SAND2016-7607C

Exceptional service in the national interest

Sandia National Laboratories

A Massively Parallel Scalable Implicit SPH Solver

SIAM Conference on Nonlinear Waves and Coherent Structures

August 10, 2015

Nathaniel Trask[1], Martin Maxey[1], Kyungjoo Kim[2], Mauro Perego[2], Michael L Parks[2], Kai Yang[3], Jinchao Xu[3], Wenxiao Pan[4], Alex Tartakovsky[4]

[1]Division of Applied Mathematics
Brown University
Providence, RI

[2]Center for Computing Research
Sandia National Laboratories
Albuquerque, NM

[3]Dept. of Mathematics
Penn State University
State College, PA

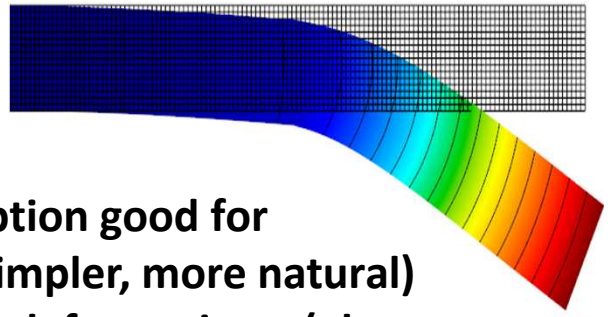
[4]Pacific Northwest
National Laboratory
Richland, WA

U.S. DEPARTMENT OF ENERGY | NNSA National Nuclear Security Administration

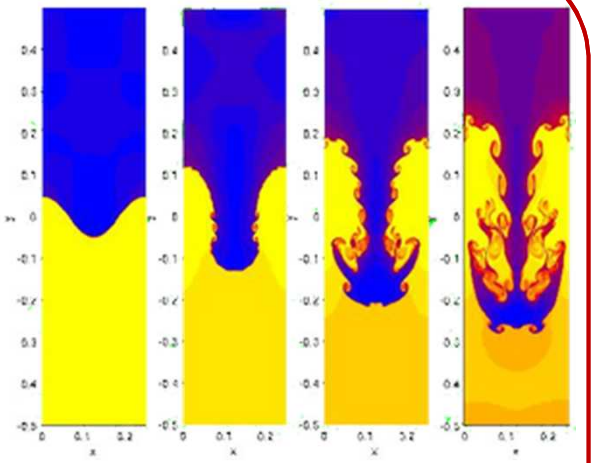
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Mesh-Based Discretizations

- Mesh-based discretizations (finite differences, finite elements, etc.) are the most common discretization method used today.
- Lagrangian description moves with material.
- Eulerian description has material moving through it.

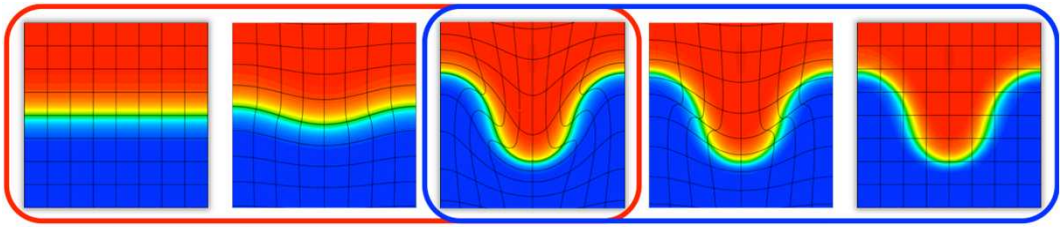


- Lagrangian description good for solid mechanics (simpler, more natural)
- Not good for large deformations (element inversion, etc.)

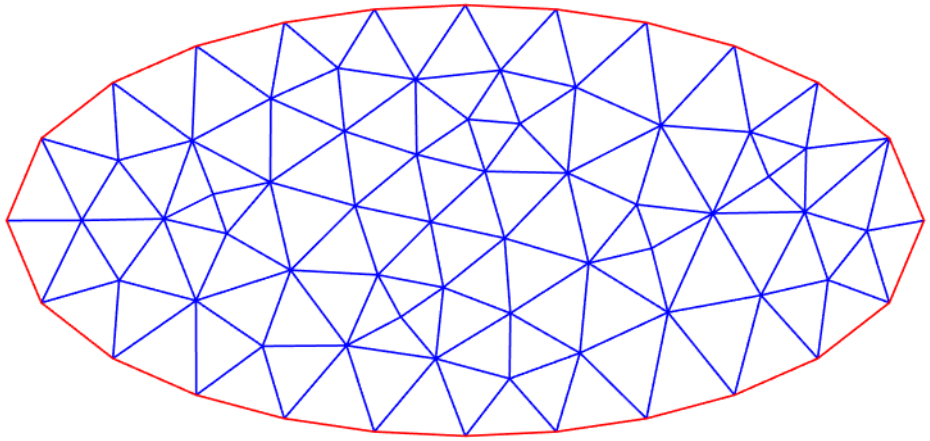


- Eulerian description good for fluid mechanics
- Not good for deformable boundaries, etc.

- Arbitrary Lagrangian-Eulerian (ALE) methods
- Highly complex codes!



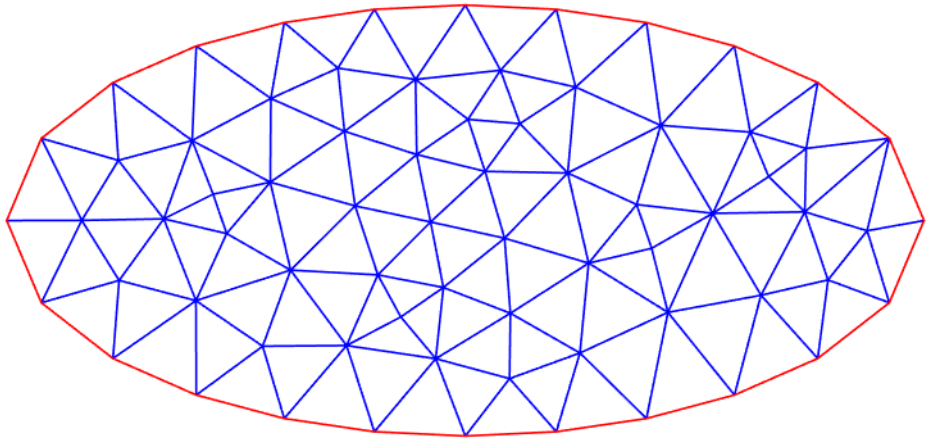
Mesh-Based vs. Meshfree Discretizations



Mesh-Based Discretization

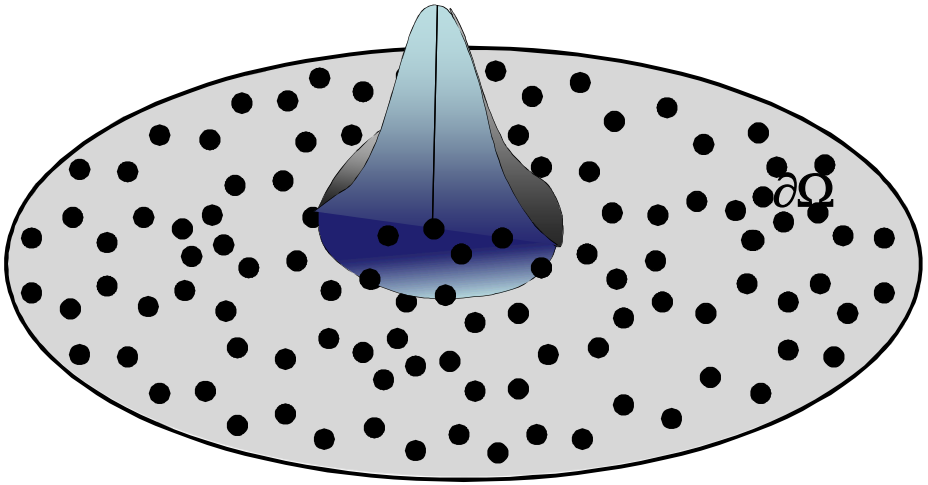
- **Computational domain defined by nodes and elements.**
- **Solution computed using polynomial approximation over each element.**
- **Lagrangian, Eulerian, ALE**

