Gasoline

Treecodes for Cosmology

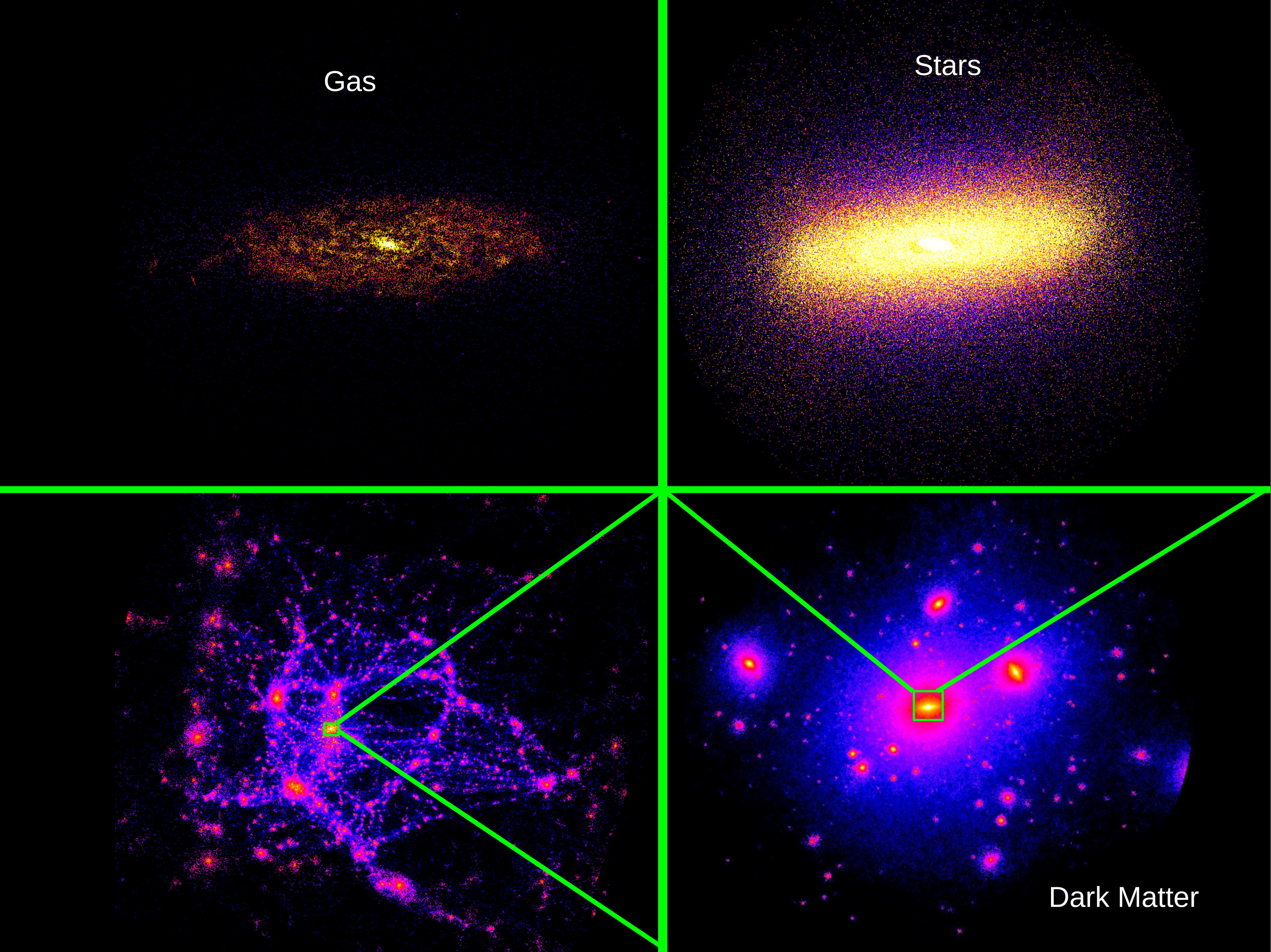
Thomas Quinn
University of Washington
N-Body Shop

Outline

- Motivation
- Multipole Expansions
- Tree Algorithms
- Periodic Boundaries
- Time integration
- Gravitational Softening
- Parallel Architecture of GASOLINE
- SPH

Motivation:
Why I am a Tree Hugger

- Dynamic Range
  - Minimum Box Size: 25 Mpc
  - Maximum Resolution: 1 kpc
  - 15 trillion grid cells!
- Geometry
  - Uniform: Large Scale
  - Spherical: Halos
  - Planar: Disk
  - All in one simulation!

