Exceptional service in the national interest

Sandia National Laboratories

Development of Scalable Parallel Implicit SPH using LAMMPS and Trilinos

Kyungjoo Kim[1]

N. Trask[2], M. Maxey[2], M. Perego[1], M. Parks[1], K. Yang[3] and J. Xu[3]

[1] *Center for Computing Research, Sandia National Labs,* [2] *Brown University,* [3] *Pennsylvania State University*

LAMMPS Users' Workshop and Symposium, August 6, 2015

U.S. DEPARTMENT OF ENERGY | National Nuclear Security Administration | Center for Computing Research

LAMMPS/Trilinos Integration

Smoothed Particle Hydrodynamics

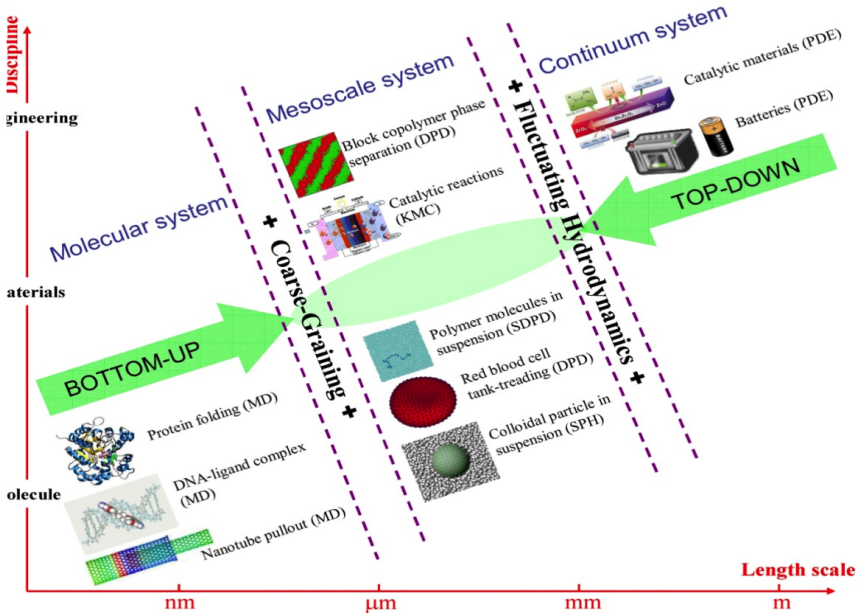
Implementation of Navier Stokes Equations

Numerical Examples

Conclusion

LAMMPS/Trilinos Integration

Collaboratory on Mathematics for Mesoscopic Modeling of Materials

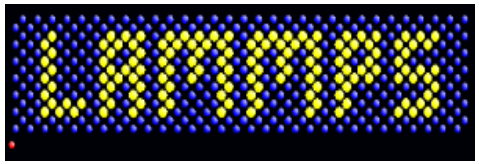


Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965
PENNSTATE 1855
Sandia National Laboratories
UCSB
BROWN PURDUE UNIVERSITY

Research focuses of CM4

- Funded by ASCR MMICC, DOE.
- Developing particle- and grid-based methods for mesoscale material processes.
- Concurrent coupling of these methods.
- Exploring fast solution techniques for exascale computing.
- Integrating mathematical and computational modesl for applications relevant to synthesis of new materials.

Overview: Massively Parallel 3D Implicit SPH Methods



Goal

- Develop large scale parallel 3D implicit simulation capability.
- Use LAMMPS, Sandia's massively parallel molecular dynamics code.
  - LAMMPS is a classical molecular dynamics code.
  - LAMMPS can simulate any particle system *e.g.*, MD, SPH, DPD, *etc.*
  - Provides modular framework easy to add new capabilities.
  - Demonstrated massively parallel scalability via MPI and spatial domain decomposition.

Problem

- LAMMPS has no capability for implicit time integration.
  - Only explicit time integration used in MD.
  - Need distributed memory parallel linear algebra infrastructure: *e.g.*, vectors, matrices, linear solvers, preconditioners, *etc.*

