$$

$$\Rightarrow U_t = -R\Lambda R^{-1} U_x + \frac{\Delta t}{2} R\Lambda^2 R^{-1} U_{xx}$$

$$\Rightarrow R^{-1} U_t = -\Lambda R^{-1} U_x + \frac{\Delta t}{2} \Lambda^2 R^{-1} U_{xx}$$

- Component-wise form:

$$\begin{aligned}
V_t &= \frac{V}{2}\left(u_{xr} + u_{xl}\right) - \frac{V}{2\sqrt{K}}\left(P_{xr} - P_{xl}\right) \\
&+ \frac{\Delta t}{4}\left[V^2\sqrt{K}\left(u_{xxr} - u_{xxl}\right) - V^2\left(P_{xxr} + P_{xxl}\right)\right] \\
u_t &= \frac{V\sqrt{K}}{2}\left(u_{xr} - u_{xl}\right) - \frac{V}{2}\left(P_{xr} + P_{xl}\right) \\
&+ \frac{\Delta t}{4}\left[V^2 K\left(u_{xxr} + u_{xxl}\right) - V^2\sqrt{K}\left(P_{xxr} - P_{xxl}\right)\right] \\
P_t &= -\frac{VK}{2}\left(u_{xr} + u_{xl}\right) + \frac{V\sqrt{K}}{2}\left(P_{xr} - P_{xl}\right) \\
&+ \frac{\Delta t}{4}\left[-V^2 K^{\frac{3}{2}}\left(u_{xxr} - u_{xxl}\right) + V^2 K\left(P_{xxr} + P_{xxl}\right)\right]
\end{aligned}$$

Octree Neighbour Search & Variable Neighbour Search

- The neighbour search algorithm is based on octrees (3D) and quadtrees (2D).
  - The use of Octree is for variable search radius for each particle, which is essential for highly compressible materials
  - The bucket search algorithm cannot be used in the case of variable search radius
  - The octree method is efficient (O(N logN)) and scales well

- We update the neighbor search radius of a particle based on the specific volume

$$r_i^{t+1} = r_i^t \left(\frac{V_i^{t+1}}{V_i^t} \right)^{\frac{1}{3}}$$

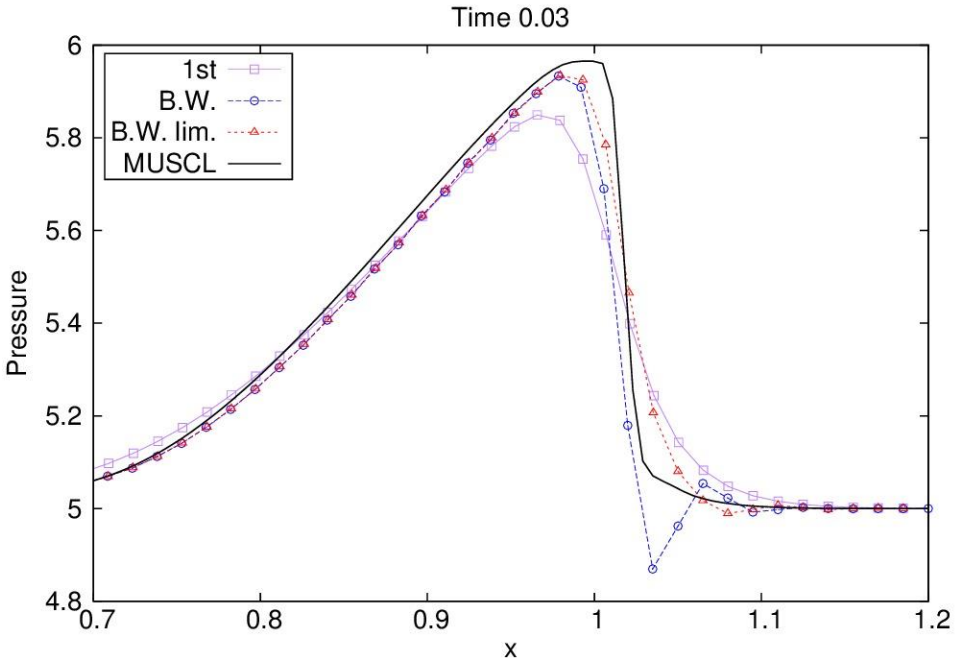
- Free surface: ghost particles with vacuum state to obtain convergence of fluid states at the vacuum interface to correct solutions of the Riemann Problem

Elimination of Numerical Dispersion and Algorithm for Free Surfaces

- Multidimensional case: Strang splitting that maintains 2nd order accuracy
- **Limiters for controlling oscillations:** detect oscillations and blend the 2nd and 1st order methods in the location of oscillations / shocks
    - No need for artificial viscosity

- Particles close to a free surface do not have full particle neighborhood. It is completed using vacuum ghost particles
- Algorithms for ghost particle states: Riemann problem for boundary particles in the direction normal to the interface for multiphase problems
    - Simplified algorithm for vacuum interface: vacuum pressure and density and GFD-based interpolation of velocitiy