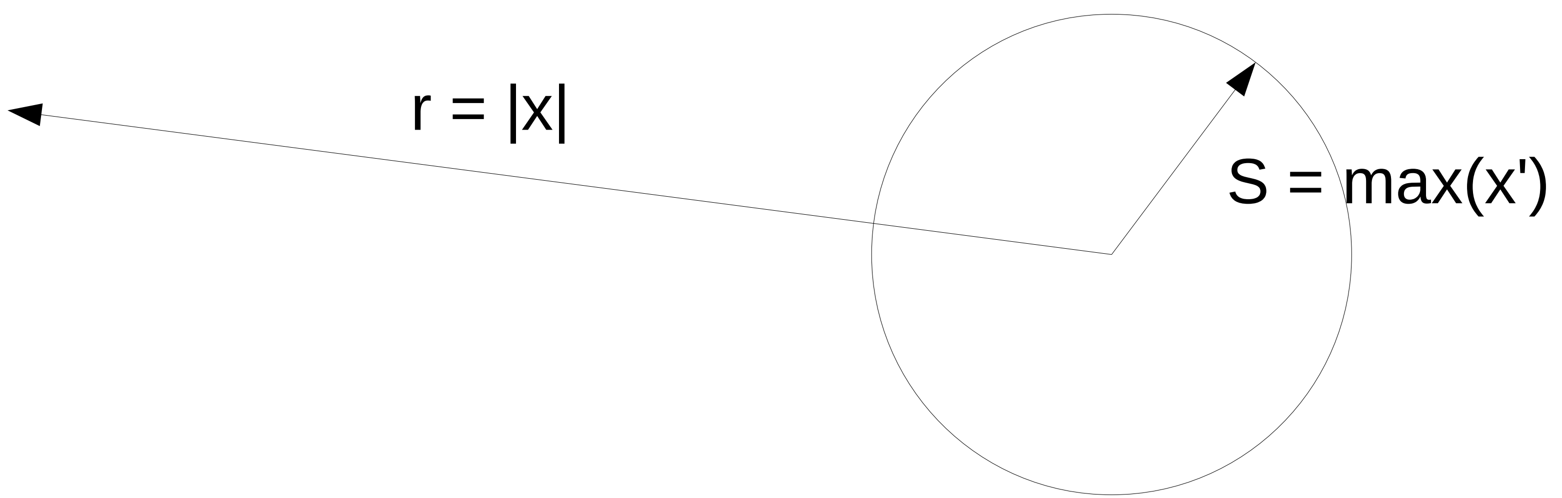

Gas
Stars
Dark Matter

Multipole Expansions

- Poisson Equation: $\nabla^2\Phi = 4\pi G\rho.$
-
- Integral Form: $\Phi(\mathbf{x}) = -G\int \frac{\rho(\mathbf{x}')}{|\mathbf{x}'-\mathbf{x}|}\mathrm{d}^3\mathbf{x}'$
- Using N-bodies: $\rho(\mathbf{r}) = \sum_{i\in V}\delta(\mathbf{r}_i - \mathbf{r})m_i$
-
- (but note softening issue)
- We naively have: $\Phi(\mathbf{x}_i) = -G\sum_j^N \frac{m_j}{|\mathbf{x}_j - \mathbf{x}_i|}$
- Expensive!

Multipole Expansions

- For x >> |x'|, $$\Phi(\mathbf{x}) = -G \sum_{lm} C_{lm} \frac{Y_l^m(\theta, \phi)}{r^{(l+1)}}$$
- Terms scale as
- $max(\mathbf{x}')^l / r^{l+1}$
- Hence “theta” = s/r is a measure of accuracy



