Adaptive Gridding in Parallel with the SPARTA DSMC Code

SAND2015-7763C

Steve Plimpton & Michael Gallis
Sandia National Labs
sjplimp@sandia.gov, magalli@sandia.gov

DSMC 2015 Conference
Sept 2015 - Kapaa, Hawaii

Center for Computing Research

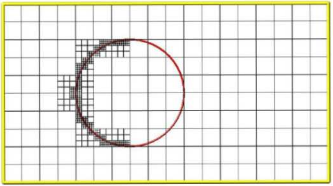
Sandia National Laboratories

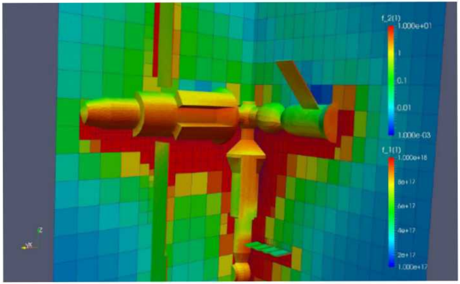
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

National Nuclear Security Administration

SPARTA

- C++, open source (Jul 2014): **http://sparta.sandia.gov**
- Short course slides: **Tutorials link** on web page
- 2d/3d, **hierarchical Cartesian grid**
- Triangulated surfaces cut/split grid cells
- Runs on large-scale parallel machines



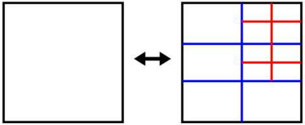


Motivation for adaptive gridding

- Ideal grid:
  - user chooses ρ and Fnum
  - every grid cell: $\sim$10 particles and $\Delta x \simeq 2\lambda$
  - good trade-off between **accuracy** and **computational cost**
- Two flavors of adaptivity: **adapt_grid** and **fix adapt**
  - static = adapt between runs; equilibrate, adapt, steady-state
  - dynamic = adapt every N steps during a run
- Why **on-the-fly** adaptivity?
  - accurately track large density variations in flow
  - optimize grid structure or particle/cell counts more quickly
  - balance accuracy/CPU trade-off even for transient flows

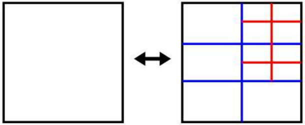
Refinement and coarsening criteria

- **Hierarchical** grid
  - top-level is single grid cell = simulation box
  - each parent cell has variable Nx by Ny by Nz child cells
  - recurse as many levels as desired (64-bit cell IDs)
  - oct-tree is 2x2x2 case, up to 15 levels
- Each child or parent cell considered **independently**



Refinement and coarsening criteria

- **Hierarchical** grid
  - top-level is single grid cell = simulation box
  - each parent cell has variable Nx by Ny by Nz child cells
  - recurse as many levels as desired (64-bit cell IDs)
  - oct-tree is 2x2x2 case, up to 15 levels
- Each child or parent cell considered **independently**



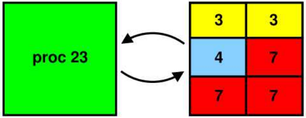
- **Allowed criteria**
  - any per-grid value, current or time-averaged
  - geometric: nearness to downwind surface
  - flow: number of particles in cell,
    mean-free-path $= \lambda = \{\sqrt{2}\pi D_{\mathrm{ref}}^2 n (T_{\mathrm{ref}}/T)^{\omega-1/2}\}^{-1}$

Parallel issues

- **Refinement** is a **local** operation
  - proc that owns a child cell can decide to refine, create new cells
  - all procs can do this without communication
- **Coarsening** may **not be local** operation
  - if one proc owns all children, then local
  - if multiple procs own children, communication needed
- Use **rendezvous** algorithm for non-local coarsening
  - owner of each child may not know who owns other children
  - everyone knows rendezvous proc, send child info to it

Parallel issues

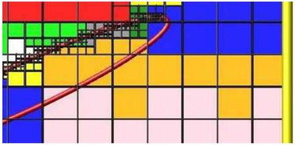
- **Refinement** is a **local** operation
  - proc that owns a child cell can decide to refine, create new cells
  - all procs can do this without communication
- **Coarsening** may **not be local** operation
  - if one proc owns all children, then local
  - if multiple procs own children, communication needed
- Use **rendezvous** algorithm for non-local coarsening
  - owner of each child may not know who owns other children
  - everyone knows rendezvous proc, send child info to it



