

Numerical simulations of gravitational dynamics

Eugene Vasiliev

Oxford University &
Lebedev Physical Institute

XII School of modern astrophysics 2016

Lecture 1. Introduction

- ▶ Motion in a gravitational field
- ▶ Two-body relaxation
- ▶ Mean-field approach
- ▶ Collisional and collisionless systems
- ▶ Historical and literature overview
- ▶ Physics of self-gravitating systems

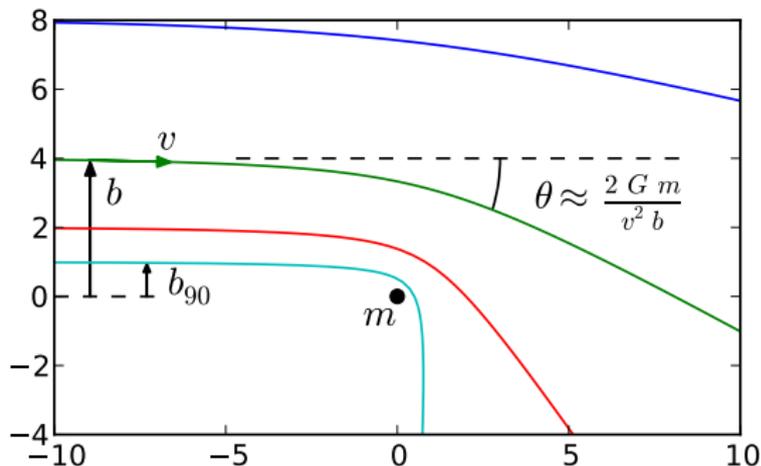
Newton's law of gravity

$$\Phi(\mathbf{r}) = -\frac{G m_1}{|\mathbf{r} - \mathbf{r}_1|}$$

Newton's law of gravity

$$\Phi(\mathbf{r}) = -\frac{G m_1}{|\mathbf{r} - \mathbf{r}_1|}$$

Gravitational deflection:
$$\delta v_{\perp} = \int_{-\infty}^{\infty} \frac{dt G m b}{[b^2 + (vt)^2]^{3/2}} = \frac{2 G m}{v b}$$



Relaxation time

A particle traveling through a uniform-density background of point masses with number density n experiences uncorrelated encounters with a rate (per unit time) $\int d\sigma n v$, where $d\sigma = 2\pi b db$ is the differential cross-section for encounters with impact parameter b . The mean-square rate of velocity change is

$$\frac{d}{dt}(\delta v)^2 = \int 2\pi b db n v \left(\frac{2Gm}{vb}\right)^2 = \frac{8\pi G^2 m^2 n}{v} \int_{b_{\min}}^{b_{\max}} \frac{db}{b}$$

Coulomb logarithm $\ln \Lambda$

The relaxation time is

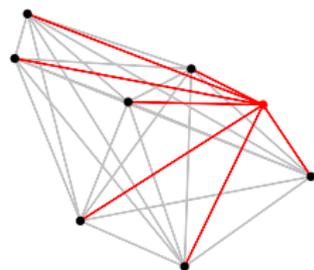
$$T_{\text{rel}} \equiv \frac{v^2}{d(\delta v)^2/dt} = \frac{v^3}{8\pi G^2 m^2 n \ln \Lambda} \sim \frac{N}{8 \ln \Lambda} T_{\text{cross}},$$

where $T_{\text{cross}} \equiv \frac{R}{v} \sim \frac{R}{\sqrt{G Nm/R}}$ is the crossing time.

The N-body problem

The equations of motion for a system of N point masses:

$$\ddot{\mathbf{x}}_i = \mathbf{a}_i \equiv - \sum_{j=1, j \neq i}^N G m_j \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|^3}, \quad i = 1..N,$$



or an equivalent system of first-order ODEs:

$$\dot{\mathbf{x}}_i = \mathbf{v}_i, \quad \dot{\mathbf{v}}_i = \mathbf{a}_i.$$

Complexity:

N particles \times N interactions per timestep Δt ,

$\Delta t \lesssim \min_{i \neq j} |\mathbf{x}_i - \mathbf{x}_j| / v \propto N^{-1/3}$ for a fixed size of the system

$\implies \propto N^{7/3}$ operations per dynamical time!

Dynamical range

- ▶ Solar system: $N \lesssim 10$, $T_{\text{cross}} \sim 1$ yr, age $\sim 10^{10}$ yr.
- ▶ Globular cluster: $R \sim 10$ pc, $v \sim 10$ km/s $\Rightarrow T_{\text{cross}} \sim 10^6$ yr;
 $N \sim 10^5 \Rightarrow T_{\text{rel}} \sim 10^{10}$ yr (typically less);
a close binary in a cluster might have an orbital period $\lesssim 1$ day!
- ▶ Nuclear star cluster in the center of Milky Way:
 $R \sim 1$ pc, $N \sim 10^6$, shortest orbital period $T \sim 10$ yr.
- ▶ Galaxy: $R \sim 10$ kpc, $v \sim 100$ km/s $\Rightarrow T_{\text{cross}} \sim 10^8$ yr;
 $N \sim 10^{11} \Rightarrow T_{\text{rel}} \gg T_{\text{Hubble}}$.
- ▶ Cosmology – Hubble volume: $R \sim 10^{10}$ pc, $N \sim 10^{12}$ galaxies,
but you'd like the galaxies to be more than just point masses...
- ▶ Dark matter halo of Milky Way:
 $M \sim 10^{12} M_{\odot}$, $m_{\text{particle}} \sim 10^{2\pm???$ GeV $\Rightarrow N \sim 10^{67}$!..

Mean-field approach

Instead of dealing with individual gravitating masses, consider the distribution function: $f(\mathbf{x}, \mathbf{v}; \alpha)$.

The number (or mass) of stars of α -th kind in the volume $d^3x d^3v$ of phase space is $f(\mathbf{x}, \mathbf{v}; \alpha) d^3x d^3v$.

f satisfies the **collisionless** Boltzmann (or Vlasov) equation (CBE):

$$0 = \frac{df}{dt} \equiv \frac{\partial f}{\partial t} + \frac{d\mathbf{x}}{dt} \frac{\partial f}{\partial \mathbf{x}} + \frac{d\mathbf{v}}{dt} \frac{\partial f}{\partial \mathbf{v}} = \frac{\partial f}{\partial t} + \mathbf{v} \frac{\partial f}{\partial \mathbf{x}} - \frac{\partial \Phi}{\partial \mathbf{x}} \frac{\partial f}{\partial \mathbf{v}}.$$

Gravitational potential Φ satisfies the Poisson equation:

$$\nabla^2 \Phi(\mathbf{x}) = 4\pi G \rho(\mathbf{x}),$$

while the density ρ is given by

$$\rho(\mathbf{x}) = \iiint d^3v f(\mathbf{x}, \mathbf{v}).$$

Mean-field approach – II

How to solve the Vlasov–Poisson system?

The characteristics of CBE are simply the trajectories of particles in the given gravitational potential:

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}, \quad \frac{d\mathbf{v}}{dt} = -\frac{\partial\Phi}{\partial\mathbf{x}}.$$

Thus one possible approach is to take N particles representing $f(\mathbf{x}, \mathbf{v})$ as samples from the probability distribution function, and evolve them using a conventional N -body method.

Important to keep in mind:

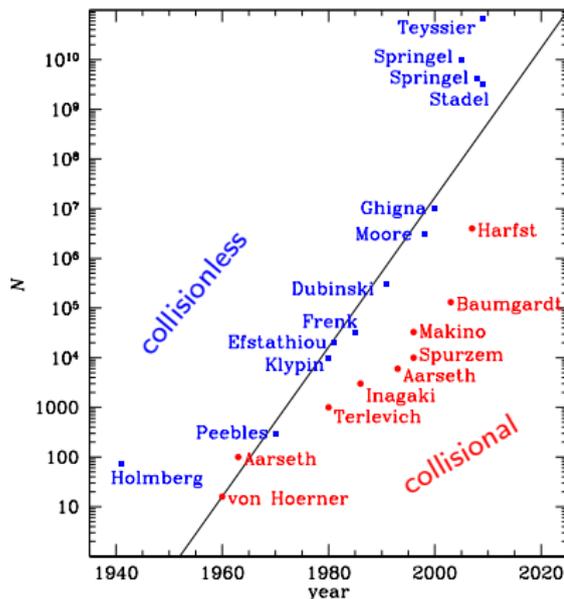
- ▶ These particles do not have physical sense themselves!
- ▶ N is a numerical parameter unrelated to the actual number of stars;
- ▶ N determines the mass and spatial resolution;
- ▶ N should be large enough that the relaxation is unimportant;
- ▶ f is estimated from the discrete samples
(limits the accuracy with which the original CBE–Poisson system is solved);
- ▶ close encounters should not be allowed.

