

# Compadre Toolkit

presented by Paul Kuberry

Sandia National Laboratories,  
Albuquerque, NM, USA



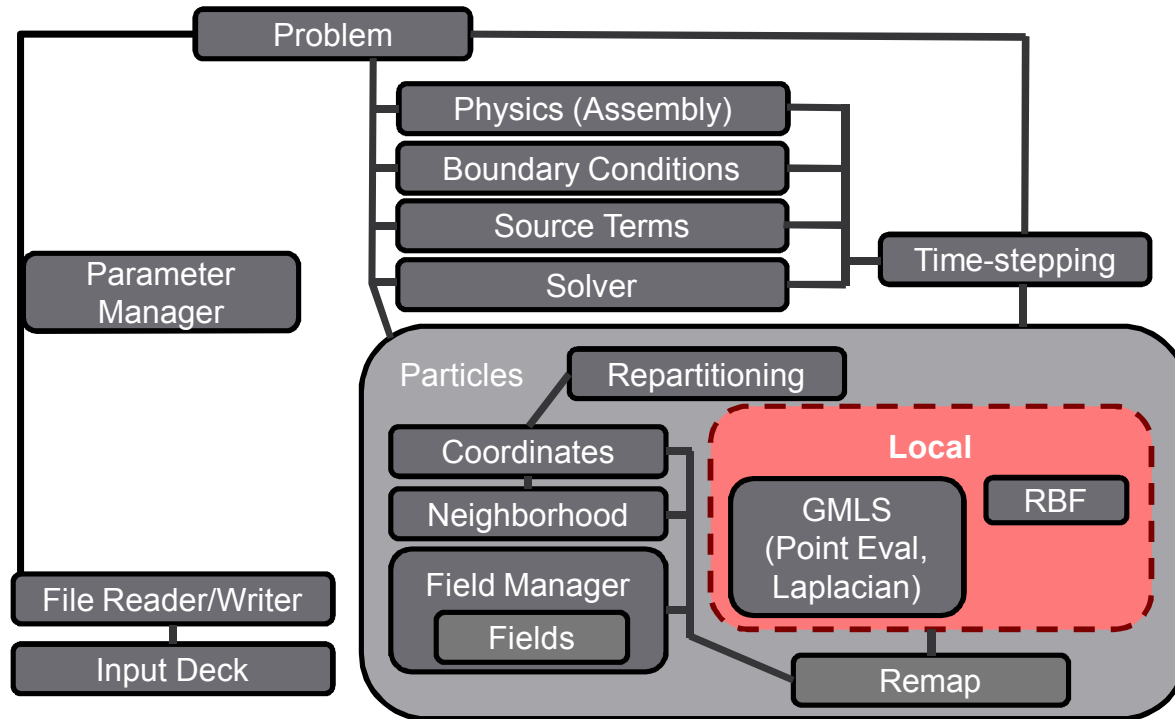
UCSD Meeting  
11/13-11/15



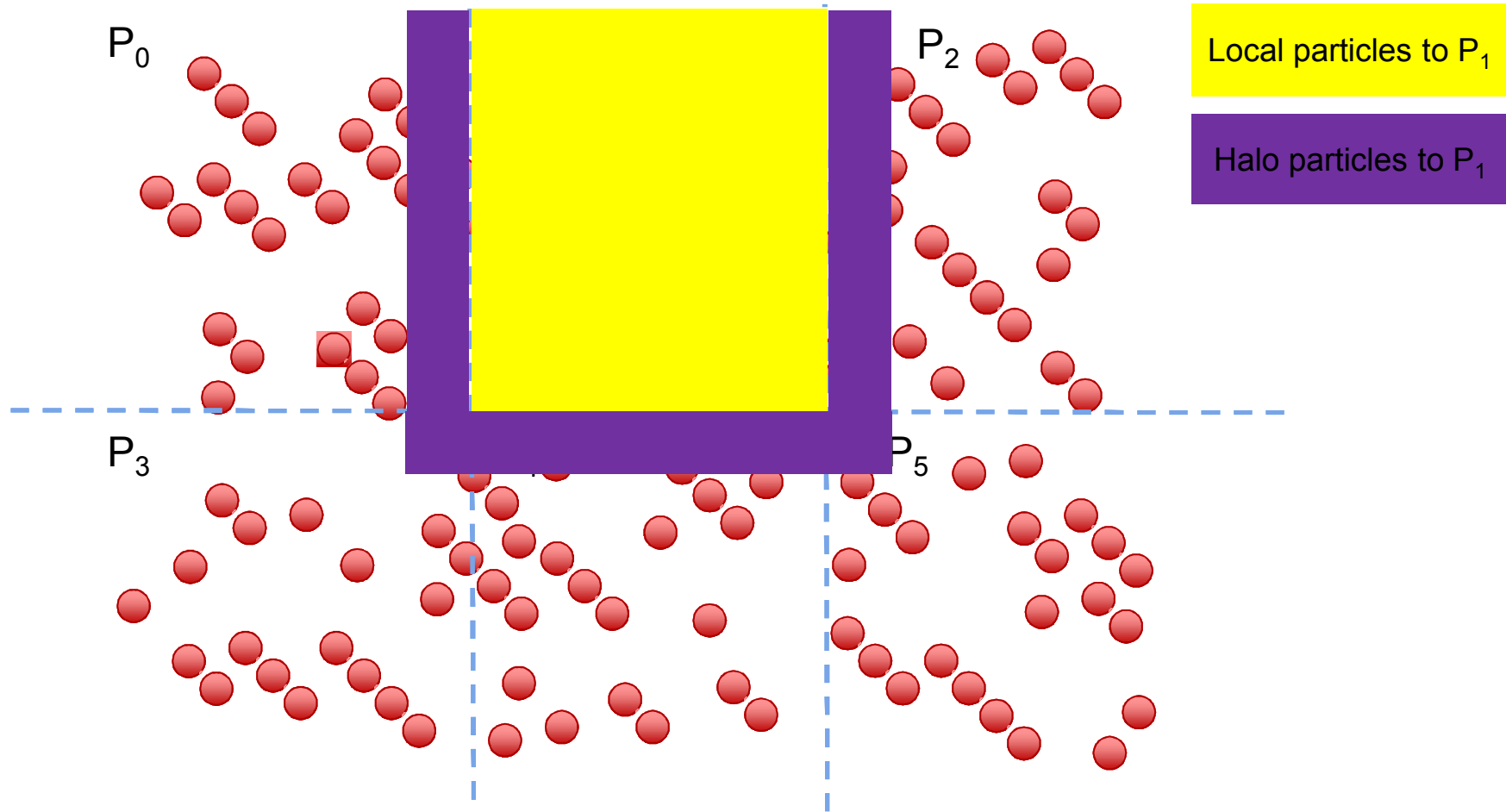
# Compadre Toolkit

- **Supports**
  - Euclidean & Spherical coordinate systems
  - Eulerian & Lagrangian simulations
  - Time-dependent simulations
  - Strong-form PDE solution (Laplacian, Stokes, Elastodynamics)
  - Neighborhood searches (k-d tree)
  - Block physics with block preconditioning (AMG, ILU, Jacobi, G-S, etc...)
- **Managing**
  - Registering fields of various dimensions ( pressure, velocity, etc.. )
  - Sets of particles
    - Data transfer between sets (remap)
    - Partitioning/repartitioning particle sets over multiple processors (Zoltan2)
- **Leverages Trilinos Tools**
  - Trilinos/Zoltan2 particles over processor partitioning
  - Trilinos solvers (Amesos2, Ifpack2, MueLu, Belos, Teko, Thyra, Stratimikos)
- Combines MPI parallelism with thread-parallelism (MPI + X) using Trilinos, Tpetra, and Kokkos
- Manages importing/exporting data in the form of VTK and Netcdf files (in parallel)
- Input deck driven simulation (XML/YAML)

