SAND2014-2697P

Fast Algorithms for Particle-Based Simulations

Stan Moore

April 3, 2014

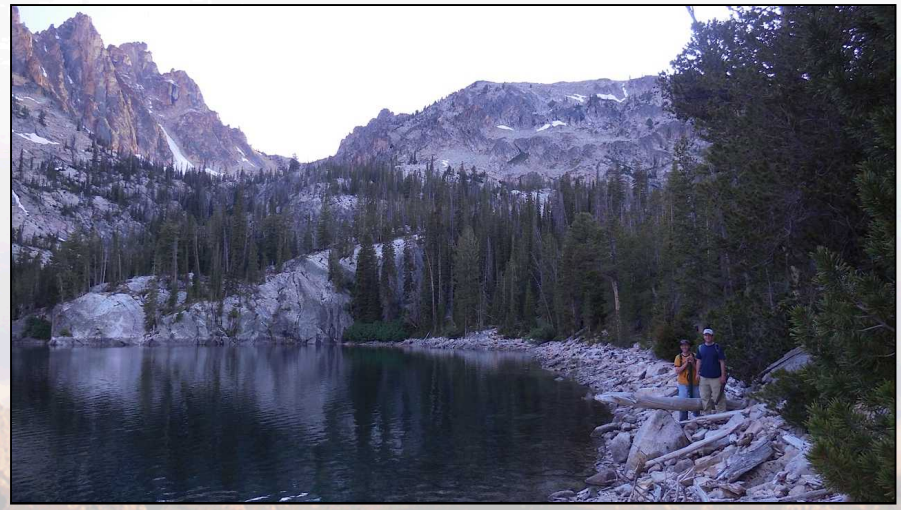
1444 Interview Candidate

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.

Sandia National Laboratories

A little about me …

- Grew up on a farm in Filer, Idaho
- Love outdoor activities (backpacking, hiking, fly fishing, etc.)
- BS Chemical Engineering, Brigham Young University, 2008
- PhD Chemical Engineering, Brigham Young University, 2012
- Dissertation topic: Developing a new method to predict chemical potential using molecular simulations



Sandia Projects

- Been at Sandia since August, 2012

Projects:

- **LAMMPS (molecular dynamics) code development:** Added and improved long-range electrostatic methods
- **Modeling of a dipolar fluid in an external electric field** (uses LAMMPS)
- **Advanced force field development (SNAP)** (uses LAMMPS, working to develop a quantum accurate SiO_2 force field)
- **Aleph (PIC-DSMC) code development:** Added an automatic mesh coarsening method for particle interactions (based on oct-tree algorithm)
- **Modeling of triggered vacuum gap switches**

Thanks to Paul Crozier for being a great mentor during my time at Sandia

Importance of Algorithms

- Computing resources limited
- Gains in processor speed not as dramatic as before
- Increasing number of processors increases power consumption
- Better algorithms can save time and money
- Example: Fast Fourier transform (FFT) vs traditional discrete Fourier transform

Introduction

- Hybrid particle-in-cell (PIC) and direct simulation Monte Carlo (DSMC) methods are frequently used to simulate low density interacting plasmas
- A single mesh is often used for both PIC and DSMC calculations
- The mesh size for PIC is often limited by the Debye length
- The collision cell size for DSMC is limited by the mean free path (can be much larger than the Debye length)
- Too few computational particles per DSMC collision cell can lead to errors
- Therefore, the optimal PIC mesh may be suboptimal for calculating DSMC collisions

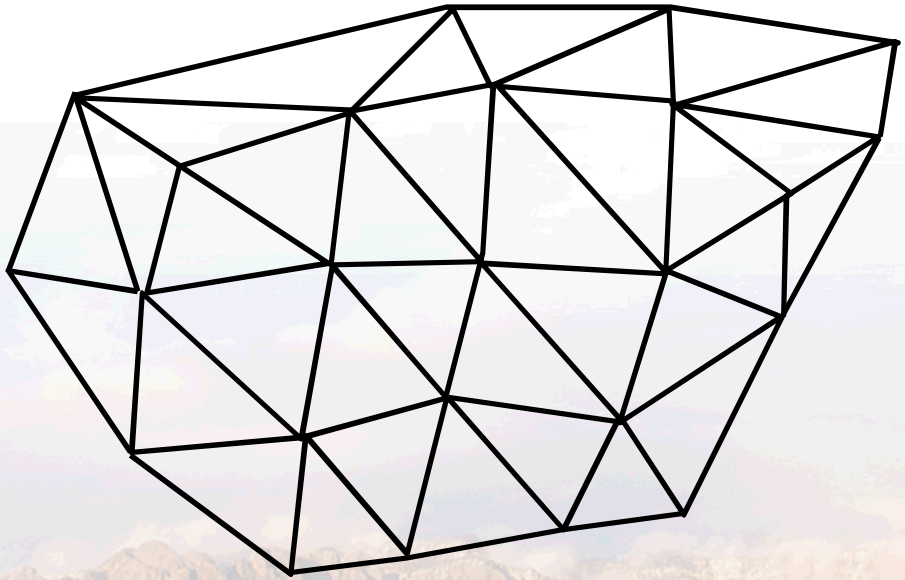
Parts of this work were done in collaboration with Paul Crozier, Chris Moore, and Matt Bettencourt

Overview of New Patching Method

- Use a fine mesh for PIC (unstructured)
- Use a rectangular grid to conglomerate many PIC elements into a single DSMC collision cell
- Size DSMC collision cells based on mean free path, λ_{mfp}
- Use oct-tree algorithm to adjust the size of DSMC collision cells on the fly

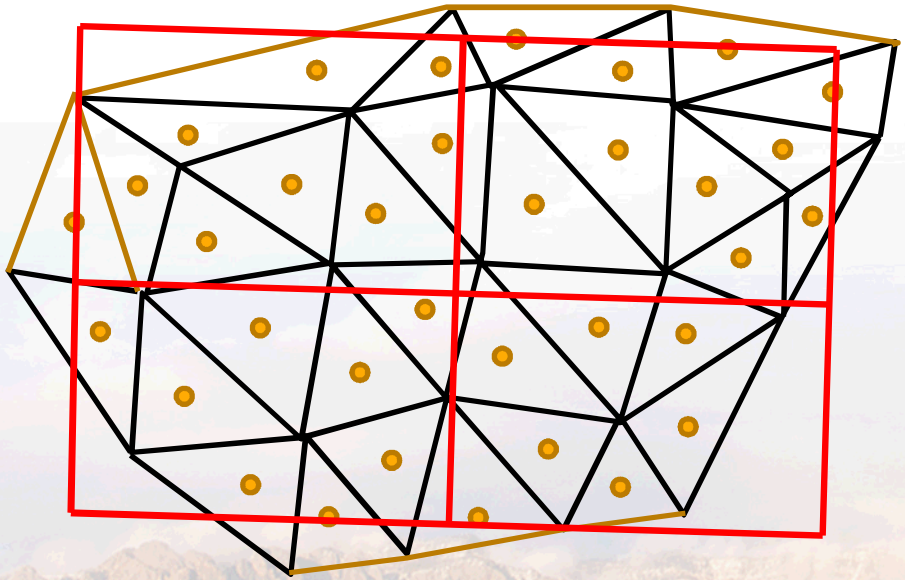
Patch Method

- Original (unstructured) PIC mesh



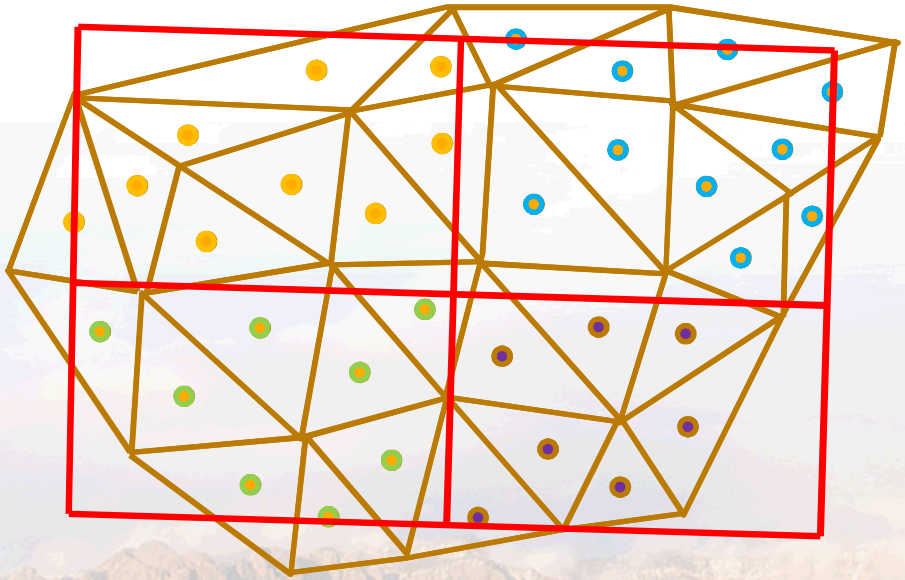
Patch Method

- Apply rectilinear grid based on element centroid

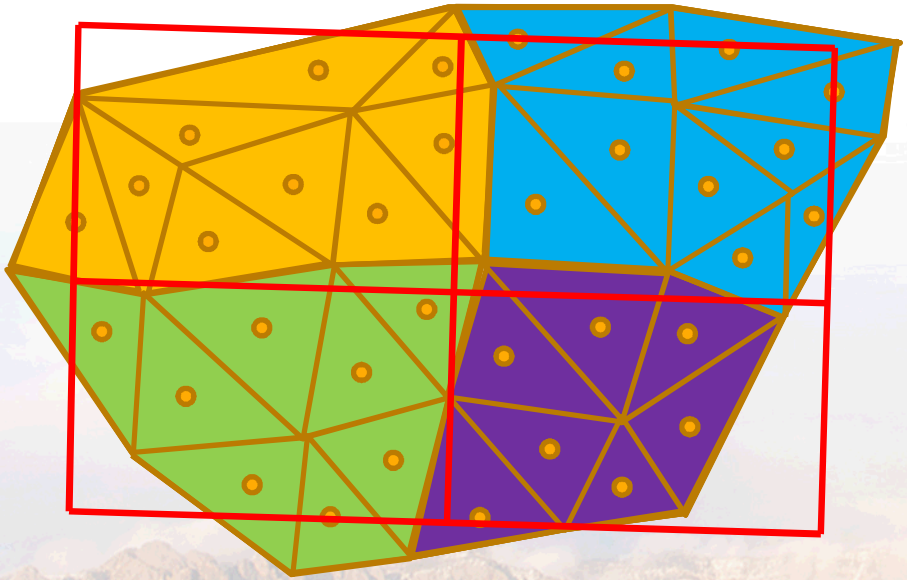


Patch Method

- Assign elements to patches (based on element centroid)

