ASTR 610
Theory of Galaxy Formation

Lecture 20: Numerical Simulations

Frank van den Bosch
Yale University, Fall 2018
Numerical Simulations

In this lecture we discuss the various methods that are used in N-body simulations of structure formation. Due to time constraints we will mainly focus on pure N-body (gravity only) simulations.

Topics that will be covered include:

- Overview of Simulation Codes
- N-body methodology
- Force Calculation
- Time Integrations
- Force Softening
- Relaxation Effects

Parts of this lecture are copied from an excellent lecture by Joshua Barnes.
Overview of Simulation Methods & Codes*

N-body (gravity only)

Collisional
($t_{\text{relax}} \ll t_{H}$)

Collisionless
($t_{\text{relax}} \gg t_{H}$)

Hydrodynamics

Particle Based
(Lagrangian)

Grid Based
(Eulerian)

Direct N-body

Tree

PM

$P^3M$

$AP^3M$

AMR

SPH

Moving grid

AMR

Nbody-6

Gadget

PKDgrav

Treecode

PMfast

ART

Ramses

AMIGA

Gadget-2

Gasoline

Arepo

Ramses

Enzo

ART

AMIGA

Flash

*NOTE: this overview is by no means exhaustive!!
Collisionless Dynamics

The equations governing a collisionless system are the Vlasov-Poisson equations:

\[
\rho(\vec{x}, t) = \int d\vec{v} f(\vec{x}, \vec{v}) \\
\Phi(\vec{x}, t) = -G \int d\vec{x}' \frac{\rho(\vec{x}', t)}{|\vec{x} - \vec{x}'|} \\
\frac{df}{dt} = \frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} - \frac{\partial \Phi}{\partial \vec{x}} \cdot \frac{\partial f}{\partial \vec{v}} = 0
\]

The direct numerical solution of the CBE, a non-linear PDE in seven dimensions, is not feasible. The main problems are:

- The 6D grid on which to solve the CBE takes too many cells (for given resolution)
- Under CBE, the DF develops ever stronger gradients; initial fluctuations are not averaged away, but mixed, leading to ever thinner layers of phase-space density.
Effect of Mixing in Collisionless System

Phase-mixing of $10^4$ points in simple 1D Hamiltonian. The fine-grained DF is initially either 0 or 1, but at late times a smooth coarse-grained DF develops...
The N-body methodology

Rather than solving the Vlasov-Poisson equations on a 6D grid, we use Monte-Carlo techniques to solve the equations of motion for a sample of N `bodies'.

Replace smooth DF with N bodies:

\[
f(\vec{x}, \vec{v}) \rightarrow \{(m_i, \vec{x}_i, \vec{v}_i)|i = 1, 2, ..., N\}
\]

\[
f(\vec{x}, \vec{v}) \approx \sum_{i=1}^{N} m_i \delta^3(\vec{x} - \vec{x}_i) \delta^3(\vec{v} - \vec{v}_i)
\]

Select \((\vec{x}_i, \vec{v}_i)\) with probability proportional to the DF, and assign all bodies equal mass:

\[
m_i = \frac{1}{N} \int d\vec{x} d\vec{v} f(\vec{x}, \vec{v})
\]
Advancing Time

Move bodies along phase flow (method of characteristics)

\[(\ddot{x}_i, \ddot{v}_i) = (\dot{v}_i, -\nabla \Phi_i)\]

Estimate potential from N-body representation using Poisson equation:

\[\nabla^2 \Phi|_{\vec{x}} = 4\pi G \sum_{i=1}^{N} m_i \delta^3(\vec{x} - \vec{x}_i)\]

This will yield the usual N-body equation for point masses. But, singular potentials are awkward, so we smooth the density field.

\[\delta^3(\vec{x} - \vec{x}_i) \rightarrow \frac{3}{4\pi} \frac{\varepsilon^2}{(|\vec{x} - \vec{x}_i|^2 + \varepsilon^2)^{5/2}}\]

Plummer (1911) smoothing; See Dehnen+01 for other smoothing kernels...

This substitution yields the following equations of motion:

\[\frac{d\vec{x}_i}{dt} = \vec{v}_i\]
\[\frac{d\vec{v}_i}{dt} = \sum_{j\neq i}^{N} \frac{Gm_j(\vec{x}_j - \vec{x}_i)}{(|\vec{x}_j - \vec{x}_i|^2 + \varepsilon^2)^{3/2}}\]

\[\varepsilon\] is called the softening length
Relaxation Time

N-body models relax (undergo gravitational `collisions’, aka `encounters’). The rate at which this two-body relaxation occurs is given by relaxation time:

\[
t_{\text{relax}} \sim \frac{N}{8 \ln(R/\varepsilon)} t_{\text{cross}} \\
t_{\text{cross}} \sim \frac{R}{\sigma_v}
\]

Since \( N \) is typically many orders of magnitude smaller in N-body representation than in real system, relaxation time of simulated system is orders of magnitude smaller than for real system.