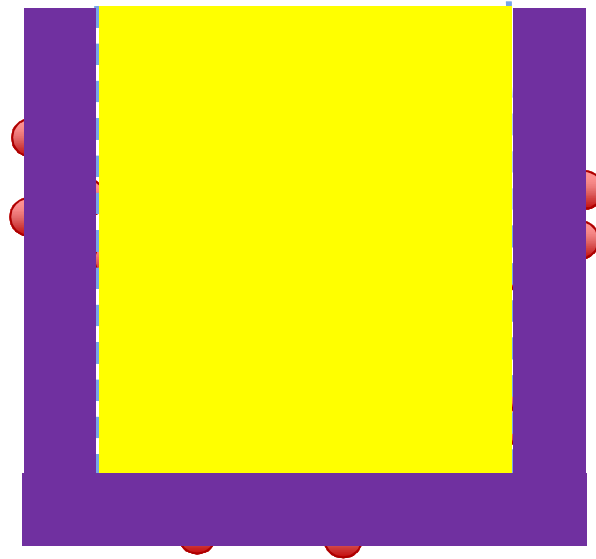
# Compadre Toolkit



# Partitioning over distributed memory



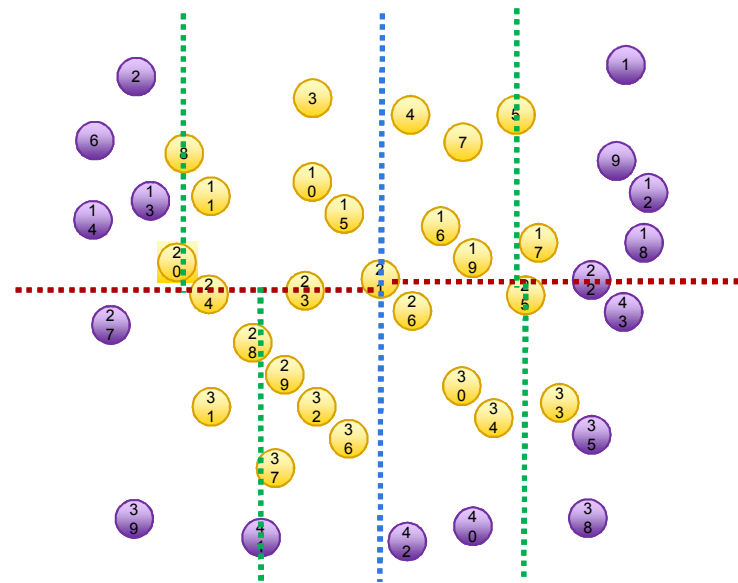
# Neighbor searches over local data



Local particles to  $P_1$

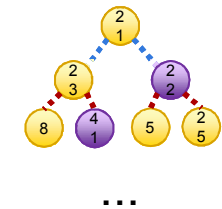
Halo particles to  $P_1$

# Neighbor searches over local data



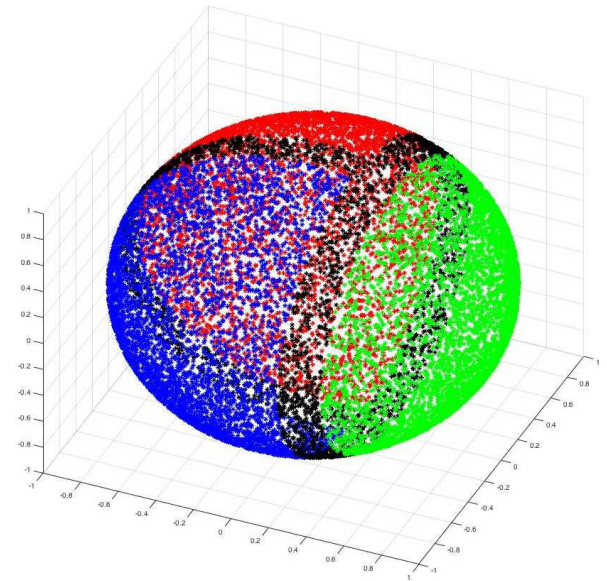
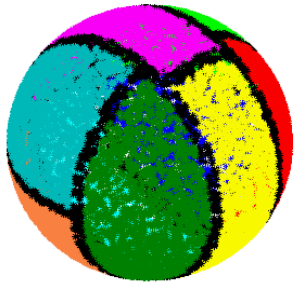
Local particles to  $P_1$

Halo particles to  $P_1$



k-d tree constructed and traversed using nanoflann to perform radius searches.

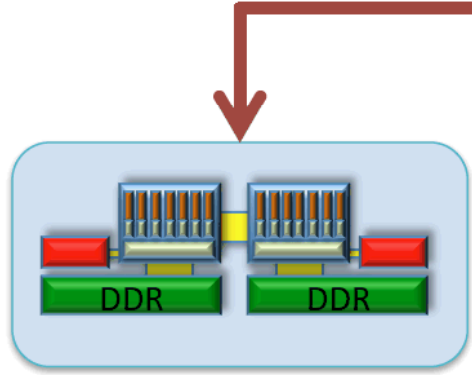
# Dynamic (re)partitioning via Zoltan2 in Trilinos



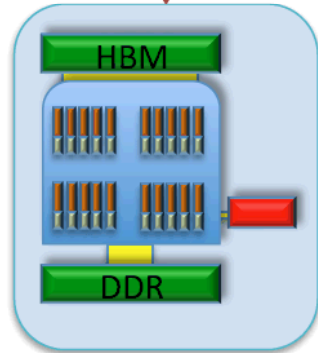
Supports various schemes for partitioning such as:

- Multi-Jagged Coordinate Partitioning
- Recursive Coordinate Bisection
- Block partitioning (simple weighted identifiers)