Mesh-Based vs. Meshfree Discretizations



Mesh-Based Discretization

- Computational domain defined by nodes and elements.
- Solution computed using polynomial approximation over each element.
- Lagrangian, Eulerian, ALE



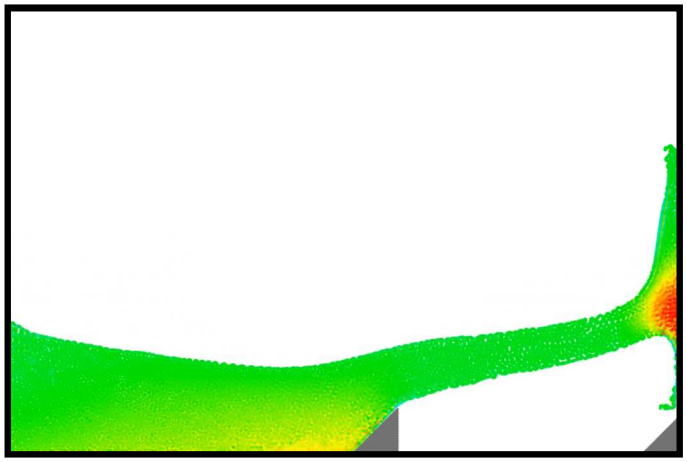
Meshfree Discretization:

- Computational domain defined by point particles (colocation points) and support of associated shape functions.
- Solution represented using shape functions at each point.
- Lagrangian, but no mesh. Suitable for large deformations and fluid flows.

- ❑ Meshfree methods sidestep many of the deficiencies of mesh-based methods, and frequently have less computational overhead.
- ❑ Meshfree methods are not a panacea and have their own shortcomings.

Smoothed Particle Hydrodynamics (SPH)

- SPH is a meshfree particle-based method to solve PDEs in fluid mechanics.
- SPH solves Lagrangian form of Navier–Stokes equations at each particle.
- This discretization gives a set of ODEs for fluid particles trajectories.
- Traditional SPH implementations yield simple-to-code explicit method.



LAMMPS Simulation of
Dam Break Problem

- SPH is useful for:
  - Modeling low Reynolds number flows in microfluidic applications
  - Colloidal suspensions
  - Fluid structure interaction (FSI)
  - No need either to adaptively fit a mesh to a moving boundary or to couple the solid boundary to a fixed Eulerian mesh

- Peridynamics fans: SPH can be shown to be a specific discretization of a peridynamic equation.*

* G.C. Ganzenmüller, S. Hiermaier, M. May, On the similarity of meshless discretizations of Peridynamics and Smooth-Particle Hydrodynamics, Comp. Struct., 150, pp. 71-78, 2015.

Smoothed Particle Hydrodynamics (SPH)

- You've probably seen SPH simulations before.

The Lord of the Rings: The Return of the King
(2003, New Line Cinema)

The Day After Tomorrow
(2004, 20th Century Fox)

- NextLimit Technologies won an Oscar in 2008 for their RealFlow code.

Smoothed Particle Hydrodynamics (SPH)

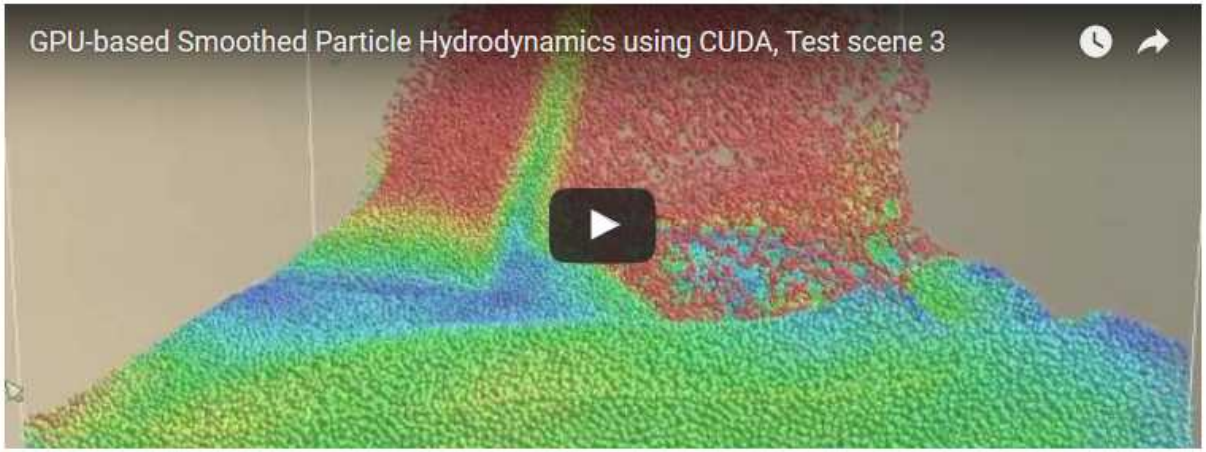
- **From an overview article* on SPH in motion pictures…**

How does SPH work?

Rather than tryi
particles. A math
determine the d

Using the density and pressure of the fluid, SPH ma le to map the force acting on each particle within the fluid. This technique provides resu quite similar to the actual fluid being modelled. And the more particles used in the simulation, the more accurate the model becomes.

"And the more particles used in the simulation, the more accurate the model becomes."



This is well known to be *false*.

* http://theconversation.com/superman-returns-but-whos-looking-after-his-water-680

SPH for Navier-Stokes

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

$$\frac{\mathbf{du}}{\mathbf{dt}} = \frac{\mathbf{1}}{\rho}\nabla \mathbf{p} + \upsilon \nabla^2 \mathbf{u} + \mathbf{g}$$

$$\nabla \cdot \mathbf{u} = \mathbf{0}$$

- Traditional SPH implementations for Navier-Stokes have some problems:
  1) Classical SPH formulations (i.e., weakly compressible SPH or WCSPH) apply an inconsistent artificial compressibility assumption to control the divergence error in the velocity field at the expense of a stiffer CFL condition -- Motivates implicit time integration.)
  2) SPH operator discretizations are inconsistent (diverge with increasing particle density).

- Our objective is to produce a consistent, efficient, 2nd order implicit SPH method.*

- We will achieve this by:
  1) Introduce 2nd order accurate incremental pressure correction scheme
  2) Introducing consistent, 2nd order accurate SPH operator discretizations
- I'll then show a massively parallel software implementation and a few examples….

