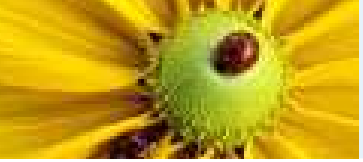
Galaxy Formation: Methods

Neal Katz
UMass Astronomy

September, 2013



The N-body Method

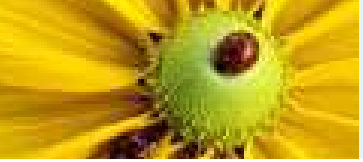
- An N-body simulation is an attempt to solve the collisionless Boltzmann equation (CBE),

$$\frac{df(\vec{x},\vec{v},t)}{dt} = \frac{\partial f}{\partial t} + \vec{v}\cdot\vec{\nabla} f - \vec{\nabla}\Phi\frac{\partial f}{\partial \vec{v}} = 0, \tag{1}$$

  in concert with the Poisson equation

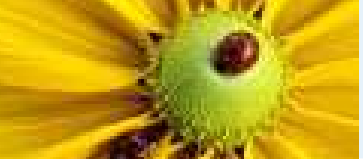
$$\nabla^2\Phi = 4\pi G \int d^3v\ f(\vec{x},\vec{v},t). \tag{2}$$

- Direct, 6-dimensional integration of f is usually impractical.
- In the N-body approach, one follows the orbits of representative mass elements, a.k.a. particles.
- An N-body simulation is an algorithm for Monte Carlo simulation of the CBE.



The N-body Method (cont)

- The particles are tracers of the density field that are used to simultaneously solve for the gravitational potential and sample the phase-space density.
- Schematically:
  - 1) Start with initial positions and velocities.
  - 2) Update positions
  - 3) Compute gravitational accelerations from particle distribution.
  - 4) Update velocities.
  - 5) Repeat.



Direct Techniques

- Determine the gravitational acceleration by directly summing over all particle pairs.
  - Scales as $O(n^2)$.
- Example using leap frog to advance one timestep, Δt:

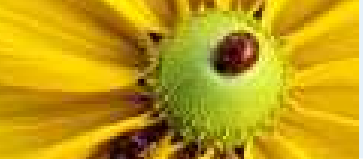
$$\vec{x}_i(t+0.5\Delta t) = \vec{x}_i(t)+0.5\Delta t\vec{v}_i(t) \tag{3}$$

$$\vec{a}_i(t+0.5\Delta t) = \sum_{j,j\neq i}\frac{Gm_j\vec{r}_{ij}}{|\vec{r}_{ij}|^3}, \qquad \vec{r}_{ij}=\vec{x}_i-\vec{x}_j \tag{4}$$

$$\vec{v}_i(t+\Delta t) = \vec{v}_i(t)+\Delta t\vec{a}_i(t+0.5\Delta t) \tag{5}$$

$$\vec{x}_i(t+\Delta t) = \vec{x}_i(t+0.5\Delta t)+0.5\Delta t\vec{v}_i(t+\Delta t) \tag{6}$$

- A symplectic integrator: maintains Hamiltonian.

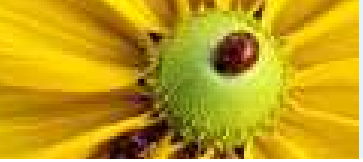


Gravitational Softening

- Introduce gravitational softening:

$$\sum_{j,j\neq i} \frac{Gm_j\vec{r}_{ij}}{|\vec{r}_{ij}|^3} \rightarrow \sum_{j,j\neq i} Gm_j \frac{\vec{r}_{ij}}{\left(|\vec{r}_{ij}|^2+\varepsilon^2\right)^{3/2}} \tag{7}$$

- This form is called Plummer Softening and ε is the gravitational softening length.
- Effectively limits the spatial resolution to 2ε.



Why Gravitational Softening?

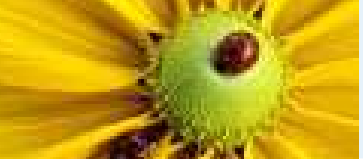
- Two main reasons to include gravitational softening:
  - Allow larger timesteps to be used to decrease run time.
  - Reduce the effects of two-body relaxation and better approximate a collisionless system.

$$\tau_r = 0.34 \frac{\sigma_{1d}^3}{G^2 m_p \rho \ln \Lambda} \tag{8}$$

$$\approx 3.9\ \mathrm{Gyr} \left(\frac{\sigma_{1d}}{100 \mathrm{km\ s^{-1}}} \right)^3 \frac{h^{-2}}{ln\Lambda} \frac{10^{11} M_\odot}{m_p} \frac{(170\rho_c)}{\rho} \tag{9}$$

  - where $\ln \Lambda$ is the Coulomb logarithm with $\Lambda \approx R_h/4\varepsilon$ and R_h is the half mass radius.

- Since ε only enters logarithmically the first reason is most important.

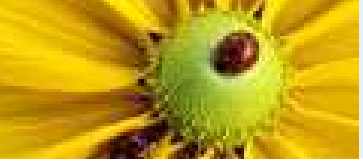


Other forms of Gravitational Softening

- The Plummer form of gravitational softening changes the law of gravity at all radii.
- Would like a form that has compact support so that gravity only changes at small scales.
- Introduce a spline softened form of the acceleration:

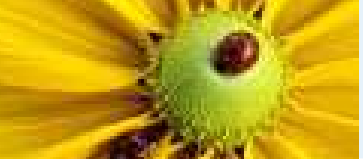
$$\vec{a}_i = \sum_{j,i\neq j} G m_j \vec{r} \begin{cases} \frac{1}{\varepsilon^3}\left[\frac{4}{3} - \frac{6}{5}u^2 + \frac{1}{2}u^3\right] & 0 \leq u \leq 1 \\ \frac{1}{|\vec{r}|^3}\left[-\frac{1}{15} + \frac{8}{3}u^3 - 3u^4 + \frac{6}{5}u^5 - \frac{1}{6}u^6\right] & 1 \leq u \leq 2 \\ \frac{1}{|\vec{r}|^3} & u \geq 2 \end{cases} \quad (10)$$

where $u = |\vec{r}|/\varepsilon$.