Ex: dark matter halo \( N_{\text{WIMP}} \sim 10^{70} \), whereas \( N_p \approx 10^9 \)

Ansatz: as long as \( t_{\text{relax}} \gg t_H \) (two-body) relaxation shouldn’t be an issue...

NOTE: (artificially short) relaxation time depends only very weakly on softening length. Softening is NOT introduced to eliminate relaxation effects; it only suppresses the large-angle `scattering' due to close encounters... For uniform density, each octave in impact parameter contributes equally to relaxation!!!
### Force Calculation

**Particle-Particle (PP):** simply sum over all other bodies

\[
\vec{F}_i = \frac{d\vec{v}_i}{dt} = \sum_{j \neq i}^{N} \frac{G m_j (\vec{x}_j - \vec{x}_i)}{(|\vec{x}_j - \vec{x}_i|^2 + \varepsilon^2)^{3/2}}
\]

**Advantage** is that this is robust, accurate and completely general.

**Disadvantage** is that for each particle, it requires a sum over all N bodies, such that cost of computing forces on all particles requires \(O(N^2)\) operations.

This method is far too slow/costly, and therefore rarely used to simulate collisionless systems. However, it is *the* method to simulate collisional systems such as globular clusters or (open) star clusters...
Force Calculation

**Particle-Mesh (PM) method:**

[Hockney & Eastwood 1988]

[1] Divide computational box (size $L$) in grid of $N_c^3$ meshes of constant size $L/N_c$


$$
\rho(q) = \frac{m}{L^3} \sum_{i=1}^{N} W(\vec{x}_i - \vec{X}_q)
$$

$q$ is mesh location

Here $W(r)$ is a (normalized) kernel function. Note that this density is a convolution of particle distribution with kernel function, and so its Fourier transform is equal to product of $\hat{W}_k$ and FT of particle density distribution.

[3] Solve for potential on mesh using the Poisson equation (in Fourier Space)

$$
-k^2 \phi_k = 4\pi G \rho_k
$$

Periodic boundary conditions $\rightarrow$ FFT

Force per unit mass at grid points:

$$
F_k = -i\phi_k \hat{k}
$$
Force Calculation

Particle-Mesh (PM) method: (Hockney & Eastwood 1988)

[4] Compute each particle acceleration using inverse CIC interpolation scheme

\[ F(\vec{x}_i) = \sum_{\vec{q}} W(\vec{x}_i - \vec{X}_q) F(\vec{q}) \]

[5] Update each particle velocity according to its acceleration

[6] Update each particle position according to its velocity

Advantages:
- Fast; computational cost for force calculation only \( O(N) \)
- Consequently, large numbers of particles can be used

Disadvantages:
- Choice of kernel function is non-trivial and can affect numerical accuracy
- Force resolution is limited by size of mesh. Large \( N_c \) required for accuracy
- As clustering develops, forces become less and less accurate
Particle-Particle-Particle-Mesh ($P^3M$) method:

pronounce: $P$-cube-$M$

$P^3M$ combines the advantages of $PP$ and $PM$ methods:

- long-range forces ($d < 2 \frac{L}{N_c}$) are computed using $PM$ (fast)
- short-range forces ($d > 2 \frac{L}{N_c}$) are computed using $PP$ (accurate)

If particles are not strongly clustered, most particle pairs are solved using $PM$.
Once particles become strongly clustered, code becomes effectively $PP$.

This problem can be avoided with the use of adaptive sub-meshes; $AP^3M$
(e.g. Couchman 1991)

With sufficiently adaptive grid, forces on all particles can be computed on a grid, in which case the code becomes essentially an $AMR$ (adaptive mesh refinement) code. Different $AMR$ codes (e.g., $ART$, $AMIGA$, $RAMSES$) mainly differ in how refinement is handled....
Tree-Algorithm

Group particles according to distance from particle under consideration.

Most efficient way to do this, such that one can use same structure for each particle, is via a `tree-structure'.

Force from each group is then replaced by its multipole expansion; since higher-order terms die off faster with distance, one only needs to keep lowest order terms (typically up to quadrupole or hexadecapole).