Solution

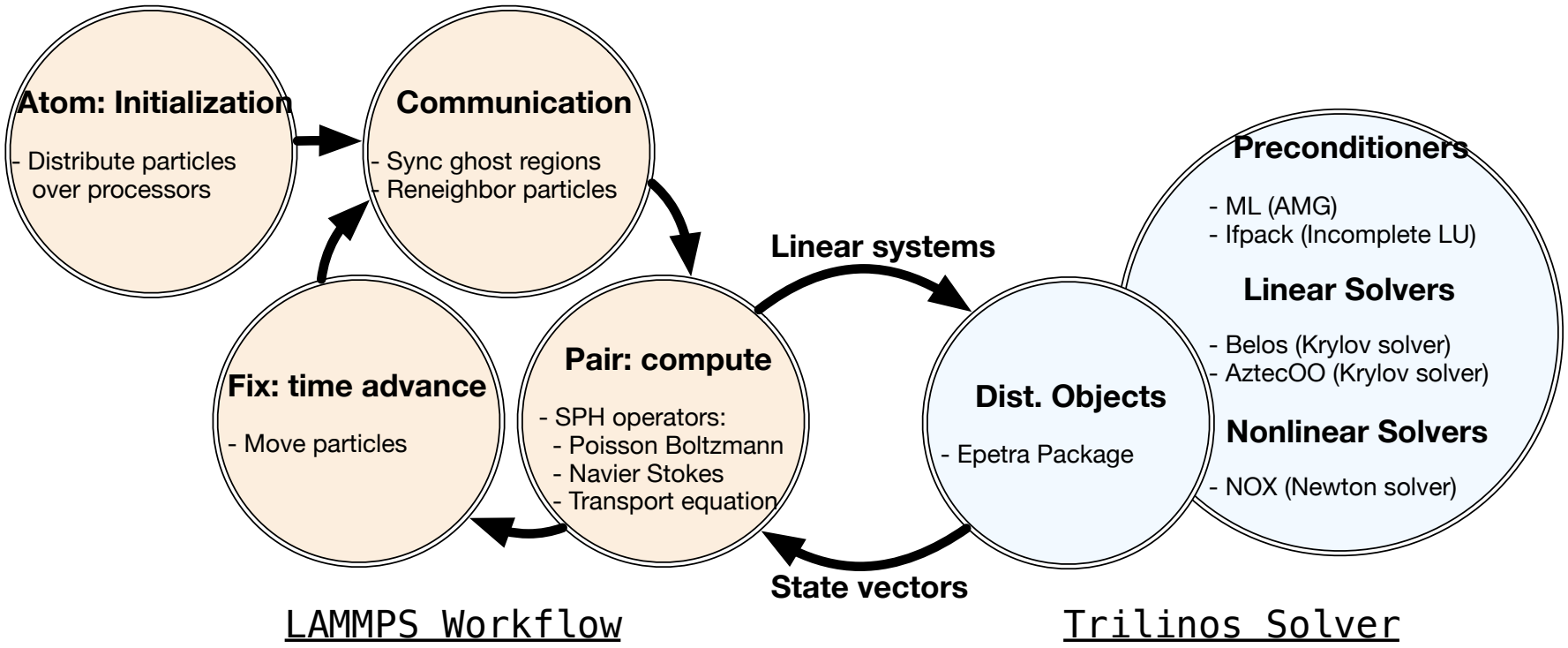
- Integrate LAMMPS with Trilinos solver packages.

Trilinos

- Open source C++ software framework for solving large scale multi-physics scientific and engineering problems: `https://trilinos.org`.
- Developed and maintained by Sandia National Labs.
- Trilinos is made of packages:
  - The current Trilinos library consists of more than 50 packages.
  - Each package is an independent piece of software but inter-operates with other packages.
  - Use a set of packages as needed, like LEGO blocks.

By Alan Chia (Lego Color Bricks) CC BY-SA 2.0 via Wikimedia Commons.

LAMMPS/Trilinos Integration



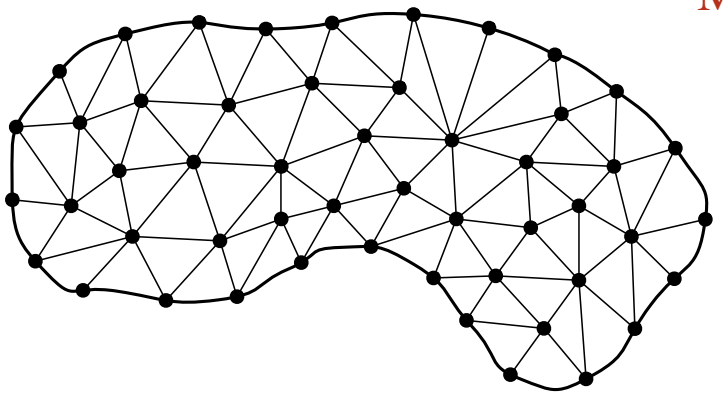
Let each code handle what it was designed to do well

- LAMMPS handles particle data, parallel data distribution, ghosting.
- Trilinos handles distributed memory linear solvers, preconditioners, *etc.*
- Developed implicit solver and time integration framework can be applied to any particle based models in LAMMPS.