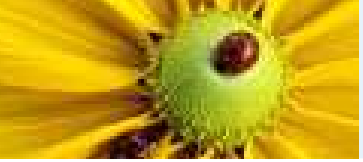
Variable Timesteps

- One can add variable timesteps to increase computational efficiency.
- Destroys symplecticness of leap frog integrator but possible to retain second order accuracy.
- Two common choices:
  - Can use power of two subdivisions of largest allowed timestep, which allows system to be resynchronized every largest timestep.
  - Can allow particles to all have different timesteps without restrictions.
    - Aarseth codes use high order integrators and scale as $O(n^{1.7})$.



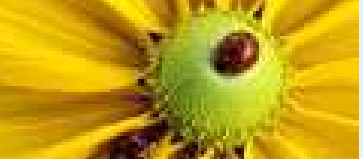
Testing the Validity of Results

- Check energy, linear momentum and angular momentum conservation.
- Try increasing the number of particles. Is the result the same?
- Try decreasing the timestep. Is the result the same?
- Try altering the gravitational softening. Is the result the same?



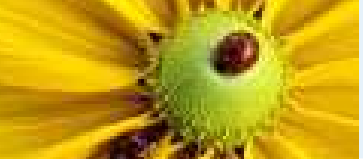
Testing the Validity of Results

- Check energy, linear momentum and angular momentum conservation.
- Try increasing the number of particles. Is the result the same?
- Try decreasing the timestep. Is the result the same?
- Try altering the gravitational softening. Is the result the same?
- All the above are necessary but not sufficient conditions.



Testing the Validity of Results

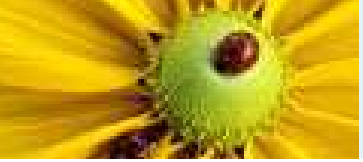
- Check energy, linear momentum and angular momentum conservation.
- Try increasing the number of particles. Is the result the same?
- Try decreasing the timestep. Is the result the same?
- Try altering the gravitational softening. Is the result the same?
- All the above are necessary but not sufficient conditions.
- Try comparing with other codes.



Testing the Validity of Results

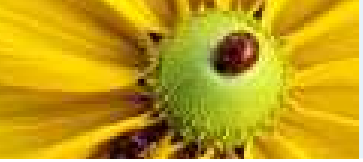
- Check energy, linear momentum and angular momentum conservation.
- Try increasing the number of particles. Is the result the same?
- Try decreasing the timestep. Is the result the same?
- Try altering the gravitational softening. Is the result the same?
- All the above are necessary but not sufficient conditions.
- Try comparing with other codes.
- Try running on problems as closely related as possible to your problem with known solutions and compare.

The best approach!



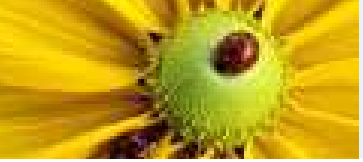
Particle Mesh Codes (PM)

- Use fast Fourier transforms (FFT's) to calculate the gravitational accelerations.
- Done on a regular rectangular M^3 grid and is $O(M \ln M)$.
- Schematically the algorithm proceeds as follows:
  - 1) Assign masses of particles to the grid to determine the density at the grid points through interpolation.
  - 2) Compute the gravitational potential at the grid points by solving the Poisson equation on the grid using an FFT.
  - 3) Compute gravitational accelerations at the grid points by finite-differencing the potential.
  - 4) Calculate the acceleration at each particle position using the same interpolation scheme used in the mass assignment to ensure momentum conservation.



Tree Codes

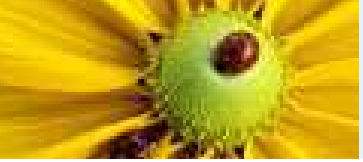
- Tree codes group together interactions with distant particles using a tree data structure.
- They are $O(n \ln n)$.
- Schematically the force calculation proceeds as follows:
  - 1) Put particles into the tree structure.
  - 2) Calculate properties of the tree nodes, e.g. the mass and center of mass of each node.
  - 3) Determine the interaction list for each particle (walking the tree).
  - 4) Sum the contributions from particles on the interaction list to determine the gravitational acceleration of each particle.



Types of Trees

Two main types of tree geometries used:

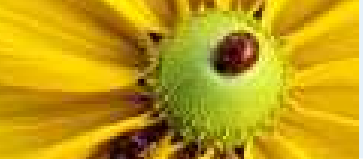
- Oct-trees–split space into successively smaller octants by volume.
  - Each node in the tree has up to 8 children.
  - Splits in all three dimensions at once.
- kd-trees–divide one dimension at a time, i.e. each node has one or two children.
  - Can divide by volume or by putting a equal number of particles in each node.
  - A kd-tree divided by number is called a balanced kd-tree.



Types of Trees

Two main types of tree geometries used:

- Oct-trees–split space into successively smaller octants by volume.
  - Each node in the tree has up to 8 children.
  - Splits in all three dimensions at once.
- kd-trees–divide one dimension at a time, i.e. each node has one or two children.
  - Can divide by volume or by putting a equal number of particles in each node.
  - A kd-tree divided by number is called a balanced kd-tree.
- Oct-tree nodes are closest to spheres and hence have the smallest errors when calculating accelerations.