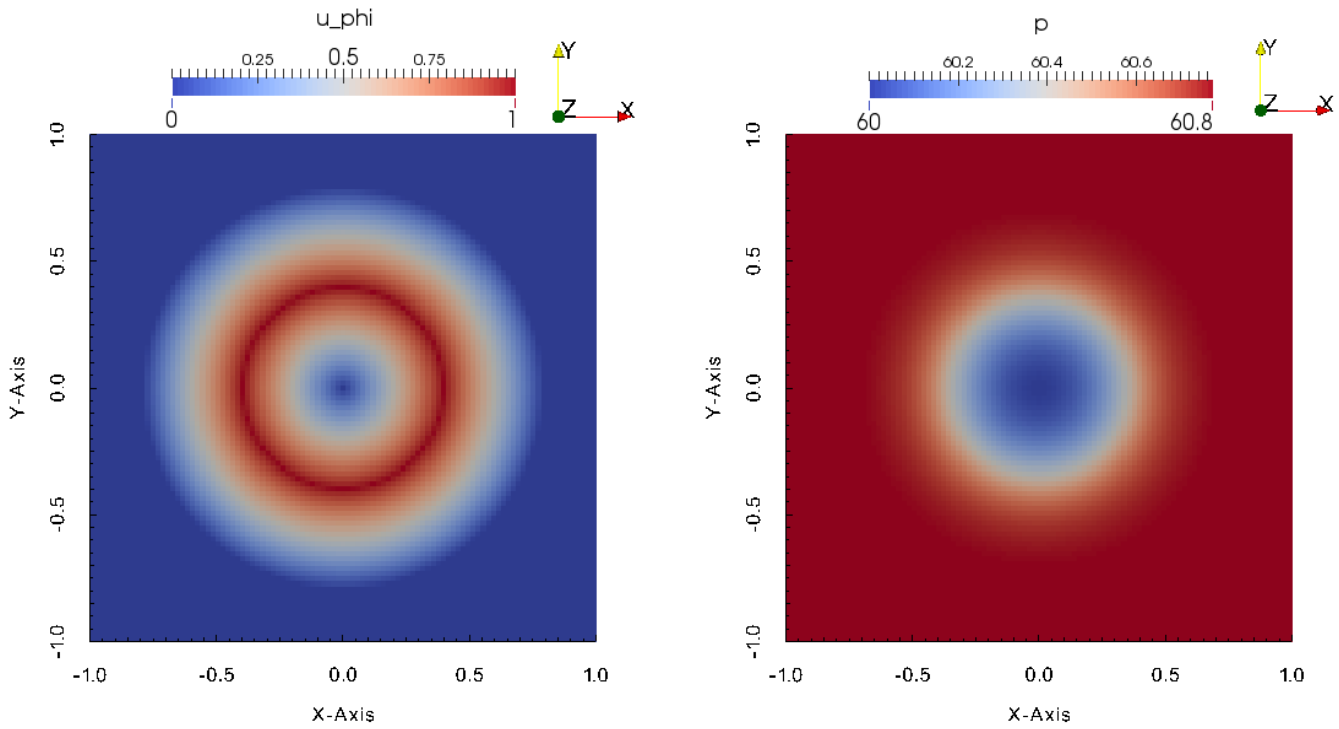
Verification: formation of shocks and effect of limiters



Number of particles	Relative $L2$-norm error	Rate of Convergence
240	0.051	NA
480	0.018	2.88
960	0.0049	3.60
1920	0.0012	4.02
3840	0.00029	4.23
7680	0.000068	4.24

Verification: Gresho vortex

Gresho vortex is a steady-state vortex in which special distribution of pressure compensates the centripetal force



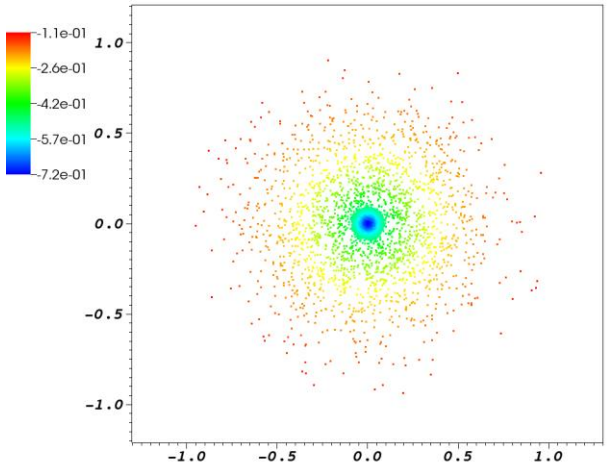
- Very difficult test for SPH codes
- Grid-based second-order codes report empirical order of convergence ~ 1.3
- Our results: 2nd order of convergence at initial stages; reduces to 1st order at late time

- Generalization to Elliptic Problems

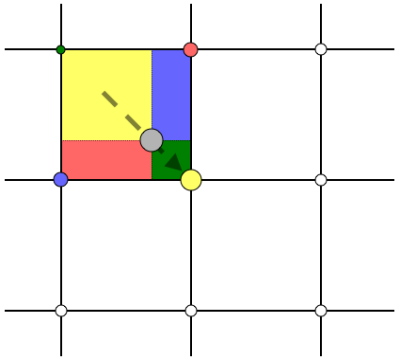
- Adaptive Particle-in-Cloud Method: a highly adaptive and artifact-free replacement for Particle-in-Cell method for Vlasov-Poisson problems

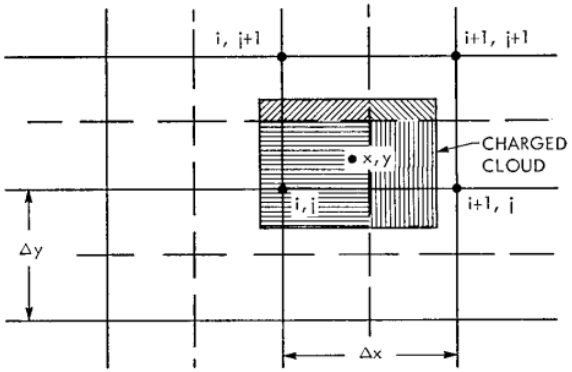
PIC Method

- Represent distribution function by macro-particles
  - Each macro-particle represents a large number of real particles