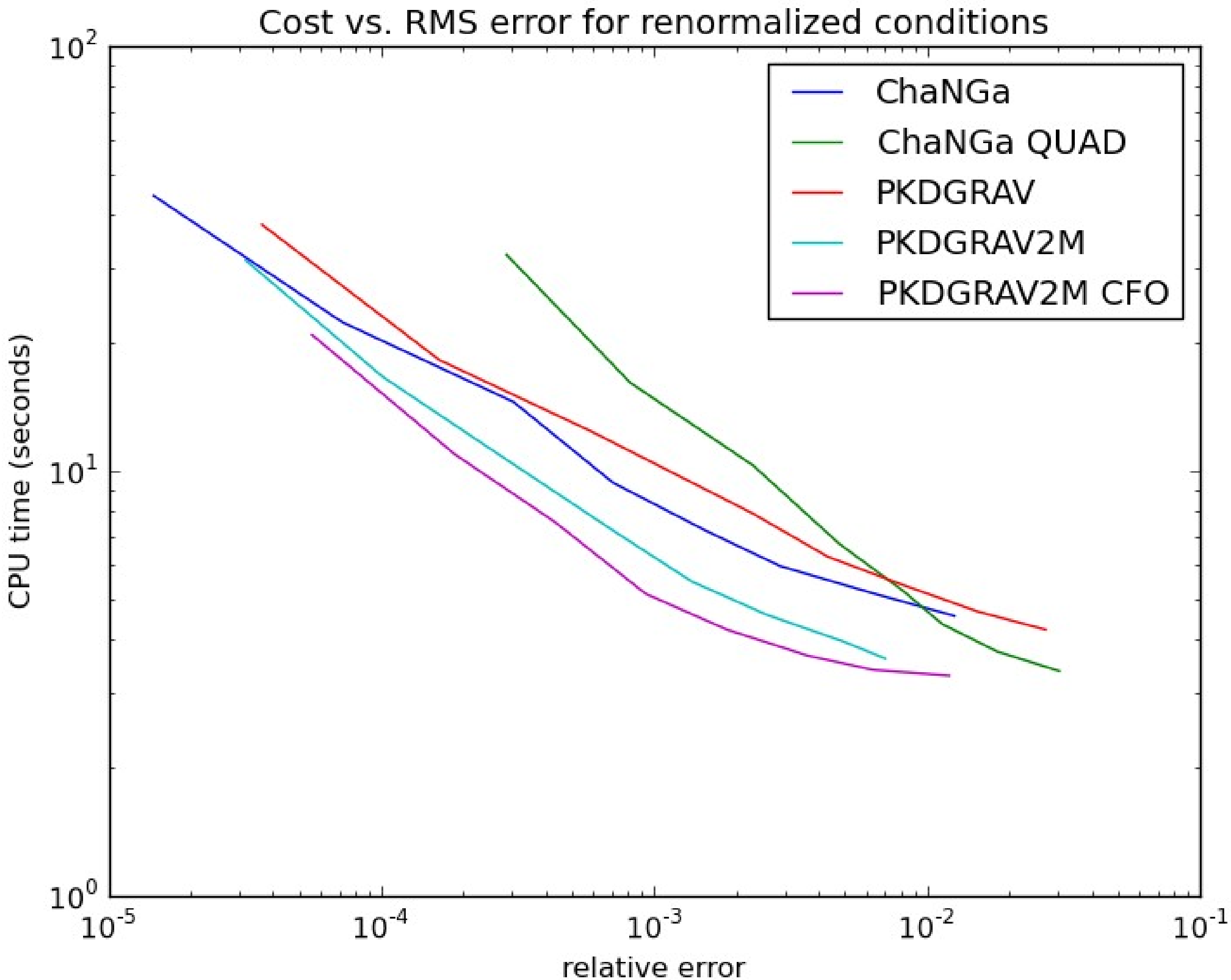
Multipole Moments

- In Cartesian coordinates:
- $$\Phi(\mathbf{x}) = -G \int d^3x' \left[\frac{\rho(x')}{|x|} + \frac{x'\rho(x') \cdot x}{|x|^3} + \frac{(3x'_i x'_j - |x'|^2 \delta_{ij})\rho(x') x_i x_j}{2|x|^5} + \cdots \right.$$
- More complicated, but quicker
  - For large x, forces due to all particles in x' evaluated with a few operations.
- Local expansions are also possible (FMM):