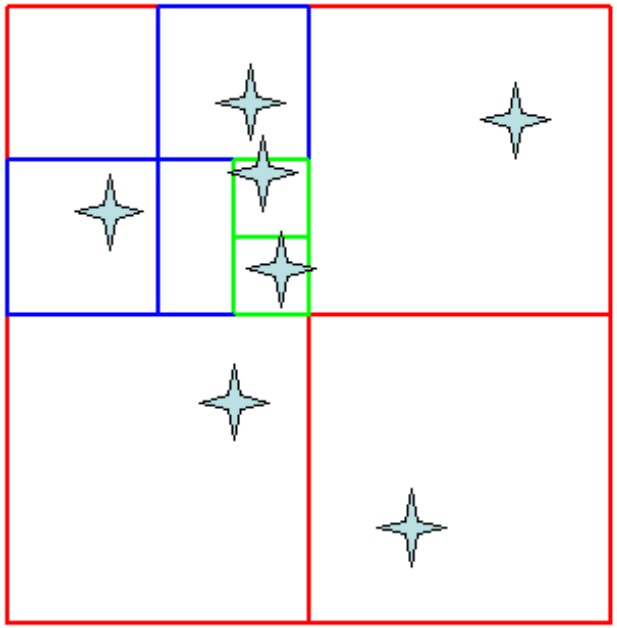


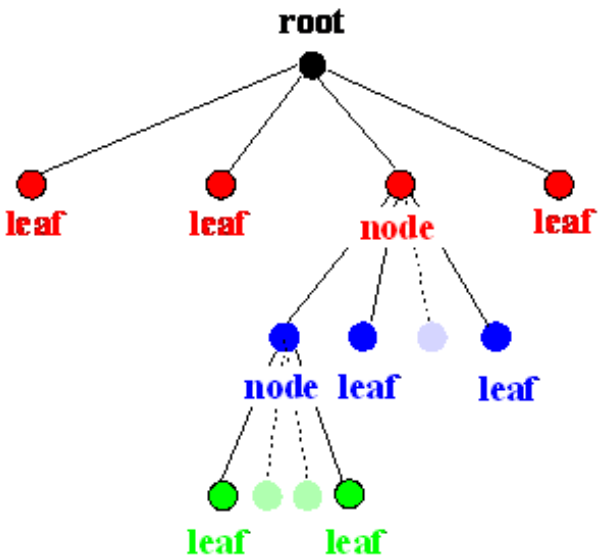
Oct-trees

Each node of the tree has:

- Pointers to 8 children (could be another node, a particle, or null if that node contains no particles).
- A pointer to its parent node (except for the root node).
- The position of the center of the node.
- The size of the node.
- The mass of the node.
- The center of mass position of the node.
- Higher order multipoles of the node.

Building Trees

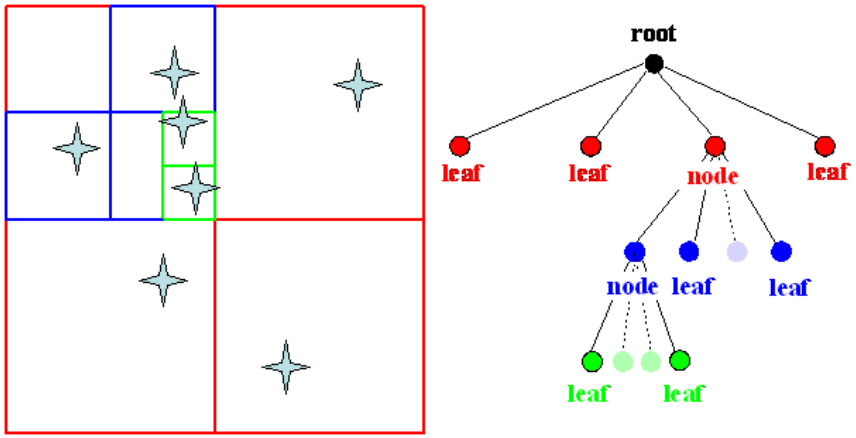




Build the tree from top down, one level at a time as follows ($O(n \ln n)$):

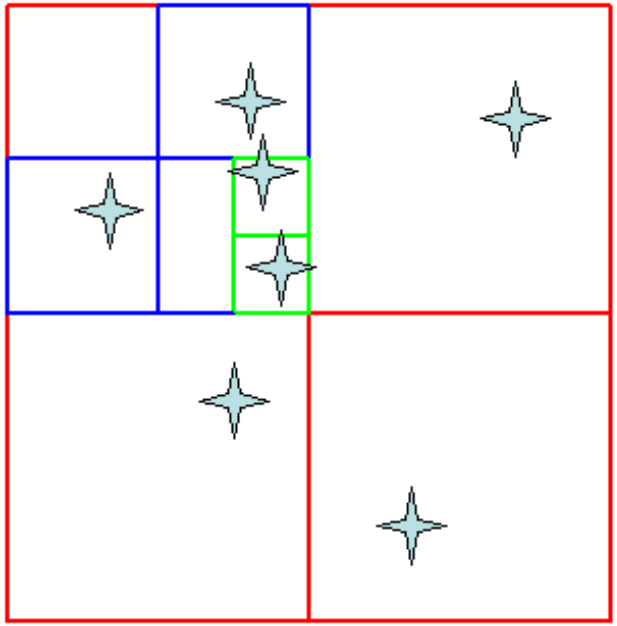
- Start with all the particles with the root node as parent, with only the root node as an active node and with all the particles active.
- Loop over the active particles and determine how many particles are in each octant of each active node.

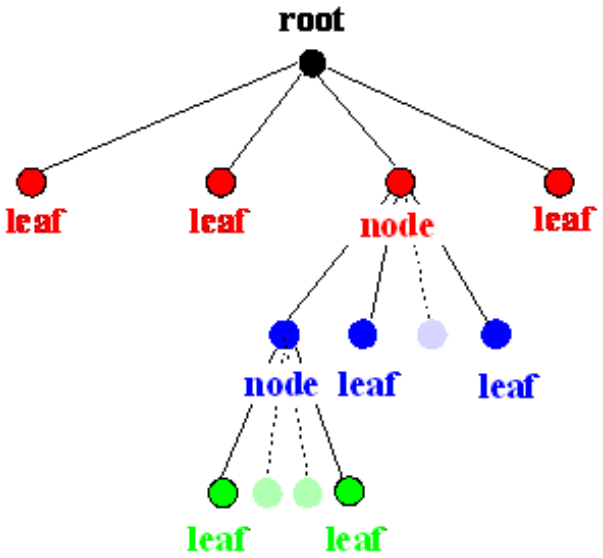
Building Trees (cont.)