Two classes of physical systems and N-body methods

- ▶ Collisional systems:
 - ▶ Relaxation *may be* (and usually is) important;
 - ▶ Correlations (e.g. binary stars) *may exist*;
 - ▶ Particles of the N -body system should* have 1:1 correspondence to real objects;
 - ▶ Forces should be computed accurately;
 - ▶ Equations of motion should be solved accurately.
- ▶ Collisionless systems:
 - ▶ Relaxation *should not* be important;
 - ▶ The physical system is described in terms of smooth functions f and Φ , not in terms of individual objects;
 - ▶ N should be as large as possible;
 - ▶ May use more approximate methods for force computation and integration of equations of motion.
- ▶ Grey zone...

History

- ▶ Analog computation [Holmberg 1941]
- ▶ First computer simulations [1960s]
- ▶ Regularization, neighbor scheme [1970s]
- ▶ Tree-code and grid codes [1980s]
- ▶ Special-purpose hardware [1990s]
- ▶ Modern parallel codes [2000s]
- ▶ Physics beyond pure gravity [2010s]



Literature

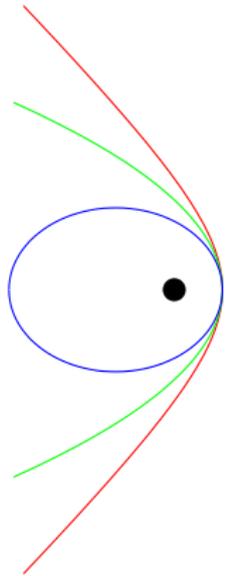
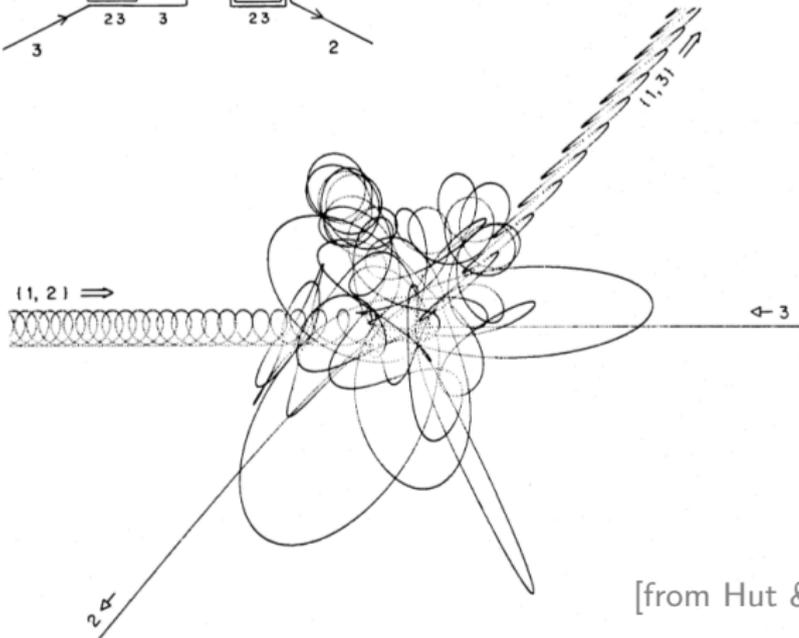
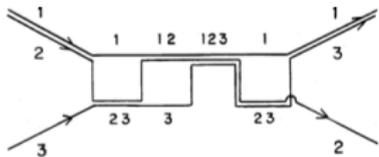
- ▶ M.Trenti, P.Hut, “Gravitational N -body simulations”, arXiv:0806.3950, or http://www.scholarpedia.org/article/N-body_simulations
- ▶ W.Dehnen, J.Read, “ N -body simulations of gravitational dynamics”, arXiv:1105.1082
- ▶ V.Springel, “High-performance computing and numerical modelling” (lecture notes), arXiv:1412.5187
- ▶ P.Hut, J.Makino, “The art of computational science”, <http://www.artcompsci.org/>
- ▶ S.Aarseth, “Gravitational N -body simulations: tools and algorithms” (2003, Cambridge Univ. press)
- ▶ S.Aarseth, C.Tout, R.Mardling, et al., “The Cambridge N -body lectures” (2008, Lecture Notes in Physics vol.760, Springer)
- ▶ D.Heggie, P.Hut, “The Gravitational million-body problem” (2003, Cambridge Univ. press)
- ▶ J.Binney, S.Tremaine, “Galactic Dynamics” (2nd ed., 2008, Princeton Univ. press)

Showcase of gravitating systems

- ▶ Two-body motion: unbound and bound;
- ▶ Three-body: chaotic; binary–single scattering;
- ▶ 1000-body: statistical equilibrium;
- ▶ 1000-body, radial orbits: instability, collective effects;
- ▶ Same for the spiral structure formation;
- ▶ Planetary systems; secular evolution, resonances;
- ▶ Thermodynamical evolution: 1000-body, core collapse;
- ▶ Galaxy interactions; dynamical friction, mergers;
- ▶ Cosmological structure formation.

N -body problem for small N

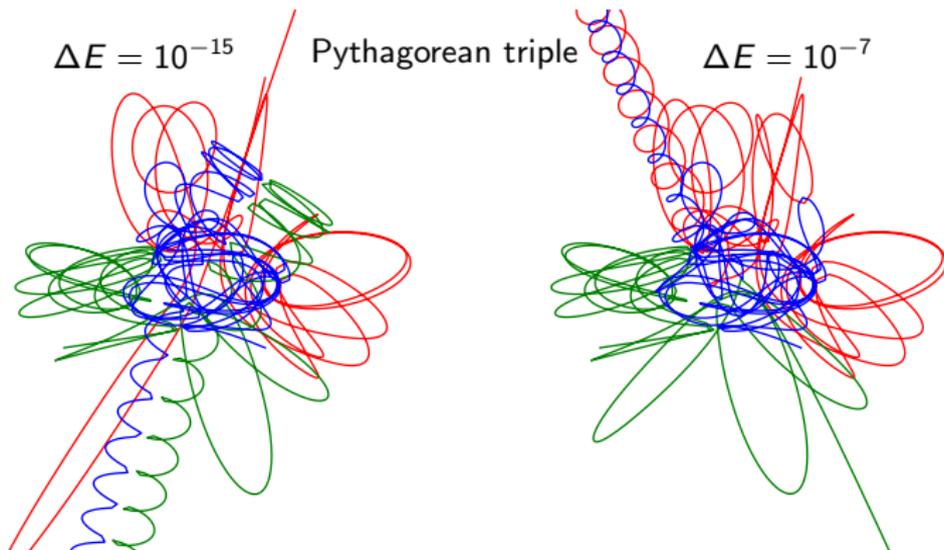
- ▶ One-body problem: piece of cake!
- ▶ Two-body problem: Kepler orbits.
- ▶ Three-body problem: here the fun starts.



[from Hut & Bahcall 1983]

Chaotic nature of N-body problem

The N -body problem is generally chaotic for $N > 2$;
the timescale for exponential divergence is $\sim T_{\text{cross}}/\ln N$.



Nevertheless, the divergence saturates at smaller scales as N increases [e.g., Kandrup & Sideris 2001], and simulations are valid in the statistical sense [e.g., Boekholt & Portegies Zwart 2015].