- **Rendezvous processor**
  - owner of parent cell, assigned in round-robin fashion
  - gathers info from all children, decide whether to coarsen
  - communicates result to all procs owning child cells

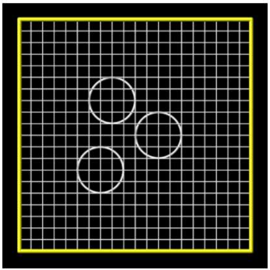
Post-adaptation operations

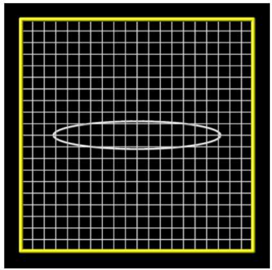
- **Cut/split** each new child cell with subset of surf elements
  - for refinement, only surfs in parent cell
  - for coarsening, union of child cell surfs
  - fast via Schwartzentruber (3d) and Weiler/Atherton (2d) algs, *Zhang & Schwartzentruber, Comp & Fluids, 2012*
- Re-build grid cell **data structures**
  - re-acquire ghost cells from neighboring processors
  - comm can cost more than adapt

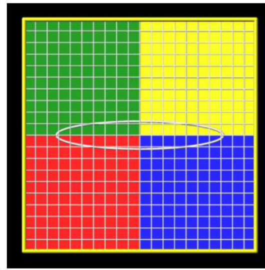


- Reset grid-based **BC** and stats
  - emission of particles from box faces
  - time ave of grid cell quantities
- **Load re-balance** to re-assign grid cells to processors
  - optional (up to user)
  - more communication of grid cells and particles

Simple 2d examples

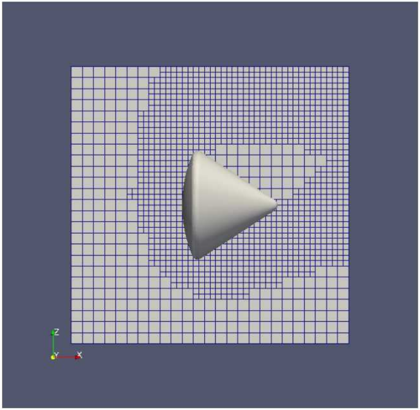






3d flow around Apollo capsule

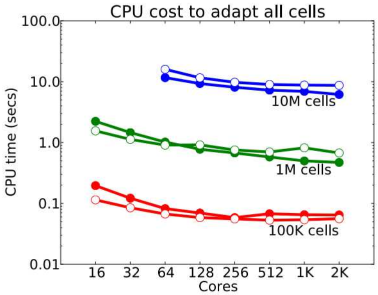
- 74 million particles, initial coarse grid $= 25^3 = 16$K cells



- Final 5-level refined grid = 7.2 million child cells
- Uniform grid at fully refined scale $= 800^3 = 512$M cells

CPU cost to adapt

- Linux cluster, node = dual 8-core SB CPUs, Infiniband
- **Stress test**: run, refine all cells, run, coarsen all cells



- Only 3-4x speed-up on 2K cores, due to **communication**
- Timestep cost: ~20 steps on 16 to ~500 steps on 2K,
  timestep speed-up is **super-linear** to 2K cores
- **Bottom line**: only a few seconds to adapt large grids

Future work

- More general **refinement rules**:
  - combined criteria: λ with minimum particle count
  - refine multiple levels at once,
    Gao, Zhang, Schwartzentruber, J Spacecraft & Rockets, 2011
- **Time average grid stats** across refine/coarsen events
  - requires additional interpolation & summation
- **Distributed storage** of parent cells
  - currently each processor owns copy of all parents
  - fine for few levels, not scalable to many-level oct-tree
  - limits size of fully adapted grid

Thanks and links

- Funding support: DOE/NNSA ASC program
- Management support: **Dan Rader** in particular

- **http://sparta.sandia.gov**
- SPARTA short course:
  http://sparta.sandia.gov/**tutorials.html**

- SPARTA **papers**:
  - Gallis, et al, *DSMC: The Quest for Speed*,
    Proc of 29th RGD Symposium, 2014.
  - Gallis, et al, *DSMC of Richtmyer-Meshkov instability*,
    Physics of Fluids, 27, 084105 (2015).