Long-range gravitational field dominated by monopole term:

\[ \Phi \simeq -\frac{Gm}{r} + \mathcal{O}(r^{-3}) \]
Force Calculation

Tree-Algorithm

In a tree-code one replaces the sum over $N$ bodies with a sum over $N_c \sim O(\log N)$ cells. Hence computational cost of force calculations is $O(N\log N)$

Different tree-codes use different tree-structures

Octree:
- each refinement, cell sizes are cut in half, such that one cell becomes 8 equal-volume cells (e.g., treecode)

kd-tree:
- each refinement, cells are divided at median along the x, y, and z axes. (e.g., PKDgrav)
Force Calculation

Tree-Algorithm

In a tree-code one replaces the sum over $N$ bodies with a sum over $N_c \sim O(\log N)$ cells. Hence computational cost of force calculations is $O(N \log N)$.

In order to compute the force on particle $i$, one proceeds as follows.

**LOOP** over all cells of the lowest refinement level.

**IF** $|\vec{x}_i - \vec{x}_c| > l_c / \theta$ **THEN**

compute force due to this cell (using low-order multipole expansion)

**ELSE**

sum the contributions of its sub-cells

**END IF**

Here $\vec{x}_c$ is the center of mass of the cell, $l_c$ is the size of the cell, and $\theta$ is a free parameter, called the opening-angle (typically $\theta \approx 0.6 - 1.0$).

Smaller $\theta$ results in more accurate forces, but at increased CPU cost.
Self-Consistent Field Method

Represent potential and density as linear sums of basis functions:

\[
\Phi(\vec{x}) = \sum_k A_k \Phi_k(\vec{x}) \quad \rho(\vec{x}) = \sum_k A_k \rho_k(\vec{x})
\]

Here \(A_k\) are coefficients and the basis functions \(\Phi_k\) and \(\rho_k\) are bi-orthogonal and satisfy the Poisson equation:

\[
I_k \delta_{kk'} = \int d\vec{x} \rho_k(\vec{x}) \left[ \Phi_{k'}(\vec{x}) \right]^* \quad \nabla^2 \Phi_k = 4\pi G \rho_k
\]

The coefficients are computed using:

\[
A_k = \frac{1}{I_k} \int d\vec{x} \rho(\vec{x}) \left[ \Phi_k(\vec{x}) \right]^* = \frac{1}{I_k} \sum_i m_i \left[ \Phi_k(\vec{x}_i) \right]^*
\]

The cost of calculating forces on all bodies is just \(O(N)\)

Orbits are integrated in smooth potential; no softening required!

Method can only be used if useful set of basis functions can be found, which is typically only the case for highly symmetric systems....
Once you know the forces, you need to integrate the equations of motion forward in time. Various time-integration methods are employed.

We begin by considering the simple `Euler method’ for a time step $\Delta t$

\[
\begin{aligned}
  x_i^{[n+1]} &= x_i^{[n]} + v_i^{[n]} \cdot \Delta t \\
v_i^{[n+1]} &= v_i^{[n]} + a_i^{[n]} \cdot \Delta t
\end{aligned}
\]

simple, but poor accuracy $O(\Delta t^2)$

NOTE: Euler method is simply based on first-order Taylor series expansion.
We can improve on accuracy by using a higher-order integration scheme.

Second-order Leapfrog Integrator

\[
\begin{aligned}
v_i^{[n+1/2]} &= v_i^{[n]} + \frac{1}{2} a_i^{[n]} \cdot \Delta t \\
x_i^{[n+1]} &= x_i^{[n]} + v_i^{[n+1/2]} \cdot \Delta t \\
v_i^{[n+1]} &= v_i^{[n+1/2]} + \frac{1}{2} a_i^{[n+1]} \cdot \Delta t
\end{aligned}
\]

accuracy $O(\Delta t^3)$

this integration scheme is sometimes called KDK, for Kick-Drift-Kick
Time Integration

The second-order leapfrog integrator is symplectic. This means that it exactly solves an approximate Hamiltonian, which implies that numerical time evolution is a canonical map preserving certain conserved quantities exactly.

Consequently, leapfrog integrator is more stable than other (non-symplectic) integrators. It is the most commonly used integrator for collisionless systems.

In principle, one may combine as many kick and drift operations as desired to raise order of integration scheme. However, it is impossible to go beyond second order without having some terms be negative (corresponding to backwards integration). This is problematic when variable time steps are required....

See Dehnen & Read (2011) for a more details
The Choice of Time Step

Given the enormous range of dynamical times involved in typical simulations, it has become essential to use variable time step schemes:

Most of these adopt a hierarchy of time steps organized in powers of two:

\[ \Delta t_n = \Delta t_0 / 2^n \]

\( n \) is called the rung of the time step

Particles can move to higher rung (smaller time step) whenever they like (meet time step criterion), but they may only move to lower rung at synchronisation points (red arrows in figure). Resulting asymmetry brakes symplectic nature of integrator...

