Gasoline

Treecodes for Cosmology

Thomas Quinn
University of Washington
N-Body Shop

Outline

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

Time stepping

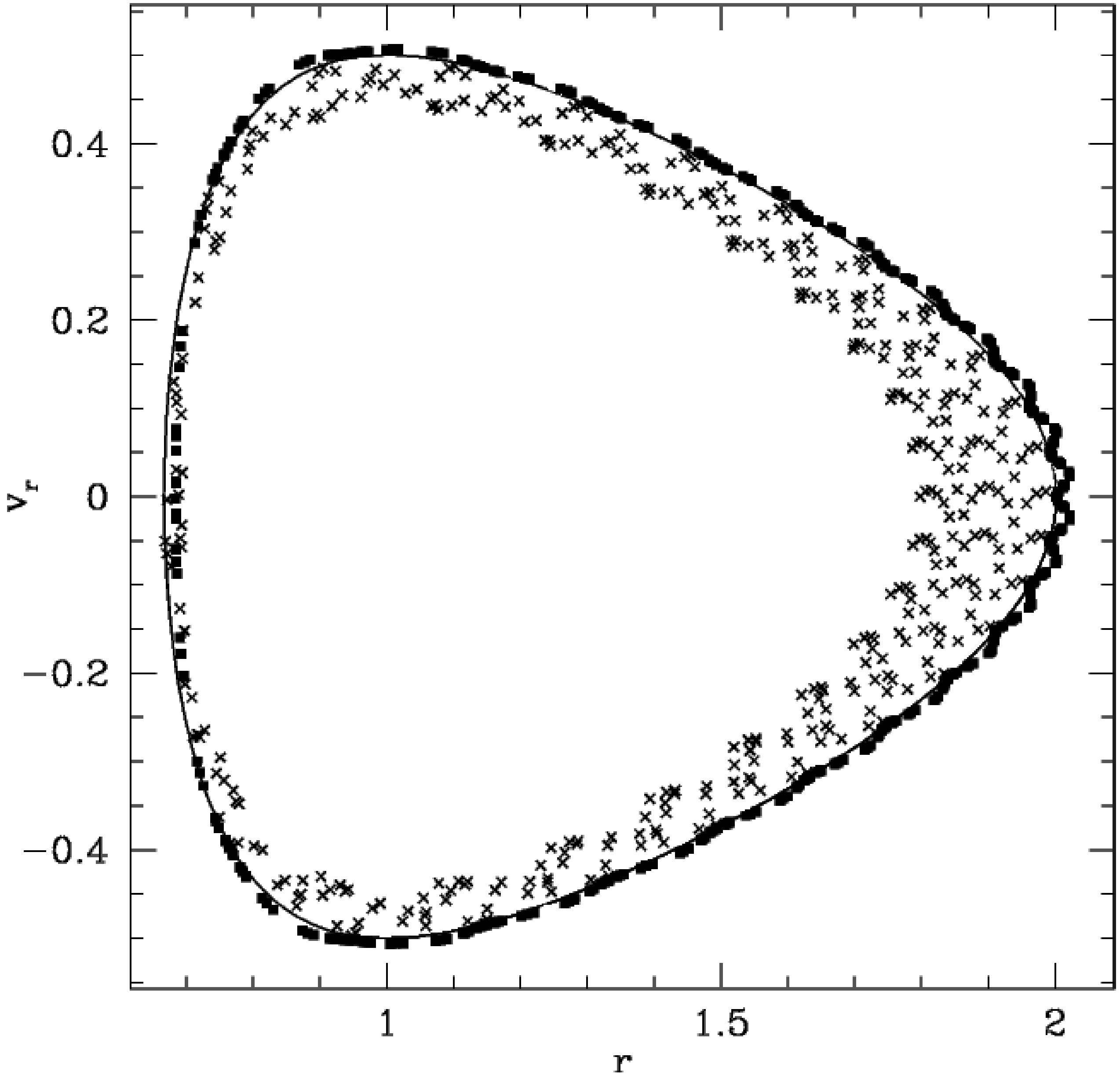
- An N-body system is Hamiltonian
  - Invariant under time translation
  - Phase space density preserved
- Preserve these properties in a numerical integration by ***Exactly*** integrating an ***Approximate*** Hamiltonian
- Operator Splitting: 1st applying part of a Hamiltonian, then applying the 2nd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

