Treecodes for Proto-Planetary Disks

Thomas Quinn
University of Washington
N-Body Shop

Outline

- Motivation
- Gravity
- Multipole Expansions
- Tree Algorithms
- Time integration
- SPH

N-body: what is N?

- Continuous distribution represented by particles
- Particles are a sample of (mass) distribution
  - f(x, v, t)
  - Explicit conservation of mass
- Even for particulate matter (dust, dark matter)
  - N(simulation) << N(physical)

Why I am a Tree Hugger

- Dynamic Range
  - Molecular cloud core size: ~ .1pc
  - Hill radius of Earth: ~ .01 AU
  - 1e19 grid cells!
- Geometry
  - Spherical: Cloud core
  - Planar: Proto-planetary Disk
  - Spherical again: planets
  - Binary stars ...

Gravity

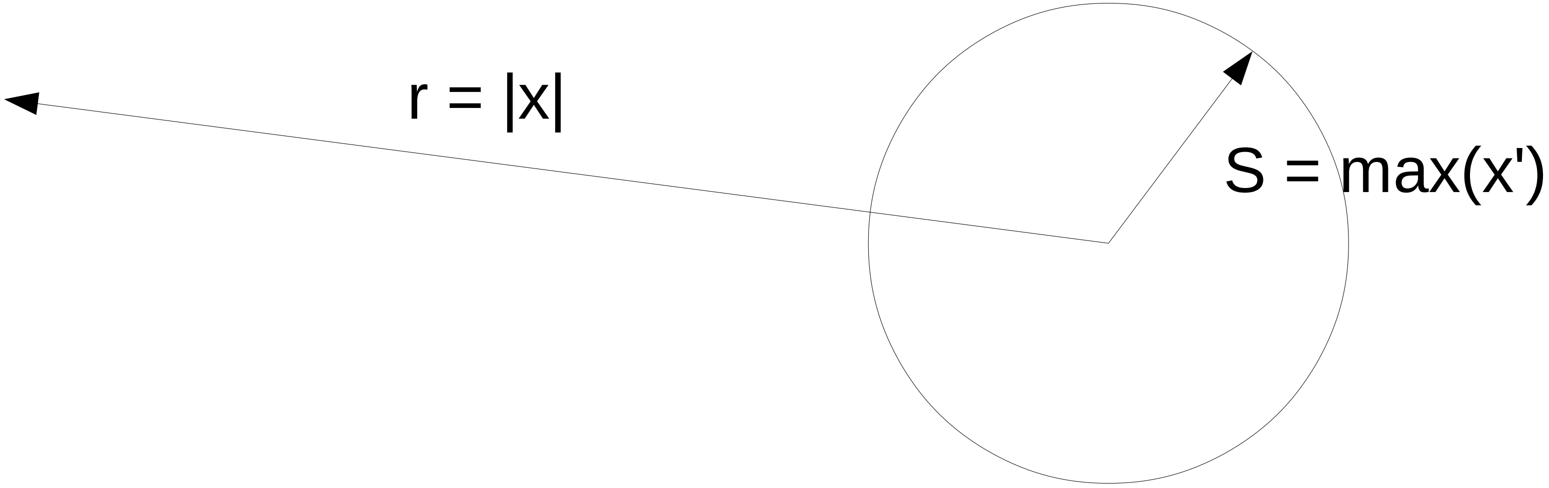
- Poisson's equation: $\nabla^2\Phi = 4\pi G\rho.$
- Elliptical PDE: solution depends on all boundaries
- Linear: if eigenfunctions can be found then solution can be constructed by expressing density in terms of these functions:
  - E.g., Fourier Transforms, Spherical harmonics
- But these can't be (easily) constructed for arbitrary points in space.

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/smoothing 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