* N. Trask, M. Maxey, K. Kim, M. Perego, M.L. Parks, K. Yang, and J. Xu, A scalable consistent second-order SPH solver for unsteady low Reynolds number flows, Computer Methods in Applied Mechanics and Engineering, 289, pp. 155-178, 2015.

Navier-Stokes

- Split into predictor/corrector steps:

Predictor (Find velocity estimate u*)

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\frac{1}{\rho}\nabla \mathbf{p}^n + \upsilon \nabla^2 \left(\frac{\mathbf{u}^* - \mathbf{u}^n}{2} \right) + \mathbf{g}$$

$$\mathbf{u}^* = \mathbf{u}_{\partial\Omega}$$

} Helmholtz

Corrector (Make velocity divergence free)

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\nabla\left(\mathbf{p}^{n+1} - \mathbf{p}^n\right)$$

$$\nabla \cdot \mathbf{u}^{n+1} = \mathbf{0}$$

$$\mathbf{u}^{n+1} \cdot n = \mathbf{u}_{\partial\Omega} \cdot n$$

This incremental projection scheme is provably $O(\Delta t^2)$ accurate in velocity & $O(\Delta t)$ in pressure.

- Divergence of corrector equations gives Poisson equation for pressure increment:

$$-\frac{1}{\rho}\nabla^2\left(\mathbf{p}^{n+1} - \mathbf{p}^n\right) = -\frac{\nabla \cdot \mathbf{u}^*}{\Delta t}$$

$$\nabla\left(\mathbf{p}^{n+1} - \mathbf{p}^n\right) \cdot n = \mathbf{0}$$

} Poisson

- Solve resulting systems with algebraic multigrid (AMG) preconditioned GMRES.

SPH Kernel Approximation Operator

- Consider approximations of f(x) with *smoothing function* W(x-y,h):

$$\langle f \rangle(x) = \int_{\Omega} f(y)\, W(x - y, h)\, dy$$

where h is a *smoothing length*.

- W possesses these properties:
  - Normalized: $\int_{\Omega} W(x - y, h)\, dy = 1$
  - Non-negative:
  - Compact support:

$$W(x - y, h) = \begin{cases} > 0 \ \text{for} \ |x - y| \le \kappa h \\ = 0 \ \text{for} \ |x - y| > \kappa h \end{cases}$$

  - Decay: Monotonic decrease with increasing |x-y|
  - "Delta function property": $\lim_{h \to 0} \int_{\Omega} f(y) W(x - y, h)\, dy = \int_{\Omega} f(y) \delta(x - y)\, dy$
  - Even: $W(x - y, h) = W(y - x, h)$
  - "Sufficiently smooth": Must be differentiable

SPH Kernel Approximation Operator

- Consider approximations of f(x) with *smoothing function* W(x-y,h):

$$\langle f \rangle(x) = \int_{\Omega} f(y)\, W(x - y, h)\, dy$$

where h is a *smoothing length*.

- With some derivation, can produce following expression for smoothed gradient:

$$\langle \nabla f \rangle(x) = \int_{\Omega} f(y)\, \nabla_x W(x - y)\, dy$$

- We can compute smoothed gradients of f by pushing derivatives on to W.
- We don't need to know ∇f, or use finite differences, or similar tactics…

- So how to discretize?

Interpolation Errors and Convergence

- Discrete approximation (suppress h from now on):

$$\langle f \rangle(x) \approx \tilde{f}(x_i) = \sum_{j \in \mathrm{supp}(W)} f(x_j)\, W(x_i - x_j)\, V_j$$

where V_j is a quadrature weight for point j.

- Straightforward discrete smoothed gradient not even 0^{th} order accurate:

$$\langle \nabla f \rangle(x) \approx \sum_{j \in \mathrm{supp}(W)} f(x_j)\, \nabla_x W(x_i - x_j)\, V_j$$

- Standard SPH discretizations (0^{th} order consistent):

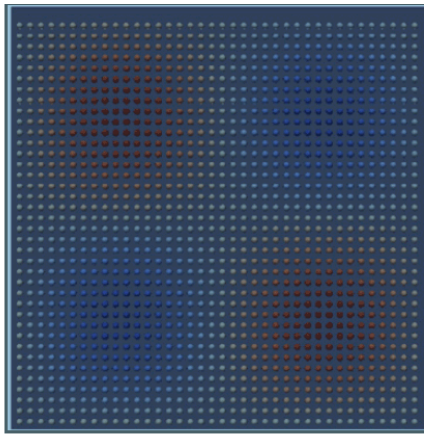
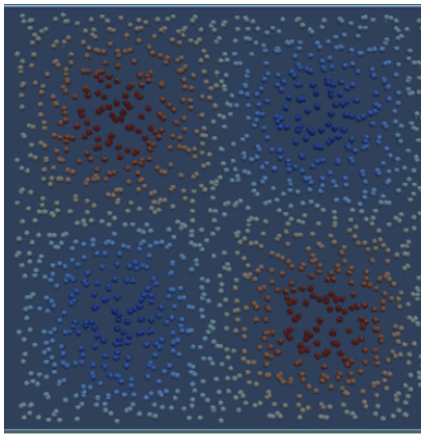
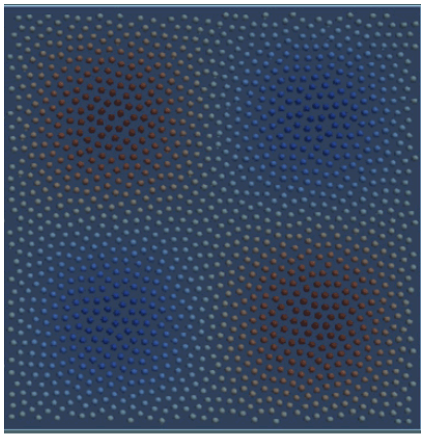
$$\langle \nabla f \rangle(x) \approx \tilde{\nabla}_0 f(x_i) = \sum_{j \in \mathrm{supp}(W)} \big(f(x_j) - f(x_i)\big) \nabla_x W(x_i - x_j)\, V_j$$

$$\langle \nabla^2 f \rangle(x) \approx \tilde{\nabla}_0^2 f(x_i) = 2 \sum_{j \in \mathrm{supp}(W)} \frac{f(x_j) - f(x_i)}{\|x_i - x_j\|} \frac{x_i - x_j}{\|x_i - x_j\|} \cdot \nabla_x W(x_i - x_j)\, V_j$$

- These standard discretizations lack higher order accuracy….