Time Stepping

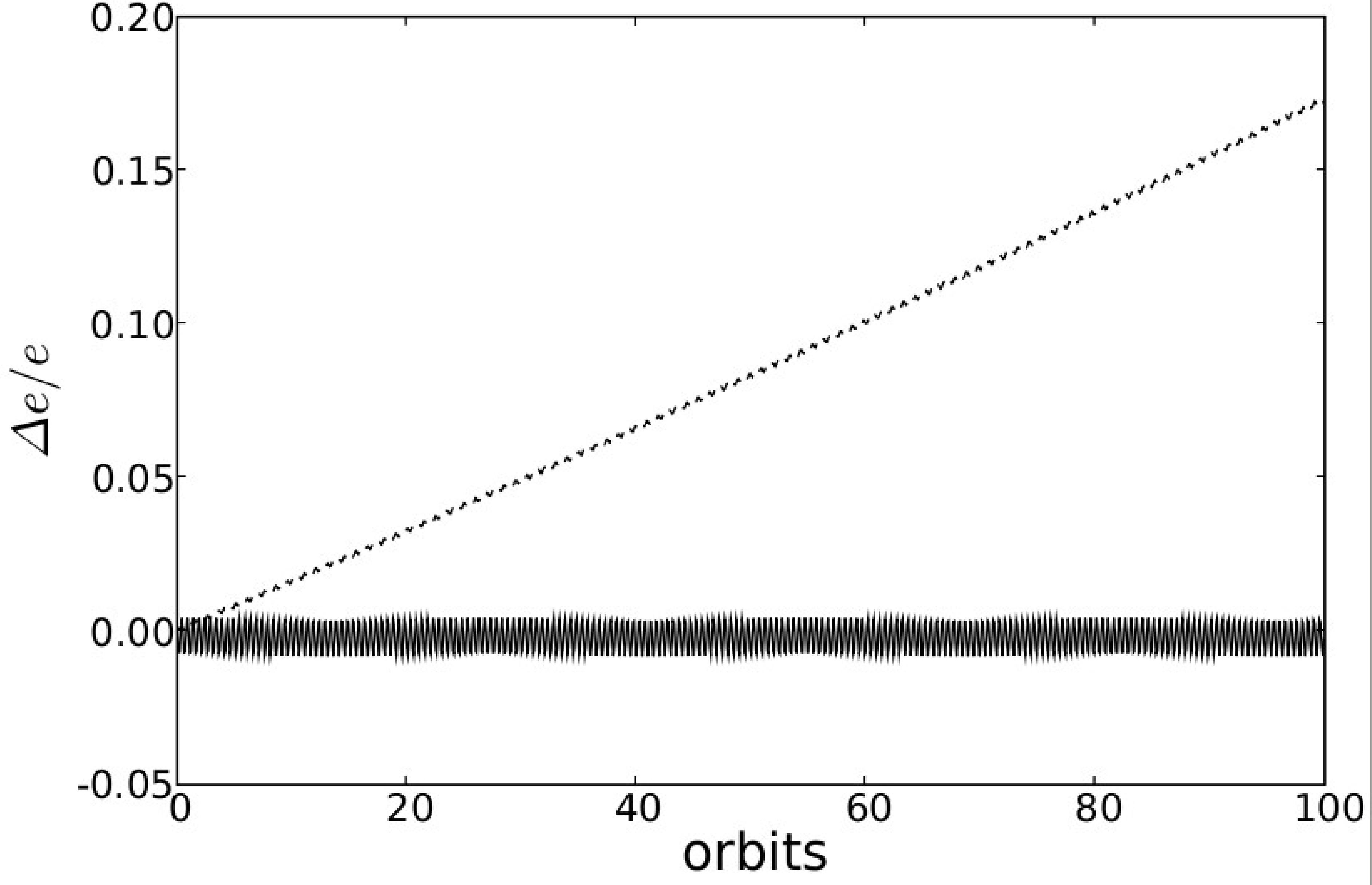
- Consider $H = H_D + H_K$
- Where $H_D = \frac{p^2}{2}; \quad H_K = \Phi(\mathbf{x})$
- Hamilton's equations give:
- $x(t) = x(t_0) + p(t - t_0); \quad p = p_0$
- $x(t) = x_0; \quad p(t) = p(t_0) - \nabla\Phi(\mathbf{x}_0)(t - t_0)$
- This is Leap Frog!
- Obeys a Hamiltonian: $H_{num} = H_D + H_K + H_{err}$