Sandia National Laboratories

Smoothed Particle Hydrodynamics

Meshfree Particle methods

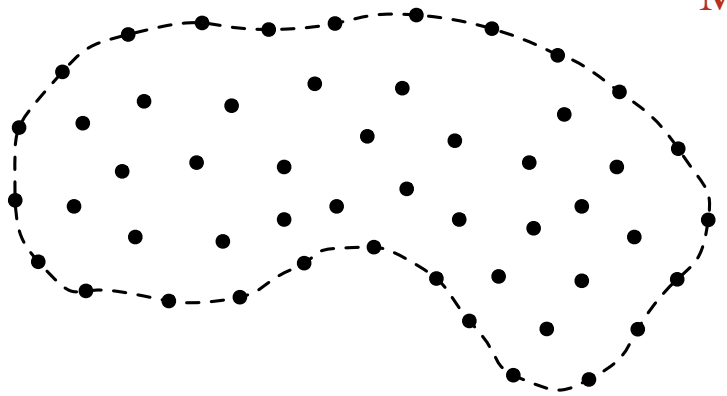


Mesh: a list of points with their connectivities.

Motivation of meshfree methods

- Generating a suitable mesh is a challenging task.
- Easier to handle large deformation, moving boundary and fluid structure interaction problems than grid-based approaches.
- By advecting points in Lagrangian form, the non-linear advection term in Navier Stokes equations can be removed.

Meshfree Particle methods

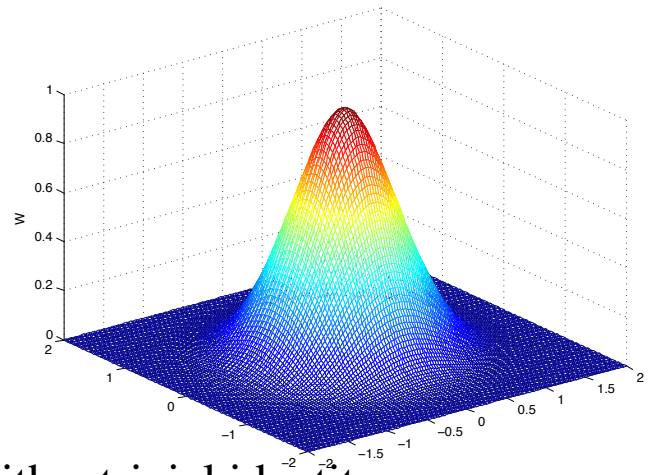


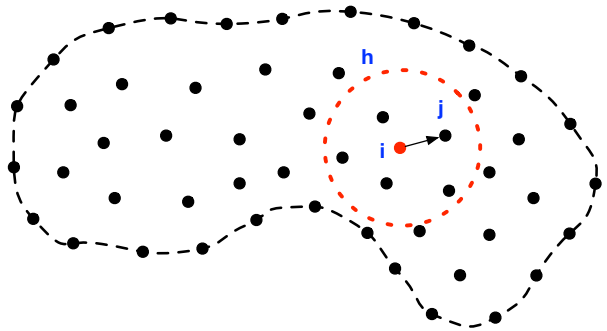
Meshfree: points are scattered on the domain.

Motivation of meshfree methods

- Generating a suitable mesh is a challenging task.
- Easier to handle large deformation, moving boundary and fluid structure interaction problems than grid-based approaches.
- By advecting points in Lagrangian form, the non-linear advection term in Navier Stokes equations can be removed.

Smoothed Particle Hydrodynamics (SPH) Interpolation





Begin with a trivial identity

$$f(x) = \int f(x')\delta(x-x')dx'.$$

Consider integral interpolants with a compact support characterized by h:

$$f(x) = \int f(x')W(x-x',h)dx' \quad \rightarrow \quad <f(x_i)> = \sum_j^N f(x_j)W(x_i-x_j,h)V_j.$$

W is an interpolating kernel with these properties:

$$\int W(u,h)du = 1 \quad \text{and} \quad \lim_{h\to 0} W(u,h) = \delta(u).$$

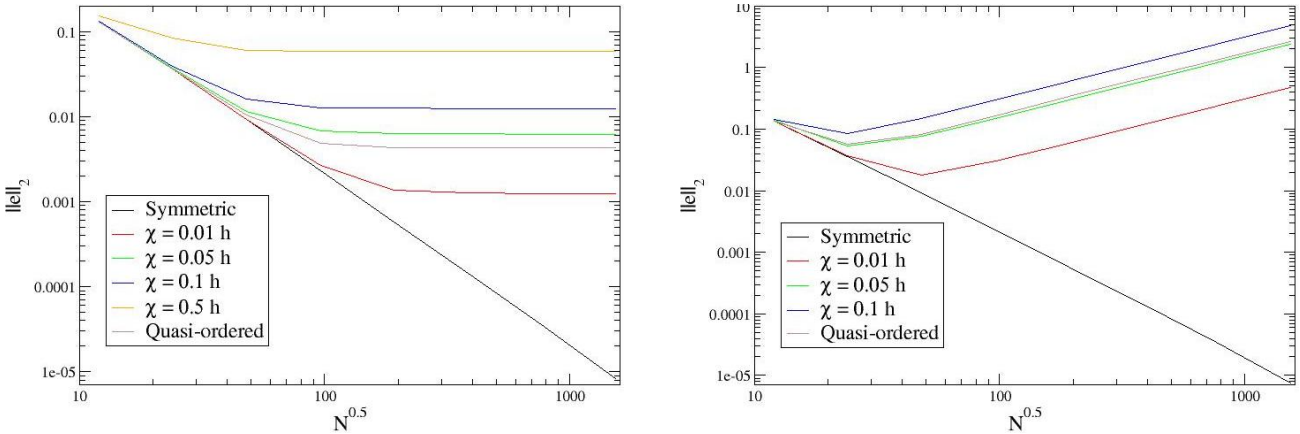
Second Order SPH Discretization

Standard SPH operators are defined as:

$$\nabla_0 f_i = \sum_j^N (f_j - f_i) \nabla_{x_i} W_{ij} V_j$$

$$\nabla_0^2 f_i = 2 \sum_j^N \frac{f_i - f_j}{r_{ij}} \mathbf{e}_{ij} \cdot \nabla_{x_i} W_{ij} V_j.$$

These operators lack 1st order consistency.



L^2 error for uncorrected gradient and Laplacian operators; χ is random perturbation applied to particles.

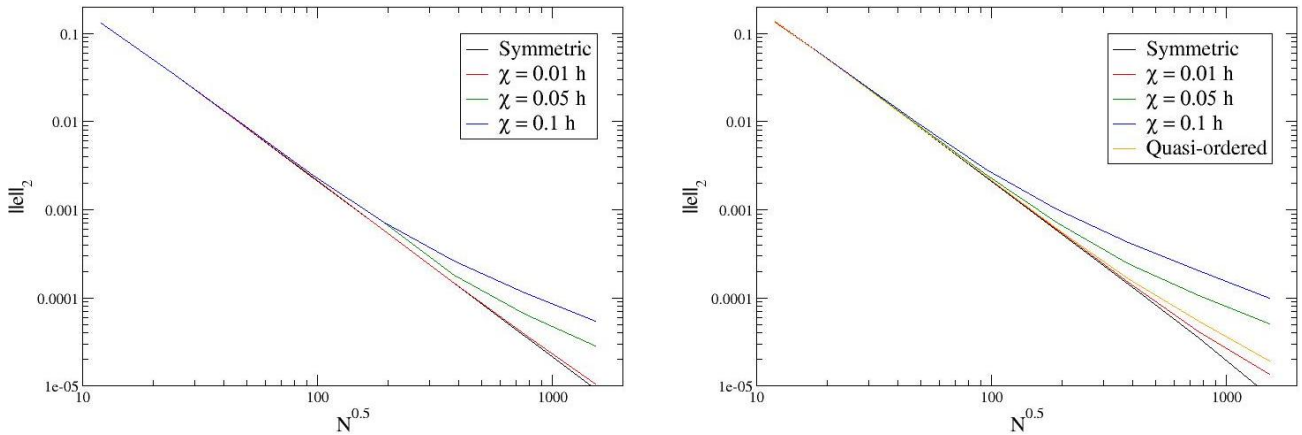
Second Order SPH Discretization

The "corrected" SPH scheme uses correction tensors to obtain 1st order consistency:

$$\nabla_1 f_i = \sum_j^N (f_j - f_i)\mathbf{G}_i \nabla_{x_i} W_{ij} V_j,$$

$$\nabla_1^2 f_i = 2\sum_j^N \left(\mathbf{L}_i : \mathbf{e}_{ij} \otimes \nabla_{x_i} W_{ij}\right) \left(\frac{f_i - f_j}{r_{ij}} \mathbf{e}_{ij} \cdot \nabla_1 f_i\right) V_j,$$

where the correction tensors **G** and **L** are derived from a Taylor expansion[2].



L^2 error for "corrected" gradient and Laplacian operators; χ is random perturbation applied to particles.

[1] N.Trask et al."A scalable consistent second-order SPH solver for unsteady viscous flows", CMAME 2015.