$$\Phi(\mathbf{x}) = -G \sum_{lm} L_{lm} Y_{lm}(\theta, \phi) r^l$$

See Greengard 1988 thesis

Multipole Accuracy Comparison

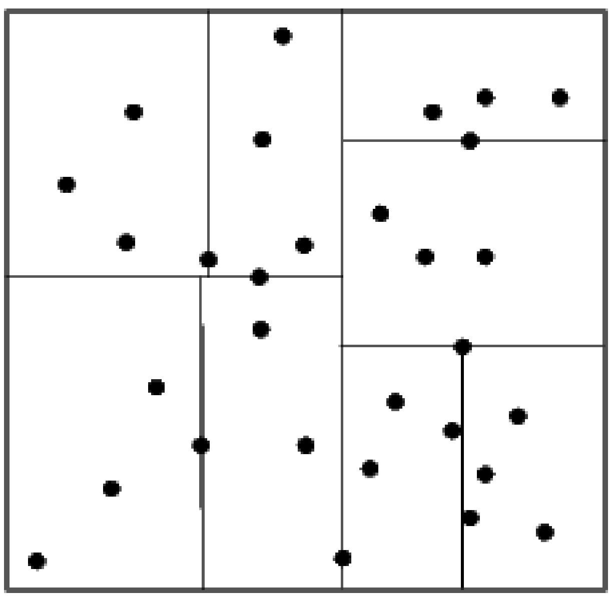


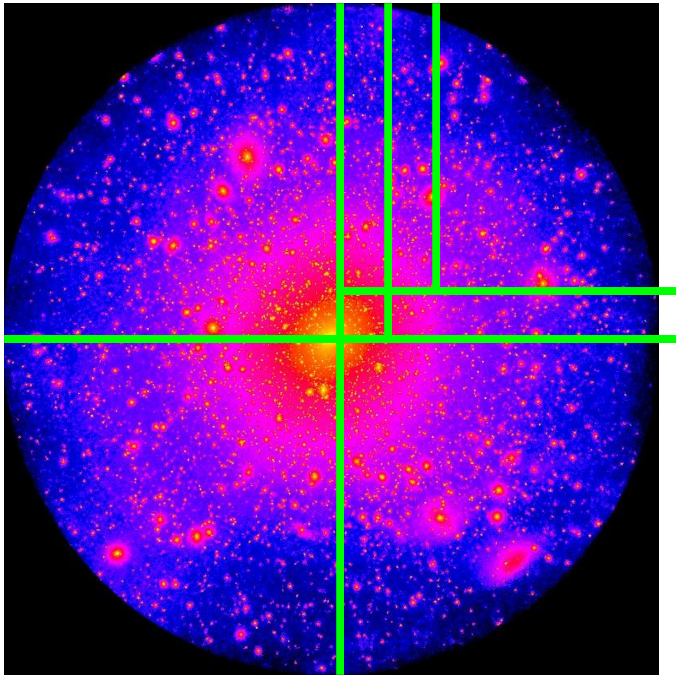
Trees: divide et impera

- “... most important nonlinear structures in computer algorithms.” (Knuth ACP)
- A root node + disjoint set of subtrees
- Leaf: a node with no subtrees
- Many representations:
  - Node + child pointers
  - Node + parent pointers
  - heap
- Divide and conquer a common paradigm:
  - e.g. Quicksort.

Spatial Trees

- Orthogonal Recursive Bisection
  - Split the median particle: a balanced tree (k-d tree)
  - Guaranteed maximum depth