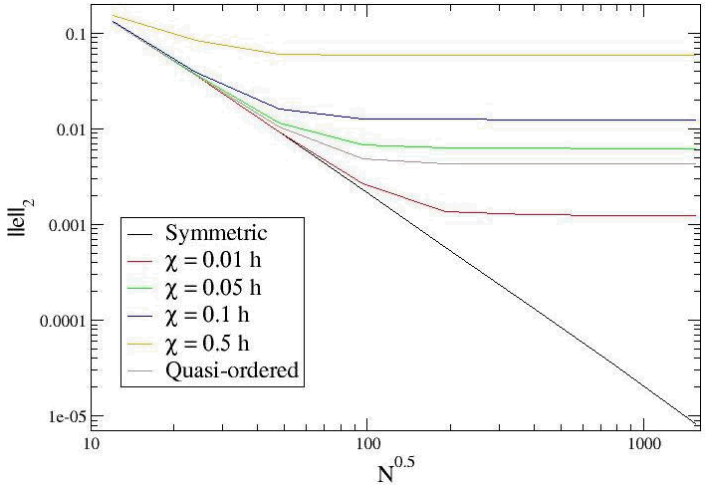
Interpolation Errors and Convergence

- Consider 3 particle distributions: Cartesian, randomly perturbed, quasi-ordered

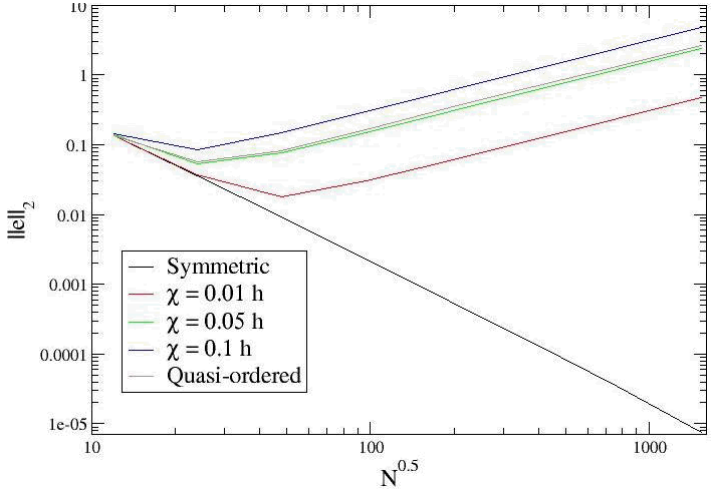




- Discrete SPH operators convergent only on perfectly cartesian grid!

$\tilde{\nabla}_0 f(x_i)$



$\tilde{\nabla}_0^2 f(x_i)$



Interpolation Errors and Convergence

- Interpolation errors of differential operators can be categorized as errors due to smoothing, quadrature, and particle anisotropy:

$$\left\| \mathbf{e}_{\text{interp}} \right\| \leq \left\| \mathbf{e}_{\text{smoothing}} \right\| + \left\| \mathbf{e}_{\text{quad}} \right\| + \left\| \mathbf{e}_{\text{anis}} \right\|$$

- There are many possible discretizations for the smoothed gradients. In general,

$$\left\| \mathbf{e}_{\text{smoothing}} \right\| \sim Ch^2 \qquad \left\| \mathbf{e}_{\text{quad}} \right\| \sim C\left(\frac{h}{\Delta x} \right)^{-\beta_1} \qquad \left\| \mathbf{e}_{\text{anis}} \right\| \sim C\frac{\chi}{h^p}\left(\frac{h}{\Delta x} \right)^{-\beta_2}$$

  where $C>0$, $\beta_1, \beta_2 \in Z^+$, $\Delta x \sim 1/N$, $p=1$ for gradient, $p=2$ for Laplacian.

- For convergence must simultaneously decrease h and increase $h/\Delta x$ (i.e. increase the number of neighbors interacting with each particle).

- If $h/\Delta x$ held constant (standard practice), error due to particle anisotropy dominates. What to do?

Interpolation Errors and Convergence

- Define corrected operators

$$\tilde{\nabla}_1 f(x_i) = \sum_{j \in supp(W)} \left(f(x_j) - f(x_i) \right) G_i \nabla_x W(x_i - x_j) V_j$$

$$\tilde{\nabla}_1^2 f(x_i) = 2 \sum_{j \in supp(W)} L_i : \frac{x_i - x_j}{\|x_i - x_j\|} \otimes \nabla_x W(x_i - x_j) \left(\frac{f(x_j) - f(x_i)}{\|x_i - x_j\|} - \frac{x_i - x_j}{\|x_i - x_j\|} \cdot \tilde{\nabla}_1 f(x_i) \right) V_j$$

where L_i, G_i are 3x3 tensors (in 3D), and can be determined from a Taylor expansion.

- L, G can be computed directly and stored for each particle at a cost of inverting and storing G, L, incurring O(N) total cost in additional computational complexity and storage.
- A simple expression exists to determine G:

$$\left(G_i^{-1} \right)^{mn} = \sum_{j \in supp(W)} \left(x_j^m - x_j^m \right) \frac{\partial}{\partial x_j^n} W(x_i - x_j) V_j$$

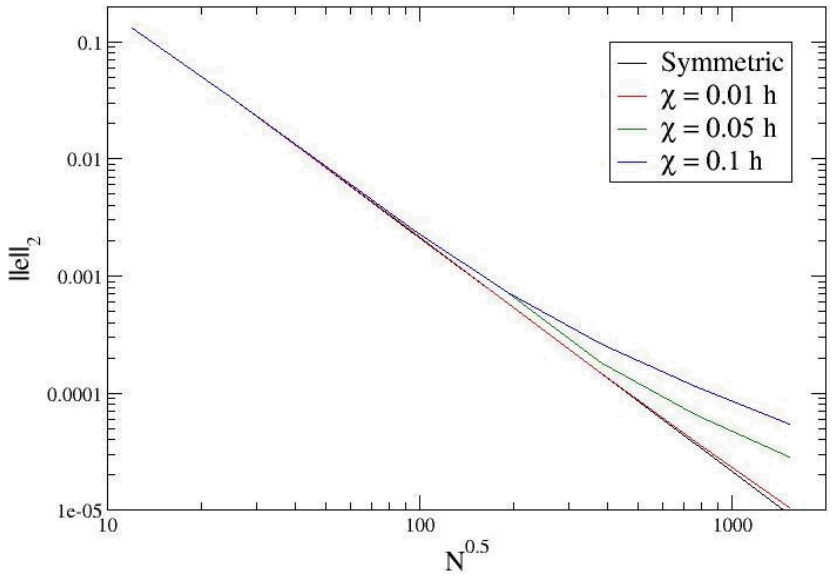
where m,n = 1,2,3.

- There is a similar (more tedious) expression for L.

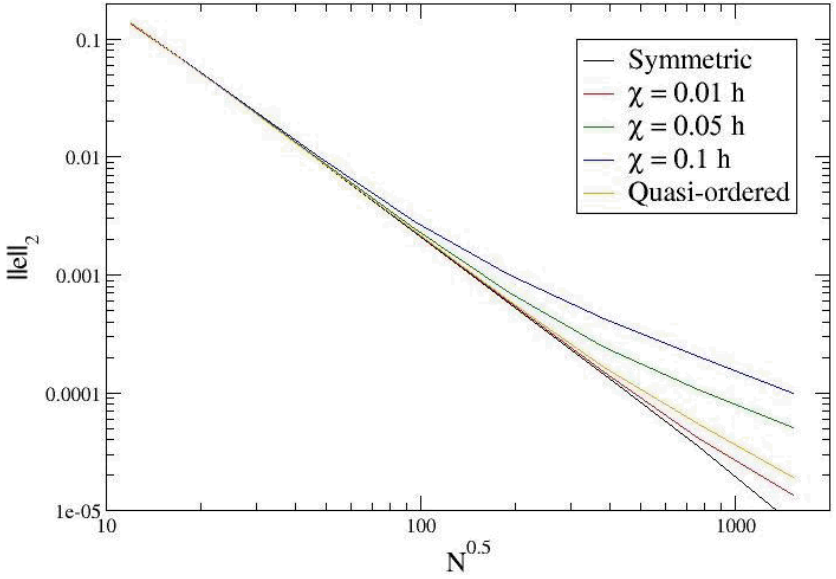
Interpolation Errors and Convergence

- The modified operators remove dependence on h/Δx (number of neighbors per particle) in the error terms e_{quad} and e_{anis}.
- This means that we can achieve $O(h^2)$ accuracy with fixed h/Δx!