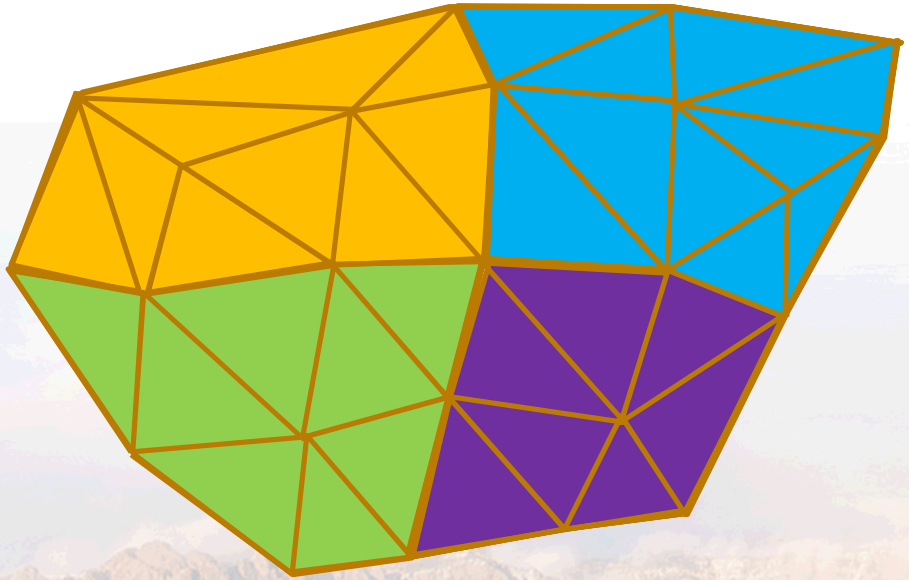


Patch Method



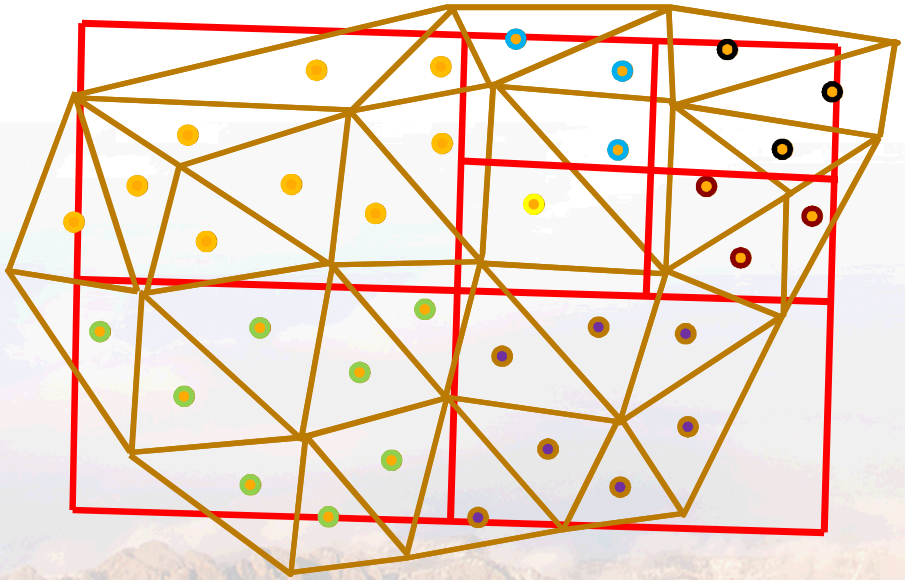
Patch Method

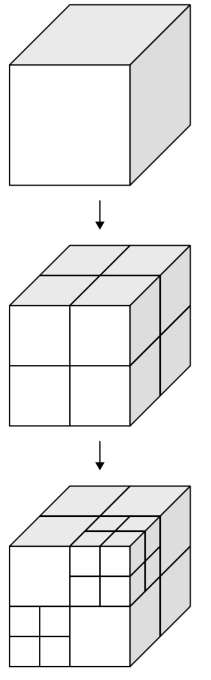
- Patched mesh
- Use patches to compute DSMC collisions



Oct-tree Refinement

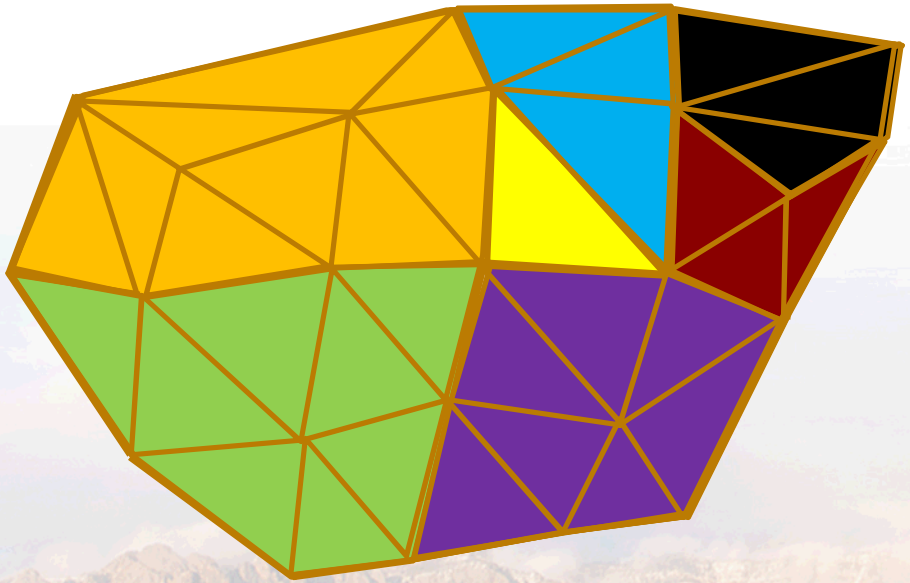
- Use oct-tree algorithm to refine mesh based on mean free path





Oct-tree Refinement

- Refined mesh



Temporal Averaging

- Calculate λ_{mfp} for each element as:

$$\lambda_{\text{mfp}} = \frac{v}{Z} n$$

v = velocity
Z = interaction frequency
n = number of particles

- With a high computational particle weighting, temporal smoothing is needed. Can use either:

$$\left\langle \lambda_{\text{mfp}} \right\rangle = \left\langle \frac{v}{Z} n \right\rangle \qquad \left\langle \lambda_{\text{mfp}} \right\rangle = \frac{\langle v \rangle}{\langle Z \rangle} \langle n \rangle$$

- Sometimes can get zero interactions in a timestep. With the first option, this leads to division by zero
- Found that the second option works much better

Automatic Sizing of Patches

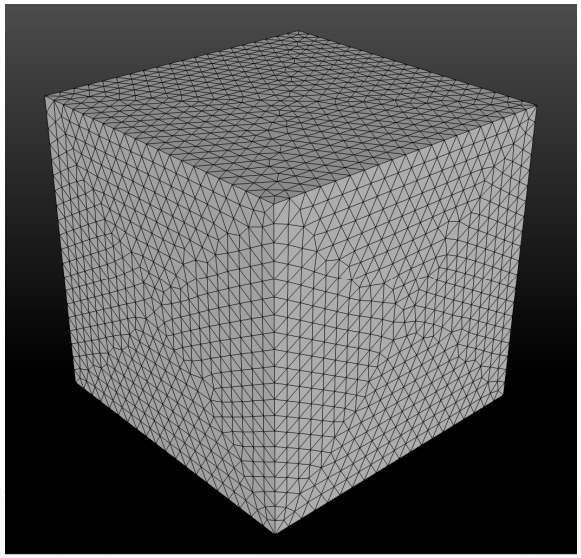
Patch size is dynamically adjusted based on the local mean free path λ_{mfp}:

1. Compute λ_{mfp} for each interaction on an elemental basis (using all species)
2. For each interaction, <u>average</u> λ_{mfp} over elements in the oct-tree cell
3. Take the <u>minimum</u> of all the average λ_{mfp} and divide by 2, use this to size patches using the oct-tree algorithm

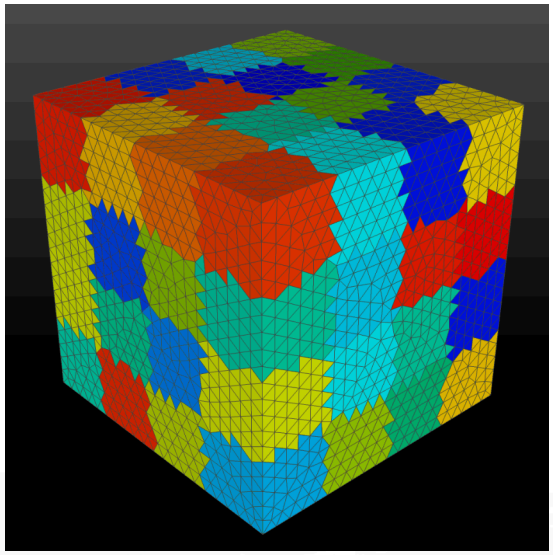
HI Test Problem

- Hydrogen iodide (HI) molecules interacting with elastic collisions
- Use variable hard sphere (VHS) interaction cross section
- T = 594.6 K, n = 10^{20} m^{-3}
- 3D cubic system, L = 5 cm
- Analytic λ_{mfp} = 3.67 cm
- 0.5*λ_{mfp} => minimum of 27 patches, but oct-tree on cube uses powers of 8 => 64 patches

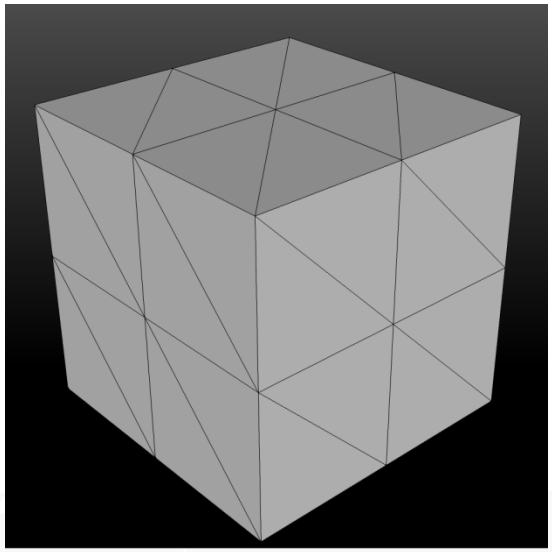
3 Meshes



fine mesh
56,557 elements
1,696,710 particles



patched mesh
64 patches
1,920 particles



coarse mesh
96 elements
2,880 particles

- **Adjust particle weighting → approximately 30 particles per element or patch**