Implementation of Navier Stokes Equations

Projection Scheme

Consider a incompressible flow governed by the Navier-Stokes (NS) equations:

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{g},$$
$$\nabla \cdot u = 0,$$

where $\mathbf{g}$ is a body force. Splitting the equations into prediction/correction steps, we get:

$$\textit{Helmholtz} \begin{cases} \dfrac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\dfrac{1}{\rho}\nabla p^n + \nu\nabla^2\left(\dfrac{\mathbf{u}^* + \mathbf{u}^n}{2}\right) + \mathbf{g} & \mathbf{x} \in \Omega, \\ \mathbf{u}^* = \mathbf{u}_{\partial\Omega} & \mathbf{x} \in \partial\Omega, \end{cases}$$

$$\textit{Corrector} \begin{cases} \dfrac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\dfrac{1}{\rho}\nabla\left(p^{n+1} - p^n\right) & \mathbf{x} \in \Omega, \\ \nabla \cdot \mathbf{u}^{n+1} = 0 & \mathbf{x} \in \Omega, \\ \mathbf{u}^{n+1} \cdot \mathbf{n} = \mathbf{u}_{\partial\Omega} \cdot \mathbf{n} & \mathbf{x} \in \partial\Omega. \end{cases}$$

Projection Scheme

Splitting the equations into prediction/correction steps, we get:

$$\textit{Helmholtz}\begin{cases}\dfrac{\mathbf{u}^*-\mathbf{u}^n}{\Delta t}=-\dfrac{1}{\rho}\nabla p^n+\nu\nabla^2\left(\dfrac{\mathbf{u}^*+\mathbf{u}^n}{2}\right)+\mathbf{g} & \mathbf{x}\in\Omega,\\ \mathbf{u}^*=\mathbf{u}_{\partial\Omega} & \mathbf{x}\in\partial\Omega,\end{cases}$$

$$\textit{Corrector}\begin{cases}\dfrac{\mathbf{u}^{n+1}-\mathbf{u}^*}{\Delta t}=-\dfrac{1}{\rho}\nabla\left(p^{n+1}-p^n\right) & \mathbf{x}\in\Omega,\\ \nabla\cdot\mathbf{u}^{n+1}=0 & \mathbf{x}\in\Omega,\\ \mathbf{u}^{n+1}\cdot\mathbf{n}=\mathbf{u}_{\partial\Omega}\cdot\mathbf{n} & \mathbf{x}\in\partial\Omega.\end{cases}$$

By taking the divergence of the second set of equations, we obtain the Poisson problem for the pressure difference:

$$\textit{Poisson}\begin{cases}-\dfrac{1}{\rho}\nabla^2\left(p^{n+1}-p^n\right)=-\dfrac{\nabla\cdot\mathbf{u}^*}{\Delta t} & \mathbf{x}\in\Omega,\\ \nabla\left(p^{n+1}-p^n\right)\cdot\mathbf{n}=0 & \mathbf{x}\in\partial\Omega.\end{cases}$$

Resulting systems of equations are solved by a preconditioned (algebraic multigrid) GMRES solver.

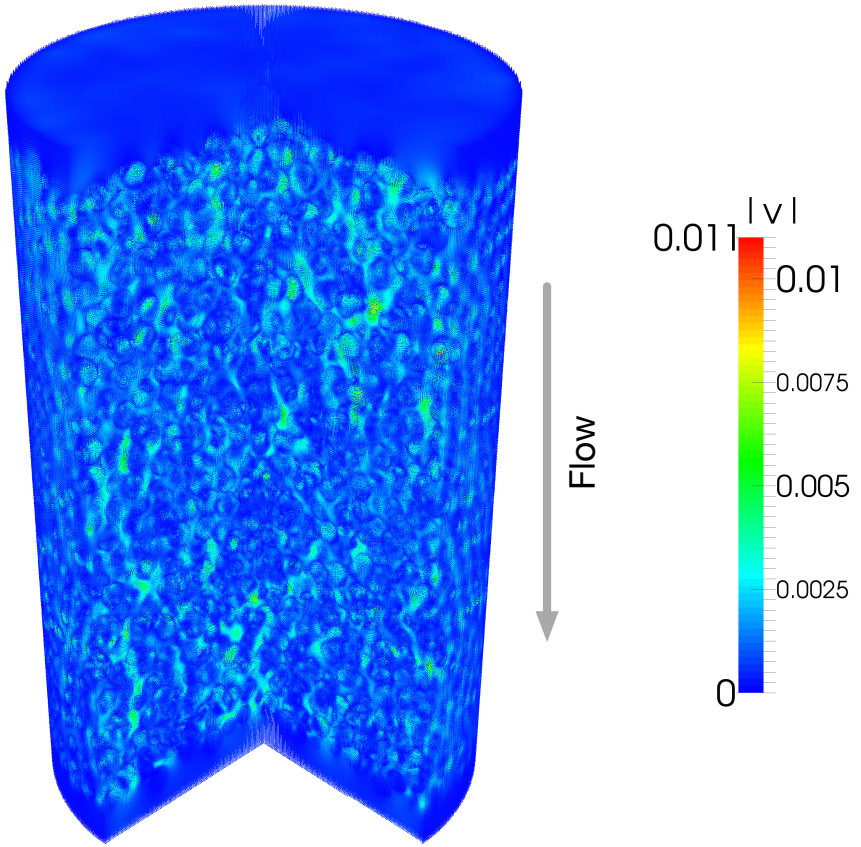
Numerical Examples

- 3D Complex geometry: Pore-scale Flow in Bead Pack.