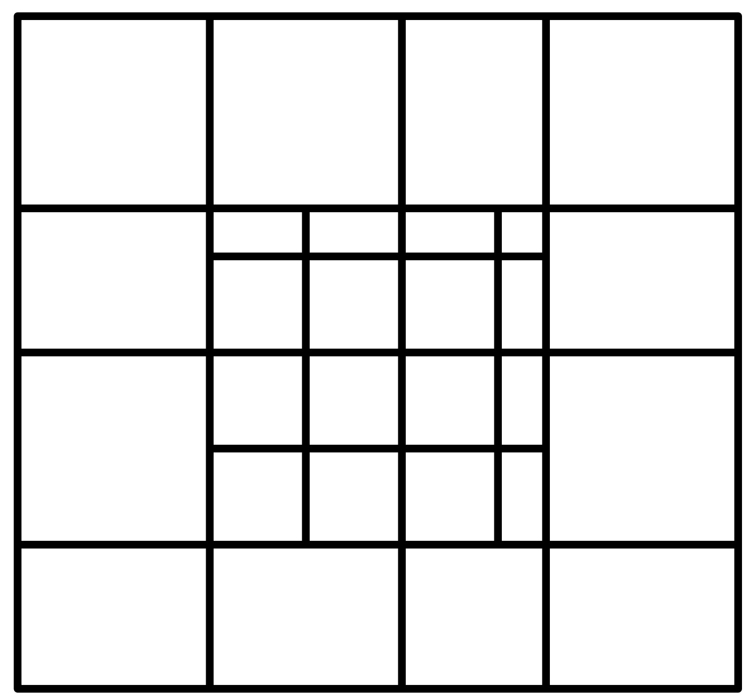


Balanced Tree Representation

- Array of nodes: nodes[N]
- Index of first child of node n = 2*n
- Index of second child of node n = 2*n + 1
- Index of parent of node n = n >> 1
- No pointers or recursion needed for traversal

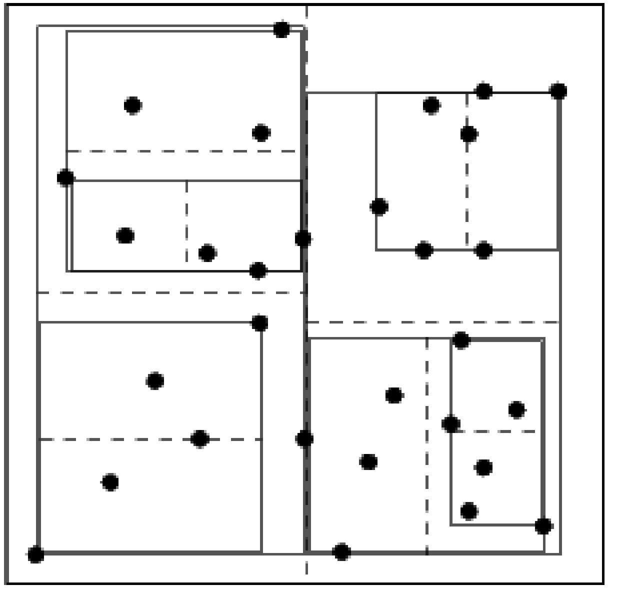
Oct Tree (or Barnes-Hut)

- See Barnes & Hut, 1986, Nature
- Split cubical node into 8 equal cubes
- Nodes are small and nearly spherical
  - Small multipole moments
- Not balanced



Spatial Binary Tree

- Bisect longest dimension of bounding box
- Split to bisect space
- Not balanced
- Adaptable to disks