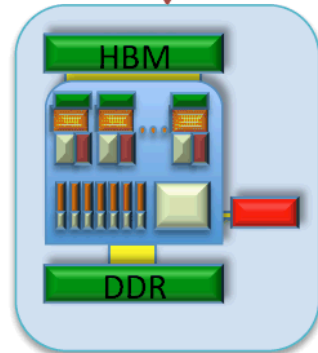
# Kokkos: Performance, Portability and Productivity



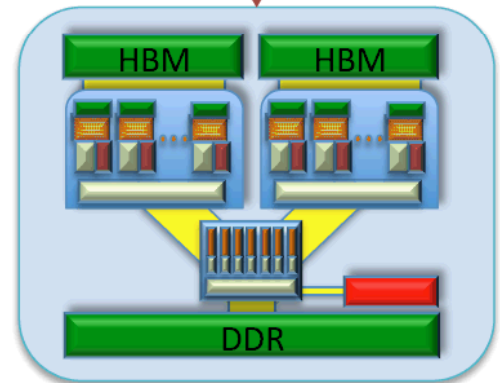
**Multi-Core**



**Many-Core**



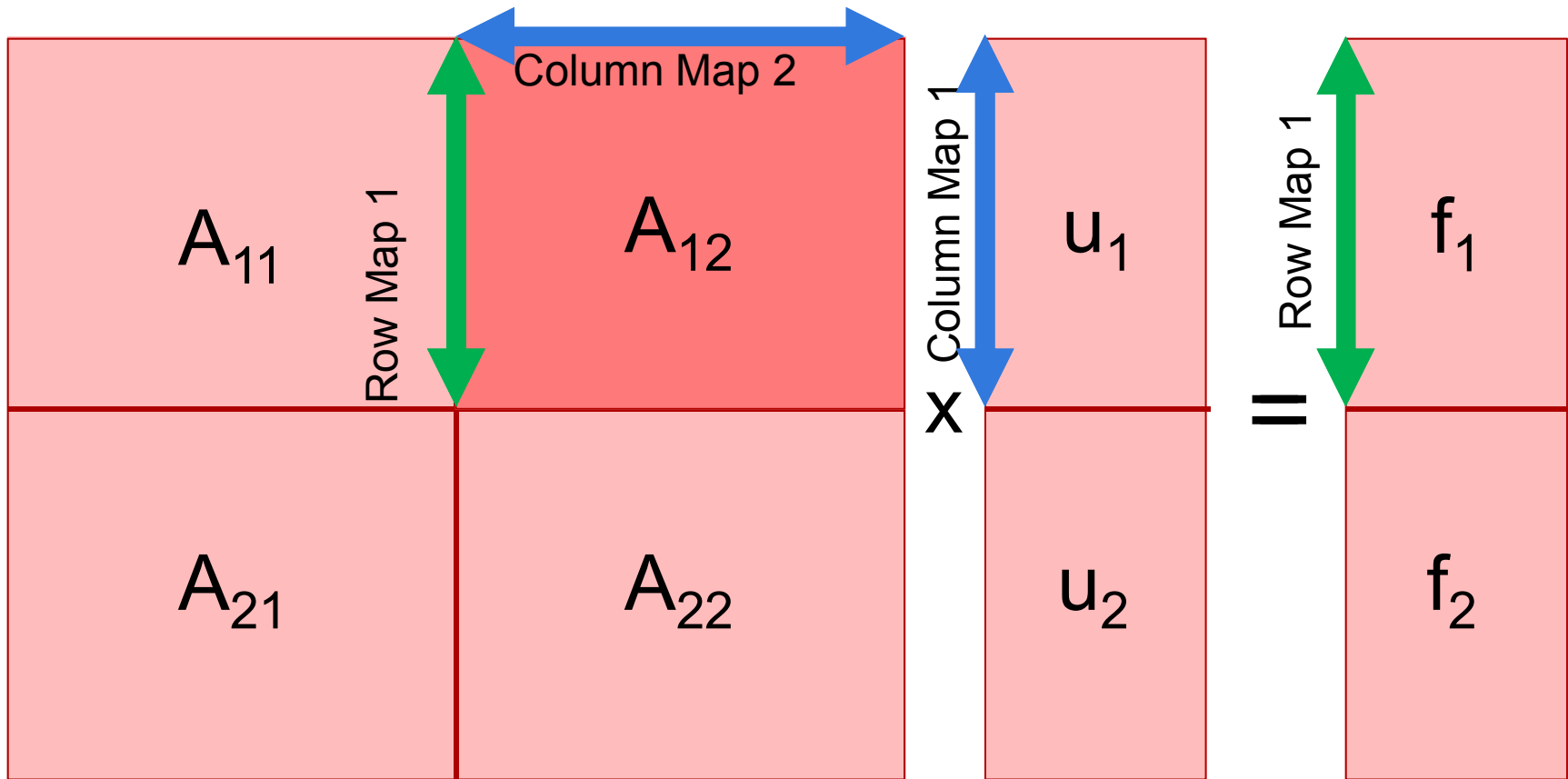