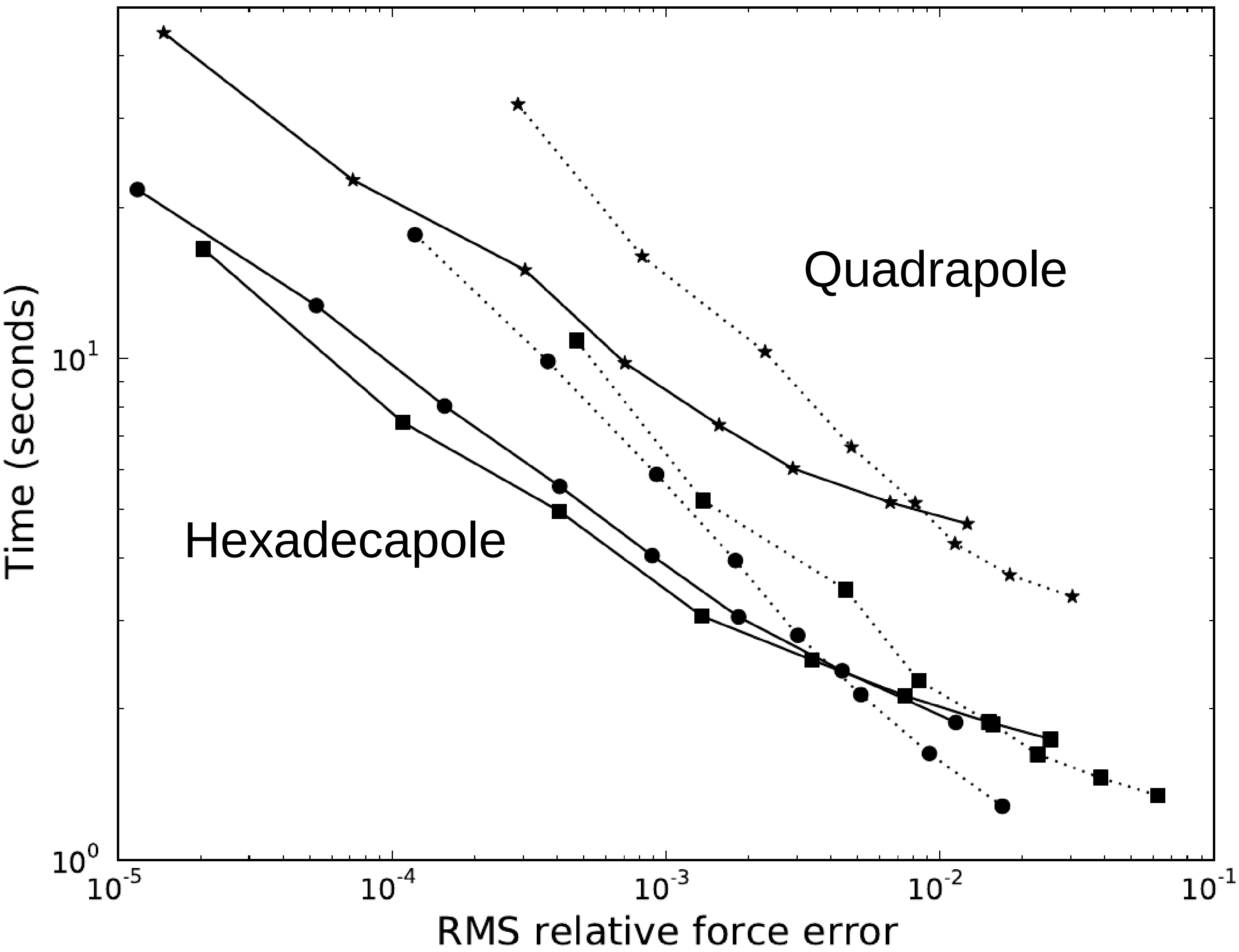
Multipole Moments

- In Cartesian coordinates:
- $\Phi(\mathbf{x}) = -G\int d^3x' \left[\frac{\rho(x')}{|x|} + \frac{x'\rho(x')\cdot x}{|x|^3} + \frac{(3x'_i x'_j - |x'|^2\delta_{ij})\rho(x')x_i x_j}{2|x|^5} + \cdot\right.$
- 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

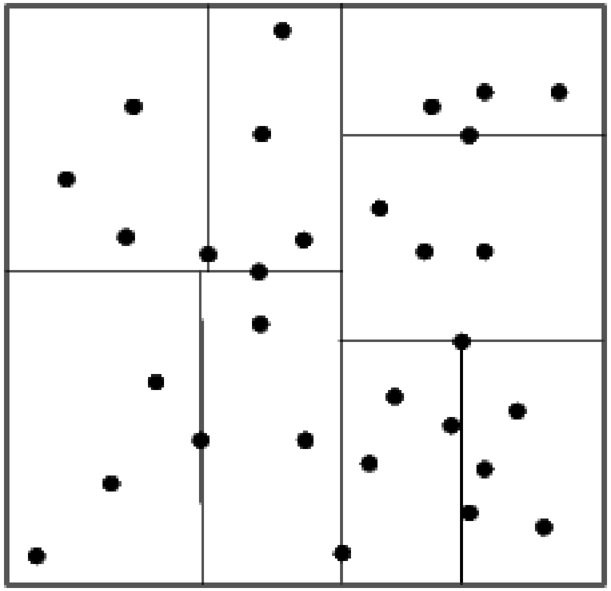


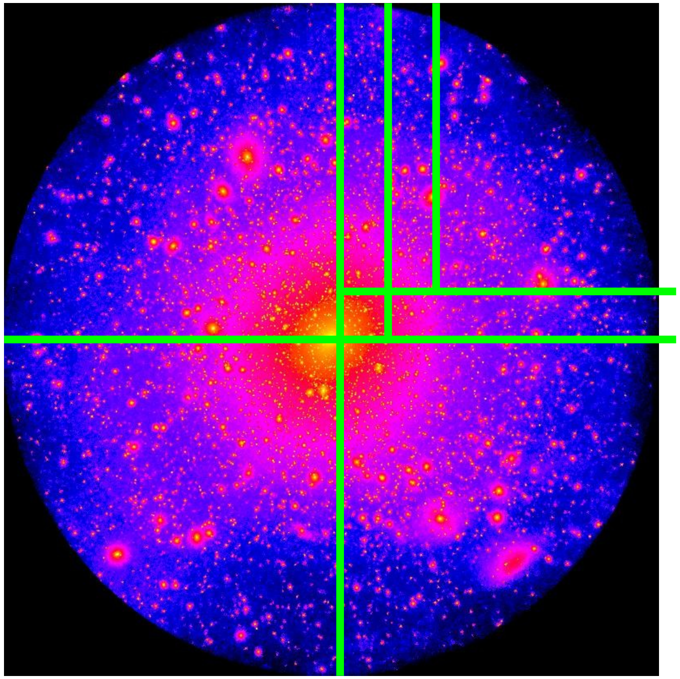
Trees: divide et impera

- “... 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



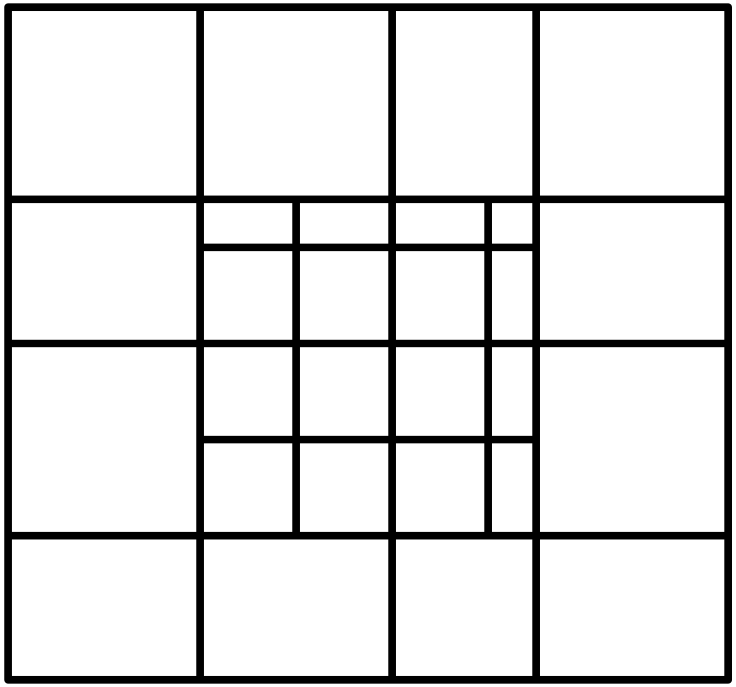


Balanced Tree Representation

- 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

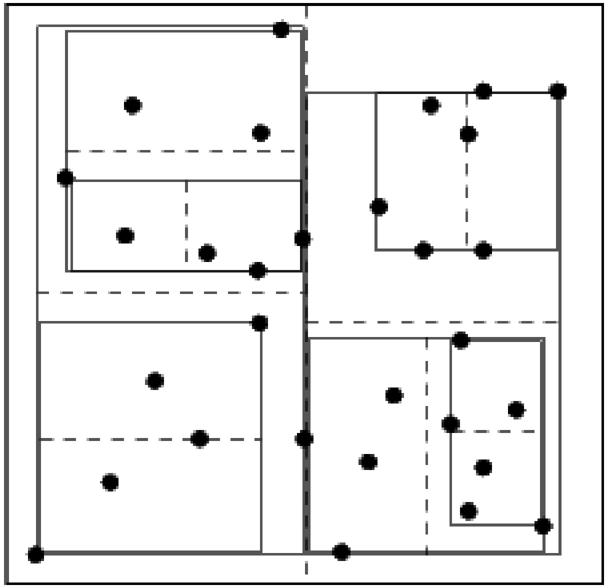
Oct Tree (or Barnes-Hut)

- 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
- More adaptable to disks