Accuracy Comparison

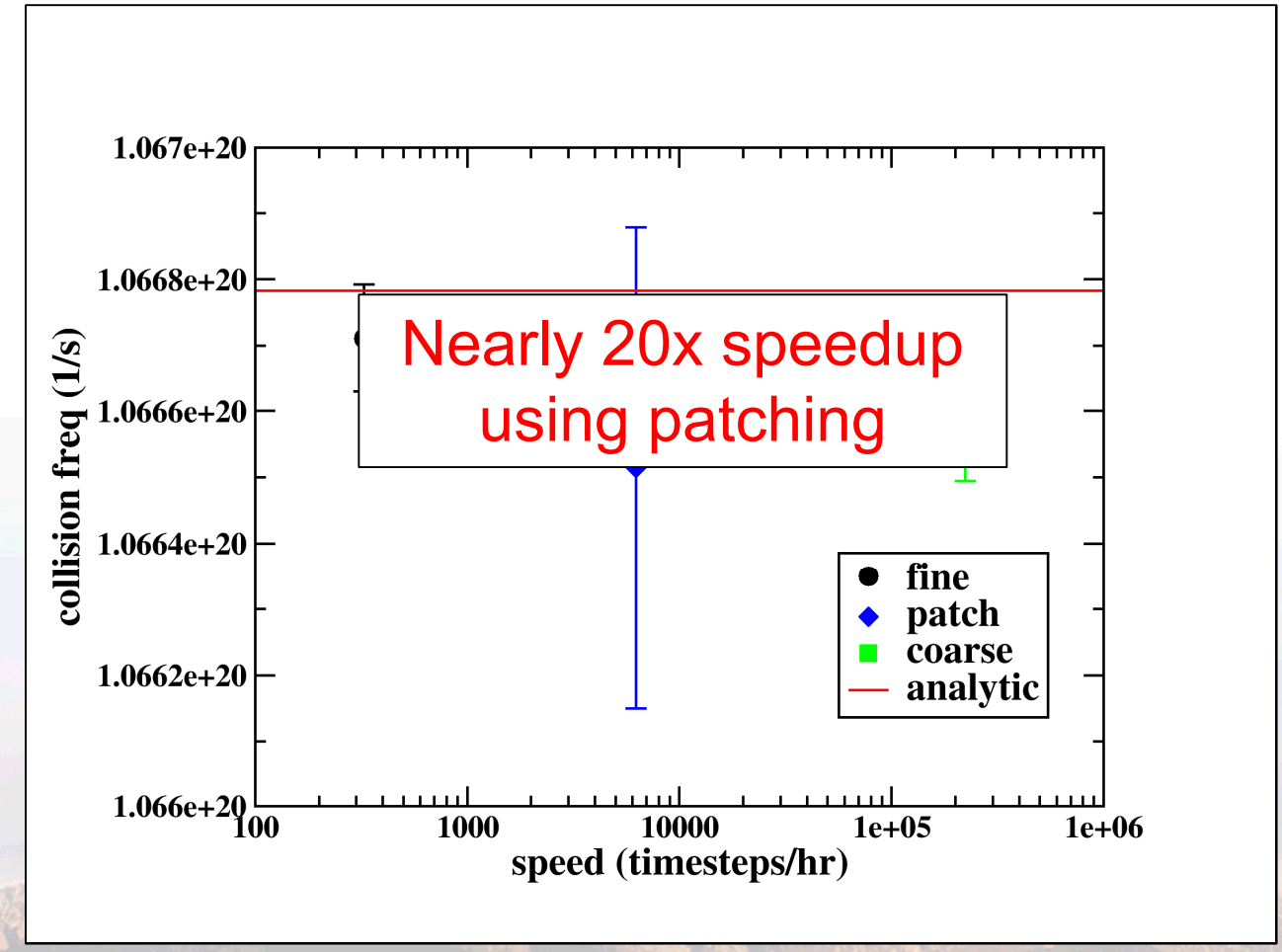
Ran each simulation for 3 hours ($\Delta t = 10^{-5}$ s) on one processor, repeated 6 times with a different random number seed

Average Collision Frequency:

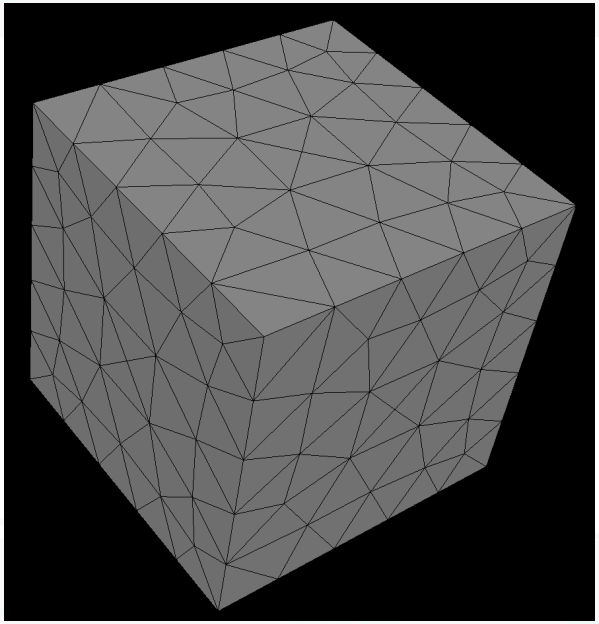
- $f_{analytic} = 1.0667 \cdot 10^{20}\ s^{-1}$
- $f_{fine} = 1.0668 \cdot 10^{20}\ s^{-1}$ → 0.007% relative error
- $f_{patch} = 1.0665 \cdot 10^{20}\ s^{-1}$ → 0.025% relative error
- $f_{coarse} = 1.0666 \cdot 10^{20}\ s^{-1}$ → 0.021% relative error

Results for the HI Test Problem

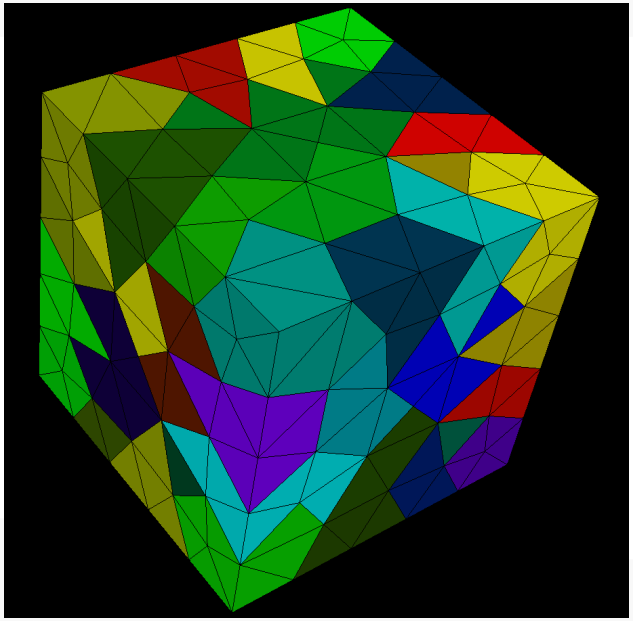
- Error bars represent 95% confidence intervals



Another 2 Meshes



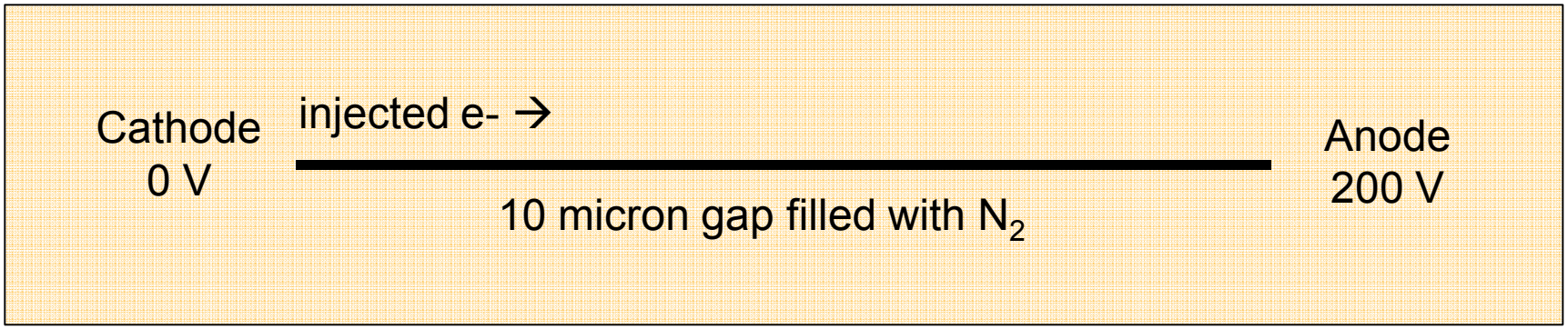
fine mesh
1,184 elements
1,223 particles



patched mesh
64 patches
1,223 particles

- **Same HI test problem (ran for 2000 timesteps)**
- **Patched mesh runs nearly 4x faster (with virtually the same accuracy)**
- **Global average (0D problem) may hide errors**

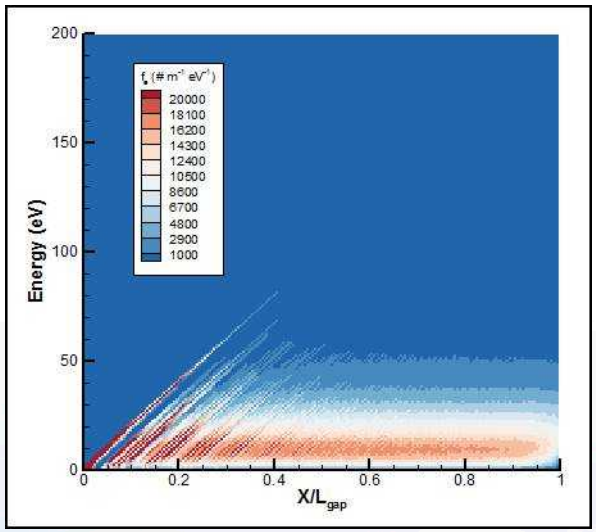
1D Inelastic Collision Problem



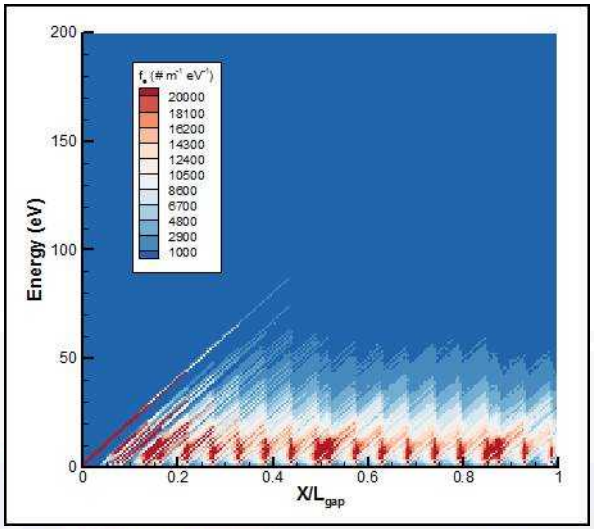
- **Electrons gain energy due to the field**
- **Electrons lose energy due to inelastic collisions with N_2 particles (fixed in space)**
- **Vary N_2 particle weight so that there are approximately:**
  1. **10 computational particles per cell**
  2. **0.1 computational particles per cell**
- **About 10 λ_{mfp} across the gap**

Electron Energy Distribution Function (EEDF)