Fixed time step is more accurate, but often too costly...

An often used time step criterion is the following

\[ \Delta t_i < \eta \sqrt{\varepsilon / |\dot{a}_i|} \]

\( \eta \) free parameter (dimensionless)

\( \varepsilon \) softening length

Not well motivated (only on dimensional grounds); many alternatives exist, and most codes allow multiple choices...
Cosmological N-body Simulations

N-body simulations are used to study non-linear dynamical evolution:

Over the years, they have been used to

- study large scale structure (vindication of CDM model)
- probe evolution of non-linear, matter power spectrum
- establish a universal (NFW) density profile of CDM halos
- predict/quantify substructure of dark matter halos
- predict/quantify mass/velocity function of CDM halos

N-body simulations are routinely used as prime tool to address fundamental questions in astrophysics:

- nature of dark energy (determine growth rate of structures)
- nature of dark matter (determines small-scale structure)

It is crucial that we continue to scrutinize simulations
Large Scale Structure: the CDM vindication

Springel+06
In what follows I focus exclusively on *collisionless* dark matter.

Goal of simulations is to solve the *Vlasov-Poisson equations*, describing the evolution of *collisionless* system.

In N-body simulations, matter field is represented by set of $N$ discrete particles. These do not represent physical objects. Rather they represent phase-space elements sampling the DF $f(x, v, t)$.

- Simulation is typically performed over a finite, periodic volume $V = L^3$.
- Simulations have finite mass resolution: $m_p = \bar{\rho} V / N$
- Simulations have finite force resolution, $\epsilon$, which is required to suppress two-body collision effects.
- Initial conditions typically only sample modes of the power spectrum below the particle Nyquist frequency $k_{\text{max}} = \pi / d$ \hspace{1cm} (d = L/N^{1/3})
How to test numerical simulations?

- **Ideal**: simulate systems for which you have an analytical solution
  - very rare in non-linear structure formation...

- Compare simulations against simulations
  - convergence; a necessary condition, but not sufficient...
    - different quantities (halo mass function, matter power spectrum, halo density profile, subhalo mass function, etc) all converge differently.

  \[
  K = \frac{\langle \delta_1 \delta_2 \rangle}{\sigma_1 \sigma_2}
  \]
  - (1 and 2 refer to different simulations with different numerical parameters)

- Look for signatures of collisional relaxation
  - mass segregation (use ≥2 particle species with different masses)
  - creation of isothermal cores in dark matter halos

ASTR 610: Theory of Galaxy Formation © Frank van den Bosch: Yale
A `natural' choice for $\varepsilon$ is the mean interparticle separation $d=L/N^{1/3}$

In what follows, $\varepsilon$ is normalized to $d$ ($\varepsilon=\varepsilon/d$)

Modern `high-res' codes ($P^3M$, tree-codes, AMR) typically use $\varepsilon \sim 0.01-0.03$

Efstathiou & Eastwood (1981):
$P^3M$ simulations become collisional
(i.e., reveal mass segregation) if $\varepsilon<0.1$

Peebles (1989): $PM$ codes require $\varepsilon \sim 1$

Melott et al. (1997):
$N$-body codes in general require $\varepsilon \sim 1$

Lively, ongoing debate whether simulations with $\varepsilon < 1$ are reliable...
(Kuhlman+96; Splinter+98; Knebe+00; Melott 07; Romeo+08; Joyce+09; Benhaim+16; Power+16)

What is the optimal softening length? (Athanassoula+00; Dehnen 01; Power+03)

At the very least, softening should be adaptive... (Iannuzzi & Dolag 11; Hobbs+15)
Warm Dark Matter simulations show `beats-on-a-string’ halos within filaments. These structures form on scales smaller than cut-off scale in power spectrum. (Bode, Ostriker & Turok 2001; Knebe+02)

Initially interpreted as due to (physical) fragmentation (Knebe+03)
Spurious Fragmentation

Beats-on-a-string halos are manifestation of spurious fragmentation.

Spacing of artificial halos equal to grid-spacing. Suggests link to regular, cubic lattice used for the initial particle load...

(Götz & Sommer-Larsen 2002, 2003)

But, spurious fragmentation also present with glass-like initial particle load, which has no preferred direction, and no long-range order.

(Wang & White 2007)

Spurious fragmentation now understood as arising from discreteness-induced velocity perturbations during early highly-anisotropic phase of structure formation.

(Hahn & Angulo 2016; Power+16)

This is exactly the artefact that has been discussed again and again, since 1990, by Melott, Shandarin and collaborators!!