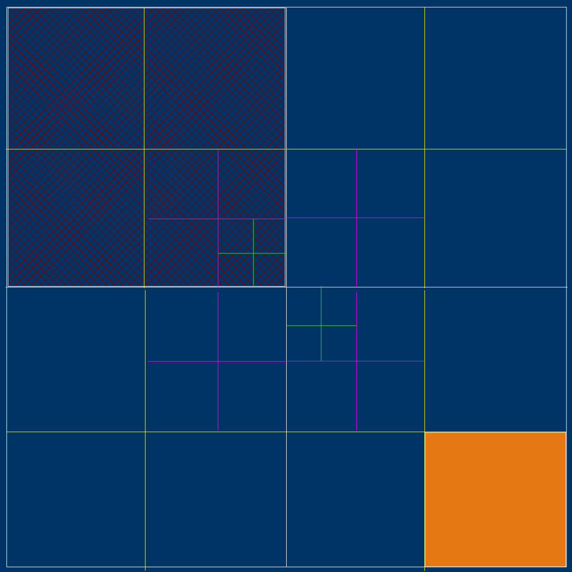


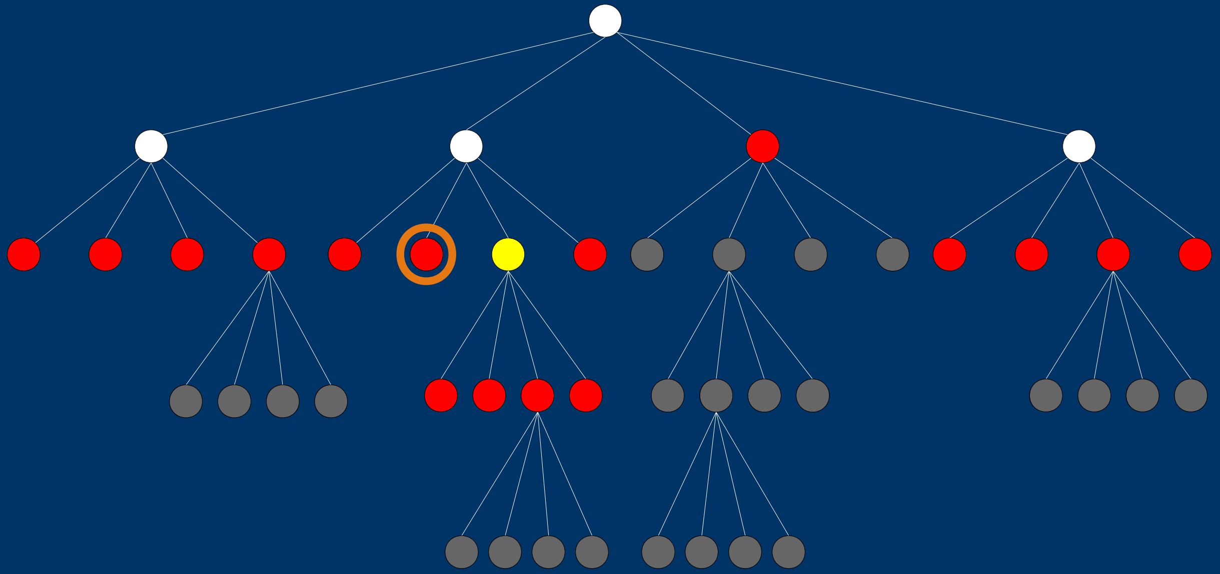
Gravity Algorithm

- Hierarchically divide space into cells
- The force on a particle is its interaction with a cell
- If the multipole expansion is valid then the interaction is the multipole expansion
- If not, the interaction is the sum of the interactions with the cells children.
- If cell is a leaf (or “bucket”) interact with the contained particles.

Basic algorithm ...

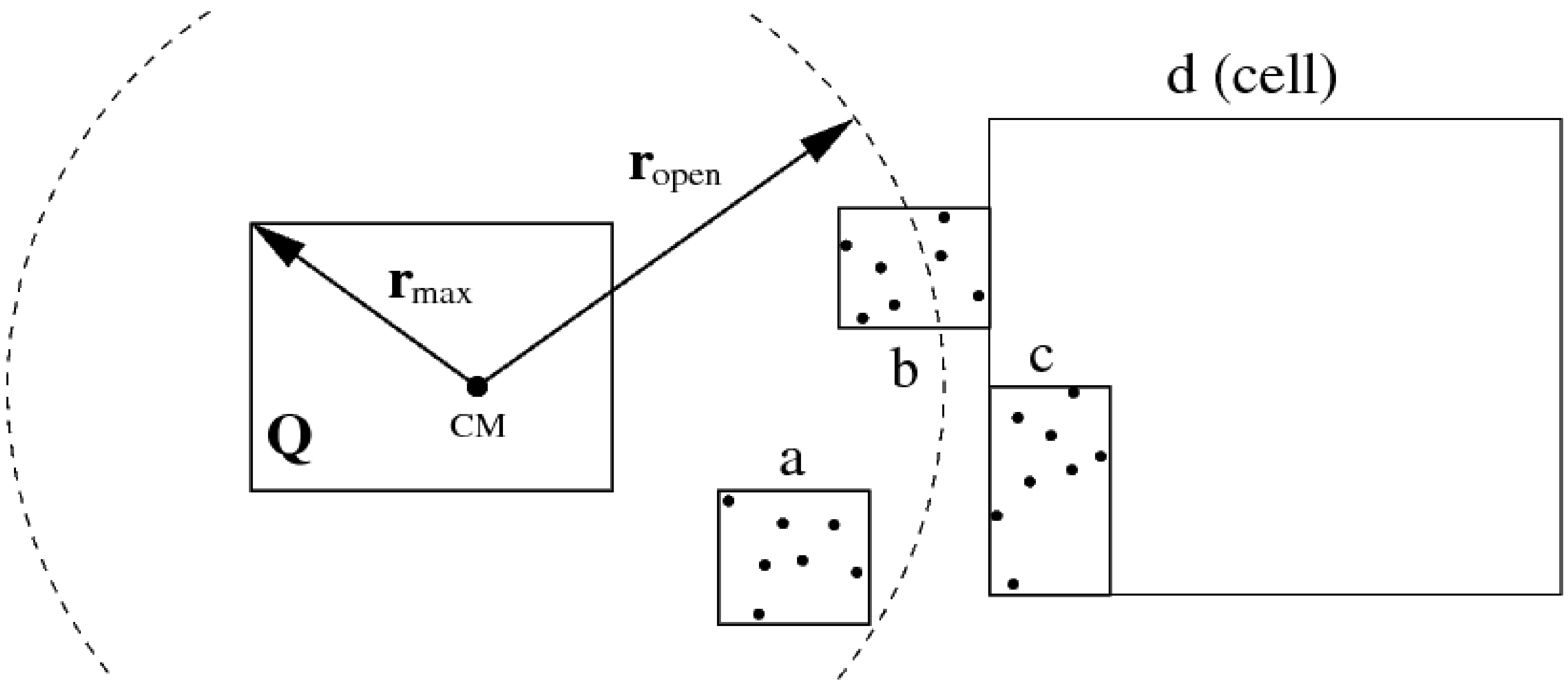
- Barnes-Hut approximation: O(*n*log*n*)
  - Influence from distant particles combined into multipole moment
  - Traversal stops when multipole is accurate