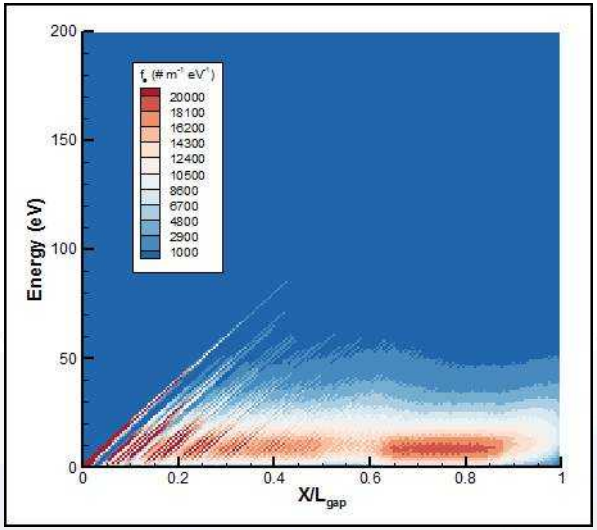
10 N_2 particles per cell



0.1 N_2 particles per cell



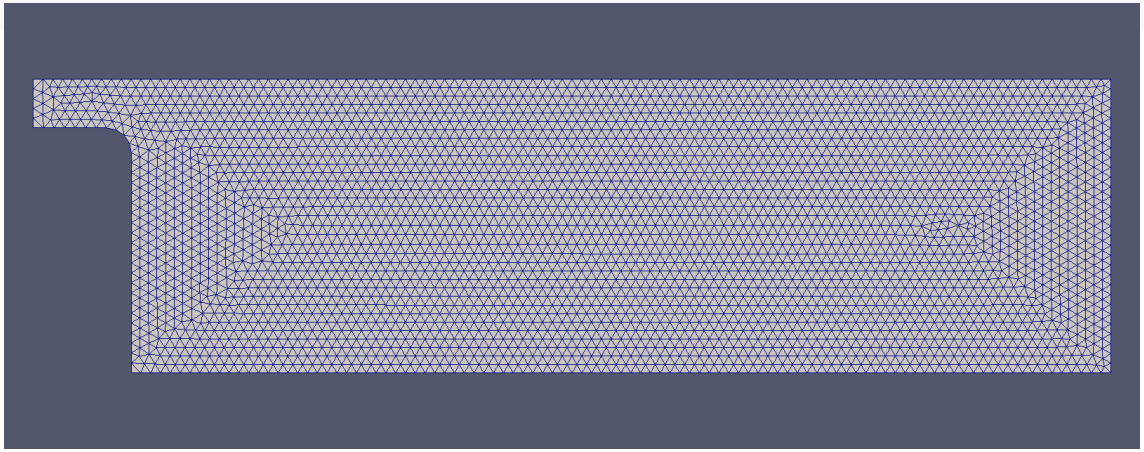
0.1 N_2 particles per cell with patching



- **Using patching gives a more realistic EEDF (crucial for simulating accurate breakdown voltages)**
- **Patching allows one to use fewer N_2 particles**

2D Vacuum Gap Breakdown

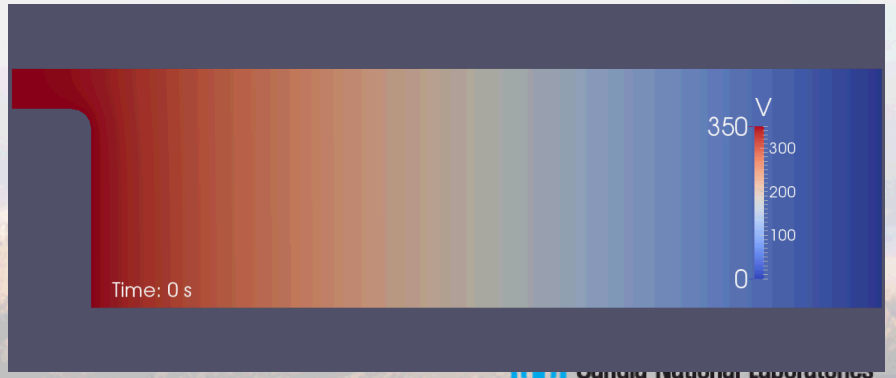
Anode
350 V



Cathode
0 V

- Air injected at high velocity and high temperature from the anode
- Low density electrons injected from the cathode
- Air ionizes and eventually will form plasma and break the gap

Species Densities (m^{-3})

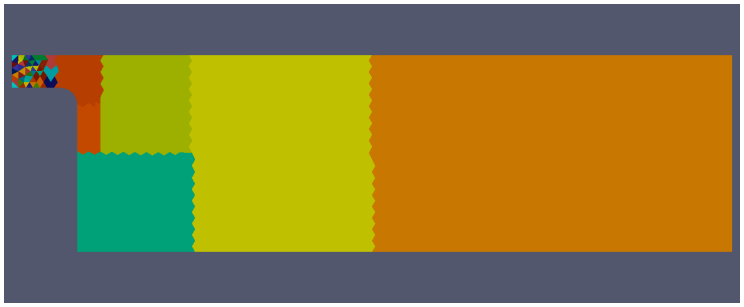


Dynamic Sizing of DSMC Collision Cells

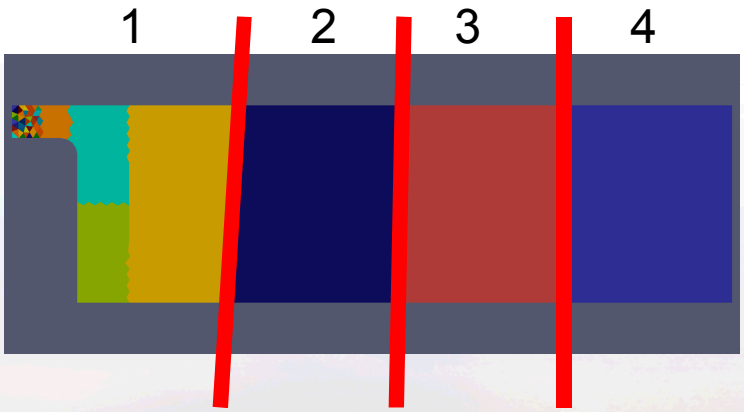
Parallelization

- **Currently each processor uses an independent oct-tree (simple):**

1 processor:



4 processors:



- **Issue: can lead to narrow/small cells (can happen with 1 proc too)**
- **In the future, could use a more robust method like inter-processor communication with ghosting, load balancing, etc.**

Conclusions

- Developed a new method (patching) to enhance hybrid PIC-DSMC simulations
- Using fewer particles with patching gives similar accuracy and uncertainty as using a fine mesh with many particles
- Allows one to dramatically speed up the simulation if the PIC mesh is too small for DSMC collisions
- Allows one to dynamically adjust the size of DSMC collision cells on the fly (based on mean free path)

Sandia Projects

- Been at Sandia since August, 2012

Projects:

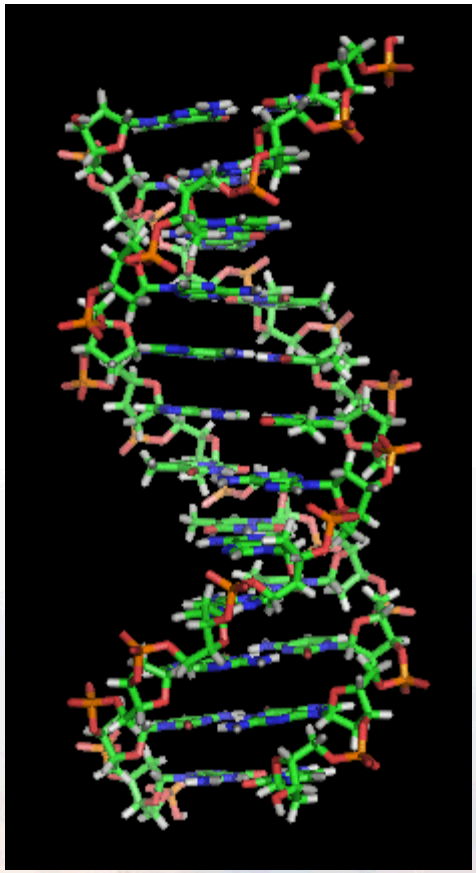
- **LAMMPS (molecular dynamics) code development:** Added and improved long-range electrostatic methods
- **Modeling of a dipolar fluid in an external electric field** (uses LAMMPS)
- **Advanced force field development (SNAP)** (uses LAMMPS, working to develop a quantum accurate SiO_2 force field)
- **Aleph (PIC-DSMC) code development:** Added an automatic mesh coarsening method for particle interactions (based on oct-tree algorithm)
- **Modeling of triggered vacuum gap switches**

Thanks to Paul Crozier for being a great mentor during my time at Sandia

Why do we care about LRE?

LRE = long-range electrostatics

- Electrostatics are extremely important in many atom-level (and coarser) models.
- Long-range part usually cannot be neglected in molecular simulations (simple cutoff can lead to artifacts).
- Large fraction of compute cycles are used in LRE calculations.
- LRE calculations represent a scaling bottleneck in many MD calculations.

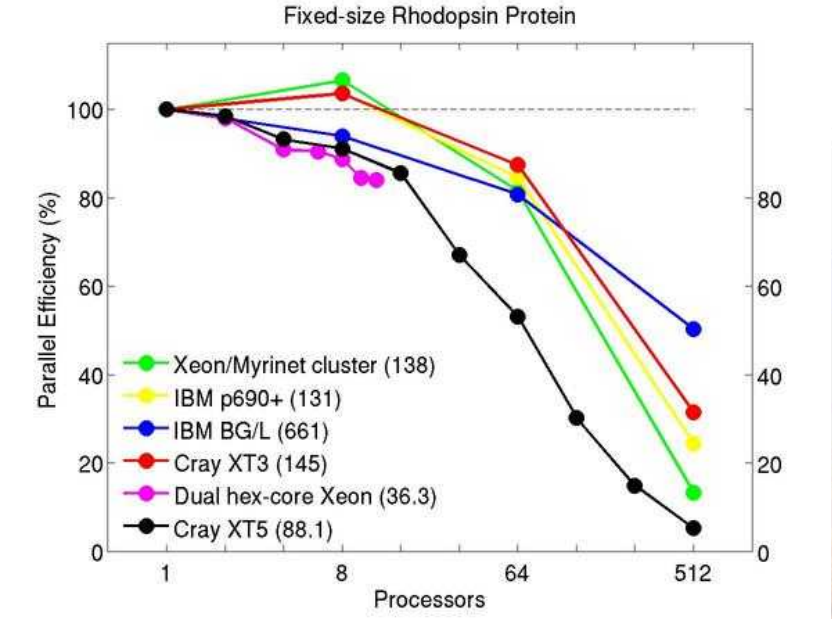


Parts of the LAMMPS long-range electrostatic work were done in collaboration with Paul Crozier, Steve Plimpton, and Stephen Bond