Benchmark: 3D Pore-scale Flow in Bead Pack[3]

The pore geometry is constructed from voxel data provided by MRI measurements.

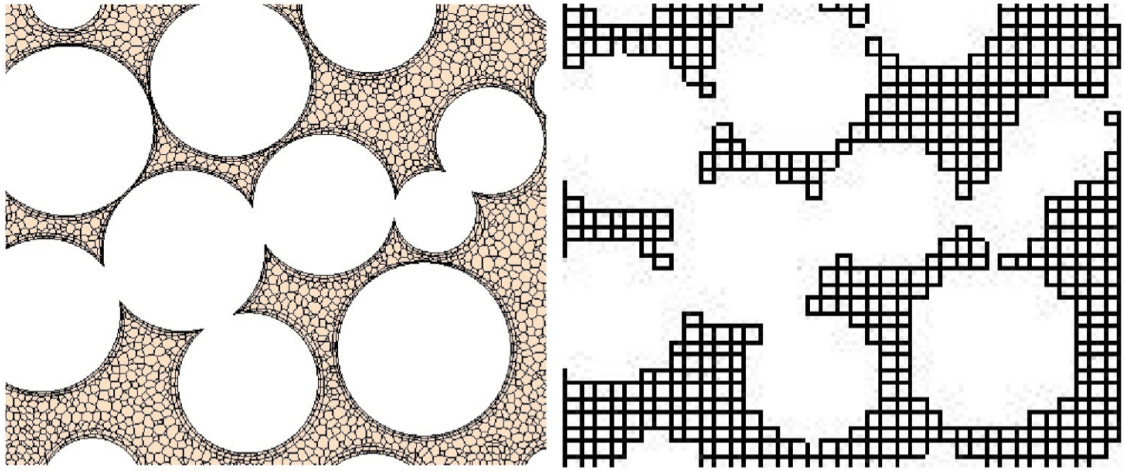
Parameter	Symbol	Value
Bead diagmeter	d (mm)	0.5
# of Beads	-	6864
Column diameter	D (mm)	8.8
Column length	L (mm)	12.8
Porocsity	ε	0.4267
Volumetric flow rate	Q (kg/s)	2.771e-5
Fluid density	ρ (kg/m^3)	997.561
Fluid dynamic viscosity	μ ($pa-s$)	8.887e-4



Steady-state solution of the flow in a bead pack.

[2] Yang. et. al., Intercomparison of 3D Pore-scale Flow and Solute Transport Simulation Methods, Advances in Water Resources, in review.