Saha & Tremaine 1992

Leapfrog vs. Runge Kutta



In a Rotating Frame



Comoving Equations of Motion

$$\dot{\mathbf{v}}' + 2H(t)\mathbf{v}' = -\frac{\nabla'\phi'}{a^3}$$

$$\dot{\mathbf{r}}' = \mathbf{v}'$$

$$\nabla'^2\phi' = 4\pi G(\rho' - \rho'_b),$$

But these can be derived from a Hamiltonian:

$$H = \frac{\mathbf{p}'^2}{2a^2} + \frac{\phi'}{a}.$$

Where p' = a^2v'

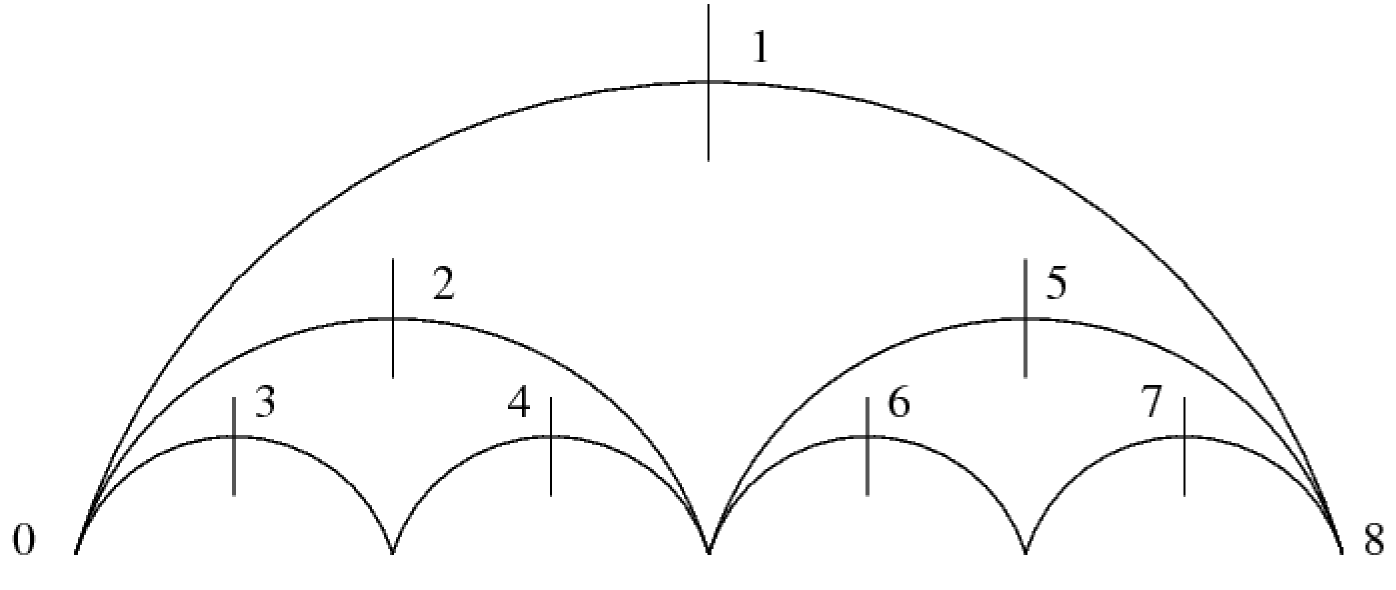
Canonical Comoving Equations

- The two pieces of the Hamiltonian can be integrated to give:
- $$D(\tau) \equiv \mathbf{r}'_{t+\tau} = \mathbf{r}'_t + \mathbf{p}' \int_t^{t+\tau} \frac{dt}{a^2}$$
- $$K(\tau) \equiv \mathbf{p}'_{t+\tau} = \mathbf{p}'_t - \nabla' \phi' \int_t^{t+\tau} \frac{dt}{a},$$
- A single timestep is taken by applying

$$K(\tau/2) D(\tau) K(\tau/2)$$

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 conserve.