Virial theorem

Collisionless Boltzmann equation: $\frac{\partial f}{\partial t} + \mathbf{v} \frac{\partial f}{\partial \mathbf{x}} - \frac{\partial \Phi}{\partial \mathbf{x}} \frac{\partial f}{\partial \mathbf{v}} = 0$.

Moment of inertia: $I \equiv \int m f r^2 d^3x d^3v$.

$$\dot{I} = \int m \left(-\mathbf{v} \frac{\partial f}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{v}} \right) d^3x d^3v = 2 \int m \mathbf{x} \cdot \mathbf{v} f d^3x d^3v.$$

$$\ddot{I} = 2 \int m v^2 f d^3x d^3v - 2 \int m \mathbf{x} \cdot \frac{\partial \Phi}{\partial \mathbf{x}} f d^3x d^3v = 4K + 2W,$$

where K and W are the kinetic and potential energy of the system.

In a dynamical equilibrium, $\ddot{I} = 0$, hence $2K + W = 0$.

Since $K + W = E$, this implies $W = 2E$, $K = -E$.

Virial ratio: $Q \equiv 2K/|W| \approx 1$ in a virial equilibrium.

N -body units

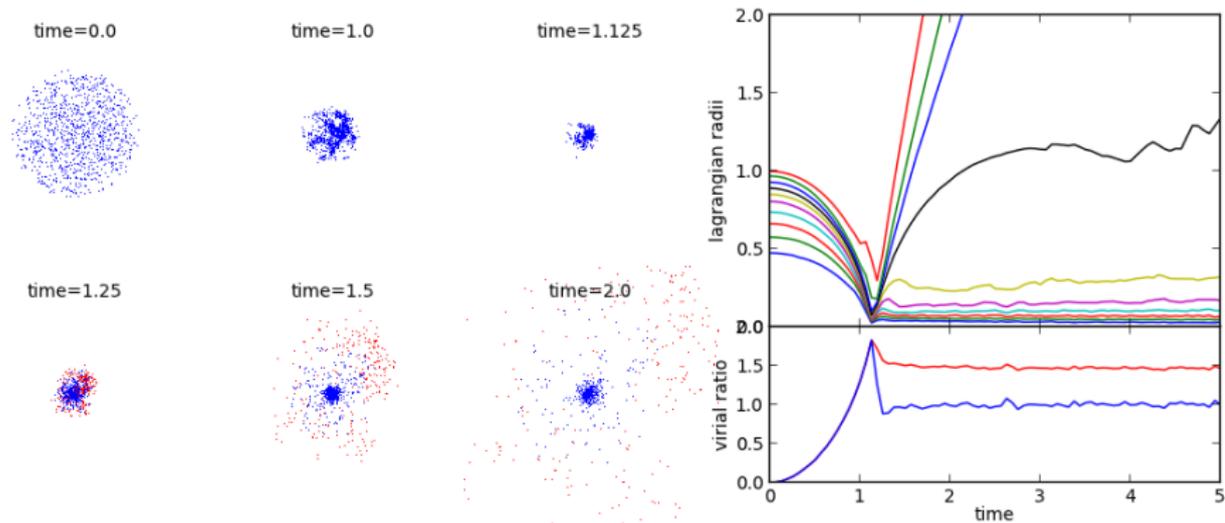
- ▶ M : total mass of the system;
- ▶ $R_{\text{vir}} \equiv -\frac{GM^2}{2W}$: virial radius;
- ▶ $v_{\text{vir}} \equiv \sqrt{\frac{2T}{M}}$: characteristic velocity $\approx \sqrt{\frac{GM}{2R_{\text{vir}}}}$ in equilibrium.
- ▶ $T_{\text{cross}} \equiv \frac{2R_{\text{vir}}}{v_{\text{vir}}}$: crossing time or dynamical time.

N -body (or Hénon) units: set $G = 1, M = 1, R = 1 \implies$
 $E = -1/4, T_{\text{cross}} = 2\sqrt{2}$.

Dynamical equilibrium

T_{cross} is the characteristic time for establishing a dynamical equilibrium, or for a perturbation to propagate through the system.

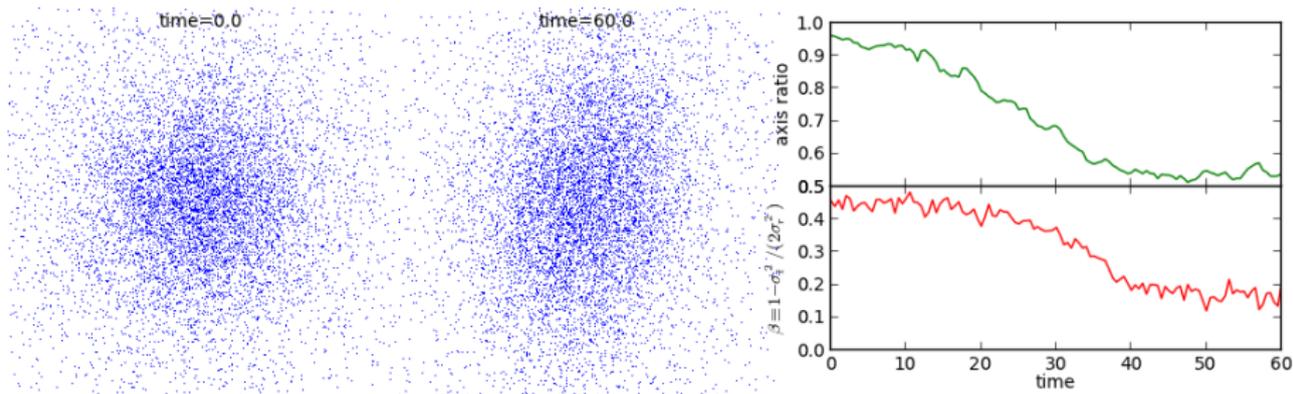
Example: cold collapse of a uniform-density sphere.



Equilibrium and stability

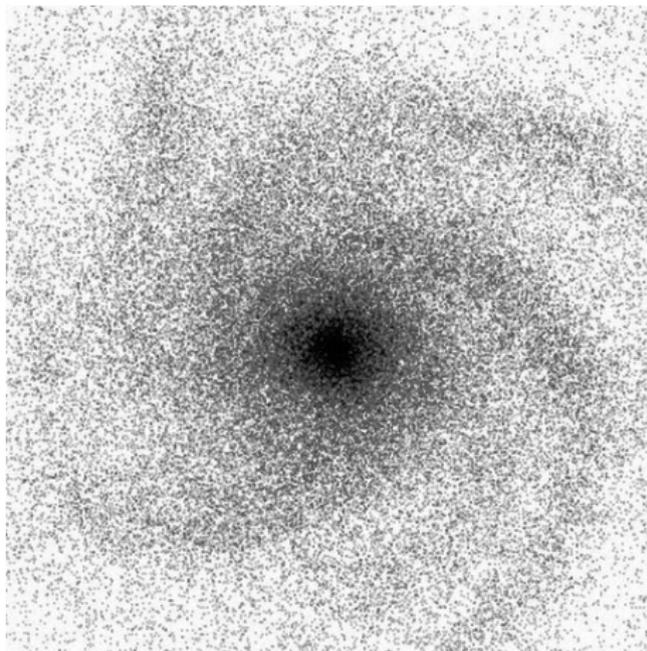
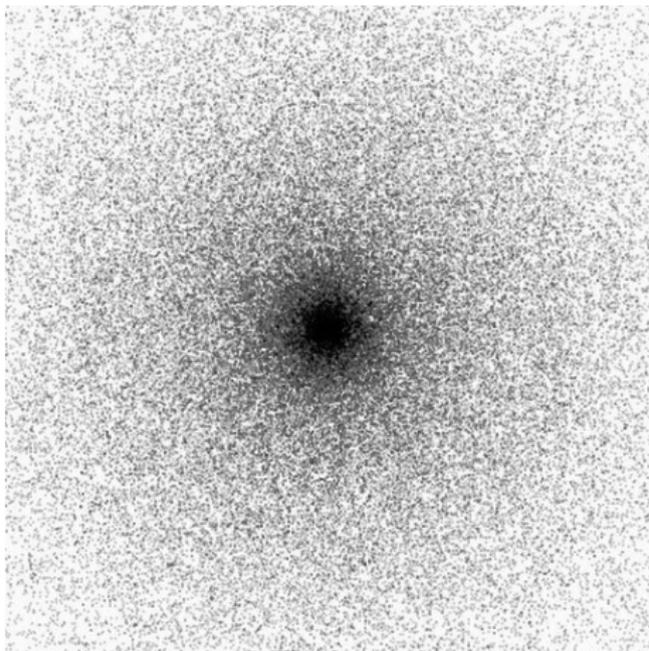
A system in a perfect dynamical equilibrium may still be unstable to collective modes.