- If an octant has more than one particle, create a node and assign it a size and a position (size is half the size of its parent node).
  - Set one parent node's children pointer to point to this node.
  - Set this nodes parent pointer to point to its parent.
  - Add this node to the new active node list.
  - Reset the parent pointer of the particles contained within this node to point to this node.
  - Add the particles in the node to the new active particle list.

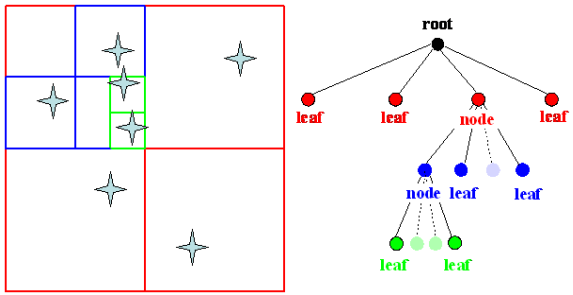
Building Trees (cont.)





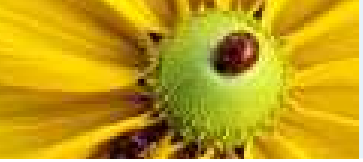
- If an octant has one particle, set one of the parent node's children pointers to point to this particle.
  - Do not add this particle to the new active particle list.
- If an octant contains no particles, set one of the parent node's children pointers to point to NULL. Repeat until there are no active nodes.

Building Trees (cont.)



Calculate the mass, center of mass position, and higher order moments of the mass distribution from the bottom of the tree up as follows $O(n \ln n)$):

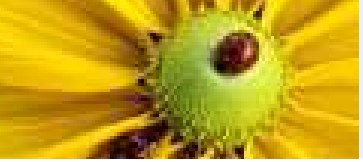
- Start with smallest size nodes, the lowest level of the tree (which were the last nodes created) and add their contributions to the node masses and center of mass coordinates (and contributions to higher order terms if they exist), e.g. $\sum m$ and $\sum m\vec{x}$.
- Continue up the tree one level at a time until the root node is reached.
- Divide $\sum m\vec{x}$ by $\sum m$ for each node to determine the center of mass coordinates (similar operations for the higher moments if they exist).



Walking Trees

Walk the tree from the top down to determine each particles interaction list ($O(n \ln n)$):

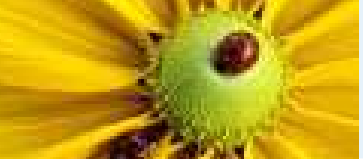
- Start with no particles or nodes on the interaction list and only the root node on the active node list.
- Loop through the active node list.
  - Add any children that are particles to the interaction list.
- If the node does satisfy the opening criteria then add it to the interaction list and remove it from the active node list.
- Repeat until there are no active nodes.



Walking Trees

Walk the tree from the top down to determine each particles interaction list ($O(n \ln n)$):

- Start with no particles or nodes on the interaction list and only the root node on the active node list.
- Loop through the active node list.
  - Add any children that are particles to the interaction list.
- If the node does satisfy the opening criteria then add it to the interaction list and remove it from the active node list.
- Repeat until there are no active nodes.
- Once the interaction list is made then the acceleration on the particle can be summed (including any higher order multipoles contributed by nodes, if such terms exist).



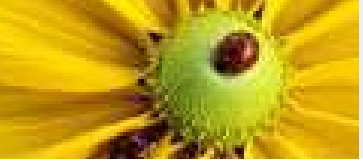
Opening Criteria

- Geometric form for θ, the opening angle,

$$\theta \equiv s/d \leq \theta_{max} \tag{11}$$

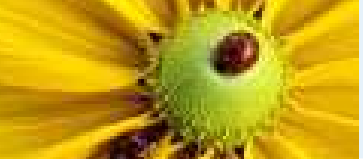
$$s = \text{size}_{node}, \qquad \text{and} \qquad d = |\vec{x}_i - \vec{x}_{node}| \tag{12}$$

- Can lead to problems if the center of mass of the node is far from the center of the node so can:
  - Replace s by $\sqrt{3}/2 \max\left\{|\vec{x}_j - \vec{x}_{cm,node}|\right\}$ where j runs over all the particles in the node and d by $|\vec{x}_i - \vec{x}_{cm,node}|$.
  - Replace d by $d - |\vec{x}_{node} - \vec{x}_{cm,node}|$.



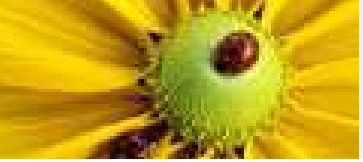
Higher Order Multipoles

- Can include higher order multipoles for the nodes to make things more accurate.
  - This allows a larger choice for the opening angle.
  - Usually it is more efficient as well, i.e. faster.
- Since the center of mass is used as the expansion center, the lowest order correction is of quadrupole order.



Higher Order Multipoles

- Can include higher order multipoles for the nodes to make things more accurate.
  - This allows a larger choice for the opening angle.
  - Usually it is more efficient as well, i.e. faster.
- Since the center of mass is used as the expansion center, the lowest order correction is of quadrupole order.
- If one constructs the interaction list carefully using multipoles of various orders for each item on the list it is possible to devise a scheme that is $O(n)$ called the Fast Multipole Method.