**APU**



**CPU + GPU**

<https://github.com/kokkos>

# Block Solver

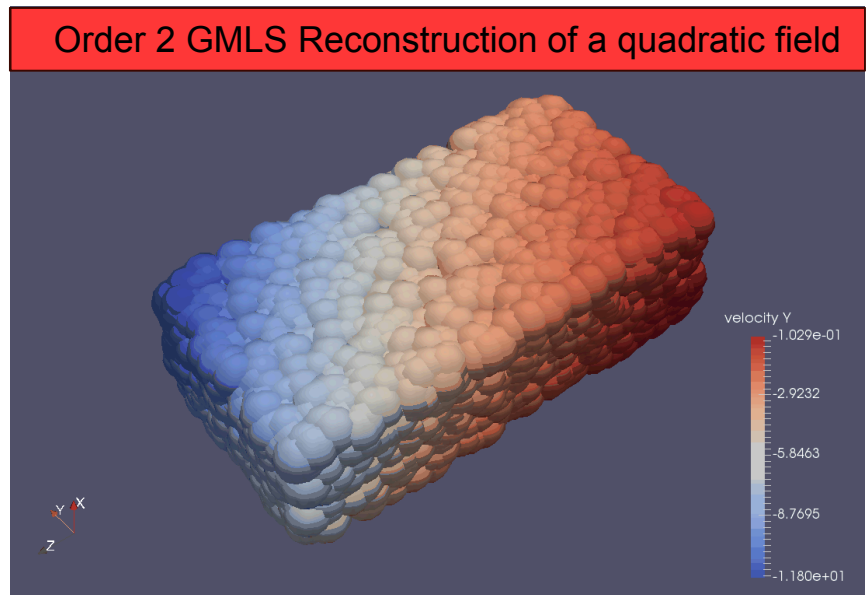
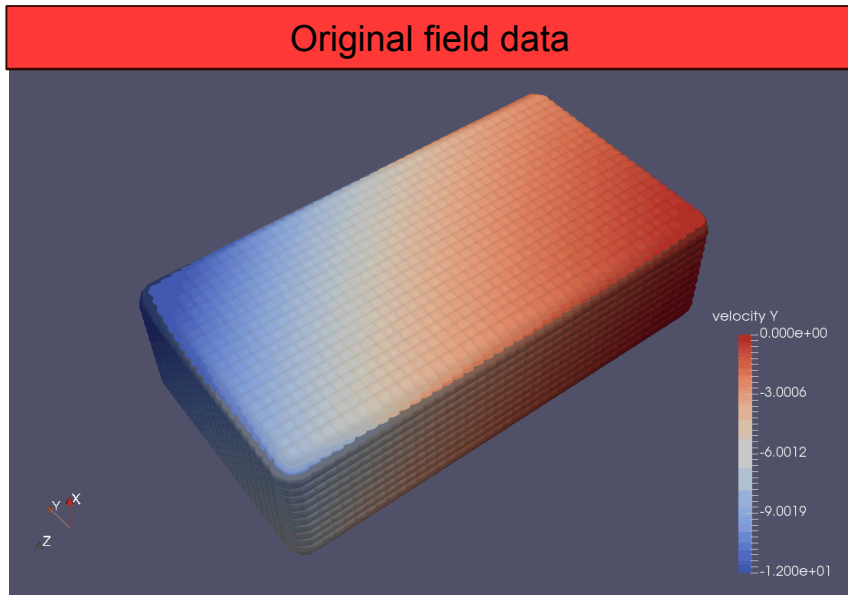


Choice between constructing A all at once,  
or  $A_{ij}$  block by block.