Example: radial-orbit instability in spherical systems.



Equilibrium and stability

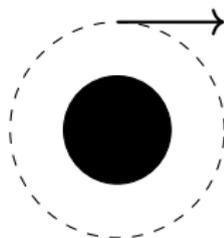
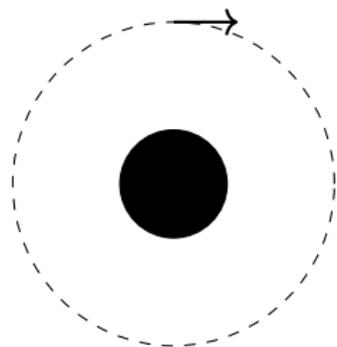
Another example: growth of spiral structure in axisymmetric disks.



Gravitational thermodynamics

Self-gravitating systems often have negative specific heat.

Example: a satellite orbiting the Earth.



$$E = -\frac{GM_{\bullet}}{r}, T = -E$$

The same phenomenon occurs in globular clusters with a core-halo structure: as the energy is transferred away from the core, it gets hotter (gravothermal instability leading to a runaway core collapse).

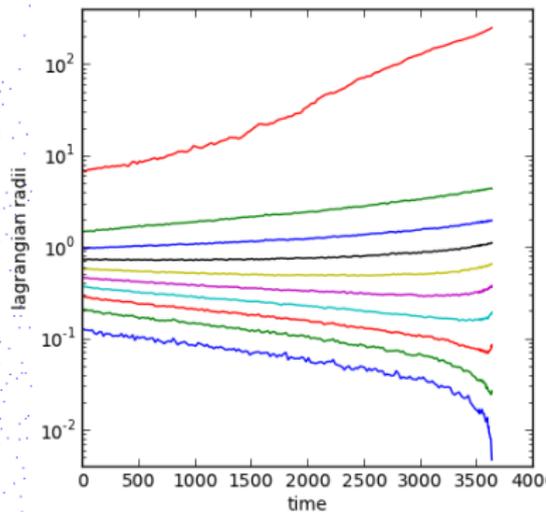
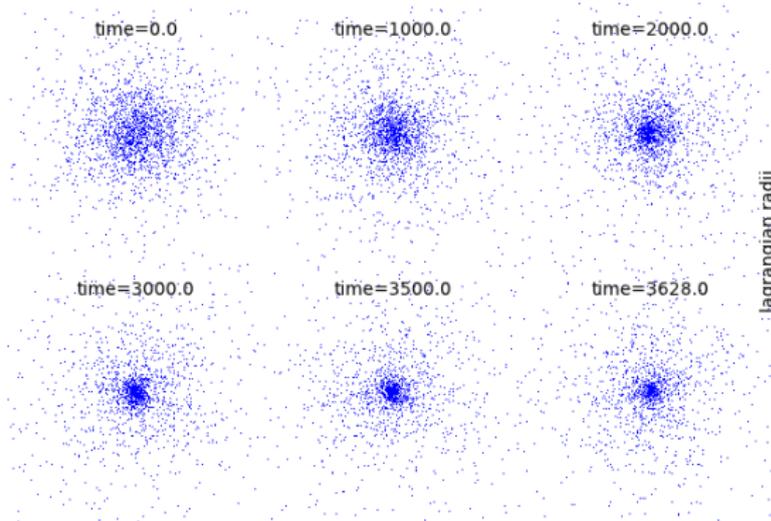
Timescale for this process is $\propto T_{\text{rel}}$.

Thermodynamical evolution

Evolution on due to two-body relaxation occurs on a timescale

$$T_{\text{rel}} \gg T_{\text{cross}}.$$

Example: core collapse of a globular cluster.



Thermodynamical equilibrium

Thermodynamical evolution is described by the *collisional* Boltzmann equation (lecture 4).

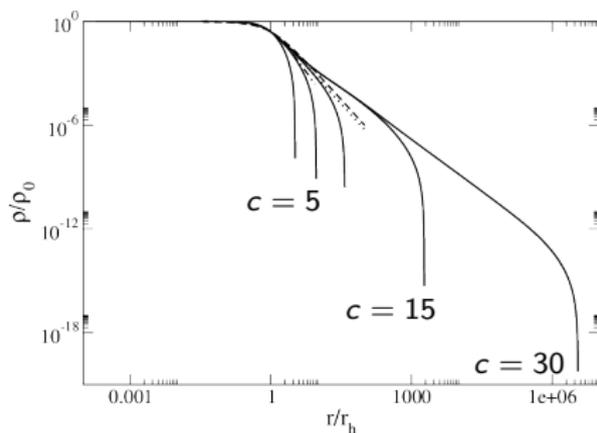
The *only* equilibrium solution is the isothermal sphere:

$$f(E) \propto \exp\left(-\frac{E}{\sigma^2}\right), \quad \rho(r) \propto r^{-2}, \quad \Phi(r) \propto \ln r.$$

Unfortunately, it has infinite mass...

A family of *lowered* isothermal models with finite mass [King 1966], defined by M , R_{core} and R_{trunc} , or concentration $c \equiv \log R_{\text{trunc}}/R_{\text{core}}$.

These models describe *some* globular clusters at early stages of evolution (before core collapse).



[from Chavanis+ 2015]

Dynamical friction

A massive body moving through a uniform-density medium creates a density wake behind itself. The gravitational pull of this

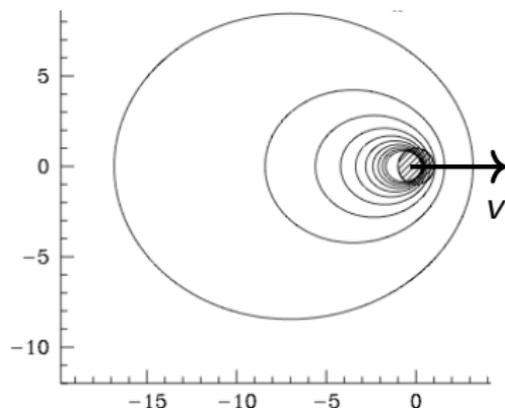
overdensity acts as a friction force: $M\dot{v} = -\frac{4\pi G^2 M^2 \rho \ln \Lambda}{v^2}$

[Chandrasekhar 1943, Mulder 1982].

In a more realistic case (motion on a quasi-periodic orbit in a finite-size system), the friction force comes from particles on orbits in resonances with the massive body [Tremaine & Weinberg 1984].

Examples:

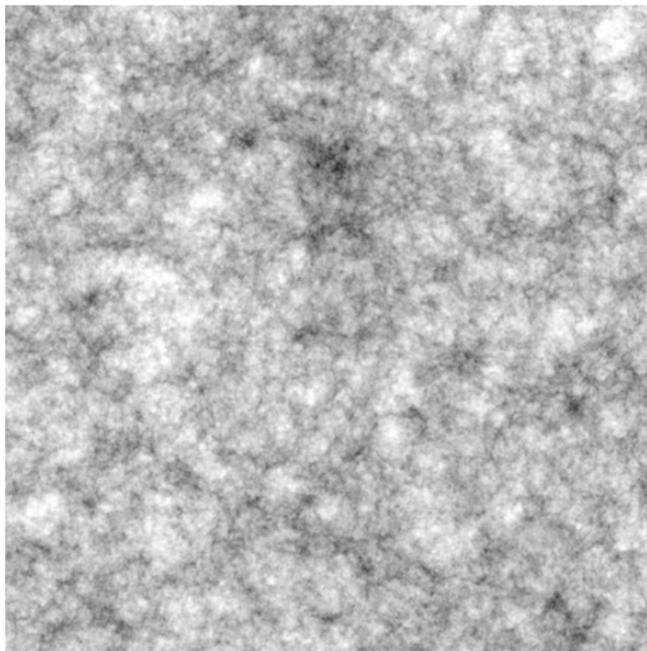
- ▶ Sinking of globular clusters and satellite galaxies.
- ▶ Slowdown of a bar in disk galaxies.
- ▶ Formation of binary supermassive black holes.
- ▶ Mass segregation in star clusters.