- Build a mesh around particles
- Deposit charge of particles on the mesh

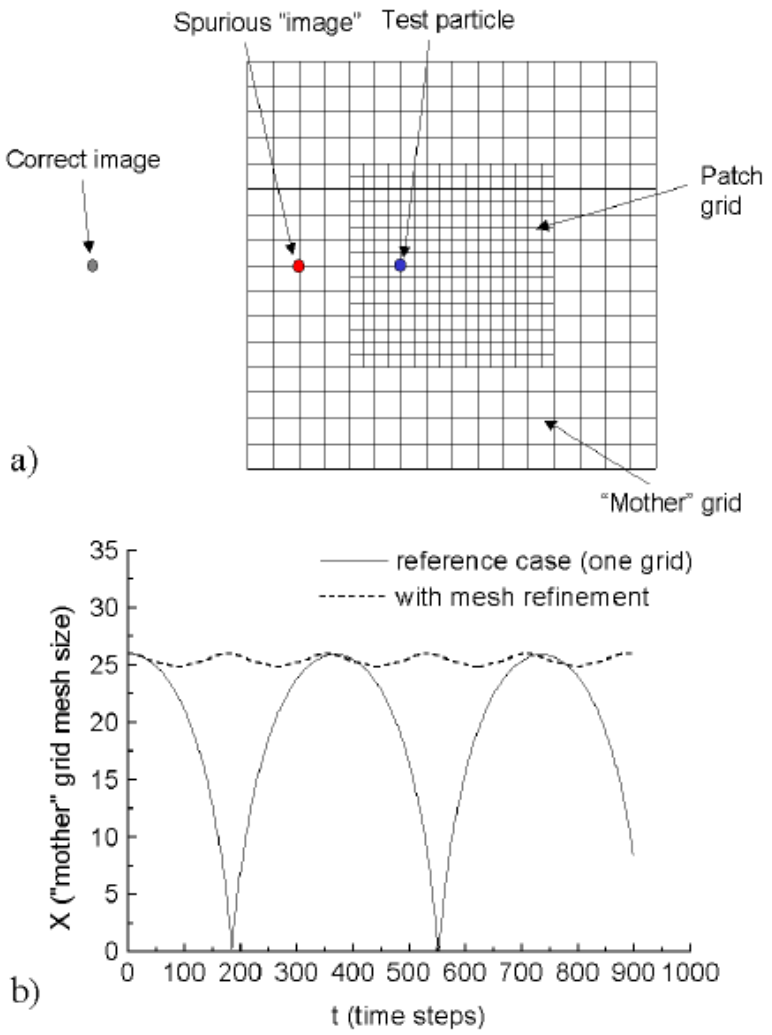




- Solve Poisson problem on the mesh using either FFT of fast linear solvers. Compute forces on the mesh
- Interpolate forces to the location of particles and propagate particles

PIC deficiencies

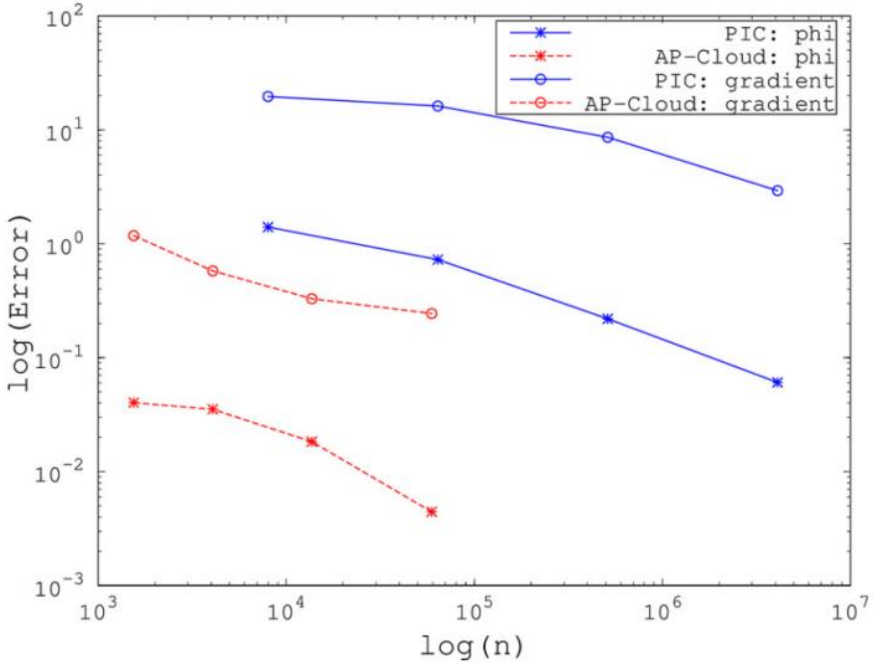
- Discretization error is not balanced with the error of charge computation
- Poor accuracy for highly non-uniform systems
- AMR-PIC (adaptive mesh refinement) introduces strong artificial forces due to spurious images (mitigation methods have been suggested)