Algebraic multigrid preconditioners, Jacobi, Gauss-Seidel, CG,  
GMLRES, etc... available through MueLu, Ifpack2, Teko, Belos, Amesos2

# Particle Insertion/Removal With Remap

Insertion requiring interpolation from existing field data using GMLS



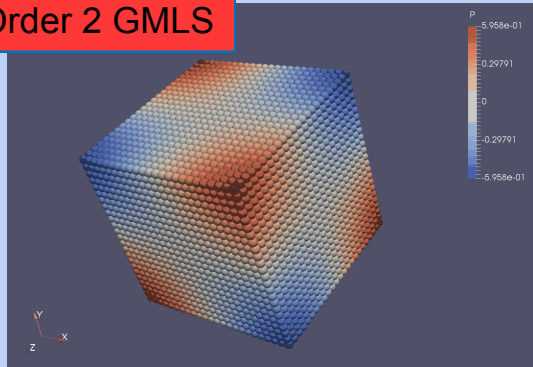
Error < 1e-16

# Poisson Solver $(-\Delta u = f)$

Manufactured Solution (Convergence):

$$\sin(x) \cdot \sin(y) \cdot \sin(z)$$

Order 2 GMLS

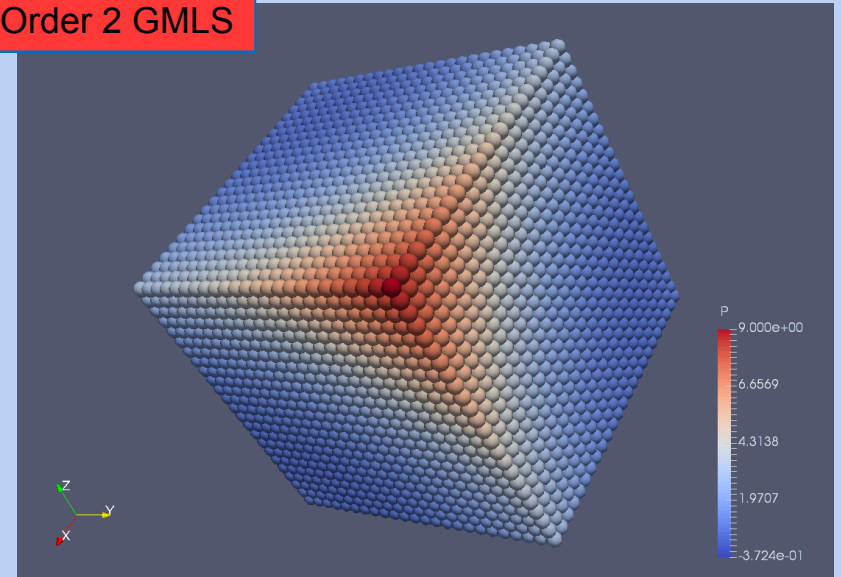


Spacing	L <sup>2</sup> Error	Rate
1/3	8.23e-4	-
1/6	1.390e-4	2.57
1/12	3.22e-5	2.11
1/24	7.88e-6	2.03
1/48	1.97e-6	2.00

Manufactured Solution (Exact):

$$x(1+x+y+z)+y(1+y+z)+z(1+z)$$

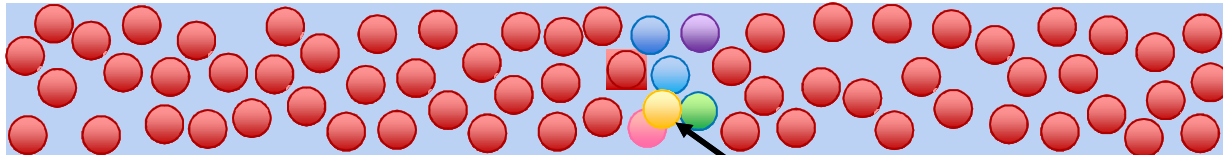
Order 2 GMLS



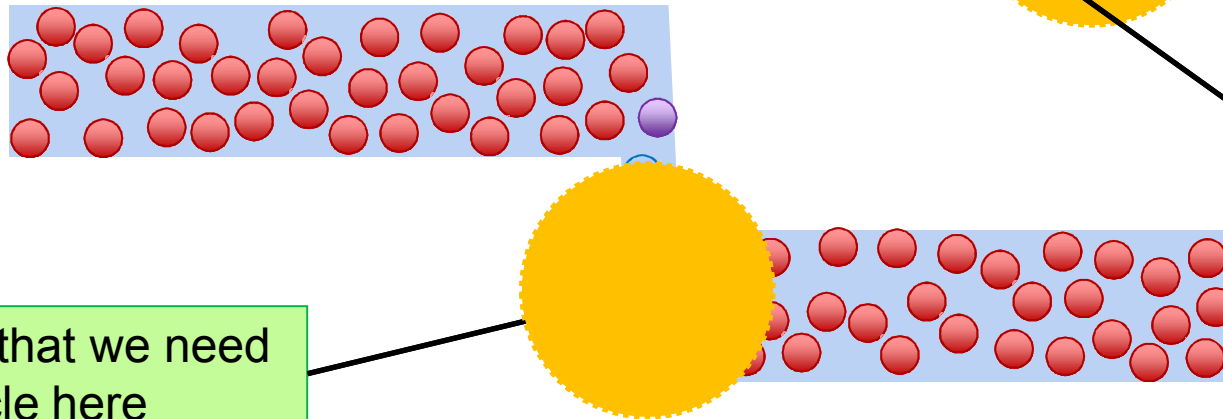
L<sup>2</sup> Error: 9.25978e-17

# Lagrangian simulation particle insertion

Material coordinates and initial configuration:



Deformed physical coordinates:



Detect that we need a particle here

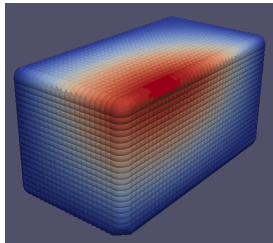
1. Using physical coordinates in neighborhood as targets, generate GMLS polynomial coefficients.

2. Use these coefficients on material coordinates of neighborhood to construct material coordinate.

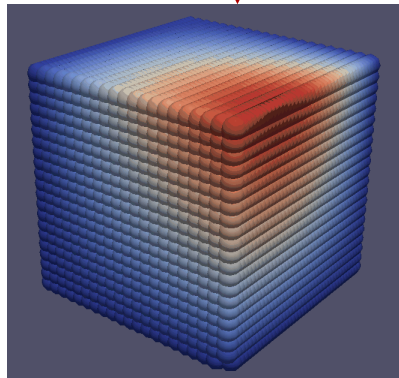
# Point Insertion Demonstration

## Physical Frame $\rightarrow$ Material Frame

Reference 1x1x2  
Hyperrectangle (n-orthotype)  
(Lagrangian/Material frame)

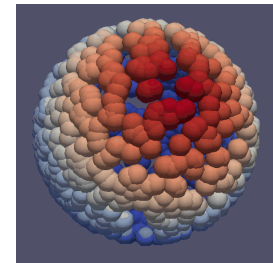


Under transformation  
 $T(x,y,z)=(2x,2y,z)$



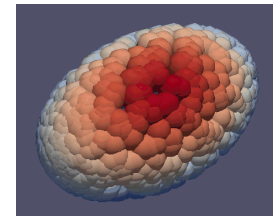
2x2x2 Hypercube  
(Physical frame)

Inserting points into the physical frame, we know the physical location, but we do not know the Lagrangian/material coordinate.



Sphere of radius = 0.75,  
centered at (1,1,1)  
(Physical frame)

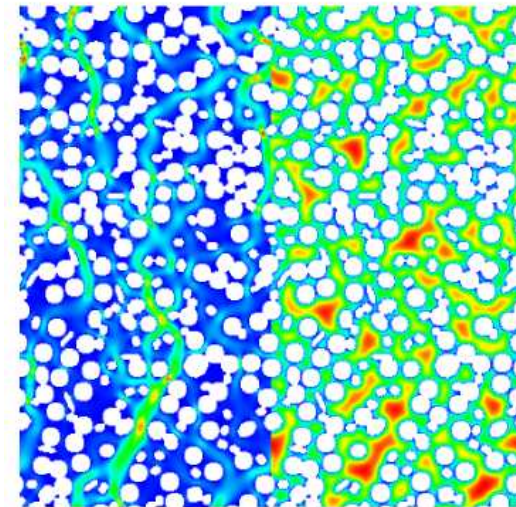
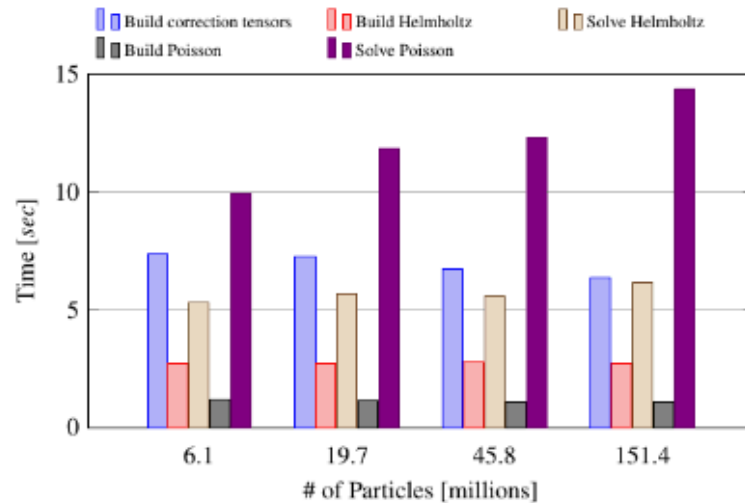
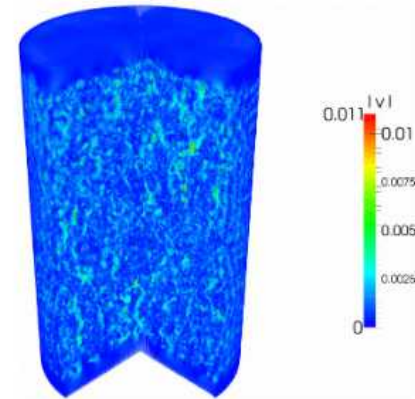
We use GMLS to get coefficients for the reconstruction in the physical frame (using physical frame coordinates for neighbor search), and then use these coefficients on the Lagrangian/material coordinate info (effectively recovering  $T^{-1}$ ).