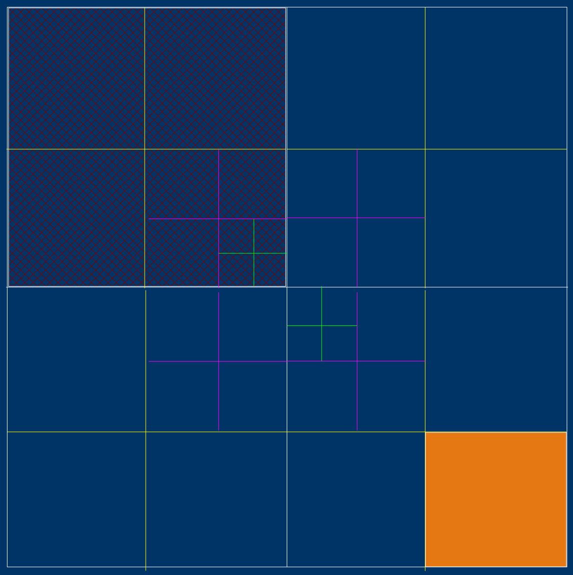


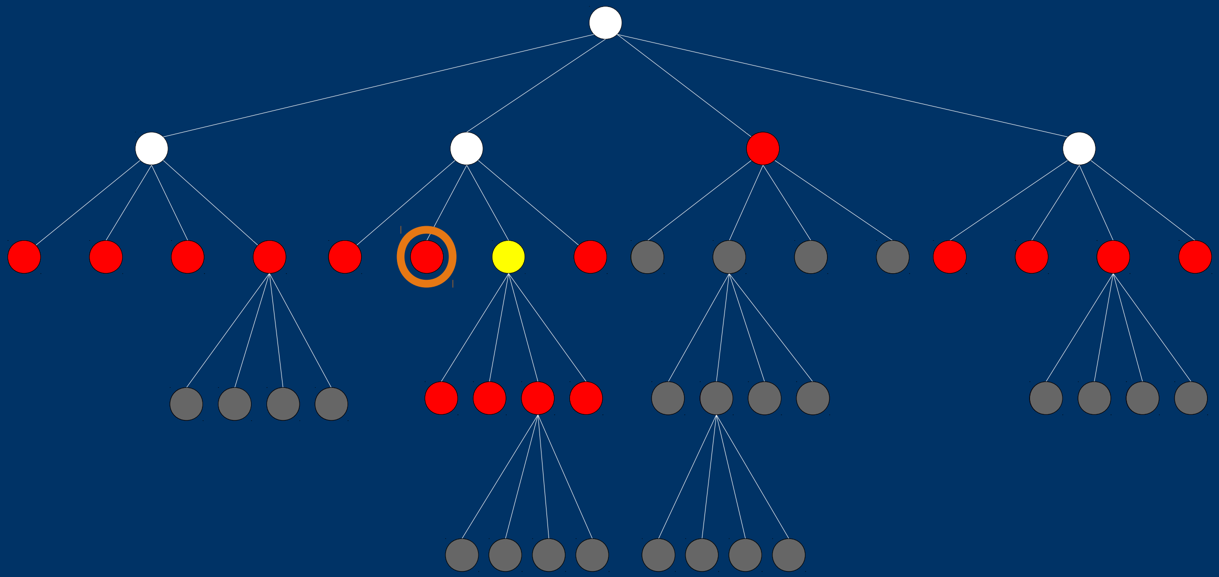
Gravity Algorithm

- 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 ...

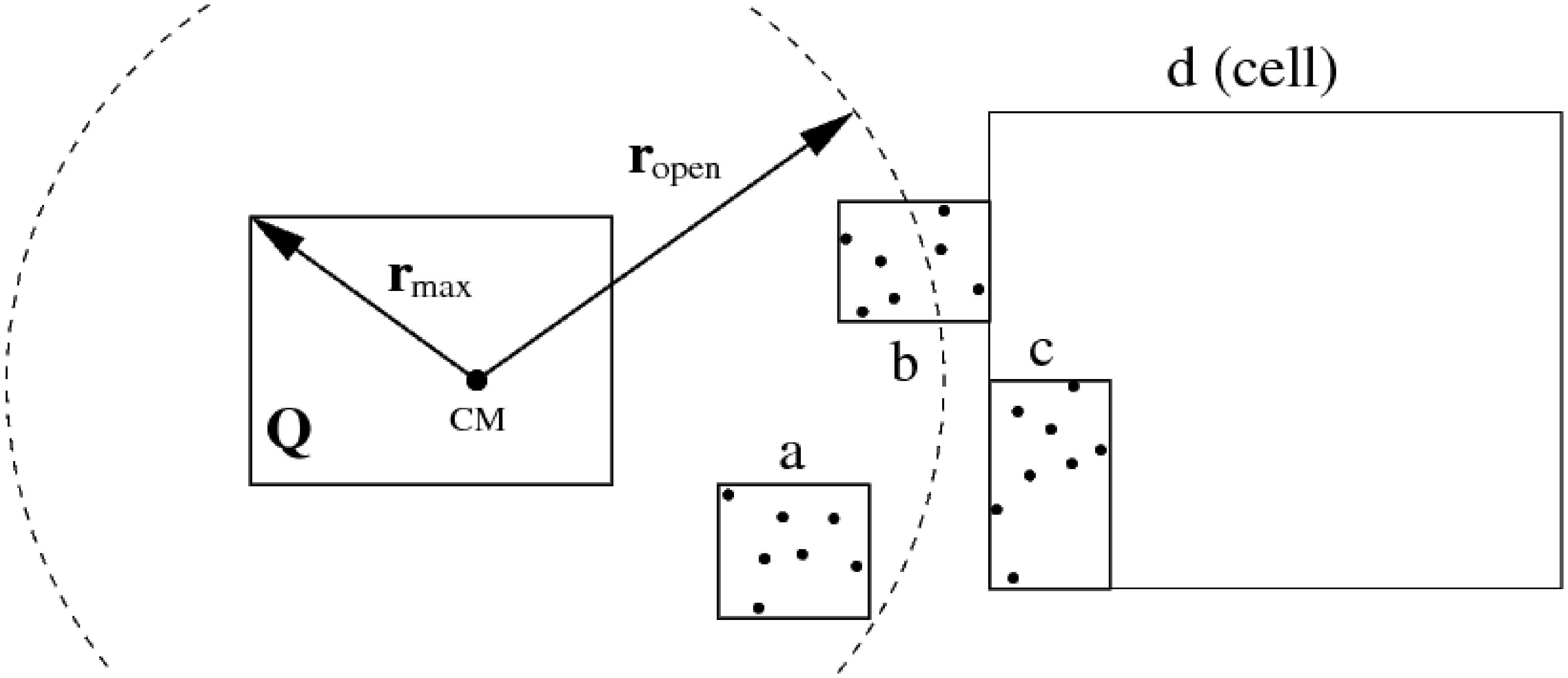
- Barnes-Hut approximation: O(*n*log*n*)
  - Influence from distant particles combined into multipole moment
  - Traversal stops when multipole is accurate