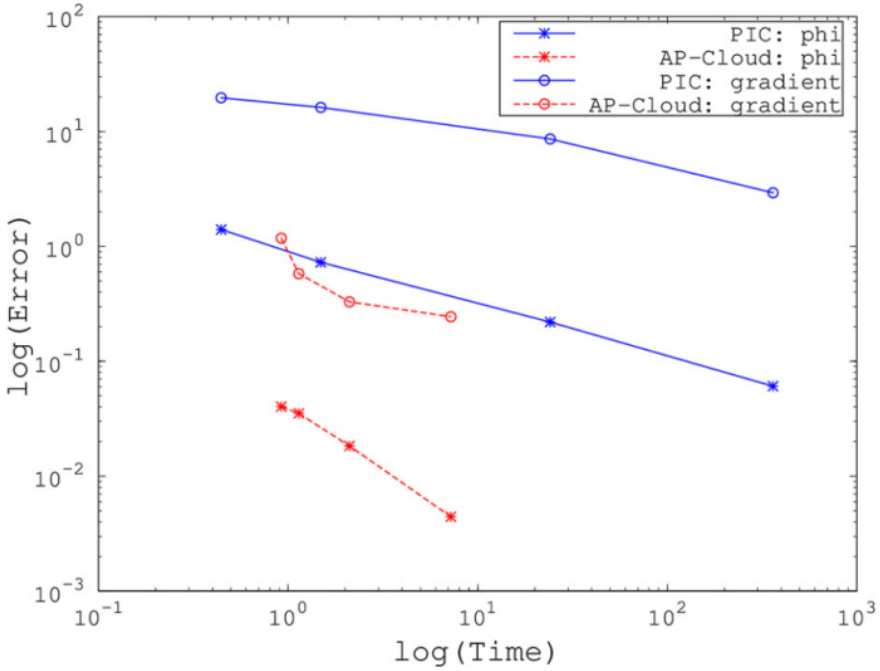


AP-Cloud Method

- Discretization PIC Cartesian mesh is replaced by **computational particles on an octree data structure**
- The density of **computational particles** is chosen adaptively
- GFD (a weighted least-square approximation) is used as discretization, integration, and interpolation framework
- The error from GFD and the source integration is balanced leading to minimization of the total error
- The method is free of artifacts typical for AMR-PIC
- Simple implementation of geometrically complex boundaries / open boundary conditions
- **Improvement of accuracy and computational speed** compared to traditional PIC. **Higher convergence order compared to PIC**

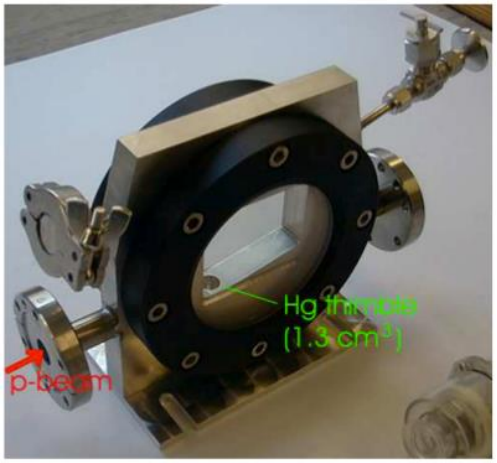
Accurcy and Performance of AP-Cloud vs PIC

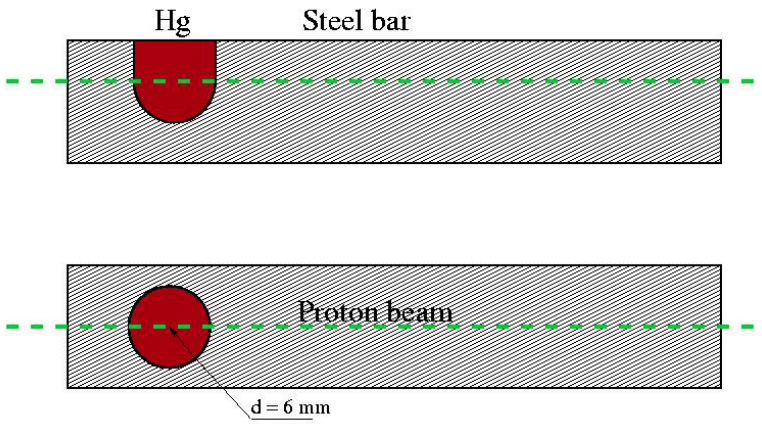




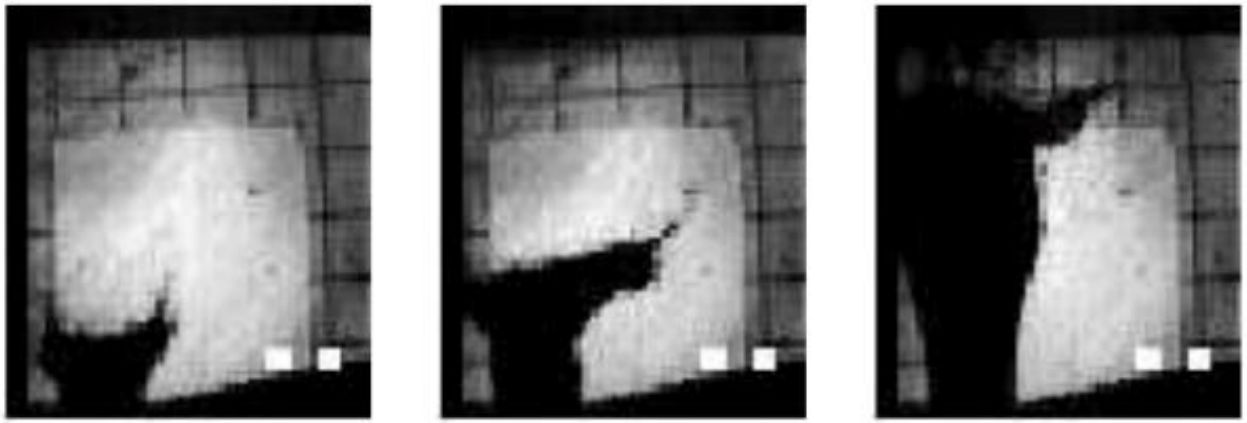
Application of Lagrangian Particle Methods to High Power Targets