The opening criterion

- An open radius is only calculated once for each cell: r_open ~ r_max/theta



Efficient Walking

- Walk a high level node
  - Determine all cells with acceptable expansions
  - These cells are also acceptable for all children
  - The other cells need to be checked when the node's children are walked.
- Go through check list with child nodes
- Construct final interaction list at bucket level.

Neighbor finding walk (SPH)

- Use a priority queue (loser tree) to track distance to k-th nearest neighbor
- Start at bucket containing particle.
- Search parent cell for uncles that overlap search ball
- Descend each child of uncle which overlaps search ball
- Replace particles in priority queue with found particles.

Inverse Neighbor Finding

- Which particles consider me a neighbor?
- Search for particles with a smoothing length that encloses me.
- Cells have bounding boxes of all smoothing radii of the particles they contain
- Walk similar to neighbor finding.
- Useful when Nactive << Ntotal

Periodic Boundaries

- For shells in an infinite Universe:
  - $F(r) \sim GM(r)/(r^2) \sim$ constant at each radius
  - Sum will not converge!
- TreePM
  - Calculate long distance forces using Mesh and FFT
- Ewald Summation
  - Split the sum into 2 converging parts.

TreePM (Gadget)

- FFT is fast (+)
- FFT has lots of communication (-)
- Available fast libraries: FFTW (+)
- Inaccuracies at the FFT grid scale (-)
  - Transition between grid and tree
- Tree walk is only local (+)

Ewald Summation

- Split Green's function:
- $$\frac{1}{r} = \frac{\operatorname{erfc}(\alpha r)}{r} + \frac{\operatorname{erf}(\alpha r)}{r}$$
- Modified Green's function:
- $$g(\mathbf{r}) = \frac{\pi}{\alpha^2 L^3} - \sum_{\mathbf{l}} \frac{\operatorname{erfc}(\alpha|\mathbf{r} - \mathbf{l}L|)}{|\mathbf{r} - \mathbf{l}L|} - \sum_{\mathbf{h} \neq \mathbf{0}} \frac{1}{h^2 \pi L} \exp\left(-\frac{\pi^2 h^2}{\alpha^2 L^2}\right) \cos\left(\frac{2\pi}{L}\mathbf{h} \cdot \mathbf{r}\right)$$
- Store differences between this and 1/r in large array.
- Perform lookup for each force calculation.

Ewald Reduced Cell Multipole

- Perform walk over fundamental cube an a number of replicas (e.g. 26 neighbors)
- Calculate forces due to Ewald sum of multipole moments of root cell of fundamental cube.
- Spatial sum modified to avoid double counting.
- Algebraically complex
- Sum is somewhat expensive (-)
- Efficient in parallel (+)

Force Calculation Overview

- Build Tree (top down)
- Calculate multipole moments and opening radius (bottom up)
- Walk nodes then buckets, constructing interaction lists
- Calculate force on particles in a bucket using interaction lists
- Calculate Ewald sum using root multipole.