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 (heap or 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) ~ GM(r)/(r^2) ~ 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:
- $$\frac{1}{r} = \frac{\mathrm{erfc}(\alpha r)}{r} + \frac{\mathrm{erf}(\alpha r)}{r}$$
- Modified Green's function:
- $$g(\mathbf{r}) = \frac{\pi}{\alpha^2 L^3} - \sum_{\mathbf{l}} \frac{\mathrm{erfc}(\alpha|\mathbf{r} - \mathbf{l}L|)}{|\mathbf{r} - \mathbf{l}L|} - \sum_{\mathbf{h} \neq \mathbf{0}} \frac{1}{h^2 \pi L} \exp\left(-\frac{\pi^2 h^2}{\alpha^2 L^2}\right) \cos\left(\frac{2\pi}{L} \mathbf{h} \cdot \mathbf{r}\right)$$
- 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

Time stepping

- An N-body system is Hamiltonian
  - Invariant under time translation
  - Phase space density preserved
- Preserve these properties in a numerical integration by ***Exactly*** integrating an ***Approximate*** Hamiltonian
- Operator Splitting: 1st applying part of a Hamiltonian, then applying the 2nd part is equivalent to an approximate Hamiltonian

Symplectic Integrators for Planetary Systems

- Split into Kepler and Interaction Hamiltonian
- Integrate Kepler (advance mean anomaly)
- Integrate interaction (planets fixed in space, but velocities are changed)
- Mixed Variable Symplectic integrator (Wisdom & Holman, 1993)

Time Stepping

- Consider $H = H_D + H_K$
- Where $H_D = \frac{p^2}{2}; \quad H_K = \Phi(\mathbf{x})$
- Hamilton's equations give:

$$x(t) = x(t_0) + p(t - t_0); \quad p = p_0$$

$$x(t) = x_0; \quad p(t) = p(t_0) - \nabla\Phi(\mathbf{x}_0)(t - t_0)$$

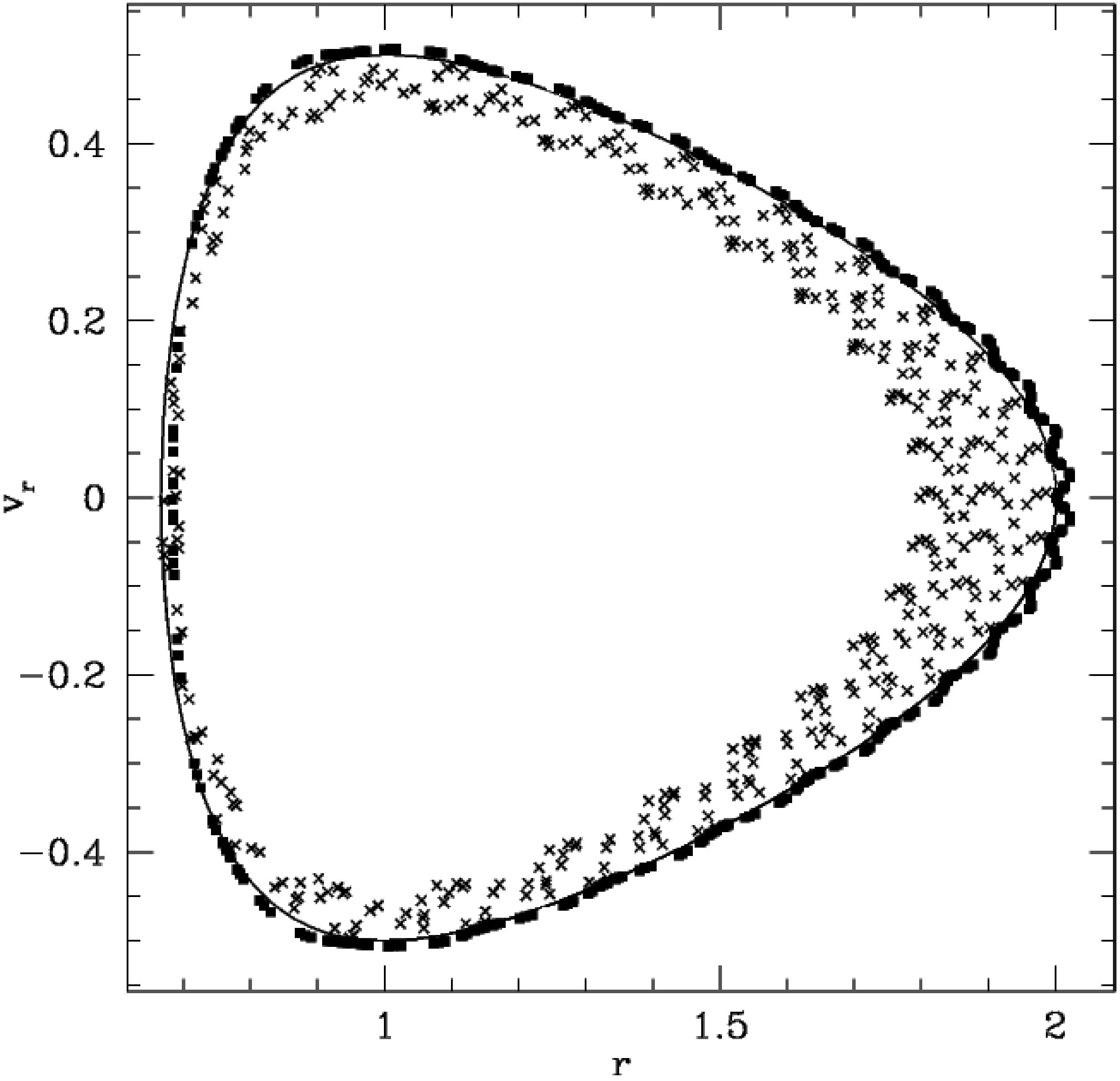
- This is Leap Frog!
- Obeys a Hamiltonian: $H_{num} = H_D + H_K + H_{err}$

Saha & Tremaine 1992

Leap Frog

- Advance positions by 1/2 timestep using old velocities (Drift)
- Advance velocities full timestep using forces from particle positions at 1/2 step (Kick)
- Advance positions by 1/2 timestep using *new* velocities (Drift)
- 2nd order in time
- Can also do Kick-Drift-Kick
- If forces are velocity dependent: need predicted velocity for the Kick