Comoving Coordinates

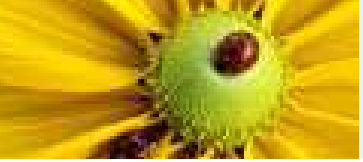
- For cosmological simulations work with coordinates comoving with the expansion, $\vec{x}$, related to proper coordinates, $\vec{r}$, and the expansion factor, $a = 1/(1+z)$:

$$\vec{x} = \frac{\vec{r}}{a} \tag{13}$$

- The momentum of a particle with mass m in these coordinates is $\vec{p} = ma^2\dot{\vec{x}}$.
- a evolves according to

$$\left(\frac{\dot{a}}{a}\right)^2 - \frac{8\pi}{3}G\rho = \frac{\Lambda}{3} - \frac{kc^2}{a} \tag{14}$$

where ρ is the mean mass density, Λ is the cosmological constant, and $k = -1, 0,$ or 1 for closed, flat, and open universes.



Comoving Coordinates (cont)

- The equations of motion for the i-th particle are

$$\dot{\vec{x}}_i = \frac{d\vec{x}_i}{dt} \tag{15}$$

$$\ddot{\vec{x}}_i + 2H\dot{\vec{x}}_i - \frac{1}{a}\vec{g}_i = 0 \tag{16}$$

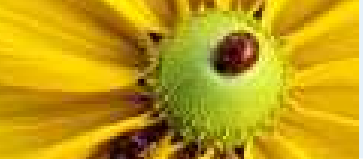
- Using leap frog to advance the particles:

$$\vec{x}_i^{n+1/2} = \vec{x}_i^{n} + 0.5\Delta t \dot{\vec{x}}_i^{n} \tag{17}$$

$$\dot{\vec{x}}_i^{n+1} = \dot{\vec{x}}_i^{n} \frac{1 - H^{n+1/2}\Delta t}{1 + H^{n+1/2}\Delta t} + \frac{\Delta t}{a^{n+1/2}} \frac{\vec{g}_i^{n+1/2}}{1 + H^{n+1/2}\Delta t} \tag{18}$$

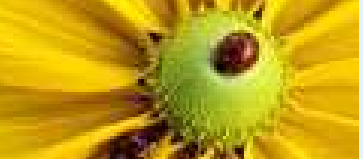
$$\vec{x}_i^{n+1} = \vec{x}_i^{n+1/2} + 0.5\Delta t \dot{\vec{x}}_i^{n+1} \tag{19}$$

where where $a^{n+1/2}$ and $H^{n+1/2}$ are the values of the expansion factor and the Hubble parameter at the intermediate time.



Boundary Conditions

- Two main choices for boundary conditions:
  - Periodic–good for simulating pieces of larger structures like in cosmological simulations.
  - Vacuum–good for simulating isolated objects like galaxies.
- The *natural* boundary conditions for:
  - Direct summation–vacuum.
  - Tree–vacuum.
  - PM–periodic.
- Can make a direct summation or tree code fully periodic using Ewald summation.



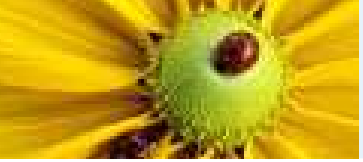
The Fluid Equations

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \mathbf{v} \cdot \nabla \rho = -\rho \nabla \cdot \mathbf{v} \tag{20}$$

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\frac{\nabla P}{\rho} - \nabla \phi \tag{21}$$

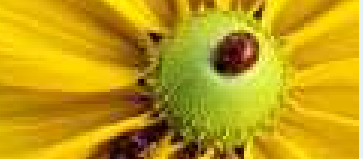
$$\frac{Du}{Dt} = \frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = -\left(\frac{P}{\rho}\right) \nabla \cdot \mathbf{v} + \frac{\Gamma - \Lambda}{\rho} \tag{22}$$

$$P = (\gamma - 1)\rho u \tag{23}$$



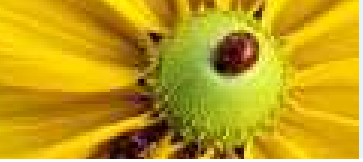
Eulerian Techniques

- The standard approach for solving the fluid equations.
- Calculate the fluid properties on a regular grid of points.
- Calculate derivatives using finite differences.
- Shortcoming: Spatial resolution and dynamic range is limited.
  - Can be overcome using Adaptive Mesh Refinement (AMR)



Smooth Particle Hydrodynamics (SPH)

- Model the fluid as a collection of fluid elements represented by N particles.
- Move the particles using Lagrangian forms of the fluid equations.
- Some thermodynamic properties are assigned to each particle, *e.g.* mass and temperature.
- Other thermodynamic properties of the fluid, *e.g.* ∇P, are estimated using local averages.
- Since the computational model consists of a finite number of fluid elements, local averages must be made over volumes of finite extent.
- Local averages are made using an interpolation method that allows any function to be expressed in terms of its values at a set of disordered points—the particles.



Interpolation

- The integral interpolant of any function $f(r)$ is

$$< f(\mathbf{r}) >= \int W(\mathbf{r}-\mathbf{r}';h)\, f(\mathbf{r}')d\mathbf{r}' \tag{24}$$

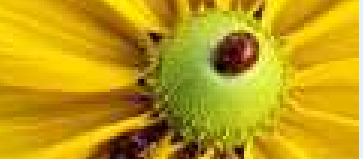
  where the integration is over all space.

- h is the smoothing length and specifies the extent of the averaging volume.