(see also Romeo+08; Joyce+09; Benhaiem+16; Power+16 and references therein)
Abundance & demographics of dark matter substructure depends sensitively on nature of dark matter: \textit{CDM vs WDM vs SIDM}.

Different models mainly differ in abundance of low mass halos, where galaxy formation is expected to be suppressed due to re-ionization.

\textbf{WDM simulations suffer from artificial fragmentation; cannot be avoided.}

\textit{State-of-the-Art `solution': remove spurious halos `by hand'}

(e.g., Schneider+13; Lovell+14; Bose+16)
If spurious fragmentation is an outcome of discreteness relaxation, shouldn’t it also be present in CDM simulations?
Collisionality gives rise to velocity perturbations that `thicken' the sheet and fragment into clumpy structures in phase-space (`spurious halos').
Spurious Fragmentation in CDM

When $\epsilon < 1$, CDM simulations reveal a `fog' of low mass halos, both in filaments and in the field: spurious or real???
Spurious Fragmentation in CDM

**Why would spurious fragmentation not occur, or not matter, for CDM?**

- **CDM** does **NOT** suffer from spurious fragmentation, because:
  - no upturn, as for **WDM**
  - agreement with **EPS**-predictions
  - convergence; running at higher force resolution yields consistent results...

**My View**

- **It does** happen, with similar abundance of spurious halos as for **WDM**
- **Real halos** dominate $\Rightarrow$ no significant impact on **CDM** mass function
- **But what about internal structure of halos? Is NFW reliable?**

---

**optimist’s view**

**reality check...**

**BUT: WDM results have also converged. But converged to garbage....**

---

**ASTR 610: Theory of Galaxy Formation**

© Frank van den Bosch: Yale
Chandrasekhar (1943) derived change in orbit by summing over all (independent) two-body scatterings with all other particles in homogeneous system.

Roughly equal contribution from every decade in impact parameter:

\[
\Gamma_{\text{relax}} \propto \frac{\ln \Lambda}{N}, \quad t_{\text{relax}} \approx \frac{N}{8 \ln \Lambda} t_{\text{cross}}, \quad \Delta = \frac{b_{\text{max}}}{b_{\text{min}}}, \quad b_{\text{min}} = b_{90^\circ} \approx \frac{2Gm_\odot}{\sigma^2}, \\
\]

Two-body relaxation generally deemed unimportant, since \( t_{\text{relax}} > t_H \), for \( N \gtrsim 100 \)

This `standard' treatment of relaxation ignores three important points:

- each halo starts out small, when it is subject to severe relaxation
- orbits are quasi-periodic, giving rise to resonant effects
- self-gravity of large-scale fluctuations [responsible for spurious fragmentation]
Discreteness-Driven Relaxation

Self-Consistent Field (SCF) method, which uses basis-functions to compute gravitational potential (no softening required) suffer from same amount of relaxation as regular N-body codes (e.g, tree-code)!!

Dominant contribution to relaxation arises from non-local, collective modes of order the size of system in question. (Weinberg 1993)
Discreteness-Driven Relaxation

Poisson fluctuations cause fluctuations in large-scale potential, which drives relaxation (akin to violent relaxation).

In SCF method, this is evident from rapid fluctuations in the amplitude of zero-th expansion coefficient.

Weinberg (1993): during the initial collapse phase, Poisson fluctuations may contribute relaxation that is factor 10-100 larger than what is predicted by local (i.e., Chandrasekhar) theory.

- Softening only suppresses impact of large-angle scattering events; not impact of small-angle scattering (large-impact parameters)
- Softening has little to no impact on these large-scale relaxation processes
- Only way to suppress these is by increasing number of particles
Progenitors of every halo start out small, and thus experience periods during which relaxation rate is large...

Even in a halo with $N=650,000$ (at $z=0$), the cumulative relaxation is such that $|\Delta E| > E$.

All knowledge about initial conditions has been erased due to two-body relaxation!! And this doesn’t even account for collective relaxation, which may well dominate...

**Fokker-Planck estimate**

$$\Gamma_{\text{relax}} = \frac{G^2 m_p \rho \ln \Lambda}{\sigma^3}$$

$$d(t) = \frac{\Delta E^2(t)}{E^2} = \int \Gamma_{\text{relax}}(t) \, dt = \sum_{i=1}^{N_{\text{step}}} \Gamma_{\text{relax}}(t_i) \Delta t$$

**Diemand+04 integrated cumulative impact of relaxation for individual particles in cosmological N-body simulation.**
Well, what do you think of simulations now?

I think they are very relaxing.