Leapfrog vs. Runge Kutta



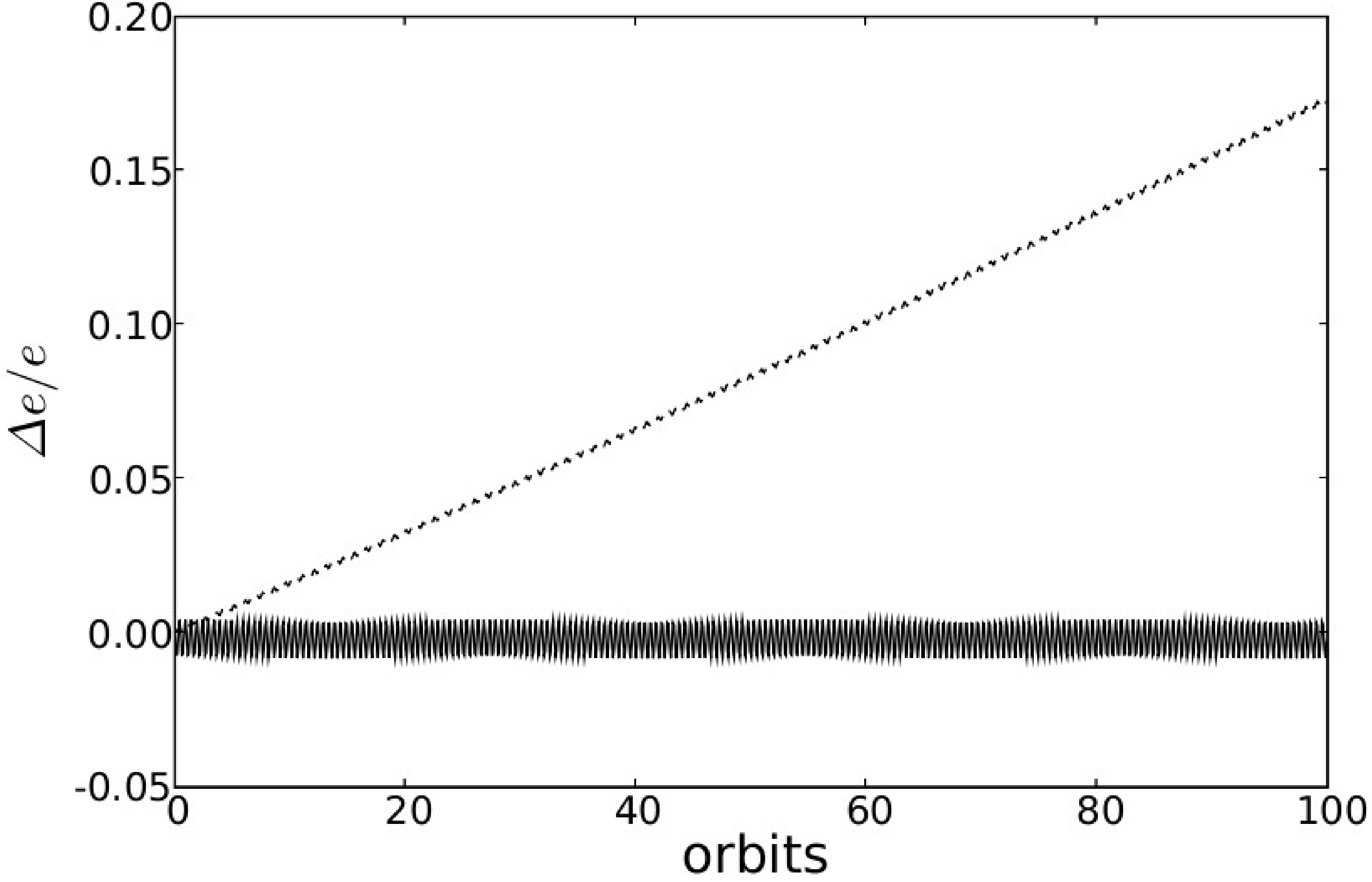
Planetismal Dynamics Example

- Local patch approximation: Wisdom & Tremaine 1988
- Hill's equations:

$$\begin{aligned}\ddot{x} - 2\Omega\dot{y} - 3\Omega^2 x &= -\frac{\partial\Phi}{\partial x}\\ \ddot{y} + 2\Omega\dot{x} &= -\frac{\partial\Phi}{\partial y}.\end{aligned}$$

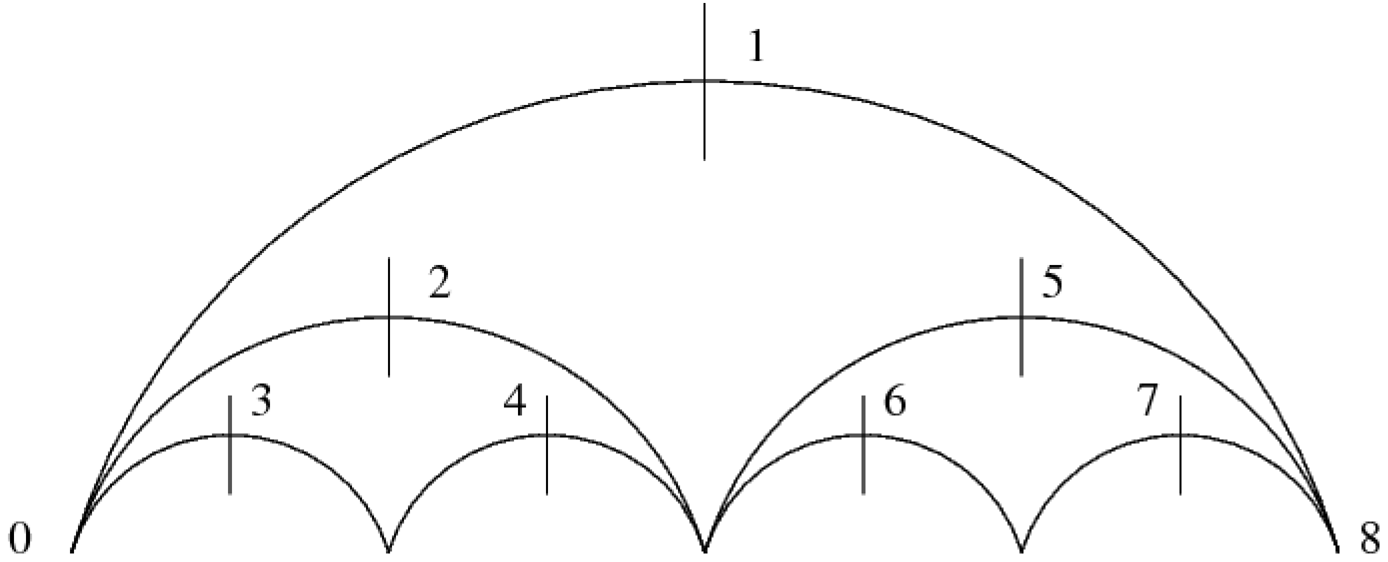
- Hamiltonian, despite v dependent forces
- Split into separately integrable parts (Quinn et al 2012)

Hill's Equations, Symplectic vs. not



Hierarchical Timestepping

- Large dynamic range in density implies large range in timescales:
- $t_{dyn} \sim \frac{1}{\sqrt{G\rho}}$
- Timesteps organized in power of 2 “rungs”



Symplectic Variable Timesteps?