- W is the smoothing kernel which must have two properties

$$\int W(\mathbf{r}-\mathbf{r}';h)d\mathbf{r} = 1 \tag{25}$$

$$\lim_{h\to 0} W(\mathbf{r};h) = \delta(\mathbf{r}-\mathbf{r}') \tag{26}$$



Interpolation (cont)

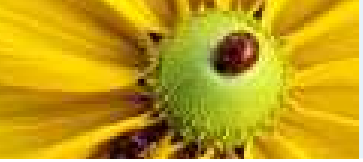
- If the values of $f(r)$ are only known at a finite number of points then

$$< f(\mathbf{r}) >= \sum_{j=1}^{N} \frac{f(\mathbf{r}_j)}{< n(\mathbf{r}_j) >} W(\mathbf{r} - \mathbf{r}_j; h). \tag{27}$$

- In particular if a mass m_j is associated with each particle then

$$< \rho(\mathbf{r}) > = \sum_{j=1}^{N} m_j W(\mathbf{r} - \mathbf{r}_j; h) \tag{28}$$

$$< f(\mathbf{r}) > = \sum_{j=1}^{N} m_j \frac{f_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j; h). \tag{29}$$



Interpolation of Gradients

- Gradients are calculated in a similar manner. By definition

$$< \nabla f(\mathbf{r}) >= \int W(\mathbf{r}-\mathbf{r}';h)\nabla f(\mathbf{r}')d\mathbf{r}'. \tag{30}$$

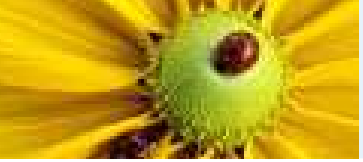
- Integrating by parts yields

$$< \nabla f(\mathbf{r}) > = \int f(\mathbf{r}')\nabla W(\mathbf{r}-\mathbf{r}';h)d\mathbf{r}' \tag{31}$$

$$< (\nabla f)_i > = \sum_{j=1}^{N} m_j \frac{f_j}{\rho_j}\nabla W(\mathbf{r}-\mathbf{r}_j;h). \tag{32}$$

- Higher accuracy is obtained by using the relation $\rho\nabla f = \nabla(\rho f) - f\nabla\rho$ giving

$$< (\nabla f)_i >= \frac{1}{\rho_i}\sum_{j=1}^{N} m_j(f_j - f_i)\nabla W(\mathbf{r}-\mathbf{r}_j;h). \tag{33}$$



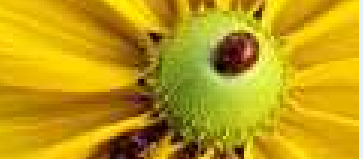
The Kernel

- $W(\mathbf{r})$ must be differentiable to at least the same order as that of the terms present in the dynamical equations.
- A common choice for the kernel is a Gaussian,

$$W(r,h) = \frac{1}{\pi^{3/2}h^3} e^{-(r^2/h^2)}. \tag{34}$$

- Another choice that has compact support is a spherically symmetric spline kernel

$$W(r,h) = \frac{1}{\pi h^3} \begin{cases} 1 - \frac{3}{2}\left(\frac{r}{h}\right)^2 + \frac{3}{4}\left(\frac{r}{h}\right)^3 & 0 \leq \frac{r}{h} \leq 1 \\ \frac{1}{4}\left(2 - \frac{r}{h}\right)^3 & 1 \leq \frac{r}{h} \leq 2 \\ 0 & \frac{r}{h} \geq 2 \end{cases}. \tag{35}$$



The Momentum Equation

- In the Lagrangian form the momentum equation is

$$\frac{d\mathbf{v}}{dt} = -\frac{\nabla P}{\rho} \tag{36}$$

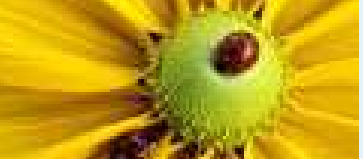
$$\frac{(\nabla \mathbf{P})_i}{\rho_i} = \sum_j m_j (P_j - P_i) \nabla_i W(r_{ij}, h). \tag{37}$$

- This form is not symmetric in i and j so would not conserve linear and angular momentum, but one can symmetrize using

$$\frac{\nabla P}{\rho} = \nabla \left(\frac{P}{\rho} \right) + \frac{P}{\rho^2} \nabla \rho \tag{38}$$

then

$$\frac{(\nabla \mathbf{P})_i}{\rho_i} = \sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W(r_{ij}, h). \tag{39}$$



The Thermal Energy Equation

- In Lagrangian form the thermal energy equation is

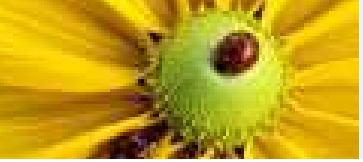
$$\frac{du}{dt} = -\left(\frac{P}{\rho}\right)\nabla\cdot\mathbf{v} + \frac{\Gamma-\Lambda}{\rho}. \tag{40}$$

- Ignoring the heating and cooling terms this can be written as

$$\frac{du_i}{dt} = \sum_{j=1}^{N} m_j \frac{P_j}{\rho_i\rho_j}\mathbf{v}_{ij}\cdot\nabla_i W(r_{ij},h). \tag{41}$$

- Often one follows the entropy A instead of the thermal energy u making energy conservation manifest and

$$u_i = \frac{P_i}{(\gamma-1)\rho_i} = (\gamma-1)^{-1}A_i\rho^{\gamma-1}. \tag{42}$$



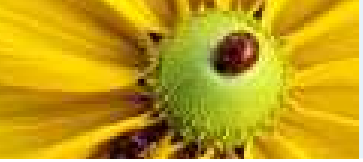
Varying Resolution

- To further increase the dynamic range it is also possible to allow h to vary spatially. Now deriving from Lagrangian Variation

$$< \rho_i(\mathbf{r}) >= \sum_{j=1}^{N} m_j W(\mathbf{r} - \mathbf{r}_j; h_i) \tag{43}$$

$$\frac{d\mathbf{v_i}}{dt} = -\sum_j m_j \left[\frac{f_i P_i}{\rho_i^2} \nabla_i W(r_{ij}, h_i) + \frac{f_j P_j}{\rho_j^2} \nabla_i W(r_{ij}, h_j) \right] \tag{44}$$

$$f_i = \left[1 + \frac{h_i}{3\rho_i} \frac{\partial \rho_i}{\partial h_i} \right]^{-1} \tag{45}$$



Artificial Viscosity

- To allow shocks to form and to prevent the interpenetration of high Mach number flows it is necessary to introduce an artificial viscosity. Add terms