Validation: Mercury Thimble Experiment





Experimental device



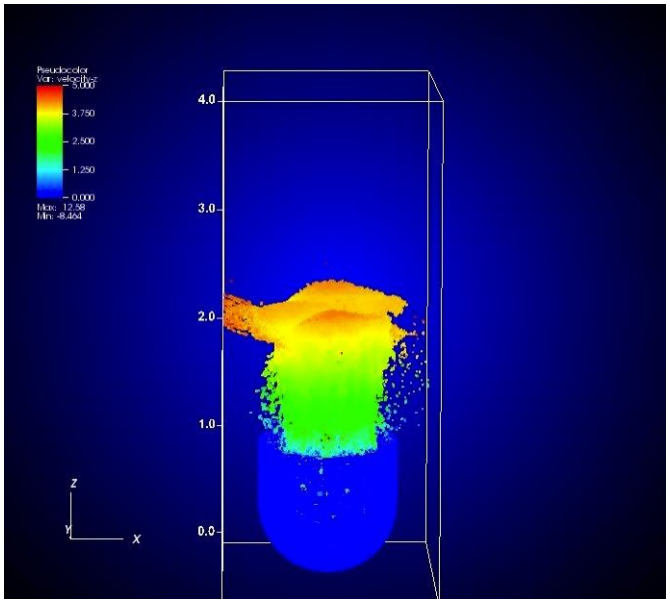
Experimental images of mercury splash at t = 0.88, 0.125, 0.7 ms after impact of 12 teraproton beam (A. Fabich)

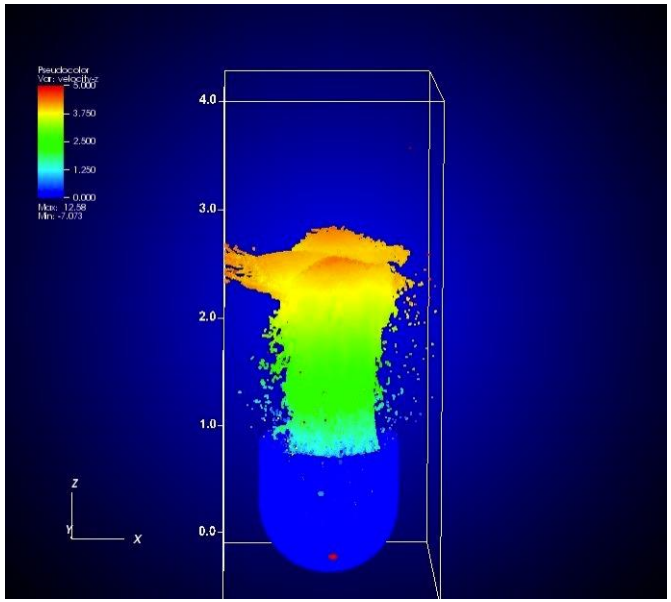
Validation: Simulation of Mercury Thimble Experiments

- EOS: analytic model for liquids that includes tension (negative pressure)
  - Coefficients of EOS calculated using mercury data tables from Sandia
  - Cavitation algorithm allows particles to separate if the pressure drops below critical pressure

- Good agreement with mercury thimble experiments (A.Fabich, Doctoral thesis)

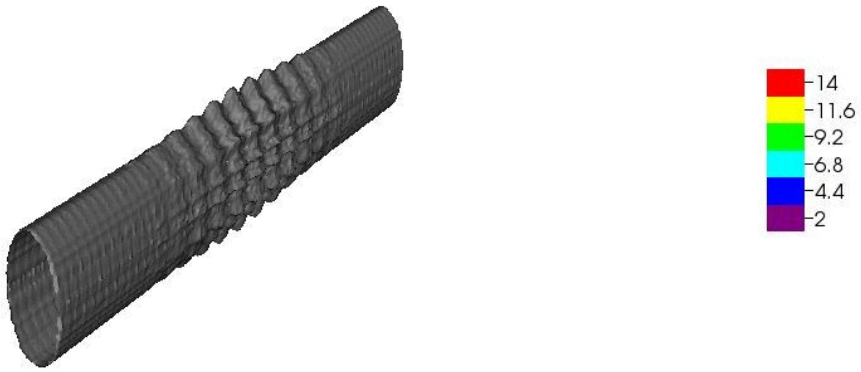
LP simulation results at t = 0.5 and 0.7 ms





Mercury Jet after Interaction with Proton Pulse in the parameter regime of muon collider

Velocities of the mercury disruption reach: ~110 m/s (shorter axis), ~40 m/s (longer axis)