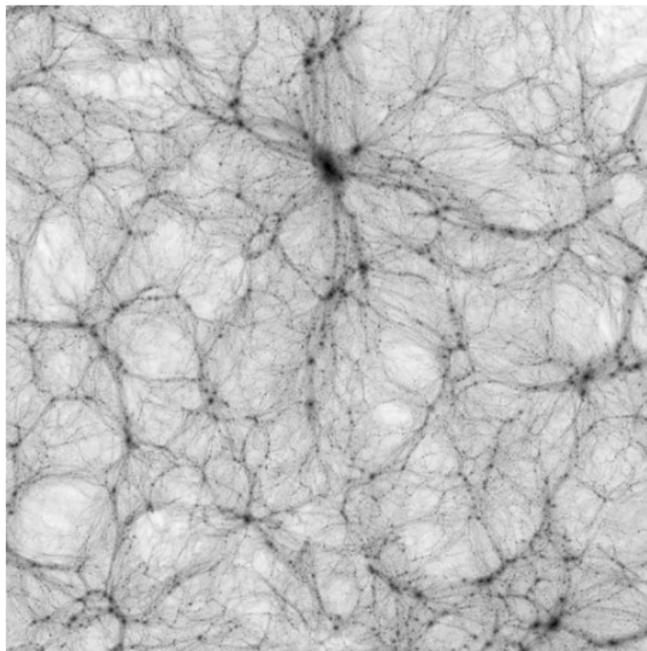
Cosmological structure formation

Hierarchical formation of large-scale structure of the Universe due to gravitational instability.

$z = 18$



$z = 0$



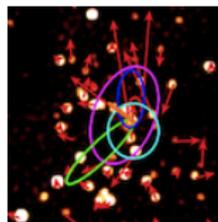
[from Springel+ 2005]

Lecture 2. Collisional systems

- ▶ Introduction
- ▶ Force computation
- ▶ Equations of motion
- ▶ Regularization
- ▶ Near-Keplerian systems
- ▶ Evolution of globular clusters

Astrophysical context

- ▶ Planetary systems: $N \sim 10$, age $\sim 10^{10} T_{\text{dyn}}$,
“collisions” do happen...
- ▶ Open clusters: $N \sim 10^3$, age $\sim 10^5$ yr;
dissolve due to dynamical effects.
- ▶ Globular clusters: $N \sim 10^6$, age $\sim 10^{10}$ yr;
the showcase of dynamical evolution.
- ▶ Dense galactic nuclei: $N \sim 10^7$,
black holes, relativistic dynamics.



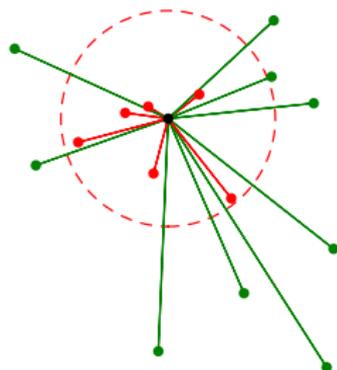
Computational context

- ▶ Force computation: direct summation, neighbor scheme, hardware acceleration, tidal field.
- ▶ Equations of motion: Hermite and symplectic integrators, error growth, choice of timestep.
- ▶ Close encounters and regularization.
- ▶ Near-Keplerian systems: higher-order and mixed-variable symplectic integrators, perturbation theory, relativistic effects in galactic nuclei.

Force computation

Brute-force direct summation:

$$\mathbf{a}_i \equiv \ddot{\mathbf{x}}_i = - \sum_{j=1, j \neq i}^N m_j \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|^3}, \quad i = 1..N,$$



The Ahmad-Cohen neighbor scheme:

- ▶ For each i -th particle, maintain the list of its neighbors (all particles within a given radius, or a given number of particles).
- ▶ Split the force into two parts: “irregular” (from neighbors, rapidly changes) and “regular” (from distant particles, changes slowly).
- ▶ Update the irregular force at each timestep;
- ▶ *Predict* the regular force also at each timestep, extrapolating the motion of distant particles using their \mathbf{x}_j , \mathbf{v}_j , \mathbf{a}_j , ...
- ▶ Fully *recompute* the regular force once in a while.

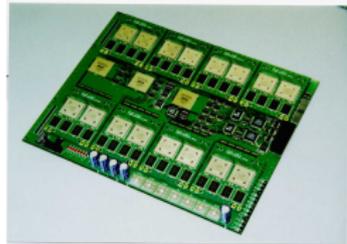
Hardware acceleration

- ▶ Special-purpose hardware (1990s–2000s) – GRAPE boards (GRAvity PipelinE):
 - ▶ Implement parallelized Newton force calculation.
 - ▶ Extension boards plugged into ordinary workstations.
 - ▶ First computer to reach 1 TFlops (1995, GRAPE-4).
 - ▶ Adapted for neighbor search and Hermite integration.
 - ▶ Originally for direct-summation codes, adapted for tree-codes, molecular dynamics...
 - ▶ Cost: $\sim 10^4$ \$/unit, highest Flops/\$ ten years ago.

- ▶ GPU boards (2000s–2010s):
 - ▶ Libraries emulating the GRAPE interface on GPU;
 - ▶ Genuinely CUDA- or OpenCL-based codes (direct-summation, tree, particle-mesh...)



Jun Makino with GRAPE-6

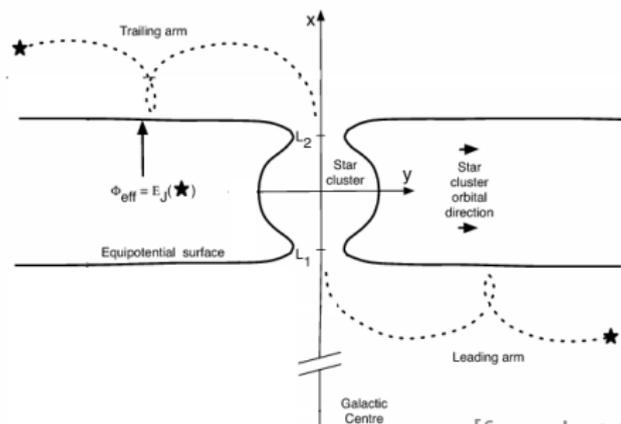


Other issues

- ▶ External tidal field:

for a globular cluster on a circular orbit in the galaxy,
equations of motion of particles in the rotating frame are

$$\ddot{\mathbf{x}} = -\nabla\Phi - 2\boldsymbol{\omega} \times \dot{\mathbf{x}} + \omega^2(3x\mathbf{e}_x - z\mathbf{e}_z).$$



- ▶ General relativity: post-Newtonian corrections.
- ▶ Fast multipole method – bring the cost of force evaluation to $\mathcal{O}(N)$ with arbitrarily high force accuracy [Dehnen 2014].

Time integration

$$\mathbf{x}_0 \equiv \mathbf{x}(t) \quad \Longrightarrow \quad \mathbf{x}_1 \equiv \mathbf{x}(t + \Delta t).$$

Simple first-order Euler scheme (no one uses it, really!):

$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{v}_0 \Delta t,$$

$$\mathbf{v}_1 = \mathbf{v}_0 + \mathbf{a}_0 \Delta t.$$

Alternatives:

- ▶ Increase the order of scheme by using several stages (e.g., Runge–Kutta);
- ▶ Increase the order of Taylor expansion by computing more derivatives (Hermite scheme);
- ▶ Integrate *exactly* an approximate Hamiltonian (symplectic methods).