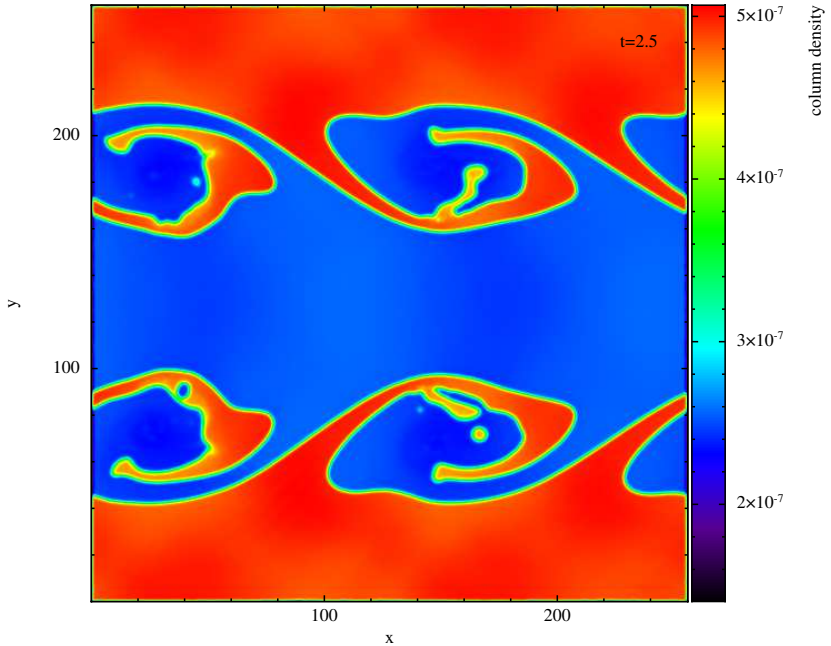
$$\left(\frac{d\mathbf{v_i}}{dt}\right)_{\text{AV}} = -\sum_j m_j \Pi_{ij} \nabla_i W(r_{ij}, h) \tag{46}$$

$$\left(\frac{du_i}{dt}\right)_{\text{AV}} = \frac{1}{2}\sum_j m_j \Pi_{ij} \mathbf{v}_{ij} \cdot \nabla_i W(r_{ij}, h) \tag{47}$$

$$\Pi_{ij} = \frac{-\alpha \mu_{ij} \bar{c}_{ij} + \beta \mu_{ij}^2}{\bar{\rho}_{ij}} \tag{48}$$

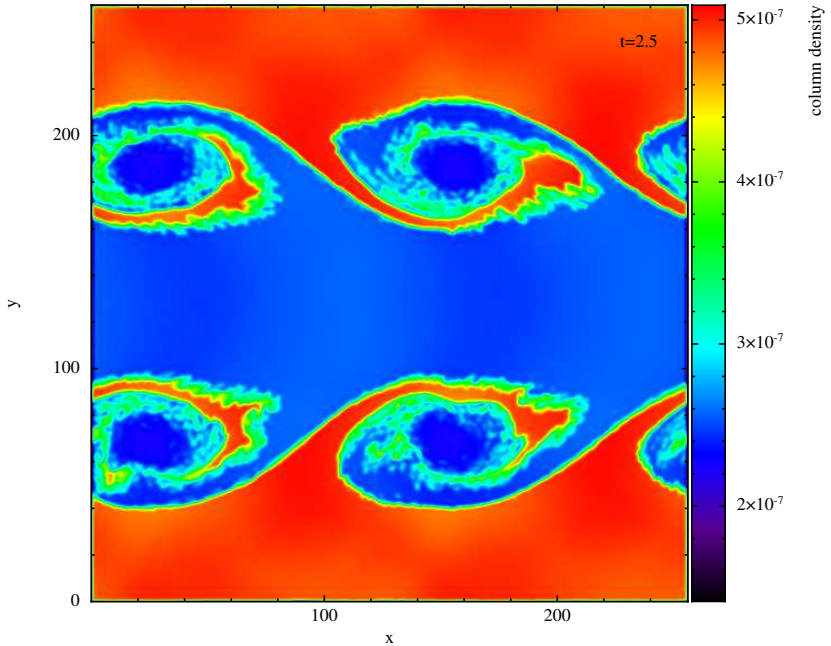
$$\mu_{ij} = \begin{cases} \frac{\mathbf{v}_{ij}\cdot\mathbf{r}_{ij}}{h_{ij}(r_{ij}^2/h_{ij}^2+\eta^2)} & \text{for} \quad \mathbf{v}_{ij}\cdot\mathbf{r}_{ij} < 0 \\ 0 & \text{for} \quad \mathbf{v}_{ij}\cdot\mathbf{r}_{ij} \geq 0 \end{cases}. \tag{49}$$

DI-SPH

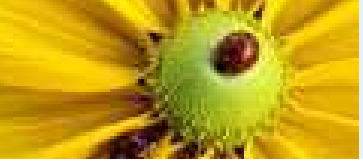


- SPH has trouble getting instabilities like the Kelvin-Helmholtz instability correct owing to artificial surface tension between phases.

DI-SPH



- SPH has trouble getting instabilities like the Kelvin-Helmholtz instability correct owing to artificial surface tension between phases.
- Added DI-SPH (Hopkins 2012) to our Gadget 3 cosmology code.
- Can get Kelvin-Helmholtz instabilities now.
- Minor changes to galaxy masses at the massive end.
- Almost no change to galaxy accretion modes.