Simulation of Tungsten Powder Target Experiments

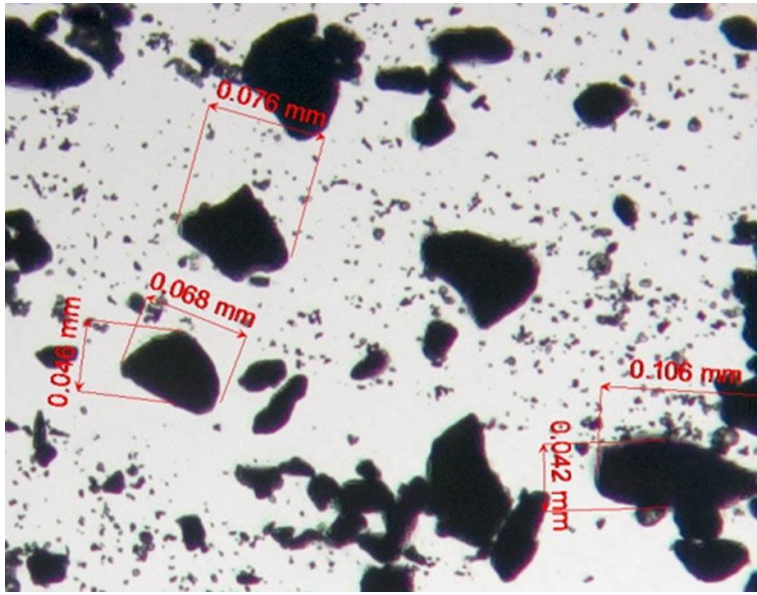
Modeling assumptions

- Tungsten powder simulations could be obtained using Molecular Dynamics (MD)

- Difficulties: large range of particle size / shape distribution, frictional forces, mixture with helium gas / compressibility, high computational cost, etc.

- I believe that continuum approach to the simulation of tungsten powder has significant advantages
  - Easy to model friction, mixture properties, etc.
  - Potential future applications may involve mixture with liquid (difficult for MD)

  - Continuum simulations have been successfully applied to many granular flow regimes



Microscope image of tungsten powder (O. Caretta et al., PRSTAB, 17, 101005, 2014.

Modeling assumptions (cont.)

- Continuum modeling has one potential difficulty: the **closure model**

- **David G. Schaeffer**, Professor at Duke University: *"Although I worked in granular flow for 15 years, I largely stopped working in this area. Part of my fascination with this field derived from the fact that typically constitutive equations derived from engineering approximations lead to ill-posed PDE. However, I came to believe that the lack of well-posed governing equations was the major obstacle to progress in the field, and I believe that finding appropriate constitutive relations is a task better suited for physicsts than mathematicians, so I reluctantly moved on"*

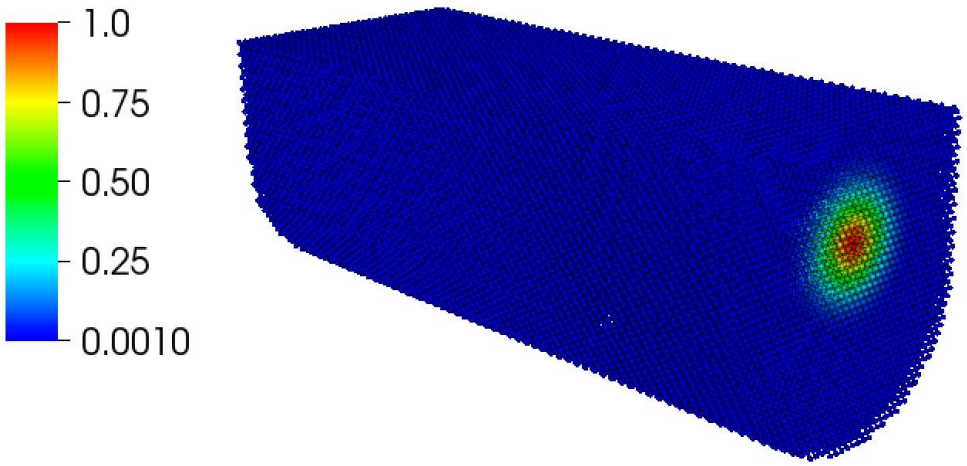
- New progress on **closure models** for granular materials has been achieved. We are in the process of implementing an improved closure model
  - More detailed experimental measurements from CERN required

- In current work, we obtained satisfactory results with a simplified model: a nonlinear modification of polytropic equation of state model

Initial geometry and energy deposition

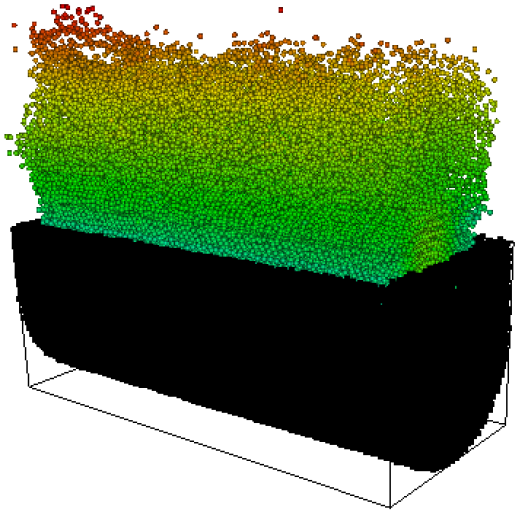
- The geometry of the trough is identical in the transverse dimension to the inner titanium trough used in the experiment. Longitudinal length was 4 cm