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
- See M. Zemp et al 2007 for an “optimal criterion”

Gravitational Softening

- Recall: we are solving the CBE, 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>
- Does not effect two body relaxation time!
- Too small: two body scattering
- Too large: lose structural detail
- Ultimately a computational cost decision

SPH advantages

- Naturally partners with a particle gravity code
- Arbitrary geometry
- Lagrangian
- Galilean invariant
- Less dissipative for density weighted quantities
- Fast
- 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
- Symmetrize explicitly

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

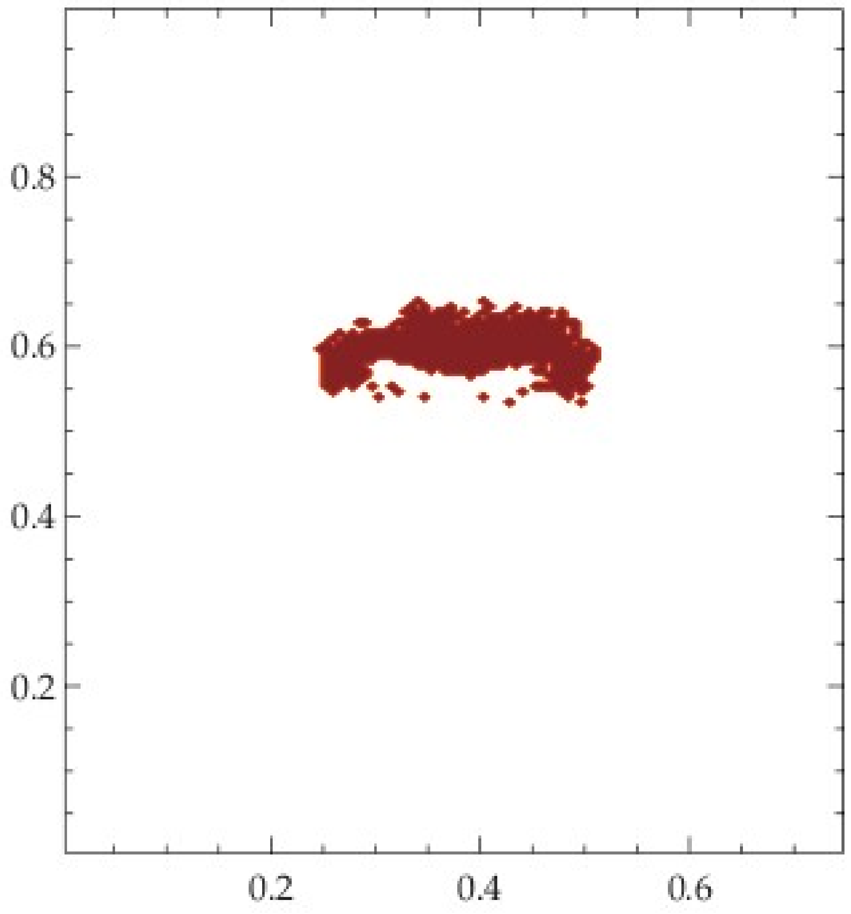
- All hydro codes introduce diffusion for stability
- SPH only has diffusion if explicitly added
- High Reynold numbers flows have turbulence below the resolution which can be modeled by diffusion (Smagorinsky 1963)