$\tilde{\nabla}_1 f(x_i)$



$\tilde{\nabla}_1^2 f(x_i)$

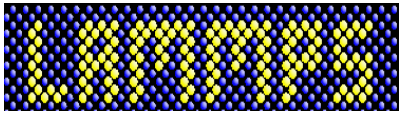


- Further (and of even more importance) the corrected operators achieve a similar accuracy with fewer neighbors (for example 20 neighbors in 2D vs. 80 neighbors).
- This gives sparser global operators (Poisson & Helmholtz) for Navier-Stokes, representing a substantial computational savings.

Massively Parallel 3D Implicit SPH

Goal: Large scale parallel 3D implicit simulation capability

- ❑ Use LAMMPS, Sandia's massively parallel molecular dynamics code
- ❑ LAMMPS can simulate any particle system (MD, SPH, DPD, PD, etc.)
- ❑ Demonstrated massively parallel scalability



lammps.sandia.gov

Problem:

- ❑ LAMMPS has no capability for implicit time integration!
- ❑ LAMMPS handles particle systems easily
- ❑ However, only explicit time integration used in MD
- ❑ Need distributed memory matrices, linear solvers, preconditioners, etc.
- ❑ Unwise/unwieldy to implement directly into LAMMPS

Solution:

- ❑ Integrate LAMMPS with Trilinos solvers!
- ❑ This has never been done before….

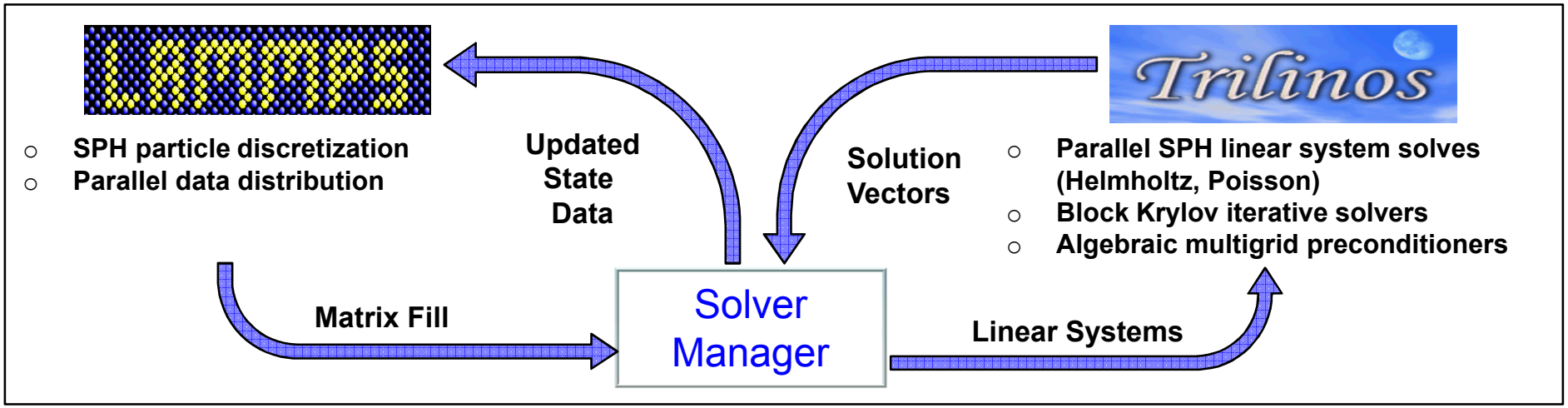
trilinos.org

Trilinos

- Object-oriented software framework for solving large complex science & engineering problems
- Trilinos is made of packages
- Not a monolithic piece of software; Use the set of packages you choose
- Each package developed by domain experts
- Like LEGO™ bricks, not Matlab™

	Objective	Package(s)
Discretizations	Meshing & Discretizations	STKMesh, Intrepid, Pamgen, Sundance, ITAPS, Mesquite
	Time Integration	Rythmos
Methods	Automatic Differentiation	Sacado
	Mortar Methods	Moertel
Services	Linear algebra objects	Epetra, Tpetra, Kokkos
	Interfaces	Thyra, Stratimikos, RTOp, FEI, Shards, Tpetra::RTI
	Load Balancing	Zoltan, Isorropia
	"Skins"	PyTrilinos, WebTrilinos, ForTrilinos, Ctrilinos, Optika
	C++ utilities, I/O, thread API	Teuchos, EpetraExt, Kokkos, Triutils, ThreadPool, Phalanx
Solvers	Iterative linear solvers	AztecOO, Belos, Komplex
	Direct sparse linear solvers	Amesos, Amesos2
	Direct dense linear solvers	Epetra, Teuchos, Pliris
	Iterative eigenvalue solvers	Anasazi
	ILU-type preconditioners	AztecOO, IFPACK, Ifpack2
	Multilevel preconditioners	ML, CLAPS
	Block preconditioners	Meros, Teko
	Nonlinear system solvers	NOX, LOCA
	Optimization (SAND)	MOOCHO, Aristos, TriKota, Globipack, Optipack
	Stochastic PDEs	Stokhos

LAMMPS/Trilinos Integration



Fuse LAMMPS and Trilinos for massively parallel 3D Implicit SPH computational framework

- 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.

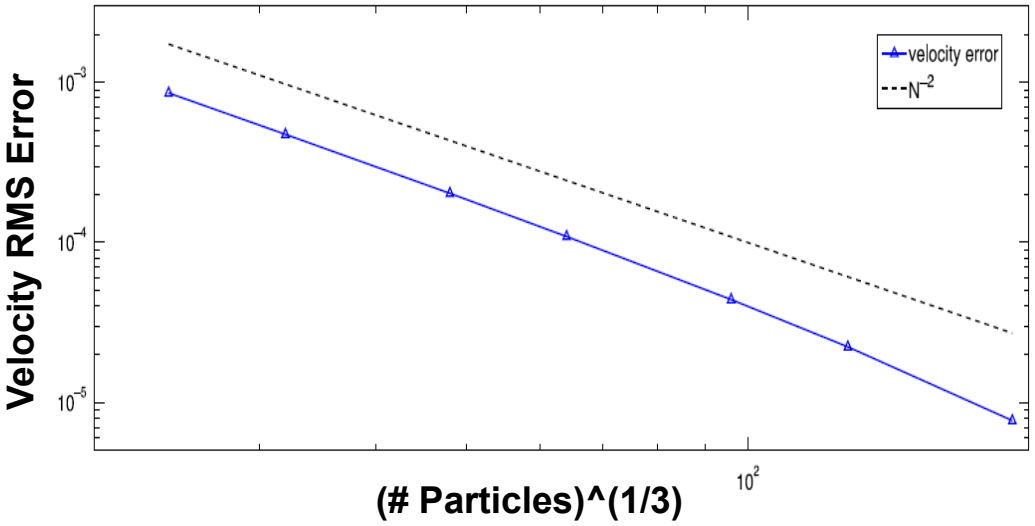
- LAMMPS "tag" vector is logically equivalent to an Epetra/Tpetra map
- LAMMPS "neighborlists" determine matrix sparsity pattern
- LAMMPS constructs and fills sparse matrices, right-hand-sides
- LAMMPS requests linear solver, preconditioner; leverages full Trilinos solver stack