- Energy deposition profiles calculated by the FLUKA code were implemented in the transverse direction. Current simulations used uniform longitudinal deposition

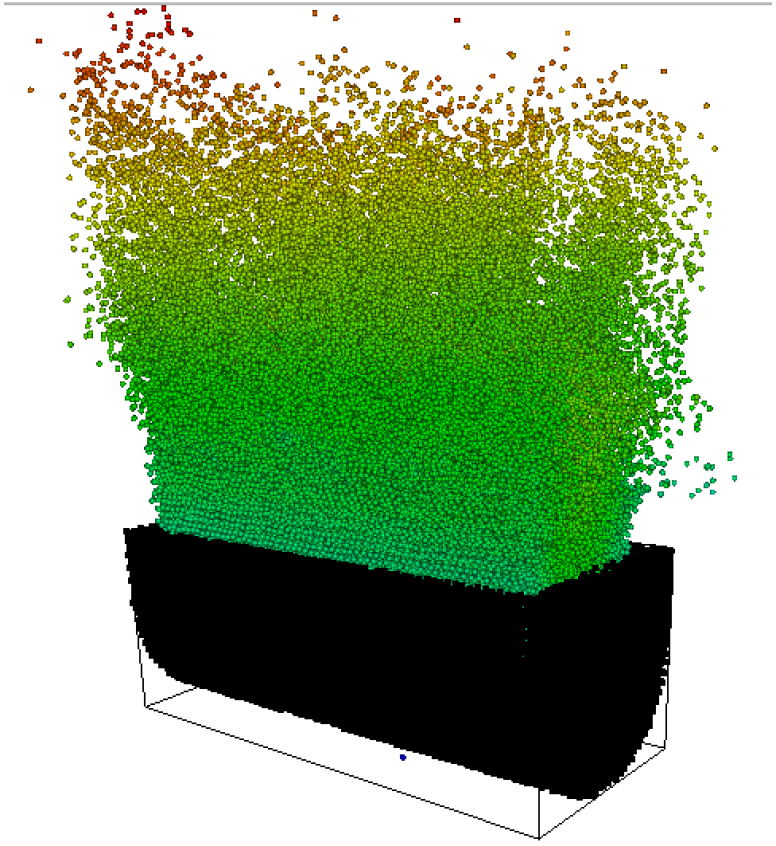


- Problem of converting energy depositions and the the consequent thermal expansions into pressure profiles. Theoretical models were employed but these contain significant uncertainties. The ultimate benchmark is obtaining experimental splash velocities.