- At a minimum, must be reversible
- Because of timestep decision, reversibility is expensive or impossible.
  - Trial timesteps and implicit step choices
  - Force splitting schemes
- At least minimize time asymmetries:
  - Make timestep choice where the acceleration is calculated.

    => KDK scheme
  - KDK more efficient and better momentum conservation.

Timestep Criteria

- EpsAccStep: dt ~ sqrt(softening/acceleration)
- DensityStep: dt ~ sqrt(1/density)
- GravStep: dt ~ sqrt($r_{ij}^3/(m_i + m_j)$)
- Courant: smoothing/sound speed
- Diffusion: dt ~ dx^2/D
- See M. Zemp et al 2007 for an “optimal criterion” for gravity

Gravitational Softening

- Recall: we are solving the BE, and particles sample f(z).
- $$\Phi(\mathbf{x}) = -GM \int \mathrm{d}^6 z' \frac{f(\mathbf{z}')}{|\mathbf{x} - \mathbf{x}'|}.$$
- The standard sum is a Monte-Carlo integral.
- The 1/|x – x'| term is not well suited to this.
- Introduce softening to minimize force error
- Truelove criterion: resolve Jeans' length
  - Pressure supports against gravitational collapse
- Does not effect two body relaxation time!
- Ultimately a computational cost decision

Resolving Jeans Mass

- Jeans Length

$$\lambda_J^2 = \frac{\pi v_s^2}{G\rho_0}.$$

- Jeans Mass

$$M_J = \frac{4\pi}{3}\rho_0(\frac{1}{2}\lambda_J)^3 = \frac{1}{6}\pi\rho_0\left(\frac{\pi v_s^2}{G\rho_0}\right)^{3/2}$$

- Need to resolve this, or not doing hydro!
- Note non-convergence of isothermal gas.

SPH advantages

- Naturally partners with a particle gravity code
- Arbitrary geometry
- Perfect advection
- Galilean invariant
- No intrinsic dissipation
- Fast (depending on neighbor finding)
- Easy to implement
- Flexibility with Equations of State

Basic principles of SPH

- Model the fluid as a collection of elements represented by particles
- Move particles using Lagrangian forms of the fluid equations
- Assign thermodynamic properties to the particles.
- Some properties determined by local averages
- Use an interpolation method to get these averages from local particles.

Interpolation

- The interpolant of any function *f(r)* is:

$$< f(\mathbf{r}) >= \int W(\mathbf{r}-\mathbf{r}';h)f(\mathbf{r}')d\mathbf{r}'$$

- *h* is the smoothing length and determines the extent of the averaging volume.
- *W* is the smoothing kernel which satisfies:

$$\int W(\mathbf{r}-\mathbf{r}';h)d\mathbf{r} = 1$$

$$\lim_{h\to 0} W(\mathbf{r};h) = \delta(\mathbf{r}-\mathbf{r}')$$

Interpolation for finite points

- In general:

$$< f(\mathbf{r}) >= \sum_{j=1}^{N} \frac{f(\mathbf{r}_j)}{< n(\mathbf{r}_j) >} W(\mathbf{r} - \mathbf{r}_j; h).$$

$$< \rho(\mathbf{r}) >= \sum_{j=1}^{N} m_j W(\mathbf{r} - \mathbf{r}_j; h)$$

Calculating Gradients

- Integration by parts can move the derivative:
- $$\langle \nabla f \rangle = \sum_{j=1}^{N} m_j \frac{f(\mathbf{r}_j)}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j; h)$$
- Better accuracy is obtained with gradients of density weighted quantities:

$$\rho \nabla f = \nabla(\rho f) - f \nabla \rho$$

The Weighting Function

- Requirements:
  - Continuous 2nd derivatives
  - Compact
  - Symmetric
- Cubic Spline satisfies these requirements
- Symmetrize explicitly
- Maintain uniform particle distribution

Finding the Neighbors

- Fixed “h”: range search
  - “Ball search”: recursively descend tree using “bounds-intersect-ball” test
- Variable “h”: kth nearest neighbor search
  - Search tree using a shrinking ball.
  - Start with nodes close to the particle considered.

(Classical) SPH equations

- Density: $\rho_i = \sum_{j=1}^{n} m_j W_{ij}.$
- Momentum
- $$\frac{d\vec{v}_i}{dt} = -\sum_{j=1}^{n} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij},$$
- Energy

$$\frac{d\ u_i}{dt} = \frac{P_i}{\rho_i^2} \sum_{j=1}^{n} m_j \vec{v}_{ij} \cdot \nabla_i W_{ij}$$

- Alternatively: Entropy Equation (comparable performance)

Artificial Viscosity

- Momentum diffusion necessary to stabilize all numerical hydro formulations.

$$\Pi_{ij} = \begin{cases} \frac{-\alpha\frac{1}{2}(c_i+c_j)\mu_{ij}+\beta\mu_{ij}^2}{\frac{1}{2}(\rho_i+\rho j)} & \text{for } \vec{v}_{ij}\cdot\vec{r}_{ij} < 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{where } \mu_{ij} = \frac{h(\vec{v}_{ij}\cdot\vec{r}_{ij})}{\vec{r}_{ij}^{\,2} + 0.01(h_i + h_j)^2}$$