Symplectic methods

The Hamiltonian of a single particle:

$$\mathcal{H}(\mathbf{x}, \mathbf{v}) = \frac{1}{2}\mathbf{v}^2 + \Phi(\mathbf{x}) = \mathcal{H}_{\text{drift}}(\mathbf{v}) + \mathcal{H}_{\text{kick}}(\mathbf{x}).$$

We know how to exactly integrate each part separately:

$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{v}_0 \Delta t \quad (\text{the drift step}),$$

$$\mathbf{v}_1 = \mathbf{v}_0 + \mathbf{a}_1 \Delta t \quad (\text{the kick step), where } \mathbf{a}_1 \equiv -\nabla\Phi(\mathbf{x}_1).$$

Because $\mathcal{H}_{\text{drift}}$ and $\mathcal{H}_{\text{kick}}$ do not commute, we have integrated *exactly* an *approximate* Hamiltonian $\mathcal{H}_{\text{approx}} = \mathcal{H} + \mathcal{H}_{\text{err}}$, where $\mathcal{H}_{\text{err}} = [\mathcal{H}_{\text{drift}}, \mathcal{H}_{\text{kick}}] \Delta t + \mathcal{O}(\Delta t^2) + \dots$

Due to antisymmetry of the Poisson bracket, if we further apply the kick and drift steps in reverse order, the first-order error term cancels out.

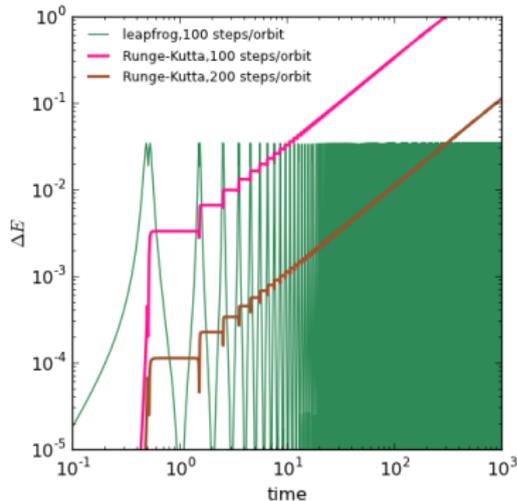
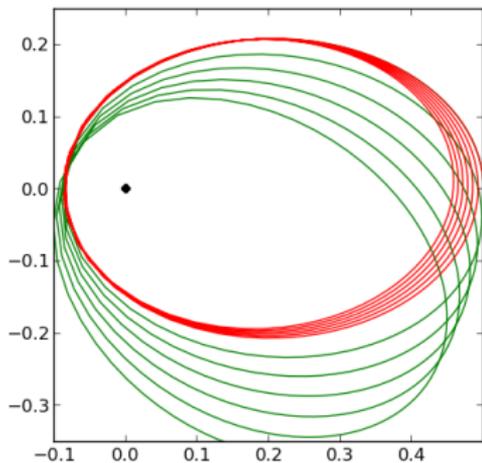
$$\left. \begin{array}{l} \mathbf{x}_{1/2} = \mathbf{x}_0 + \mathbf{v}_0 \Delta t/2 \\ \mathbf{v}_{1/2} = \mathbf{v}_0 + \mathbf{a}_{1/2} \Delta t/2 \\ \mathbf{v}_1 = \mathbf{v}_{1/2} + \mathbf{a}_{1/2} \Delta t/2 \\ \mathbf{x}_1 = \mathbf{x}_{1/2} + \mathbf{v}_1 \Delta t/2 \end{array} \right\} \implies \begin{array}{l} \mathbf{x}_{1/2} = \mathbf{x}_{-1/2} + \mathbf{v}_0 \Delta t \\ \mathbf{v}_1 = \mathbf{v}_0 + \mathbf{a}_{1/2} \Delta t \end{array}$$

Leapfrog integrator

The leapfrog scheme is the simplest 2nd-order symplectic integrator for Hamiltonian systems, i.e., it integrates exactly the motion in an approximate (surrogate) Hamiltonian $\mathcal{H} + \mathcal{H}_{\text{err}}$, with $\mathcal{H}_{\text{err}} \propto \Delta t^2$.

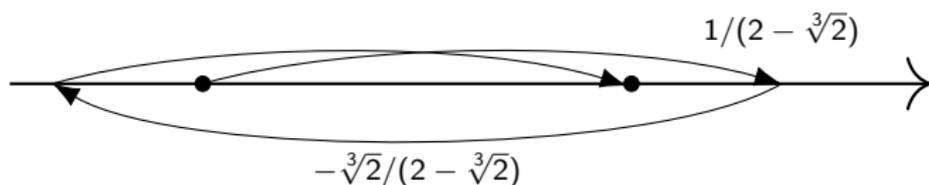
The advantage is that it preserves all integrals of motion of the surrogate system, which are $\mathcal{O}(\Delta t^2)$ away from the real integrals: the latter only have periodic oscillations but no secular drift.

symplectic (leapfrog: green) vs. non-symplectic (4th order Runge–Kutta: red)



Other symplectic integrators

One may construct higher-order symplectic schemes by concatenating several leapfrog steps in a carefully chosen proportion [Yoshida 1990]:



Another application of symplectic schemes is for near-Keplerian systems, where the Hamiltonian is split into the Kepler (drift) part and the interaction (kick) part, each one integrated exactly using its own set of coordinates (mixed-variable symplectic)

[Wisdom & Holman 1991, Saha & Tremaine 1992].

Higher-order integration schemes

$$\mathbf{x}_0 \equiv \mathbf{x}(t) \quad \Longrightarrow \quad \mathbf{x}_1 \equiv \mathbf{x}(t + \Delta t).$$

Taylor expansion:

$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{v} \Delta t + \frac{1}{2} \mathbf{a} \Delta t^2 + \frac{1}{6} \mathbf{j} \Delta t^3 + \frac{1}{24} \mathbf{s} \Delta t^4 + \frac{1}{120} \mathbf{c} \Delta t^5 + \dots$$

acceleration jerk snap crackle

How to compute higher derivatives?

- ▶ Polynomial fit to n previous timesteps (multistep methods);
- ▶ Compute directly from particle positions:

$$\mathbf{j}_i \equiv \ddot{\mathbf{x}}_i = - \sum_{j=1, j \neq i}^N m_j \left[\frac{\mathbf{v}_i - \mathbf{v}_j}{|\mathbf{x}_i - \mathbf{x}_j|^3} - 3([\mathbf{x}_i - \mathbf{x}_j] \cdot [\mathbf{v}_i - \mathbf{v}_j]) \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|^5} \right]$$

Hermite method

Taylor-expanding \mathbf{x} , \mathbf{v} , \mathbf{a} , \mathbf{j} and eliminating \mathbf{s} and \mathbf{c} , we get

$$\mathbf{x}_1 = \mathbf{x}_0 + \frac{1}{2}(\mathbf{v}_0 + \mathbf{v}_1)\Delta t + \frac{1}{12}(\mathbf{a}_0 - \mathbf{a}_1)\Delta t^2 + \mathcal{O}(\Delta t^5),$$

$$\mathbf{v}_1 = \mathbf{v}_0 + \frac{1}{2}(\mathbf{a}_0 + \mathbf{a}_1)\Delta t + \frac{1}{12}(\mathbf{j}_0 - \mathbf{j}_1)\Delta t^2 + \mathcal{O}(\Delta t^5).$$

However, these expressions are implicit \implies iterative solution.