Stellar Mass Functions

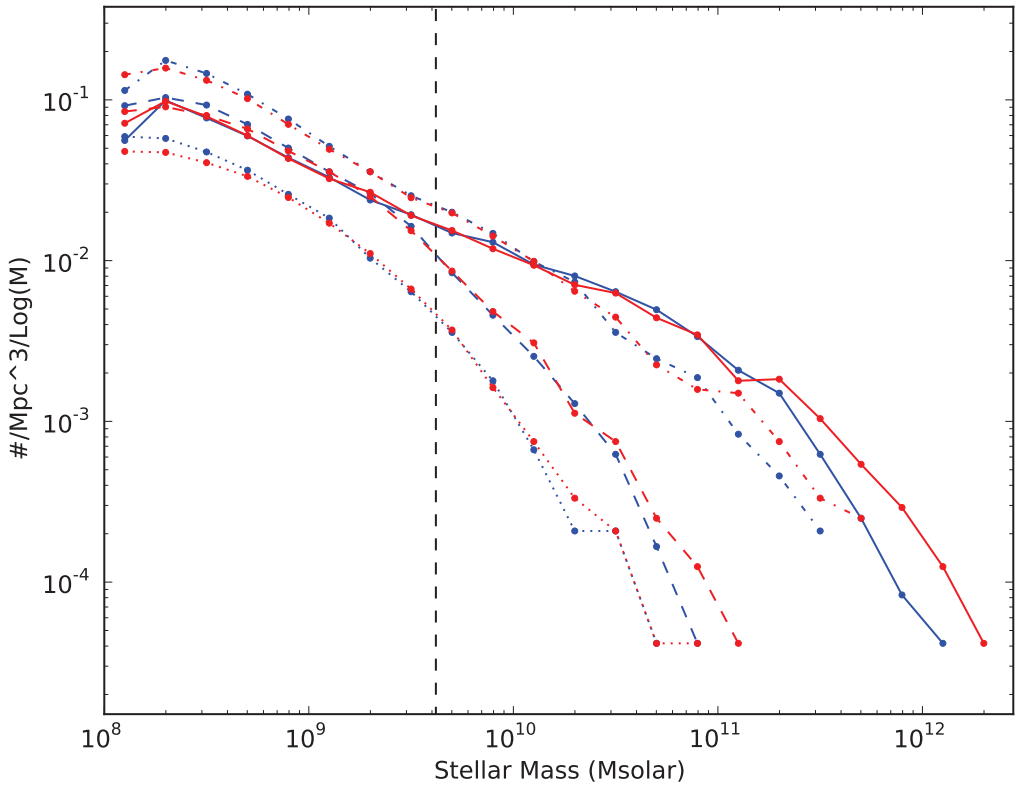


Fig. 2.— Stellar mass function at different redshift for the two simulations. Blue lines are from non-DISPH run. The dashed black line shows the mass limit of 64 gas particles. The solid, dash-dotted, dashed and dotted lines show the smf at redshifts 0, 1, 3, 4 respectively.

Accretion Modes

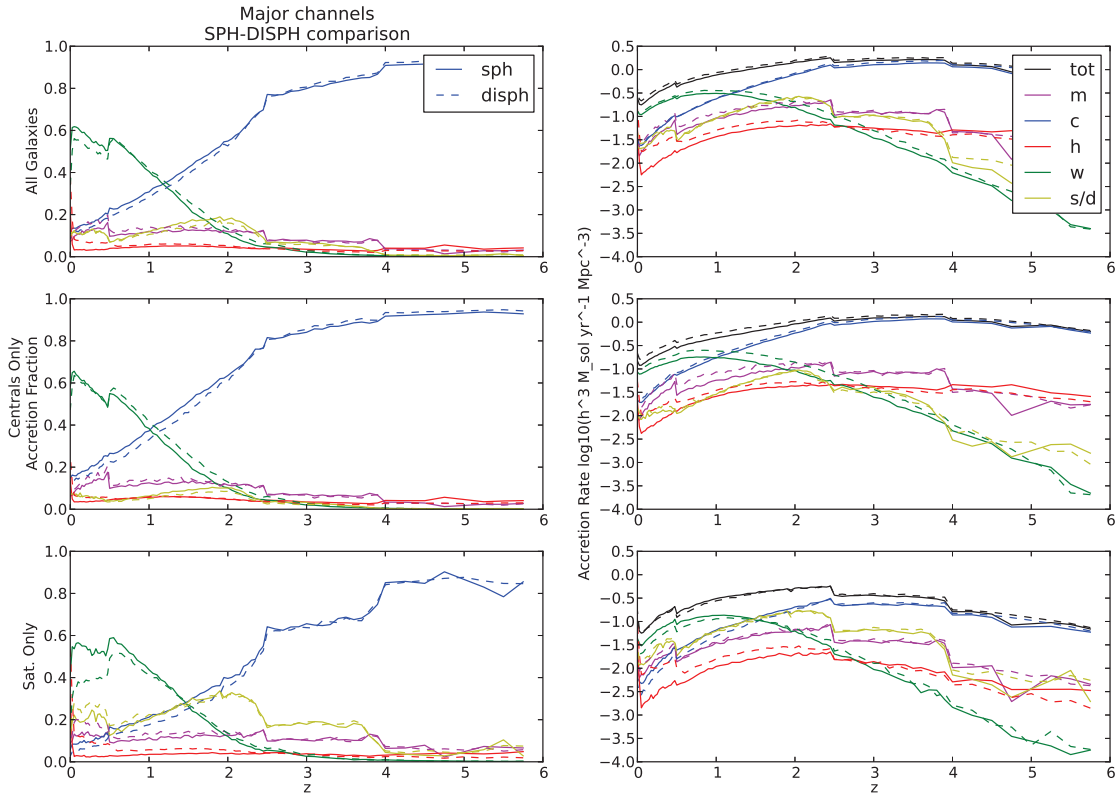


Fig. 2.1: Accretion split into the five major channels plotted as a function of redshift: cold mode (blue), hot mode (red), wind (green), stripped/disrupted (yellow), and mergers (magenta). On the left these are plotted as the fraction of accretion and the right panels are represented as an accretion rate, in $\log_{10}\left(\frac{h^3 M_{sol}}{Mpc^3 yr}\right)$. The total accretion rate is plotted in the right panels in black. This is done seperately for all galaxies (top two panels), central galaxies only (middle panels), and satellite galaxies (bottom panels). This is done for both the SPH simulation (solid line) and the DISPH simulation (dashed line).