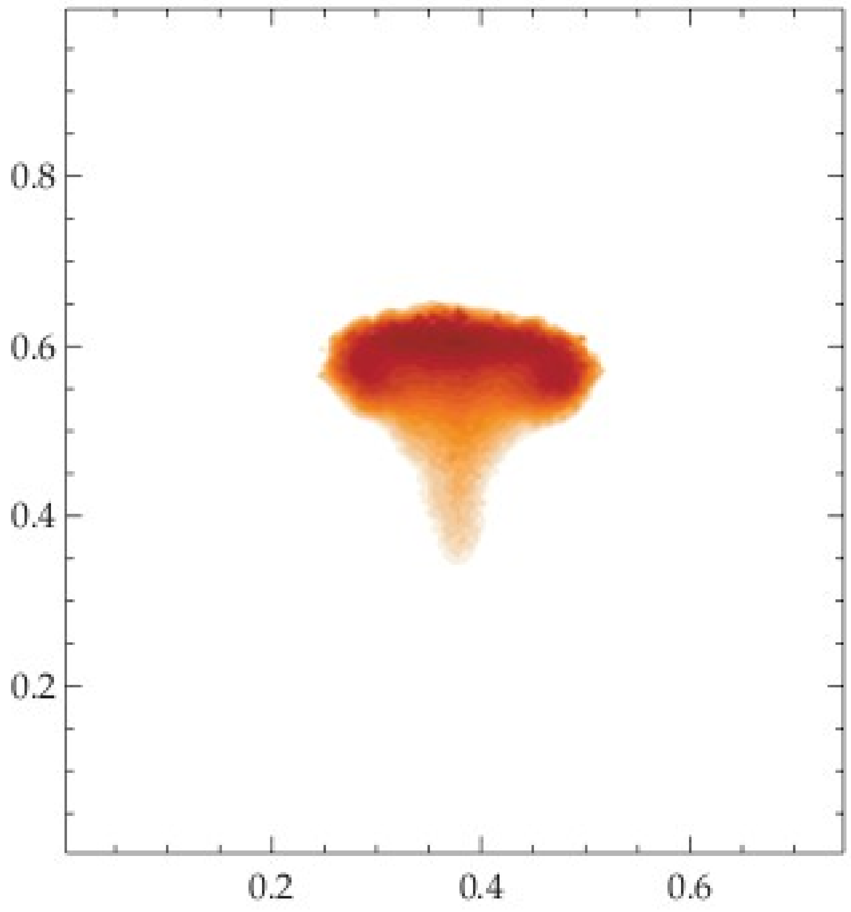
$$\frac{\mathrm{d}\,u_i}{\mathrm{d}t} = \frac{P_i}{\rho_i^2}\sum_j m_j(\boldsymbol{v}_i - \boldsymbol{v}_j)\cdot\nabla_i W_{ij}$$

$$- \sum_j m_j Q_{ij}(u_i - u_j)\frac{(\boldsymbol{r}_b - \boldsymbol{r}_a)}{|\boldsymbol{r}_b - \boldsymbol{r}_a|^2}\cdot\nabla_i W_{ij},$$