- ▶ Compute $\mathbf{a}_0, \mathbf{j}_0$ at the beginning of timestep.
- ▶ *Predict* the values of \mathbf{x}, \mathbf{v} at the end of timestep:

$$\mathbf{x}_p = \mathbf{x}_0 + \mathbf{v}_0 \Delta t + \frac{1}{2}\mathbf{a}_0 \Delta t^2 + \frac{1}{6}\mathbf{j} \Delta t^3,$$

$$\mathbf{v}_p = \mathbf{v}_0 + \mathbf{a}_0 \Delta t + \frac{1}{2}\mathbf{j}_0 \Delta t^2.$$

- ▶ Compute predicted $\mathbf{a}_p, \mathbf{j}_p$ at the end of timestep.
- ▶ *Correct* the values of \mathbf{x}, \mathbf{v} at the end of timestep:

$$\mathbf{v}_1 = \mathbf{v}_0 + \frac{1}{2}(\mathbf{a}_0 + \mathbf{a}_p)\Delta t + \frac{1}{12}(\mathbf{j}_0 - \mathbf{j}_p)\Delta t^2,$$

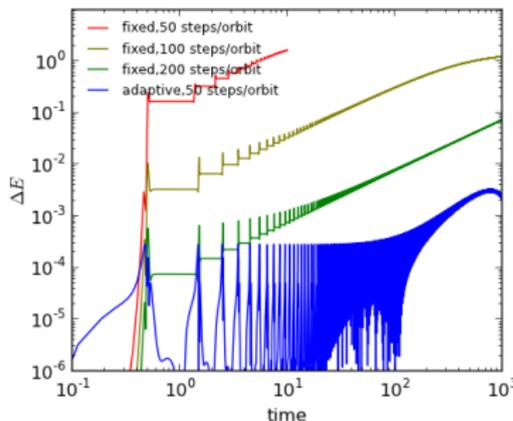
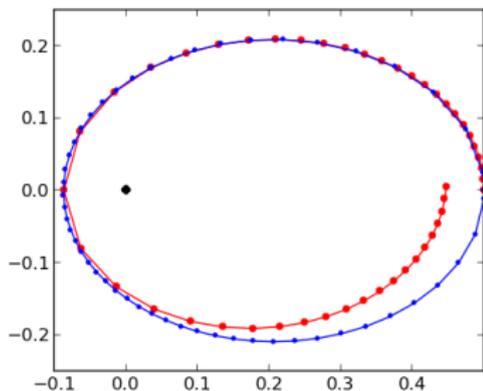
$$\mathbf{x}_1 = \mathbf{x}_0 + \frac{1}{2}(\mathbf{v}_0 + \mathbf{v}_1)\Delta t + \frac{1}{12}(\mathbf{a}_0 - \mathbf{a}_p)\Delta t^2.$$

(note the reversed order and the use of corrected \mathbf{v} in the last line).

Choice of timestep

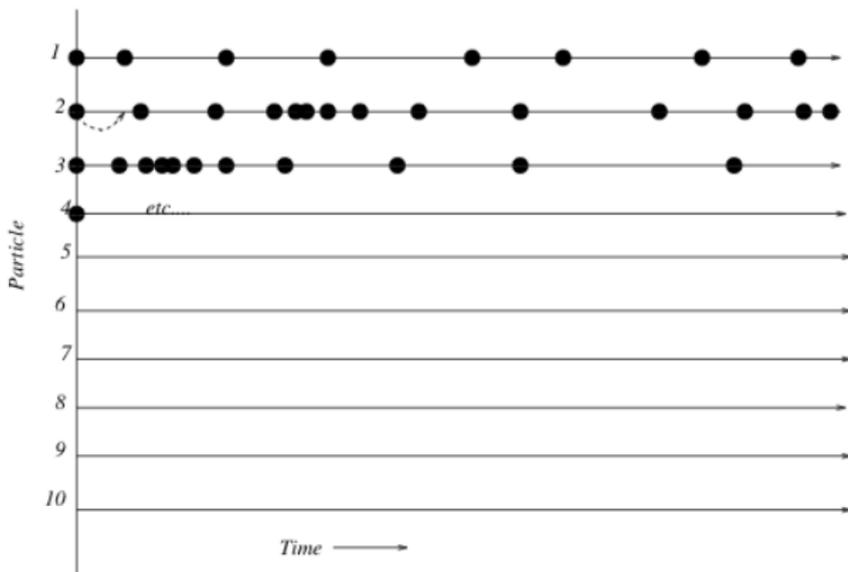
- ▶ Δt should be smaller than the “time-to-collision”:
$$\Delta t \leq \eta \min_{i \neq j} \frac{|\mathbf{x}_i - \mathbf{x}_j|}{|\mathbf{v}_i - \mathbf{v}_j|},$$
 where $\eta < 1$ is the accuracy parameter.
- ▶ Convergence of Taylor expansion requires that successive term should be progressively smaller: $\mathbf{a} \Delta t \ll \mathbf{v}, \mathbf{j} \Delta t \ll \mathbf{a}, \dots$
- ▶ A combination of higher derivatives works best:

$$\Delta t \leq \eta \left(\frac{|\mathbf{a}| |\ddot{\mathbf{a}}| + |\dot{\mathbf{a}}|^2}{|\dot{\mathbf{a}}| |\ddot{\mathbf{a}}| + |\ddot{\mathbf{a}}|^2} \right)^{1/2} \quad [\text{the Aarseth criterion}]$$



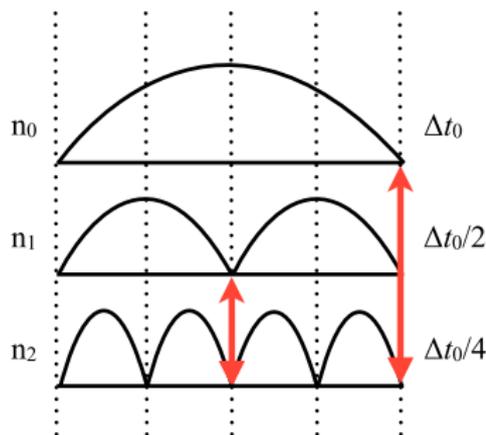
Individual timesteps

- ▶ Each particle follows its own optimal timestep.
- ▶ Positions of other particles at the given time are extrapolated.
- ▶ Minimized the number of timesteps taken, but extrapolation incurs additional computational overhead.
- ▶ Used in early schemes based on polynomial extrapolation.



Block timesteps

- ▶ Hierarchy of levels: $\Delta t_k = 2^{-k} \Delta t_0$.
- ▶ Timestep of each particle is rounded down from its optimal value to the nearest power of two.
- ▶ Particles at the same level of hierarchy are advanced simultaneously, reducing the cost of extrapolation.
- ▶ A particle can reduce its timestep at any time, but increase only at the synchronization point of the next level.
- ▶ Used in most present-day schemes.
- ▶ Disadvantage of any variable-timestep scheme is that the choice of timestep is not time-symmetric (deteriorates energy conservation); restoring [approximate] time-symmetry requires extra effort.



Close encounters and regularization

Close encounters are those with deflection angle $\gtrsim 90^\circ$,
i.e., with impact parameter $b \lesssim b_{90} \equiv m/v^2$.

They cause troubles because...

1. $|\mathbf{x}_1 - \mathbf{x}_2|$ suffers from roundoff errors;
2. Δt required to resolve the encounter is very small.

The solutions are:

1. Switch to Jacobi coordinates (center-of-mass and relative offset):

$$\mathbf{X} \equiv \frac{m_1 \mathbf{x}_2 + m_2 \mathbf{x}_1}{m_1 + m_2}, \quad \mathbf{x} \equiv \mathbf{x}_1 - \mathbf{x}_2,$$

and write the equations of motion as

$$\ddot{\mathbf{x}} = -(m_1 + m_2) \frac{\mathbf{x}}{|\mathbf{x}|^3} + \mathbf{a}'_1 - \mathbf{a}'_2,$$

where $'$ stands for a sum over all other particles except 1 and 2.

2. Transform coordinates and time to remove the $1/r$ singularity.

Two-body regularization

1d example:

Motion in dimensionless variables is described by $\ddot{x} = -1/x^2$.

Introduce fictitious time τ so that $dt = x d\tau$, and denote $' \equiv d/d\tau$.

The dominant Keplerian motion then becomes

$$x'' = x'^2/x - 1 = 2hx + 1,$$

where $h \equiv \dot{x}^2/2 - 1/x$ is the binding energy of the binary.

Replacing x by $u \equiv \sqrt{x}$, this simplifies even further: $u'' = \frac{1}{2}hu$.