Benchmark: 3D Pore-scale Flow in Bead Pack



StarCCM+ *TETHYS*

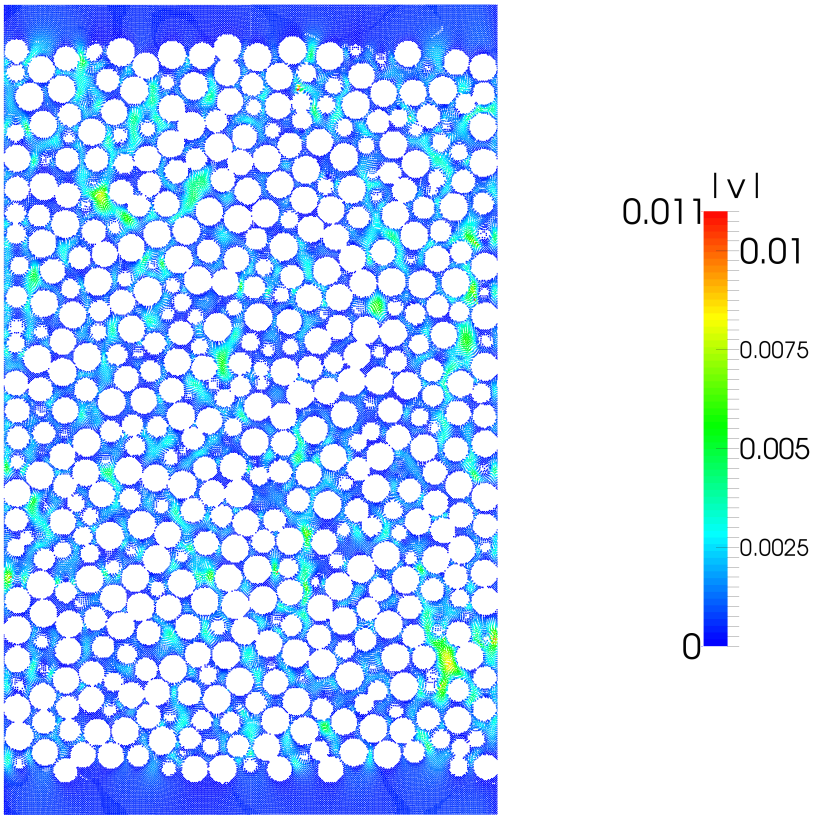
Computational cost of different methods

Code	Mesh	40 [μm]		20 [μm]		Description
		Time	Machine	Time	Machine	
StarCCM+	Tet	15 hrs	4 cpus	-	-	Finite Volume, CD-adapco
TETHYS	Hex	4 hrs	480 cpus	9 hrs	1600 cpus	Finite Volume, PNNL
iRMB-LBM	Hex	4.5 hrs	1 gpu K40c	61.07 hrs	2 gpus K40c	Lattice Boltzmann Tech. Univ. Braunschweig
ISPH	-	0.17 hrs	960 cpus	0.21 hrs	7680 cpus	SPH, SNL

Benchmark: 3D Pore-scale Flow in Bead Pack

Pressure drop along the axial direction

Code	Resolution	ΔP [Pa]	Diff [%]
Reference[4]	-	14.29	-
StarCCM+	40 μm	13.61	4.48
ISPH	40 μm	13.26	4.76
TETHYS	40 μm	13.32	6.79
TETHYS	20 μm	13.19	7.70
iRMB-LBM	40 μm	15.20	6.37
iRMB-LBM	20 μm	16.26	13.79



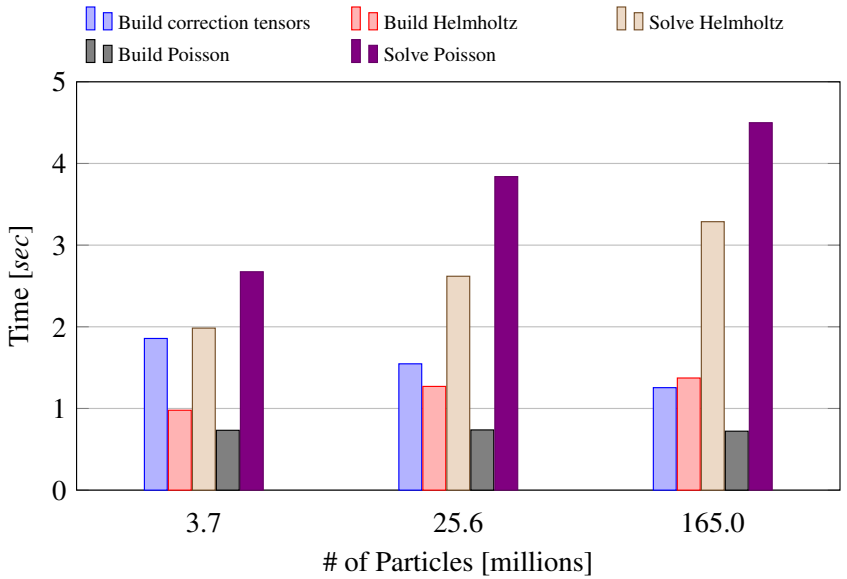
Velocity profile of a vertical cross-section.

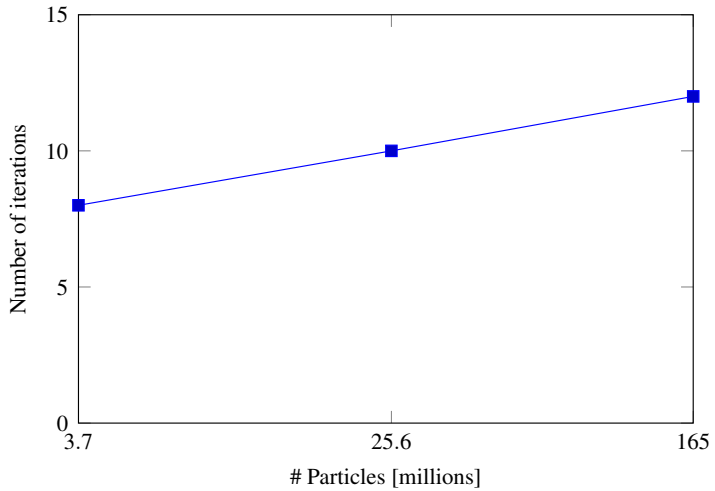
[3] B.Eisfeld and K.Schnitzlein, The influence of confining walls on the pressure drop in packed beds, Chemical Engineering Science, 2001.