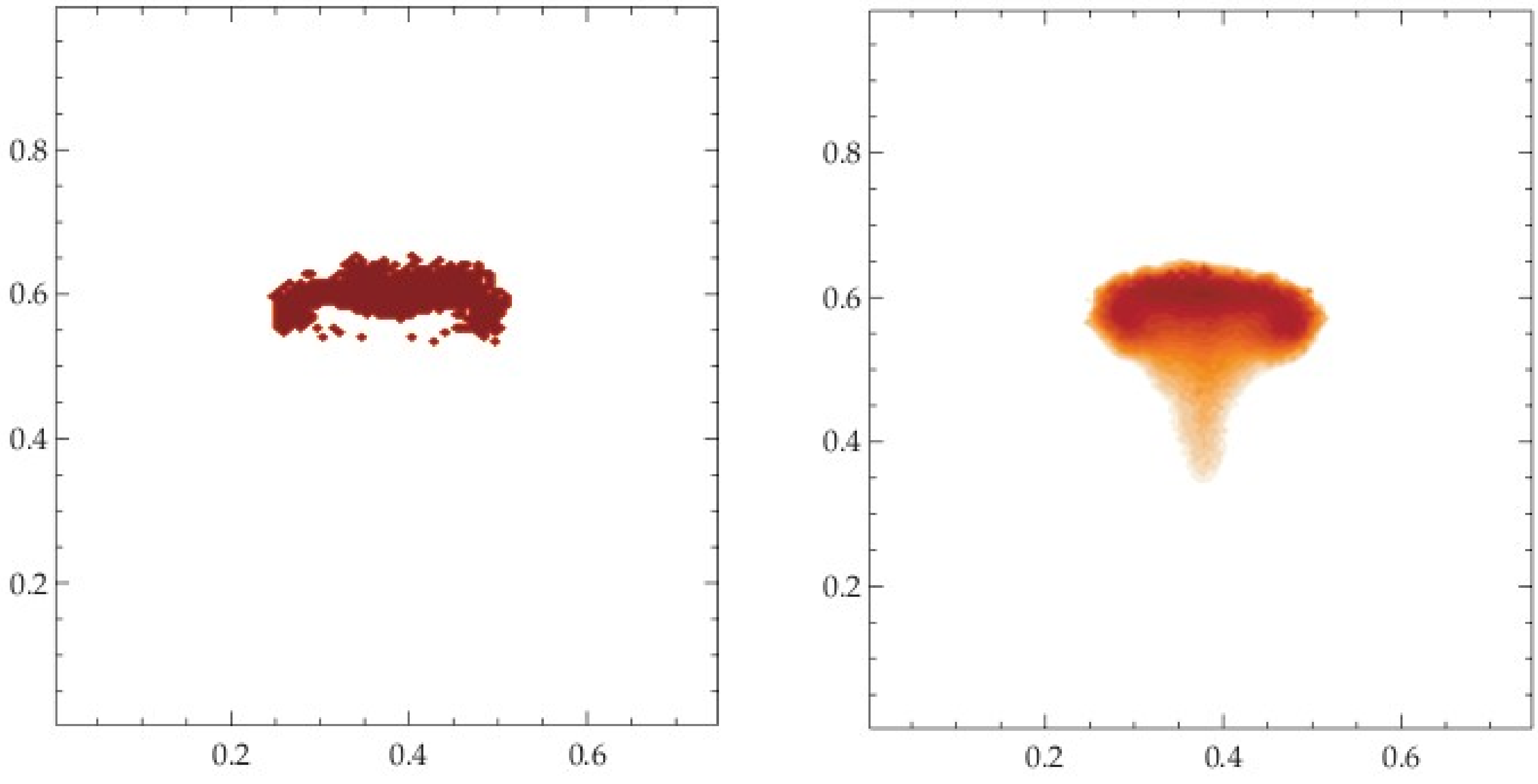
Artificial Viscosity & Diffusion

- All hydro codes introduce diffusion for stability
- SPH only has diffusion if explicitly added
- High Reynold numbers flows have turbulence below the resolution which can be modeled by diffusion (Smagorinsky 1963)

$$
\begin{aligned}
\frac{\mathrm{d}\, u_i}{\mathrm{d}t} = {} & \frac{P_i}{\rho_i^2} \sum_j m_j (\boldsymbol{v}_i - \boldsymbol{v}_j) \cdot \nabla_i W_{ij} \\
& - \sum_j m_j Q_{ij} (u_i - u_j) \frac{(\boldsymbol{r}_b - \boldsymbol{r}_a)}{|\boldsymbol{r}_b - \boldsymbol{r}_a|^2} \cdot \nabla_i W_{ij},
\end{aligned}
$$

$$
Q_{ij} = C \frac{|\boldsymbol{v}_i - \boldsymbol{v}_j|(h_i + h_j)}{\rho_i + \rho_j}.
$$

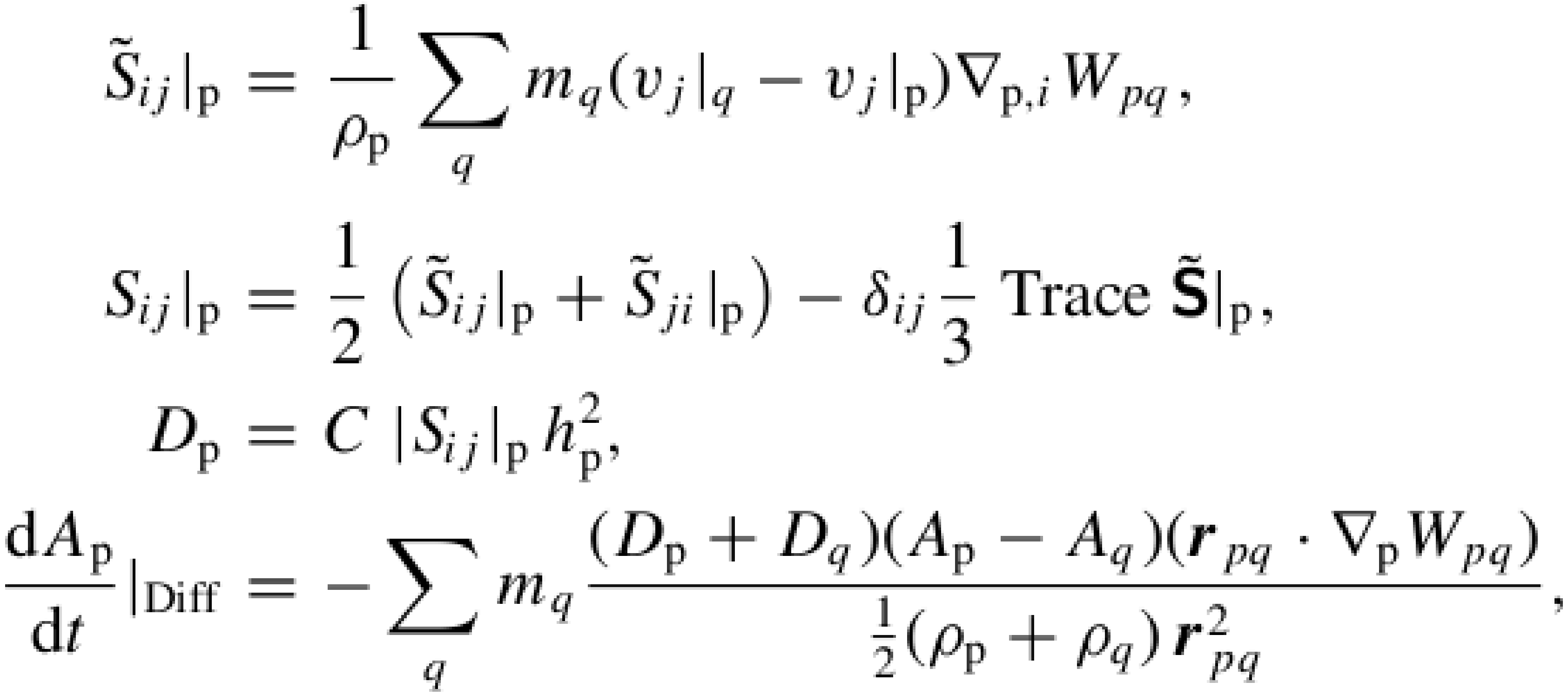
Bubble comparison



Wadsley et al 2008

Metal Diffusion

- Turbulence should also diffuse metals.
- For a scalar, A:



$$\tilde{S}_{ij}|_{\mathrm{p}} = \frac{1}{\rho_{\mathrm{p}}} \sum_{q} m_q (v_j|_q - v_j|_{\mathrm{p}}) \nabla_{\mathrm{p},i} W_{pq}\,,$$

$$S_{ij}|_{\mathrm{p}} = \frac{1}{2}\left(\tilde{S}_{ij}|_{\mathrm{p}} + \tilde{S}_{ji}|_{\mathrm{p}}\right) - \delta_{ij}\frac{1}{3}\,\mathrm{Trace}\,\tilde{\mathbf{S}}|_{\mathrm{p}}\,,$$

$$D_{\mathrm{p}} = C\;|S_{ij}|_{\mathrm{p}}\,h_{\mathrm{p}}^{2},$$

$$\frac{\mathrm{d}A_{\mathrm{p}}}{\mathrm{d}t}|_{\mathrm{Diff}} = -\sum_{q} m_q \frac{(D_{\mathrm{p}} + D_q)(A_{\mathrm{p}} - A_q)(\boldsymbol{r}_{pq} \cdot \nabla_{\mathrm{p}} W_{pq})}{\frac{1}{2}(\rho_{\mathrm{p}} + \rho_q)\,\boldsymbol{r}_{pq}^{2}}\,,$$

Shen, Wadsley & Stinson 2010

Cooling

- Cooling timescales can be short compared to a dynamical time
- Implicit (stiff) solver for thermal energy, assume work and density are constant.
- Chemical reaction networks have similar properties: use same solver for chemistry/thermal energy concurrently. (CHEMEQ2, Mott, D.R. & Oran, E.S., 2001)

SPH disadvantages

- Low order
  - No Riemann solver to handle shocks, etc.
  - Grad 1 != 0 !?!
- Poor particle distribution => large errors
- Kernel choices matter
  - E.g. Clumping instability
- Robust
  - Keeps going even if the results are garbage

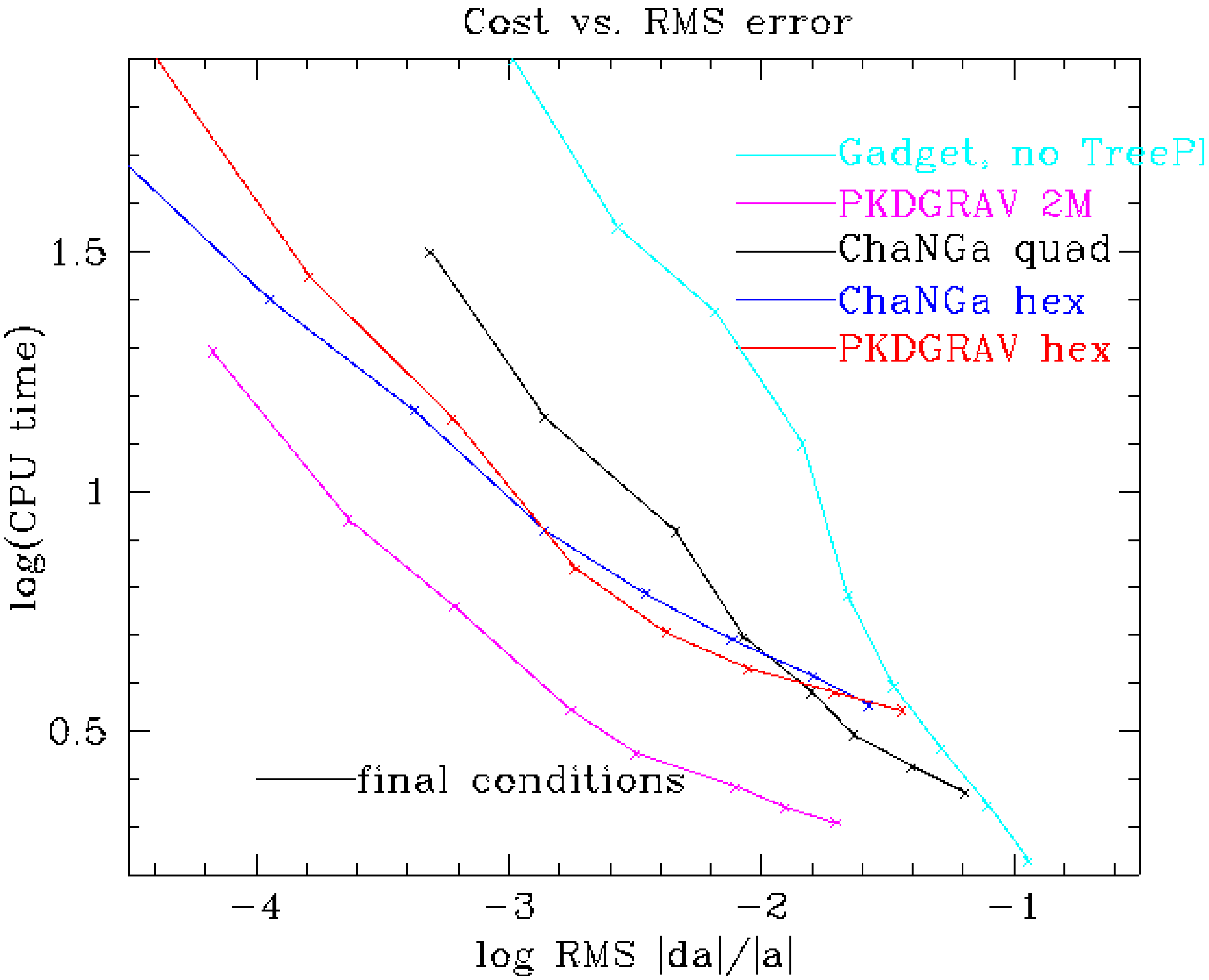
Summary

- Continuous media can be represented by collections of particles.
  - Advantages
    - Lagrangian, adaptable
  - Disadvantages
    - Low order, noise
- Trees are useful for organizing the resulting unstructured data.
  - Naturally adaptable
  - Usually have good worst case performance

Timestep Overview

- Adjust timesteps
- “Kick” velocities
- “Drift” Particles
- Domain Decompose
- Build tree, calculate Moments
- Calculate gravity forces
- Calculate SPH forces (predicted v & u needed)
- “Kick” velocities

Old Comparison


Cost vs. RMS error
Gadget, no TreePl
PKDGRAV 2M
ChaNGa quad
ChaNGa hex
PKDGRAV hex
final conditions
1.5
1
0.5
log(CPU time)
−4
−3
−2
−1
log RMS |da|/|a|