Evolution of splash: uniform initial height



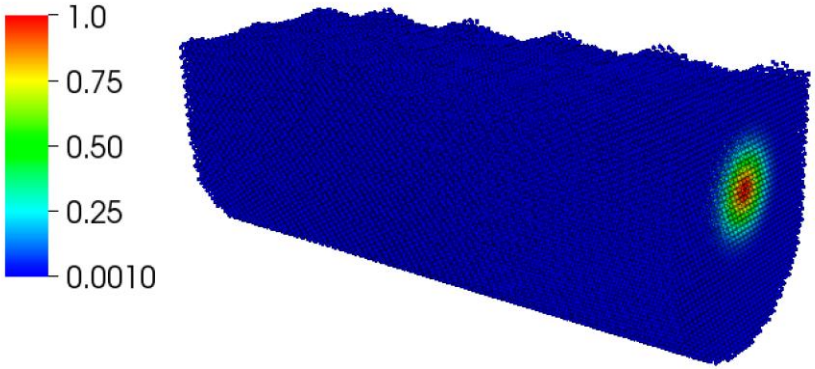
1.8 cm height of powder lift



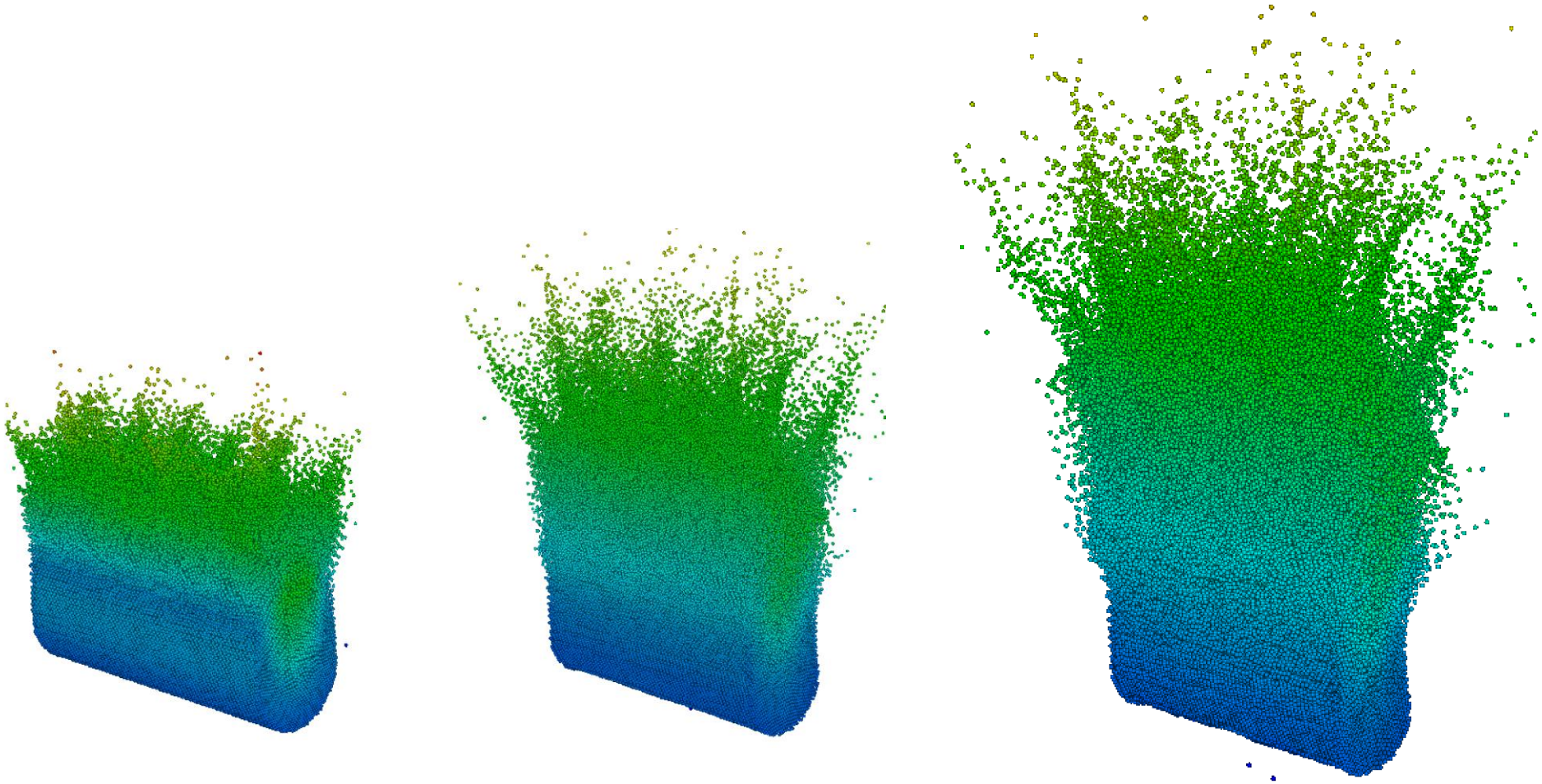
4 cm height of powder lift

Initialization of particle distribution with perturbed initial height

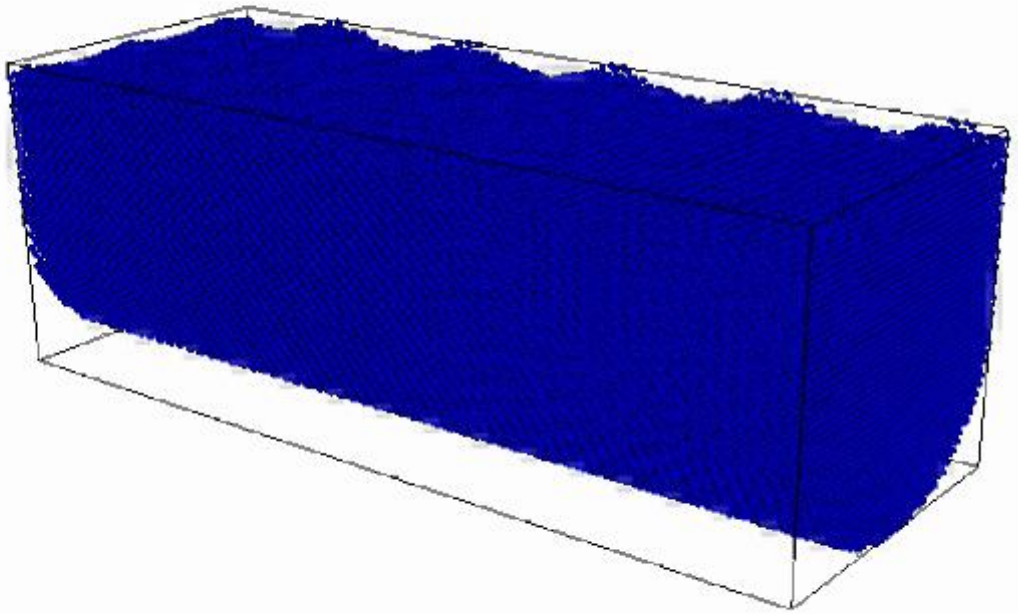
- To emulate initial experimental conditions after multiple shots, small amplitude surface perturbations were applied to the initial particle distribution.



Evolution of splash: perturbed initial height



Evolution of tungsten powder splash with velocity of 0.45 m/s.



Summary of powder target simulations

- Numerical simulations obtained velocities of the powder splash in the experimental range: 0.1 – 0.5 m/s
- Uniform height of the initial power distribution resulted in close to uniform lift of powder; initial surface perturbations led to spikes of the powder material
- Given a very simplified closure model, we ontained reasonably good agreement with experiments
- Additional work on the closure model (with input from experiments) is needed in order to obtain a robust simulation model which that can be successfully used for predictive simulations in beyond-experimental regimes

Conclusions

- Recently developed Lagrangian Particle (LP) method for hydrodynamics type systems presents a significant improvement in accuracy compared to SPH

- Generalization to elliptic problems. Adaptive Particle-in-Cloud is a method for optimal solutions of Vlasov-Poisson problems traditionally solved with PIC, that replaces the PIC mesh with adaptively chosen computational particles

- Applications to high power mercury target problems

- Application of the LP method to tungsten powder targets
  - Satisfactory results were obtained with a simplified closure problem
  - Work on more accurate close model is underway; CERN experimental input would greatly help this work