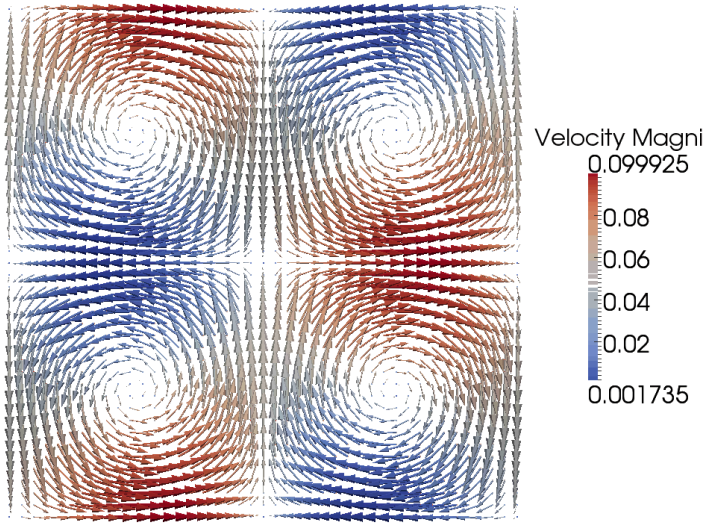
Taylor-Green Vortex

Verification Problem (2D, Extruded to 3D)

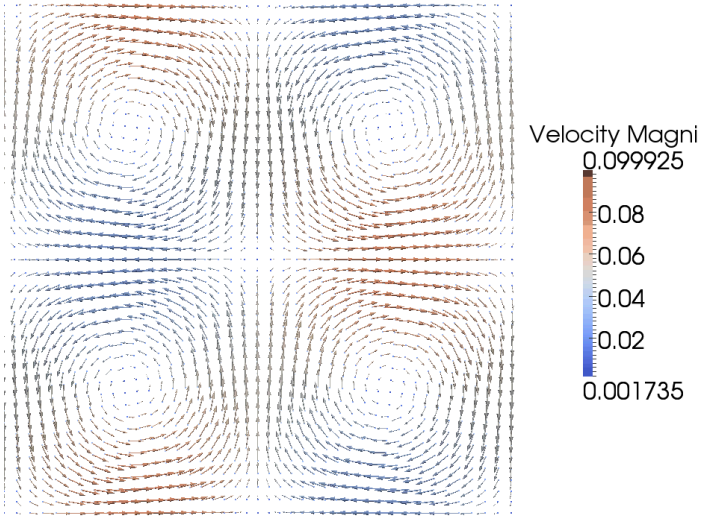
$$\frac{d\mathbf{u}}{dt} + \frac{1}{\rho}\nabla p - \frac{\eta}{\rho}\Delta\mathbf{u} = \mathbf{f}, \quad \text{in } \Omega$$
$$\nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega$$
$$\mathbf{u} = 0, \quad \text{on } \partial\Omega$$

$$u_x = U_0 e^{-2\nu\pi^2 t}\sin(\pi x)\cos(\pi y)$$
$$u_y = -U_0 e^{-2\nu\pi^2 t}\cos(\pi x)\sin(\pi y)$$
$$p = \frac{U_0^2}{4}e^{-4\nu\pi^2 t}(\cos(2\pi x) + \cos(2\pi y) + 2)$$