The motivation: FFTs don't scale very well

(and HPC core counts are growing quickly)

- LAMMPS originally had two methods for computing long-range electrostatics: Ewald and particle-particle/particle-mesh (PPPM)
- Ewald summation is fastest for small systems (or very high accuracy), but expensive for large systems
- PPPM relies on FFTs, which don't scale well on many processors:



http://lammps.sandia.gov/bench/rhodo.fixed.jpg

Multilevel-summation method (MSM) background

- Multi-grid method (but not iterative); split potential and approximate the slowly varying part on a hierarchy of grids
- No FFTs are required, so the communication cost of MSM is expected to scale better than PPPM on large core counts
- PPPM scales with number of atoms as O(N log N) while MSM scales as O(N)
- MSM may be faster for large problems running on large core counts

D. J. Hardy, Ph.D. thesis, University of Illinois at Urbana-Champaign (2006).

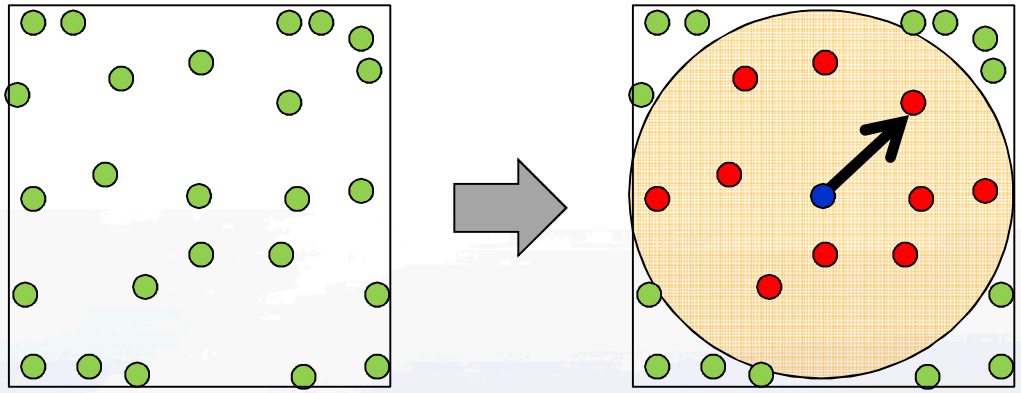
D. J. Hardy, J. E. Stone, and K. Schulten, Parallel Comput. **35**, 164 (2009).

MSM Algorithm pieces

1. Short-range part
2. Anterpolation
3. Direct sum
4. Restriction
5. Prolongation
6. Interpolation

Algorithm pieces

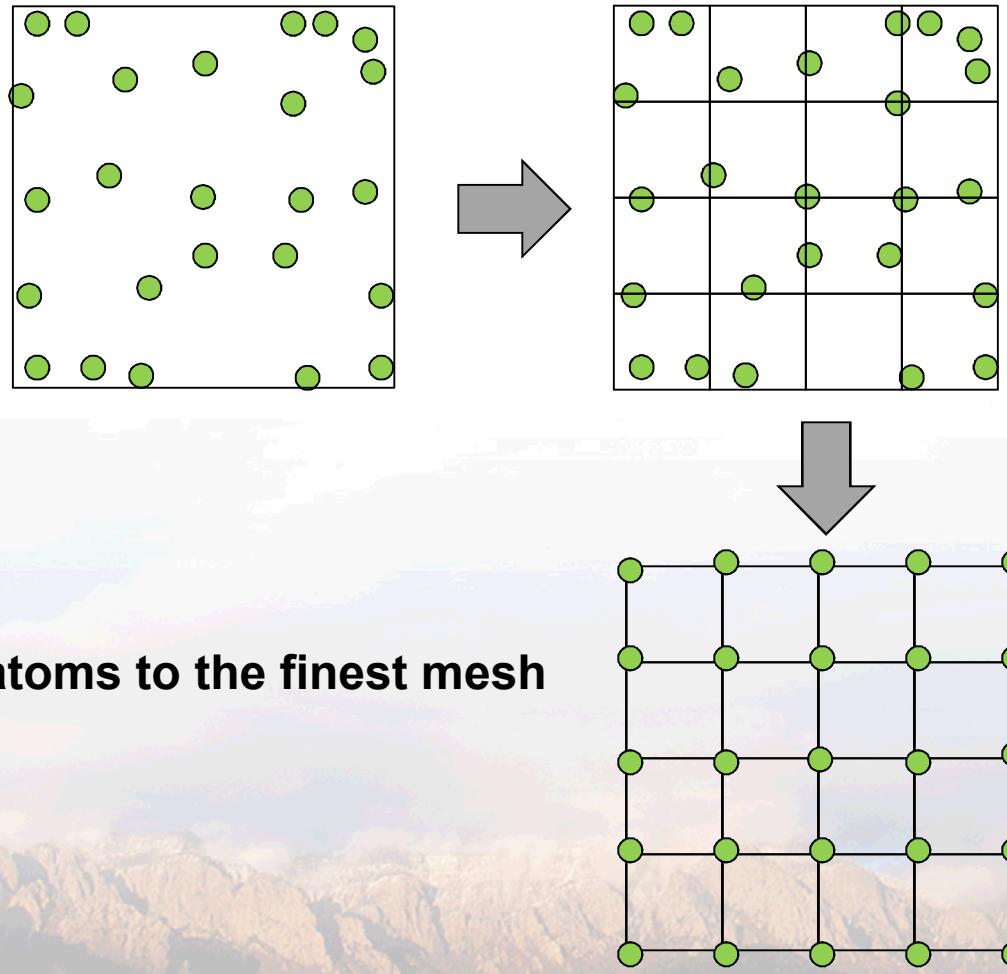
1. **Short-range part**
2. Anterpolation
3. Direct sum
4. Restriction
5. Prolongation
6. Interpolation



Particles interact within a spherical cutoff to get short-range forces, energy, and pressure

Algorithm pieces

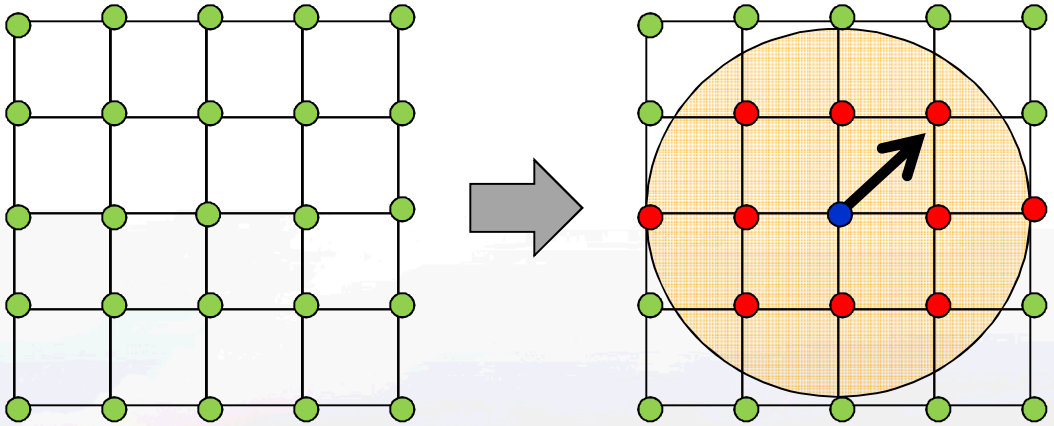
1. Short-range part
2. **Anterpolation**
3. Direct sum
4. Restriction
5. Prolongation
6. Interpolation



Interpolate charges from atoms to the finest mesh

Algorithm pieces

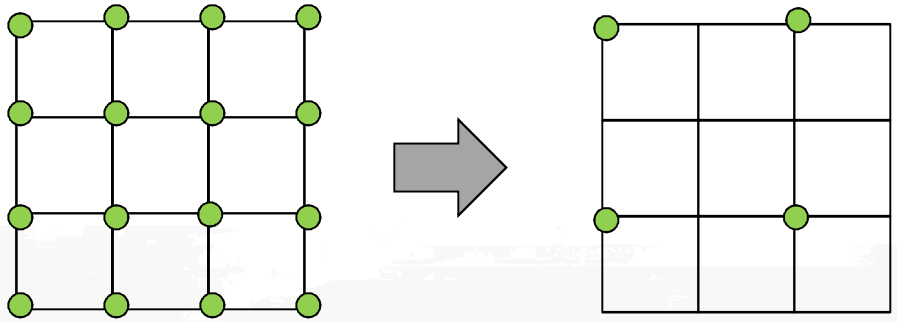
1. Short-range part
2. Anterpolation
3. **Direct sum**
4. Restriction
5. Prolongation
6. Interpolation



Mesh points interact within a spherical cutoff to get long-range electric field, energy, and pressure