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

Bubble comparison





Wadsley et al 2008

Metal Diffusion

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

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

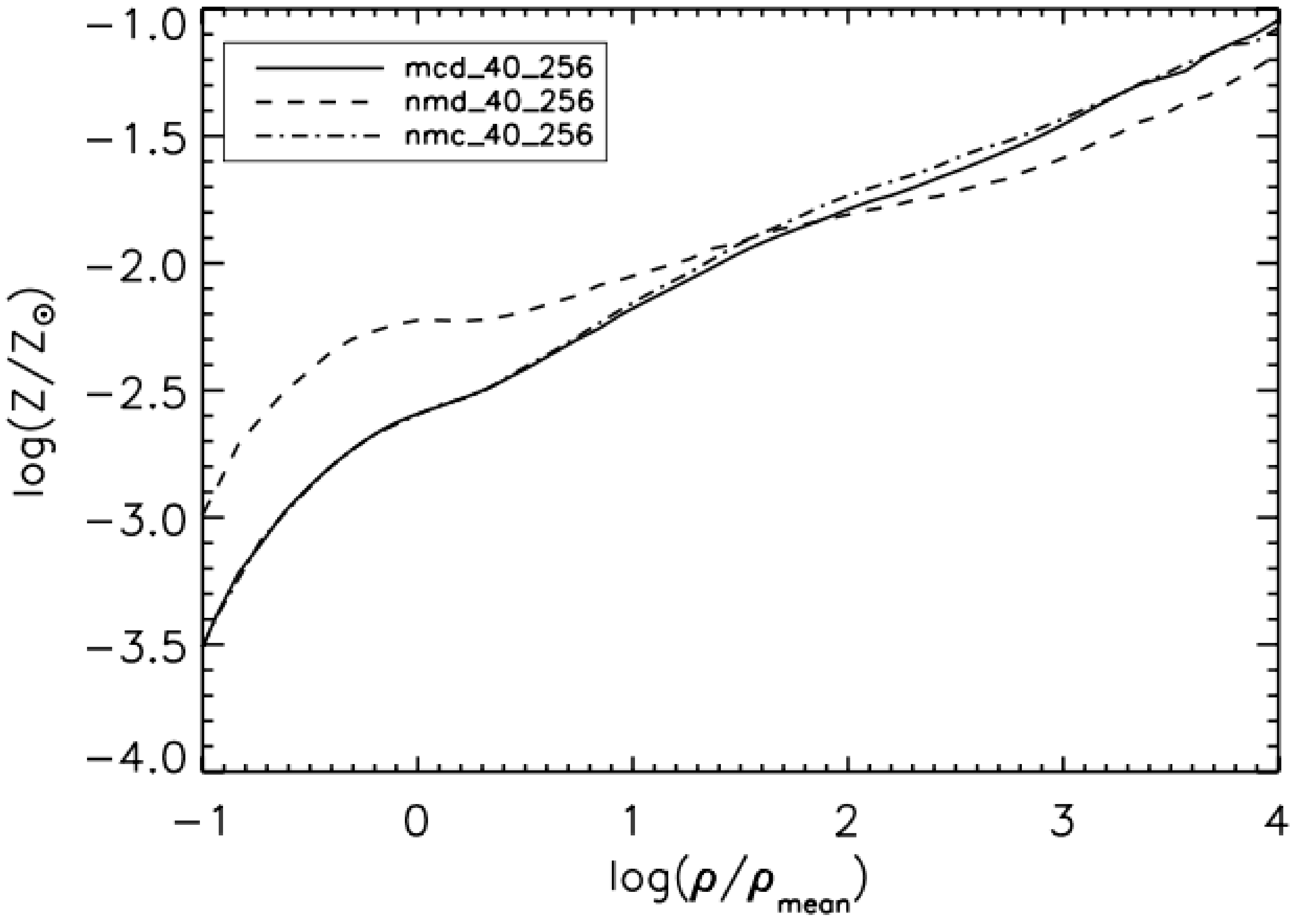
$$S_{ij}|_{\mathrm{p}} = \frac{1}{2}\left(\tilde{S}_{ij}|_{\mathrm{p}} + \tilde{S}_{ji}|_{\mathrm{p}}\right) - \delta_{ij}\frac{1}{3}\,\mathrm{Trace}\,\tilde{\mathbf{S}}|_{\mathrm{p}},$$