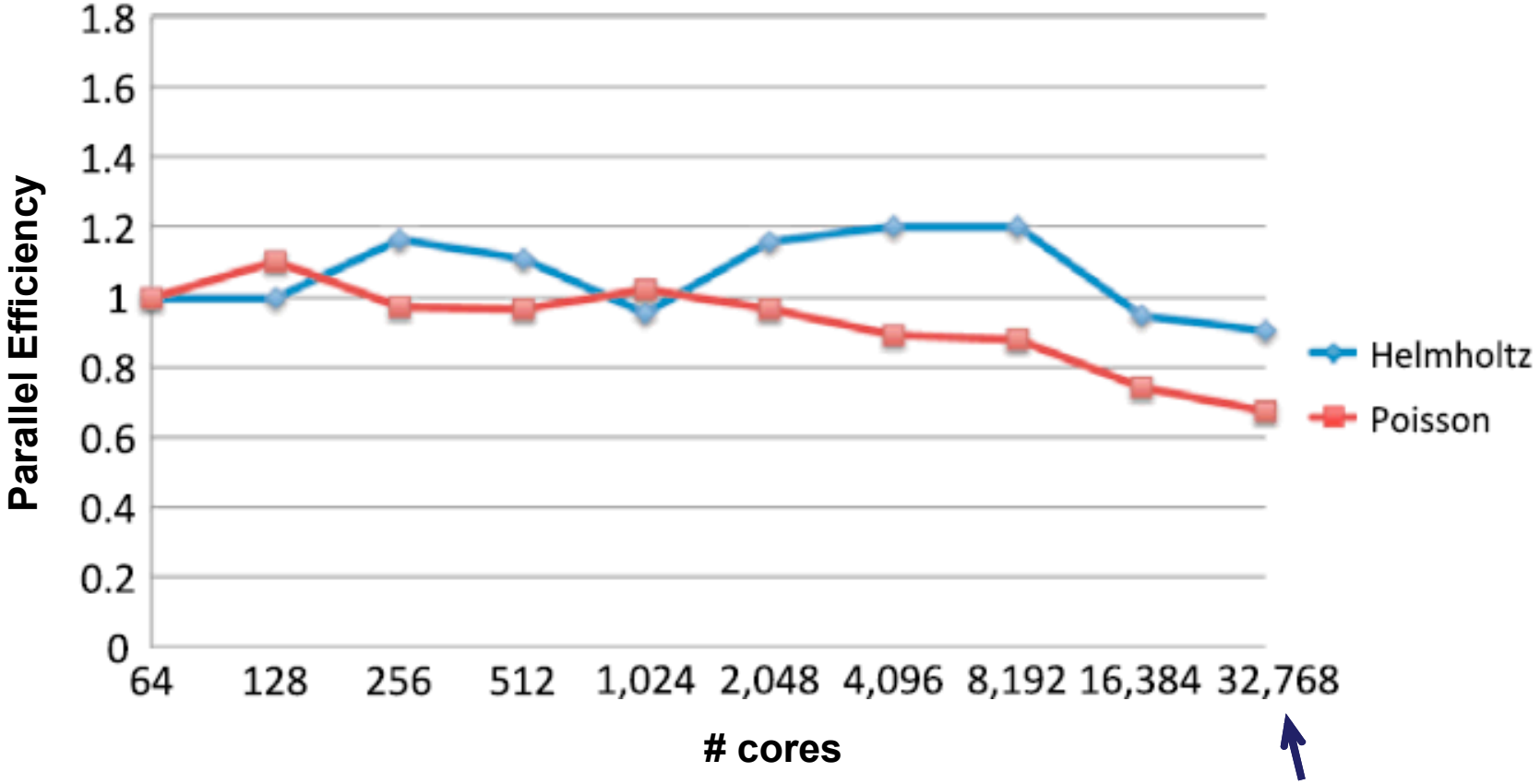
t = 0.000



t = 0.375

Parallel Scalability

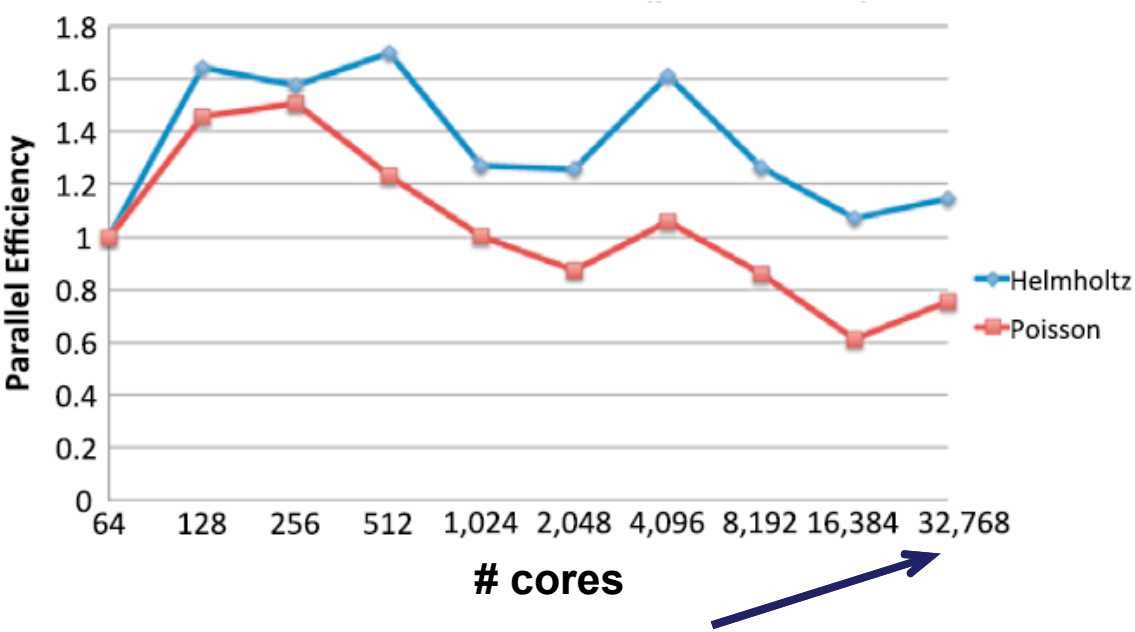
- Weak scaling results (Simulations on Hopper @ NERSC, Cray XE6)
- h = 1.5 Δx
- 4096 particles/core
- Smoothed aggregation AMG



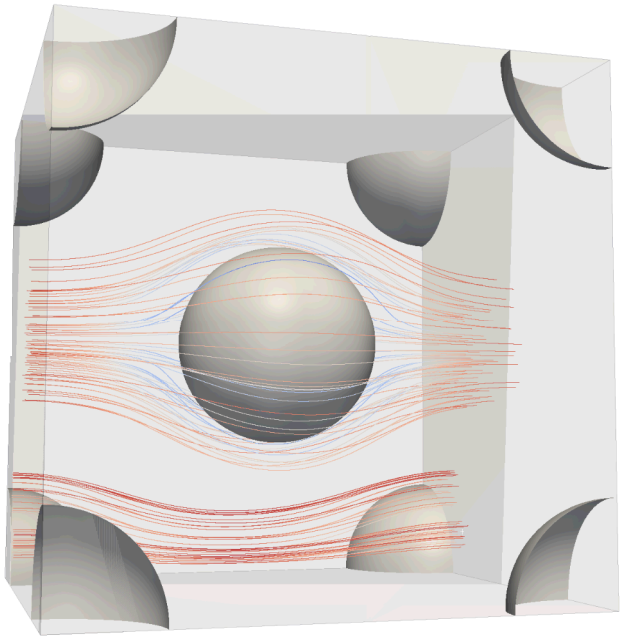
~133M Particles!

Flow Past Periodic 3D Spheres

- Flow through complex geometry
- 3D periodic array of spheres in BCC lattice driven by external force
- h = 0.8 Δx; 4096 particles/core; Smoothed aggregation AMG



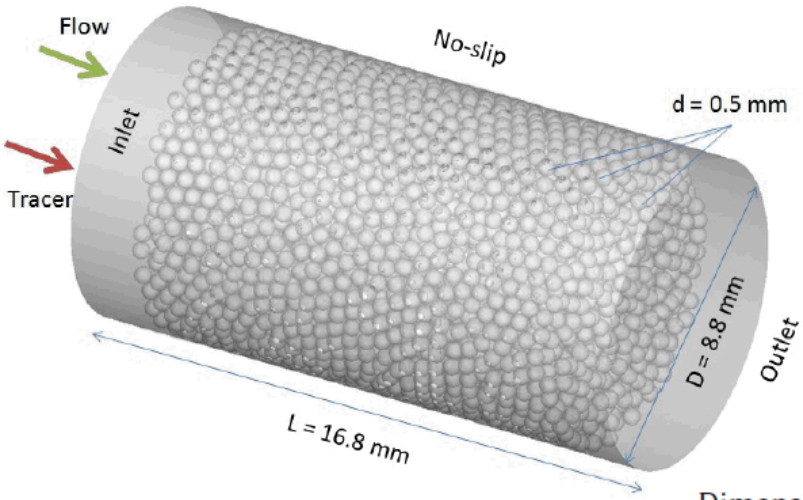
~133M Particles!



Selected particle streamlines (color indicates velocity magnitude)

Pore Scale Flow in Bead Pack*

- ❑ Simulate porous media flow at pore scale
- ❑ High-resolution magnetic resonance imaging of 6864 packed polystyrene beads used to construct pore geometry