Algorithm pieces

1. Short-range part
2. Anterpolation
3. Direct sum
4. **Restriction**
5. Prolongation
6. Interpolation

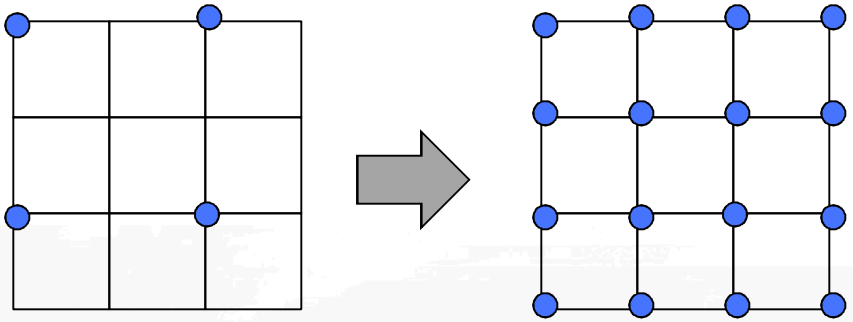


Interpolate charge from finer mesh to coarser mesh

Repeat steps 3 and 4 until finished on coarsest mesh

Algorithm pieces

1. Short-range part
2. Anterpolation
3. Direct sum
4. Restriction
5. **Prolongation**
6. Interpolation

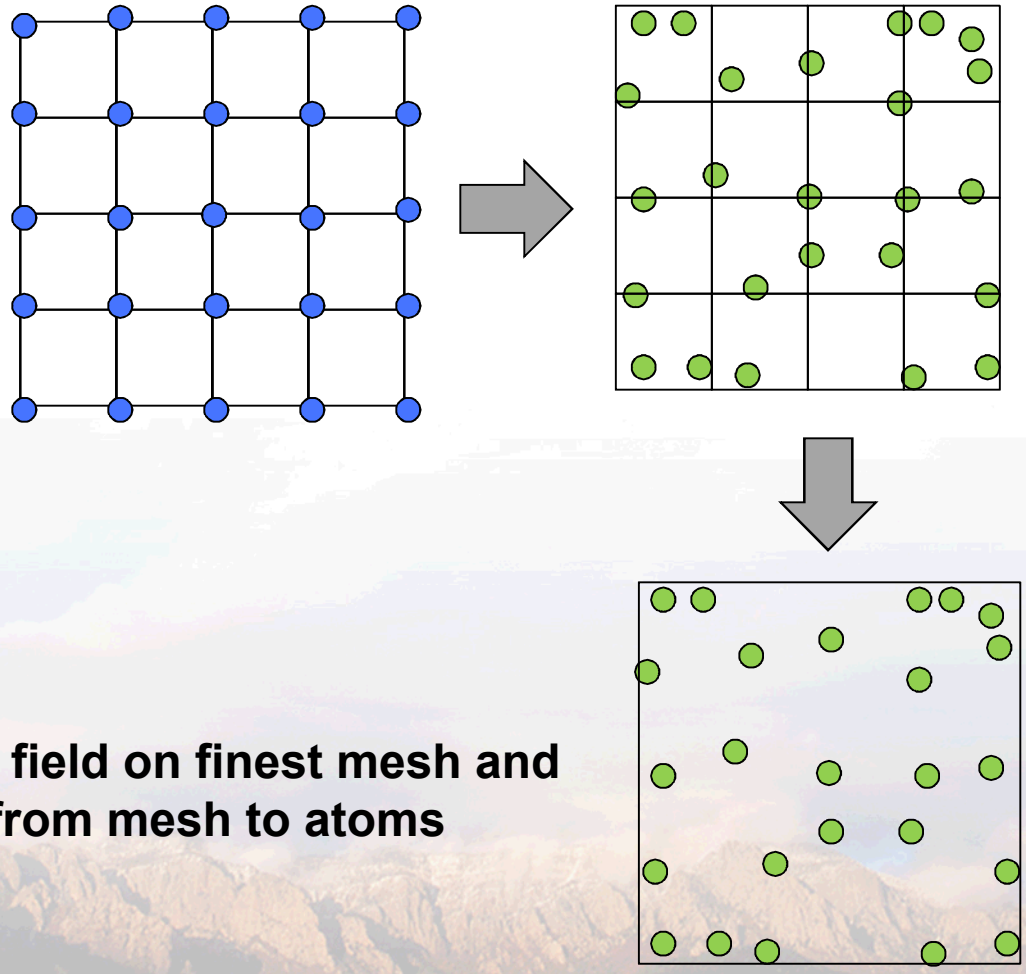


Interpolate electric field from coarser mesh to finer mesh

Repeat step 5 until the finest mesh is reached

Algorithm pieces

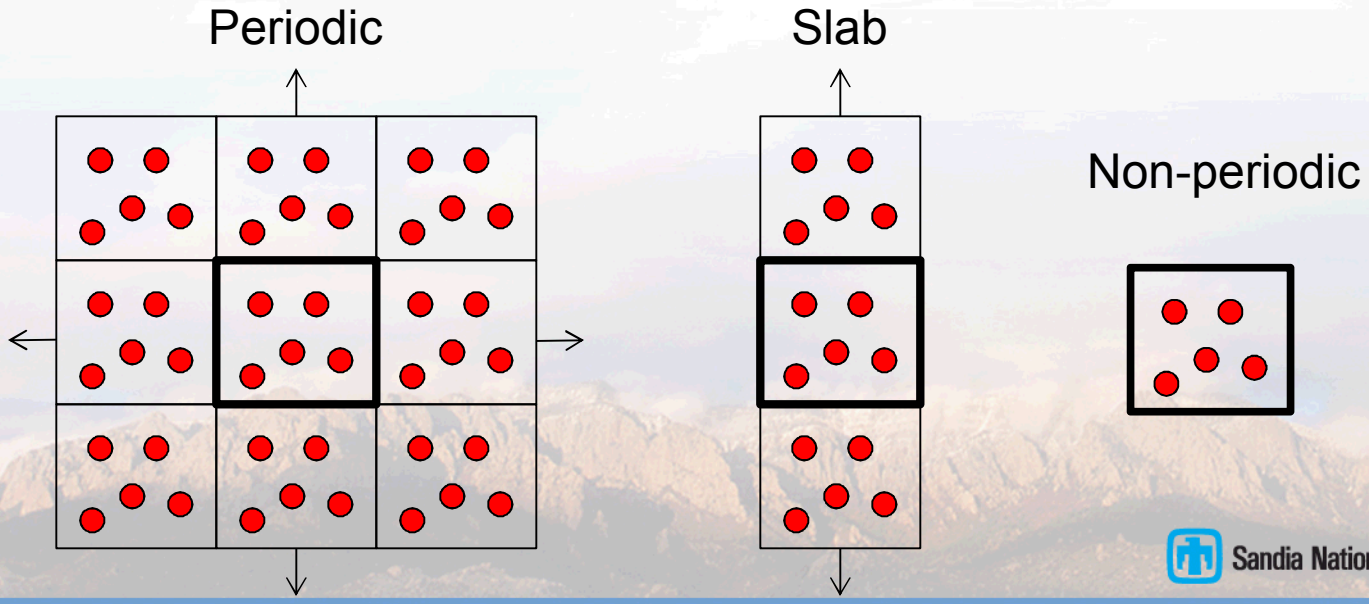
1. Short-range part
2. Anterpolation
3. Direct sum
4. Restriction
5. Prolongation
6. **Interpolation**



Compute force from electric field on finest mesh and back-interpolate force from mesh to atoms

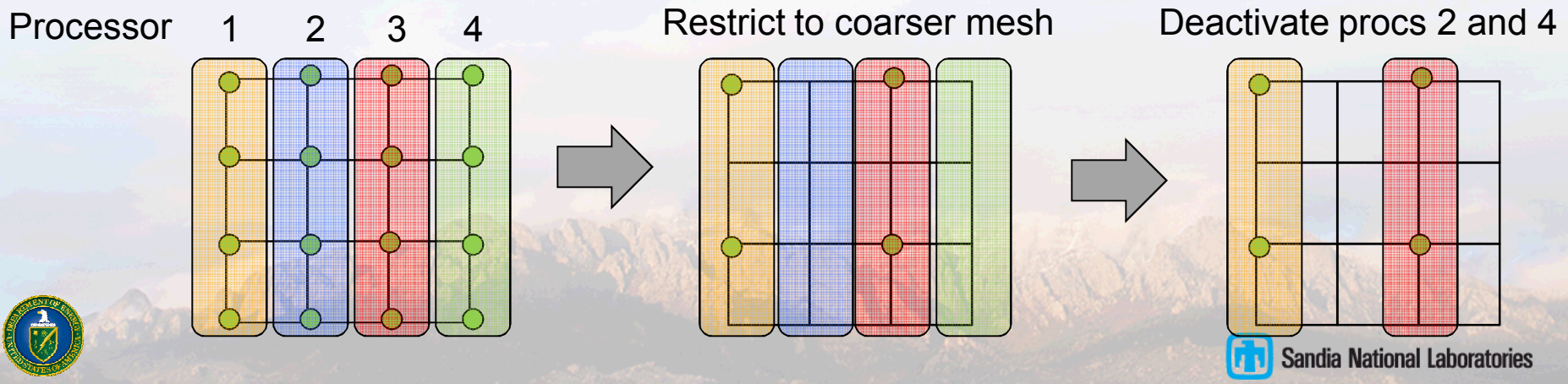
Non-Periodic BCs

- MSM works for both periodic and non-periodic boundary conditions
- Ewald and PPPM only work for periodic or slab (periodic in *x* and *y* and non-periodic in *z*) boundary conditions



Parallelization Strategy

- Challenge: lots of work on finest grid, very little work on coarsest grid
- Use same domain-decomposition layout on all levels (simple)
- Inactive processors don't participate in MPI communication routines
- Use neighbor point-to-point communication for fine grids
- Use MPI AllReduce for coarse grids

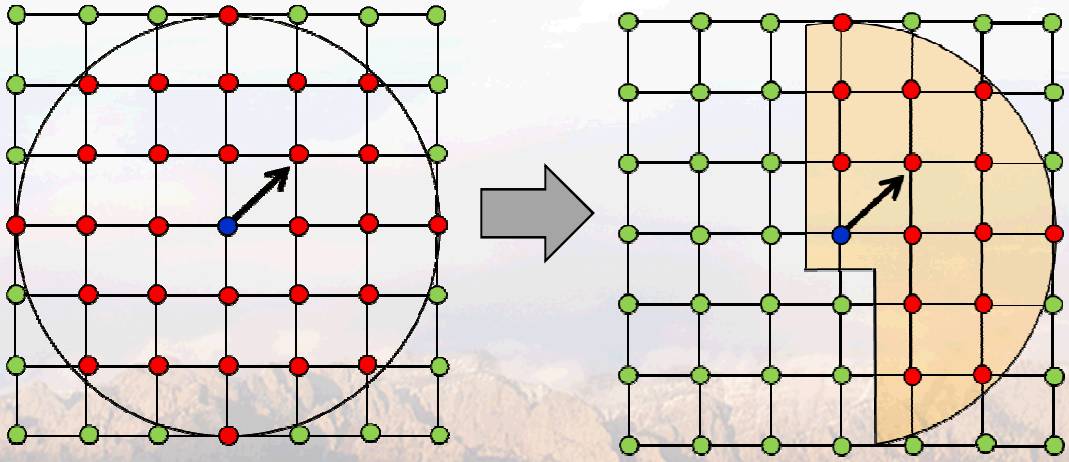


Enhancements to MSM

- Error estimator (important for comparing to other methods like PPPM)
- Pressure calculation
- Added heuristic to estimate optimal parameters, including automatic adjustment of Coulombic cutoff (based on work by Hardy)
- Per-atom energy/virial
- Fast scalar pressure
- OMP threaded version of MSM (Axel Kohlmeyer)

Improving Single-Core Performance

- Use hemisphere (instead of full sphere) for direct sum interactions to avoid double computations
- Using a hemisphere can also (sometimes) reduce the amount of communication needed
- Added various other code optimizations
- Compared to Hardy's NAMD-lite code, LAMMPS MSM was 60% faster for periodic and 25% faster for non-periodic (1 processor, two point-charges in a box, order 4)



[David J. Hardy, *NAMD-Lite*, http://www.ks.uiuc.edu/Development/MDTools/namdlite/, University of Illinois at Urbana-Champaign, 2007.]