Ellipsoid  
(Lagrangian/Material frame)

# Anticipated Scalability (Based on ...)

Discretization			Per Processor		
N	$dx$	# Particles	# Processor	# Particles	Load balance
128	6.875e-05	6,083,687	432	14,083	1.0003
192	4.583e-05	19,701,287	1,440	13,682	1.0004
256	3.437e-05	45,803,537	3,432	13,347	1.0007
384	2.291e-05	151,438,991	11,376	13,313	1.0006



Refs: Trask JCP 2016, W Pan JCP 2017

# Other Features/Info

- **Conditional Building of Toolkit**
  - VTK, netcdf, GSL, boost are all optional
    - Cmake configuration checks for their existence, also for CMAKE variables or shell variables containing locations of TPLs
    - Some combination of the above must be used to perform any meaningful computation  
( can not load in files / write files without either VTK or netcdf )
- **Ctest for testing various configurations for executables and classifying them by labels assigned**
  - i.e. ``ctest -L remap`` focuses on tests that use remap
  - 27 tests at this point, exercising capabilities including remap, insertion, solving problems exactly, convergence rates under refinement, Lagrangian simulation insertion of points into various frames.

# File Structure

## Compadre Core

<p><b><u>Configuration / Utilities:</u></b></p> <p>Compadre_Typedefs          Compadre_GlobalConstants          Compadre_AnalyticFunctions          Compadre_FileIO          Compadre_ParameterManager          Compadre_XyzVector</p> <p><b><u>Problem/Assembly/Solution:</u></b></p> <p>Compadre_ProblemT          Compadre_BoundaryConditionsT          Compadre_PhysicsT          Compadre_SourcesT</p> <p>Compadre_SolverT          Compadre_DirectSolverT          Compadre_IterativeSolverT</p> <p><b><u>Partical/Coordinate/Field Management</u></b></p> <p>Compadre_ParticlesT</p> <p>Compadre_CoordsT          Compadre_EuclideanCoordsT          Compadre_SphericalCoordsT</p> <p>Compadre_DOFManager          Compadre_FieldManager          Compadre_FieldT</p>	<p><b><u>Tree Searches:</u></b></p> <p>Compadre_NeighborhoodT          Compadre_nanoflannInformation          Compadre_nanoflannPointCloudT</p> <p><b><u>Particle Injection/Remap:</u></b></p> <p>Compadre_RemapManager</p> <p><b><u>Examples:</u></b></p> <p>CompadreGMLSTest          CompadreTrilinosTest          Compadre_KdTreeTest          Compadre_ParallelReaderTest          Compadre_RemapTest          Compadre_SimpleCoordsUnitTest          Compadre_SimpleMPIUnitTest          Compadre_SphReaderTest          Compadre_SpherePoissonTest</p> <p><b><u>Tests:</u></b></p> <p>Examples tested under 16 different configurations of processor/threads</p>
--	---

## Problems

Compadre\_GMLS\_PinnedLaplacian\_BoundaryConditions  
 Compadre\_GMLS\_PinnedLaplacian\_Operator  
 Compadre\_GMLS\_PinnedLaplacian\_Sources

Compadre\_PinnedGraphLaplacian\_BoundaryConditions  
 Compadre\_PinnedGraphLaplacian\_Operator  
 Compadre\_PinnedGraphLaplacian\_Sources

## GMLS

GMLS\_Config.h.in (cmake configuration)

GMLS\_gsl  
 GMLS\_particle\_gsl (LU or SVD solve)

GMLS\_boost  
 GMLS\_particle\_boost (LU solve ONLY)

Questions?