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

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

Shen, Wadsley & Stinson 2010

Metal distribution

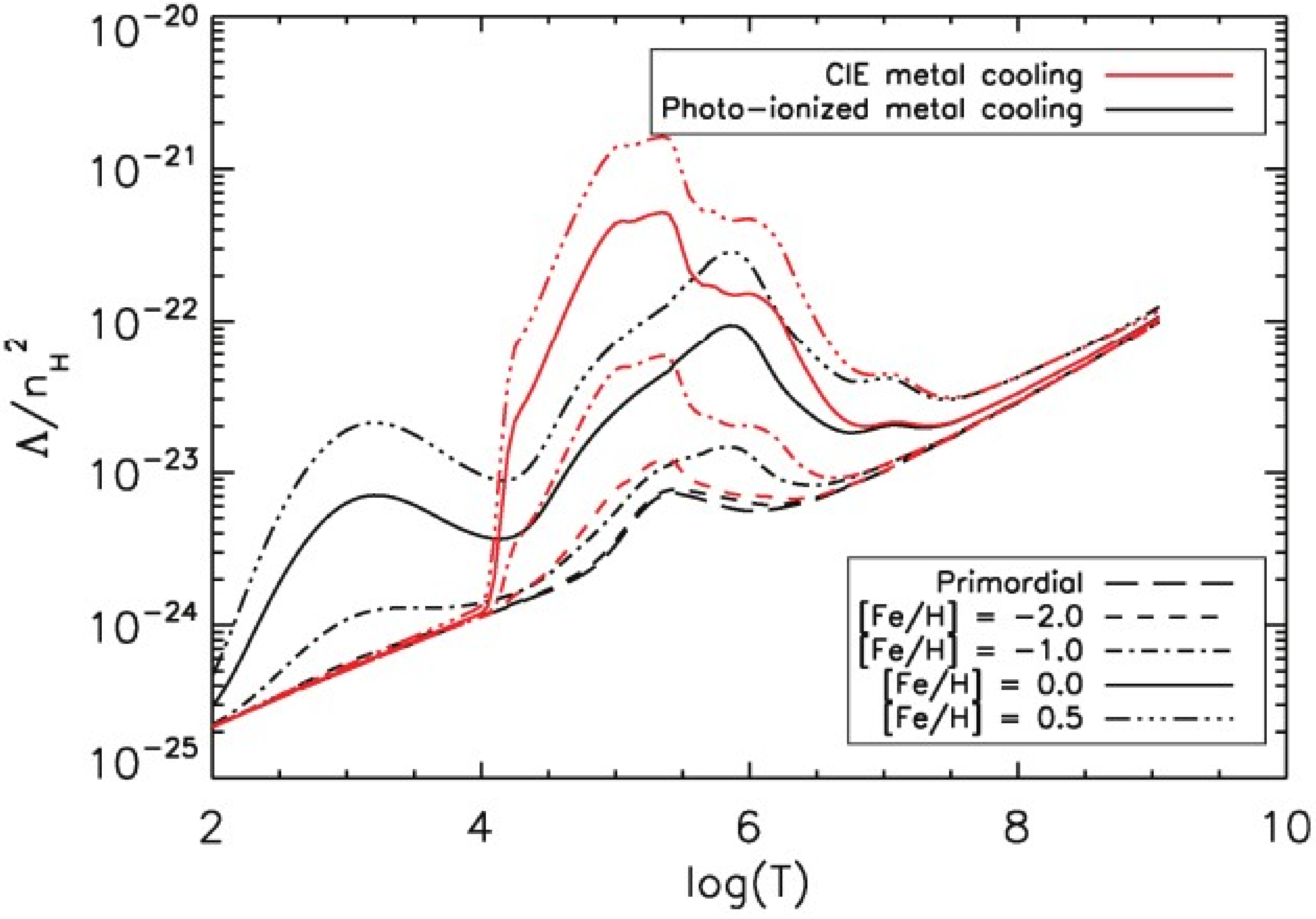


Cooling

- Cooling timescales can be short compared to a dynamical time
- Implicit (stiff) solver for thermal energy, assume work and density are constant.
- Addition of non-equilibrium metal cooling:

  Enhances low T cooling in the presence of UV

Metal Cooling



Parallel Architecture of GASOLINE

- Master layer: overall flow control; serial
- Processor Set Tree layer (PST): parallel glue
- Parallel KD layer (PKD):
  - Access to particle/tree data
  - Serial, runs simultaneously on all processors
- Machine Dependent layer (MDL):
  - Interface to parallel primitives
  - Implemented in MPI, Posix-threads, PVM, serial, ...