Weak scalability of 3D Implicit SPH

≈ 30k Particles per processor

- In theory, AMG convergence factor is independent of the problem size.
- Here, we observe the # of iterations grows moderately with respect to the # of DOFs.





Conclusion

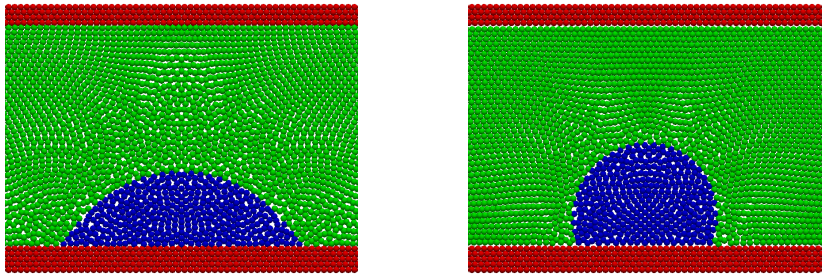
- Demonstrated scalable parallel Implicit SPH method.
- With local correction operators, our ISPH method delivers efficient and accurate solutions that are comparable to other numerical methods.
- Implicit time integration allows to use a large time step.
- Trilinos interface can be applied to problems arising from any particle-based models in LAMMPS.

State of the Code

- Discretizations:
  - implemented second order SPH;
  - implemented MLS with arbitrary order of approximation and ALE scheme.
- Highly scalable parallel code:
  - demonstrated the weak scalability up to 134 million particles with 32k cores;
  - applied the implicit SPH method to solve a real problem which demands highly intensive computation.
- Muti-physics capabilities:
  - provide capability to solve electro-kinetic flows coupled with the Poisson Boltzmann equation.
- Boundary conditions:
  - Morris mirroring technique with Holmes modification for Dirichlet BCs;
  - continuous boundary force method proposed by *Pan et al.* for Robin BCs;
  - partial slip boundary (Robin) condition with no-penetration (Dirichlet) on normal directions;

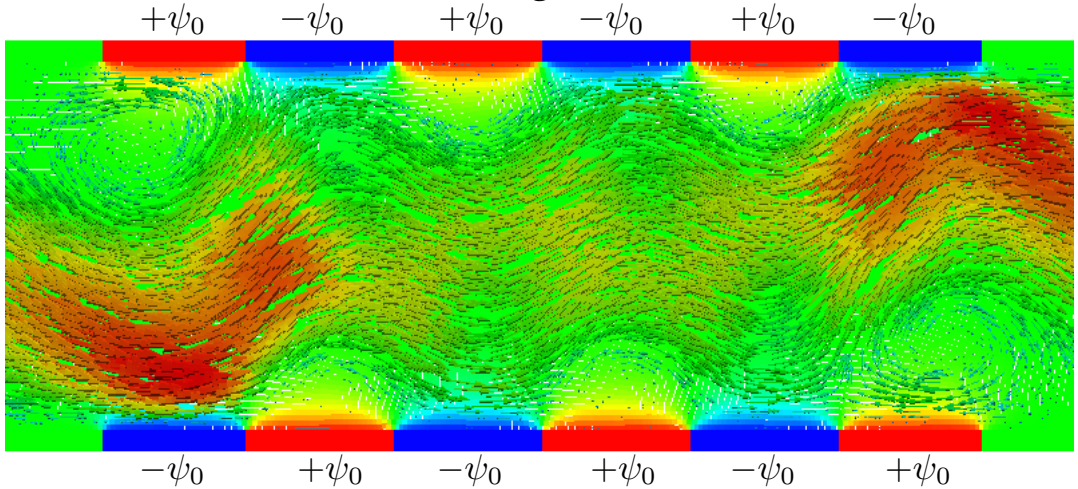
On-going and Future work

- Multi-phase flow: Continuum Surface Force (CSF) and Pairwise Force (PF) model.



θ=*60* θ=*120*

- Muti-physics capabilities: adding improved physics description and coupling strategy to solve electro-kinetic flows *e.g.*, DFT and PNP.



Microfluid mixing channel with alternating electric potentials on walls.

Thank you

Sandia National Laboratories

This code is a researh code and we look for more collaborations for interesting application problems.

Acknowledgement

This work is supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program as part of the Colloboratory on Mathematics for Mesoscopic Modeling of Materials (CM4), under Award Number DE-SC0009247.

This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.