Fast (Scalar) Pressure Calculation

- Calculation of the 6-component pressure tensor is expensive with MSM (increases cost by ~2x)
- Often only scalar pressure [i.e. $1/3*(P_{xx} + P_{yy} + P_{zz})$] is needed
- For Coulombic systems, can use a virial "trick" to relate energy to scalar pressure (much cheaper)
- For SPC/E system, reduces overall cost by 20% (short-range part has some overhead)
- Can use scalar pressure to run isotropic barostat

LRE speed and scalability tests

Chama

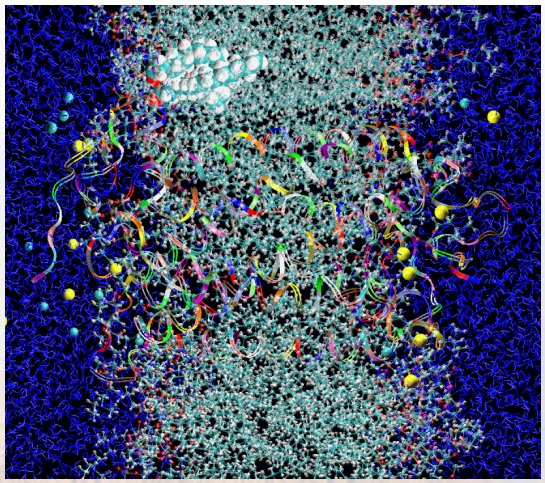
- Sandy-Bridge (2.6 GHz)
- 2012
- 16 cores/node
- 1,232 nodes
- 19,712 cores
- Infiniband 4X QDR, Fat Tree, Qlogic

Redsky

- Nehalem (2.93 GHz)
- 2009
- 8 cores/node
- 2,816 nodes
- 22,528 cores
- Infiniband 4X QDR, 3D Torus, Mellanox

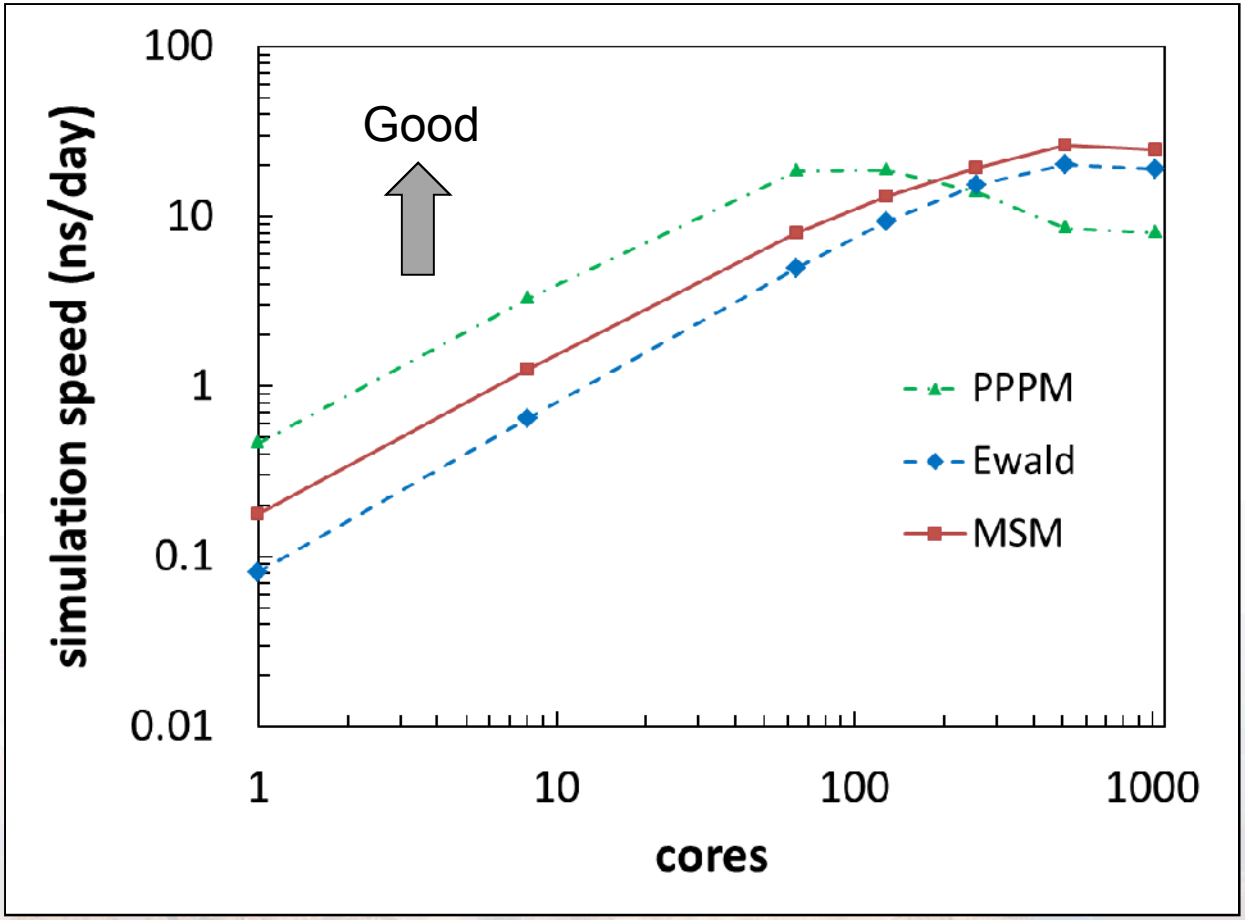
Rhodopsin benchmark

- NVT dynamics
- 1e-4 relative accuracy
- 32k atoms, replicable
- 2 fs timestep size



Redsky, 32k atoms, 10^{-4} accuracy

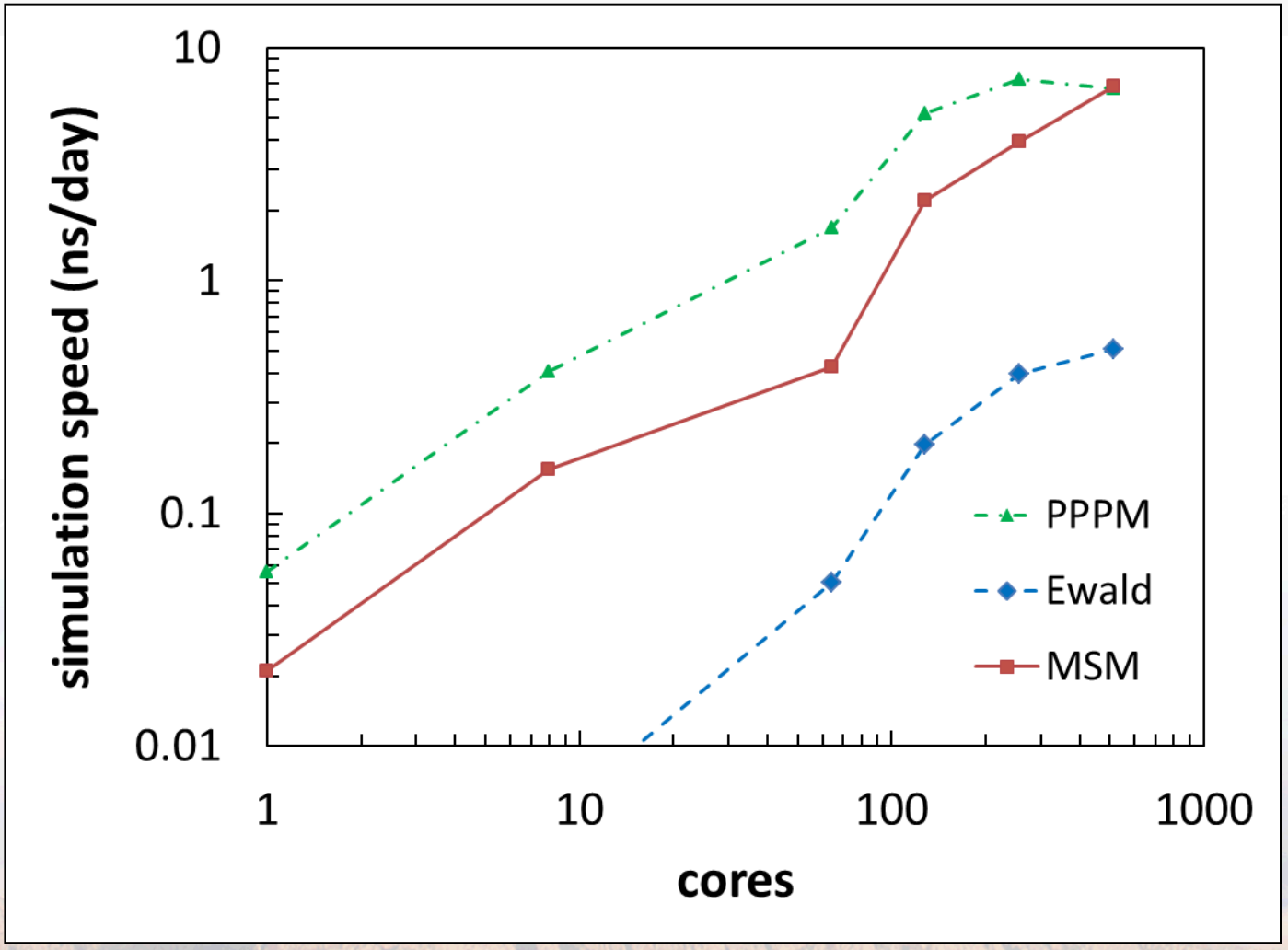
- PPPM is fastest at low core count
- MSM is fastest at high core count
- MSM scales better than PPPM since it doesn't rely on FFTs
- Maxes out at 26 ns/day