MDL

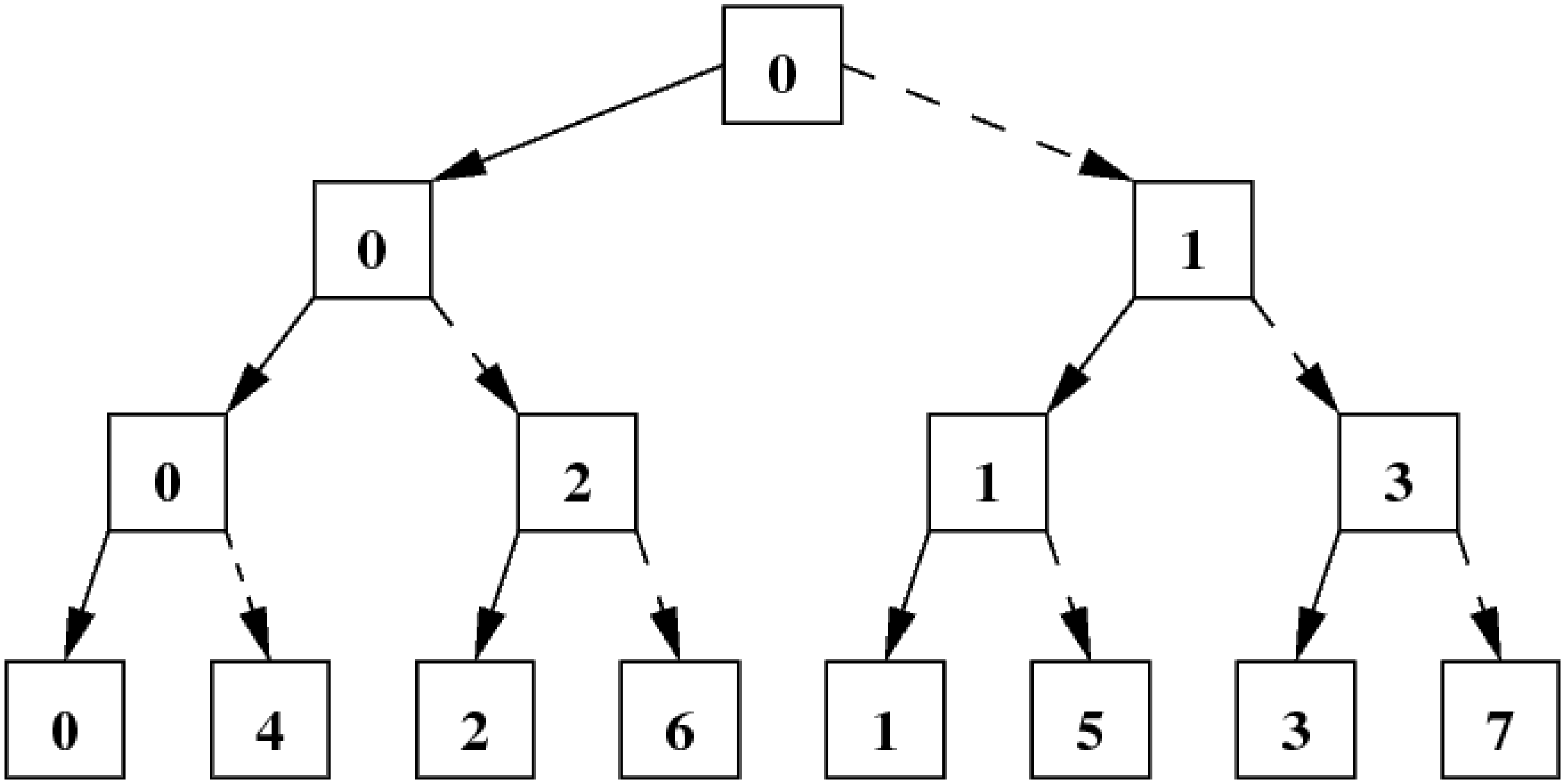
- Allows for easy portability/performance tuning
- Implements:
  - Asynchronous remote procedure call
  - Memory swapping
  - Read-only and “combining” remote memory access
  - Diagnostic utilities

MDL Cache

- Software cache of remote data
  - Amortizes access of remote data
  - Avoids excess memory use
- Read-only
- Combiner: commutative/associative operations
  - e.g.: sum, maximum
  - Necessary for symmetric SPH

PST layer

- Balanced, binary tree of processors.
- Organizes parallel dispatch and top level tree



Domain Decomposition

- Domains are complete subtrees: domain tree coincides with top level of gravity tree
- ORB tree used to balance work
- “Root find” at each level to find split that balances work
- “Non-Active” particles split separately based on memory

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

Outline

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