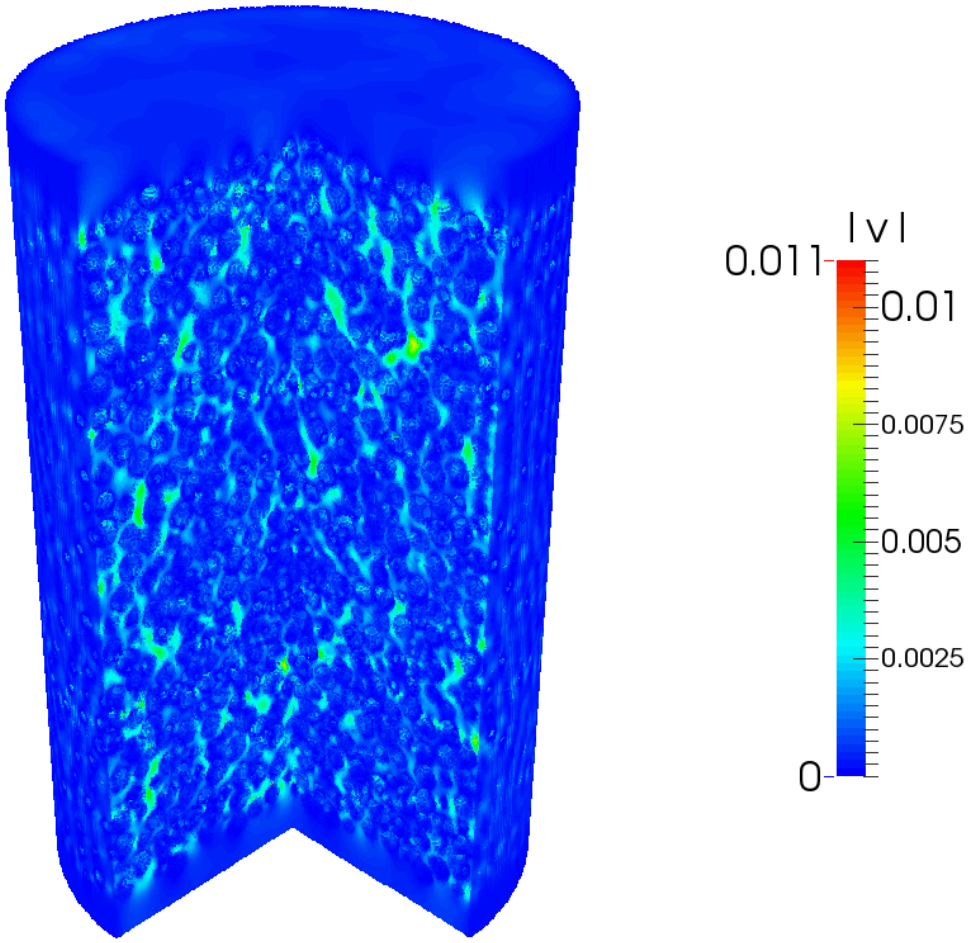
Dimensions and parameters of the column.

Parameter	Symbol	(units)	Value
Length of the simulated domain	L_c	(mm)	16.8
Length of the sphere pack	L_b	(mm)	12.8
Diameter of the column	D	(mm)	8.8
Diameter of the beads	d_p	(mm)	0.5
Porosity	ϵ		0.4267
Volumetric flow rate	Q	(kg/s)	2.771×10^{-5}
Fluid density	ρ	(kg/m^3)	997.561
Fluid dynamic viscosity	μ	(Pa s)	8.887×10^{-4}

* X. Yang, Y. Mehmani, W.A. Perkins, A. Pasquali, M. Schönherr, K. Kim, M. Perego, M.L. Parks, N. Trask, M.T. Balhoff, M.C. Richmond, M. Geier, M. Krafczyk, L-S. Luo, A.M. Tartakovsky, and T.D. Scheibe, Intercomparison of 3D Pore-scale Flow and Solute Transport Simulation Methods, Advances in Water Resources, In Press, 2015.

Pore Scale Flow in Bead Pack*

- 165M particles; $\Delta x = 20\ \mu m$
- Simulations on Edison @ NERSC (Cray XC30)
- 7680 cores; 13 minutes wall-clock run time



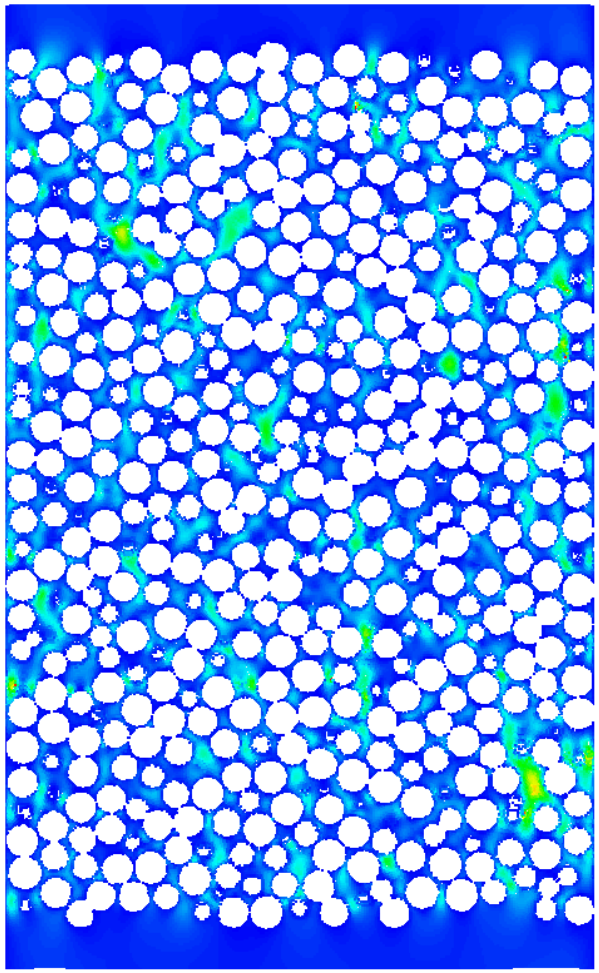
Steady state simulated result (cut view)
(color indicates velocity magnitude)

* X. Yang, Y. Mehmani, W.A. Perkins, A. Pasquali, M. Schönherr, K. Kim, M. Perego, M.L. Parks, N. Trask, M.T. Balhoff, M.C. Richmond, M. Geier, M. Krafczyk, L-S. Luo, A.M. Tartakovsky, and T.D. Scheibe, Intercomparison of 3D Pore-scale Flow and Solute Transport Simulation Methods, Advances in Water Resources, In Press, 2015.

Pore Scale Flow in Bead Pack*

- Compare multiple codes (including this one) on same problem.
- QoI: Pressure drop along axial direction

Code	Resolution	ΔP [Pa]	Diff [%]
Reference	-	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



Steady state simulated result (cross section)
(color indicates velocity magnitude)

* X. Yang, Y. Mehmani, W.A. Perkins, A. Pasquali, M. Schönherr, K. Kim, M. Perego, M.L. Parks, N. Trask, M.T. Balhoff, M.C. Richmond, M. Geier, M. Krafczyk, L-S. Luo, A.M. Tartakovsky, and T.D. Scheibe, Intercomparison of 3D Pore-scale Flow and Solute Transport Simulation Methods, Advances in Water Resources, In Press, 2015.

Summary

- Review of mesh-based and meshless methods
- SPH review
- Developed scalable, efficient, parallel implicit 2nd order accurate SPH method
  - Introduced consistent, 2nd order accurate SPH operator discretizations
  - Introduced 2nd order accurate incremental pressure correction scheme
- Massively parallel implementation (LAMMPS+Trilinos)
- Demonstrated weak scaling on largest ever implicit SPH simulations
- Applications:
  - Verification: Taylor-Green Vortex
  - Flow past periodic 3D spheres
  - Pore scale flow in bead pack (using MRV data from physical experiment)

- Acknowledgements: This work is supported by the Applied Mathematics Program within the Department of Energy (DOE) Office of Advanced Scientific Computing Research (ASCR) as part of the Collaboratory on Mathematics for Mesoscopic Modeling of Materials (CM4).

Parallel Scalability

- Weak scaling results (Simulations on Hopper @ NERSC, Cray XE6)
- h = 1.5 Δx
- 4096 particles/core
- Smoothed aggregation AMG

~133M Particles!