10 Angstrom cutoff, default parameters, MSM order 10

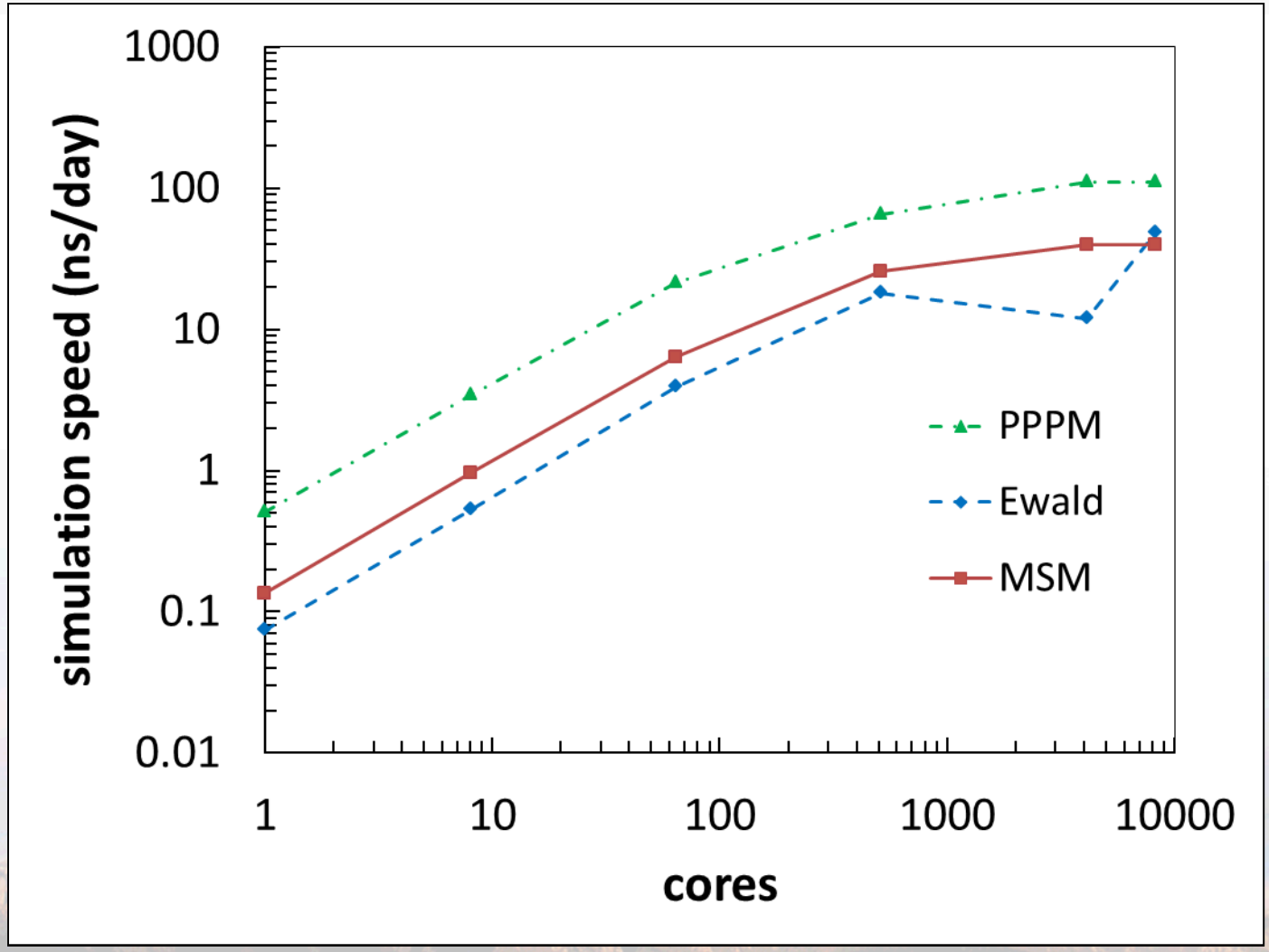
Redsky, 256k atoms, 10^{-4} accuracy

- 8x as many atoms as before
- Ewald chokes
- PPPM FFT bottleneck pushed out to larger core count
- Scaling: MSM O(N), PPPM O(N log(N)), Ewald O($N^{1.5}$)



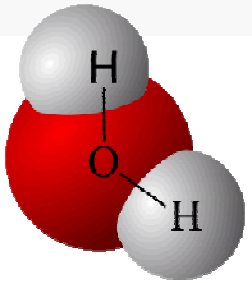
Chama, 32k atoms, 10^{-4} accuracy

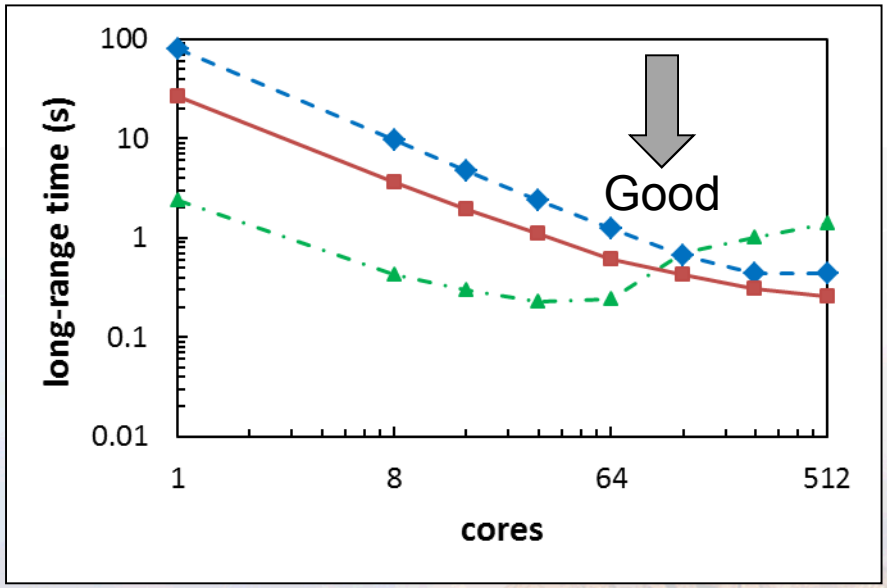
- Chama gives better overall performance
- PPPM wins, but we can't expect more scaling
- Maxes at 110 ns/day

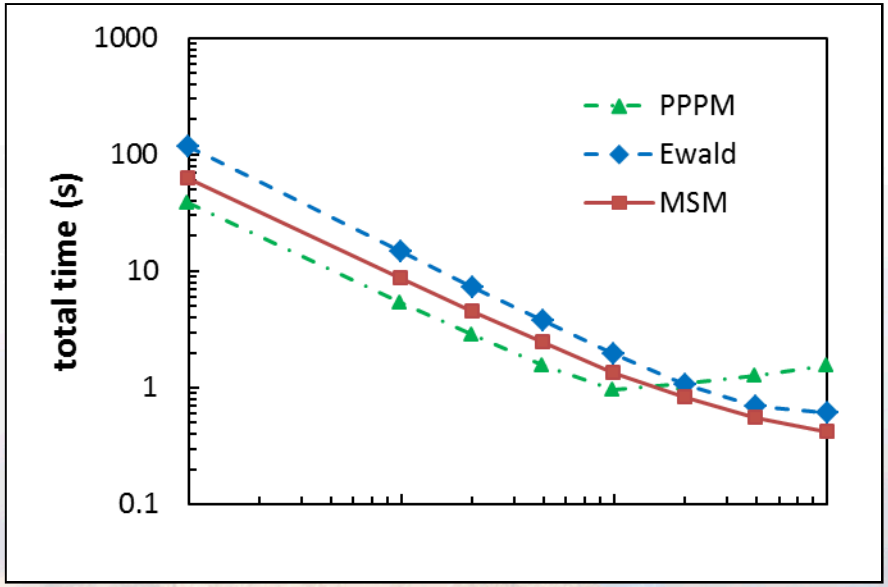


SPC/E Water Benchmark

- 36,000 atoms (strong scaling on Redsky)
- NVT, pressure computed every 50 timesteps
- 1e-3 accuracy

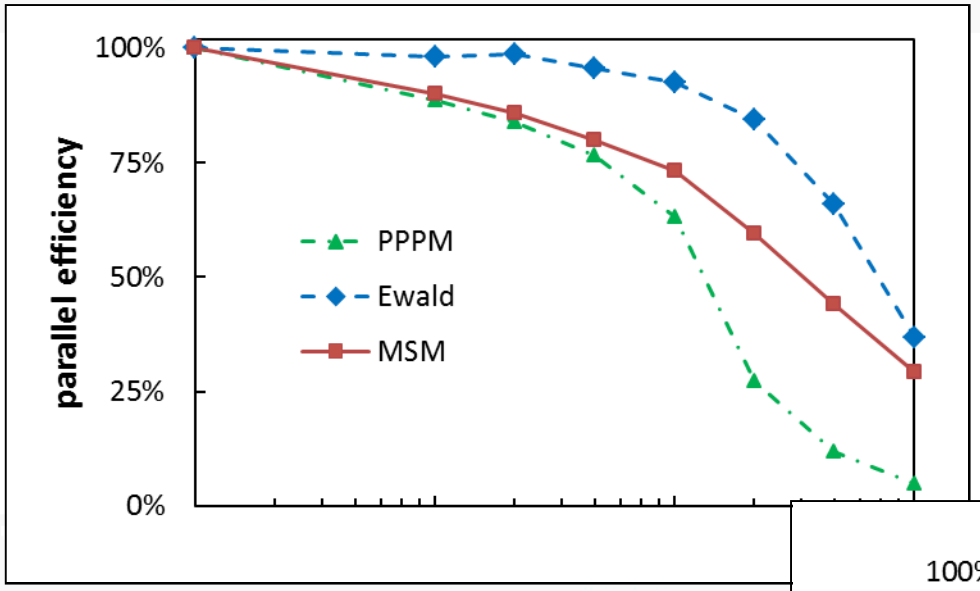


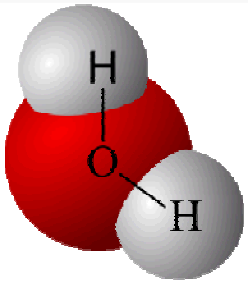


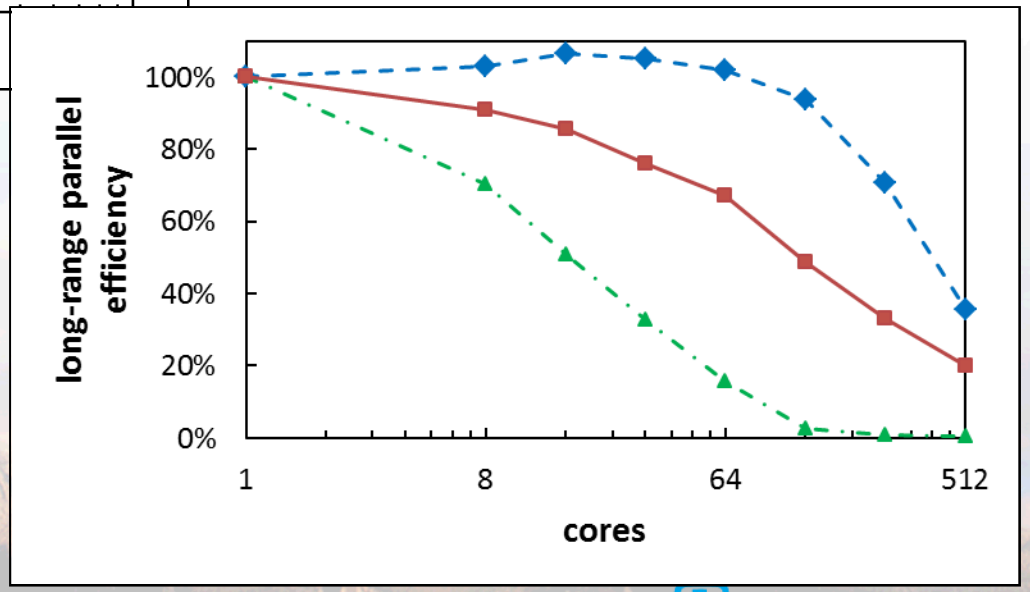


10 Angstrom cutoff, default parameters, MSM order 8

SPC/E Water Benchmark







Other Enhancements to LRE in LAMMPS

- Per-atom energy/virial for kspace
- Compute group/group for PPPM and Ewald
- Triclinic for kspace
- Ewald/disp for point-dipoles
- Staggered PPPM (up to 4x faster for high accuracy)

Good for PPPM on large core counts:

- Fix verlet/split
- 2 FFT PPPM

Conclusions

- **MSM works well for:**
  - large core counts where many-to-many communication is expensive
  - non-periodic BCs
  - lower accuracy (1e-4 and below)
  - large numbers of atoms (at least in theory due to better O(N) scaling)
- **Ewald works well for small systems**
- **PPPM is good for many practical systems and is hard to beat**

Thank You

Questions or Comments?

Sandia National Laboratories