Thus the singular Keplerian motion is transformed to a regular harmonic oscillator; perturbations from other particles are taken into account (the choice of timestep depends on the strength of perturbation).

2d: Levi-Civita [1920] transformation (seldom used in practice).

3d: Kustaanheimo–Stiefel [1965] regularization (4d auxiliary coords).

Multiple regularization

- ▶ Multiple coupled KS transformations [Heggie 1974, Mikkola 1985] – inefficient for large number of interacting particles ($\#$ of equations grows as $\sim 4N^2$).
- ▶ Wheel-spoke [Zare 1974] and chain regularization [Mikkola & Aarseth 1993] – implement KS between $N - 1$ pairs of adjacent particles, not between all $N(N - 1)$ pairs.



- ▶ Algorithmic regularization [Mikkola & Tanikawa 1999, Preto & Tremaine 1999, Mikkola & Aarseth 2002]: instead of $H \equiv T + U$, use a modified Hamiltonian $\tilde{H} = \ln T - \ln U$, also used with chain to improve accuracy.

Near-Keplerian systems

Mixed-variable symplectic (MVS) integrators

[Wisdom & Holman 1991, Saha & Tremaine 1992]:

$$H = H_{\text{Kep}} + \epsilon H_{\text{perturb}} = \left[\sum_{i=1}^N \left(\frac{\mathbf{p}'_i{}^2}{2m'_i} - \frac{m_i m_0}{|\mathbf{x}'_i|} \right) \right] - \left[\sum_{0 < i < j} \frac{m_i m_j}{|\mathbf{x}'_{ij}|} \left(1 - \frac{\mathbf{x}'_i \mathbf{x}'_j}{|\mathbf{x}'_{ij}|^2} \right) \right].$$

The dominant motion of each particle around the central body is solved exactly using Kepler orbital elements and an appropriate Kepler solver (the *drift* step), and the perturbations from other particles are taken into account in the *kick* step, using Cartesian coordinates.

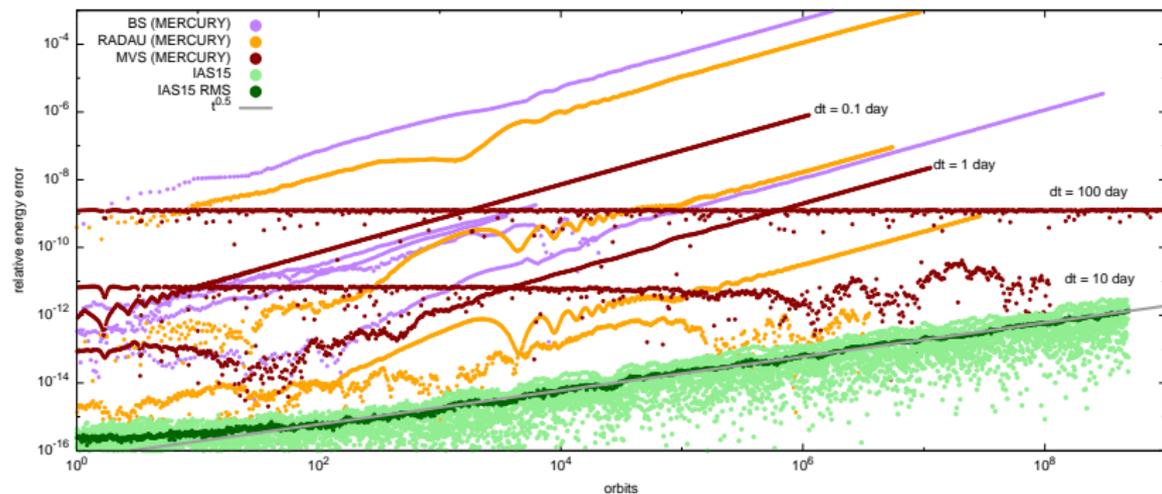
The limitation of MVS is that the perturbations must remain small, i.e., close encounters are not tracked accurately.

To deal with the latter, hybrid symplectic integrators switch to another integration method (typically Bulirsch–Stoer) during a close encounter [e.g. Chambers 1999].

Cautionary note about symplectic integrators

Symplectic integrators conserve energy *apart from roundoff errors*, which is the dominant error in long-term integration.

A high-order *non-symplectic* integrator [Rein & Spiegel 2015] with a carefully implemented floating-point scheme and unbiased rounding may have a better long-term energy conservation ($\Delta E \propto \sqrt{t}$):



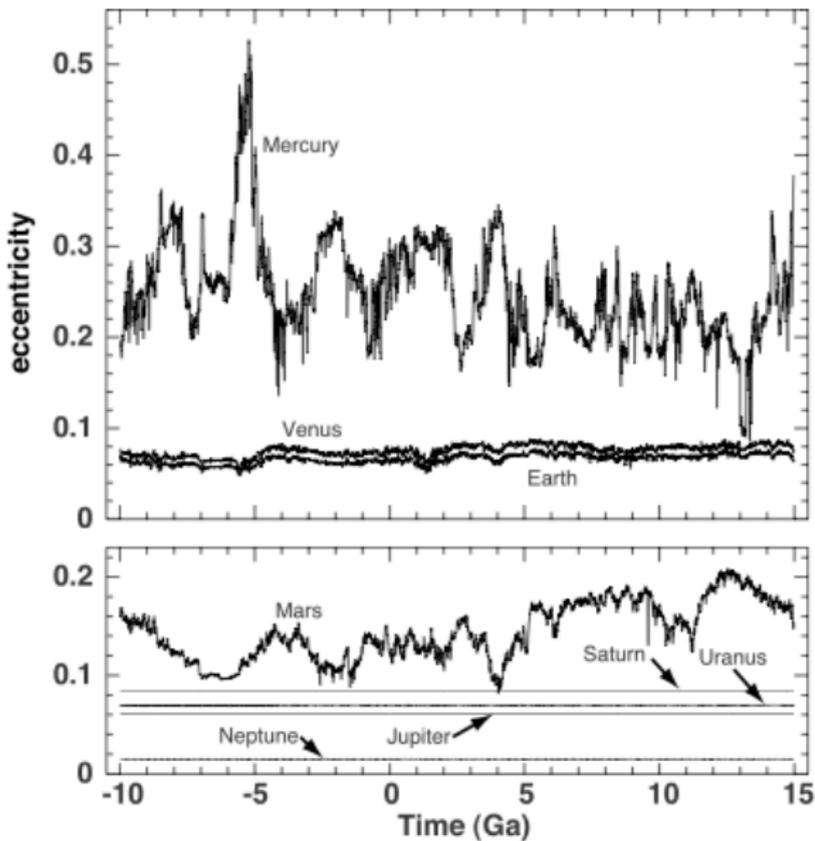
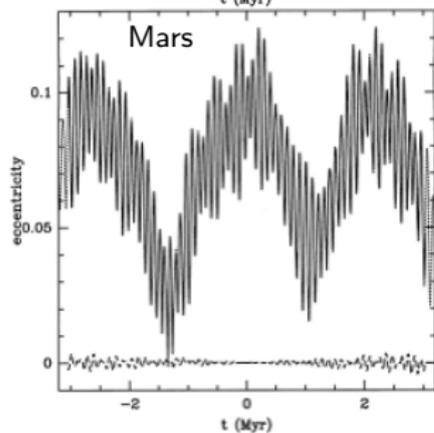
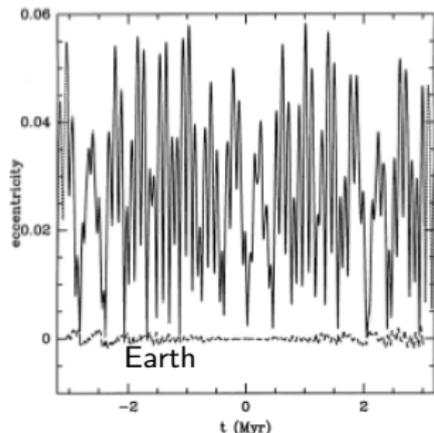
Orbit-averaged dynamics

Near-Keplerian motion around a central mass can be described in terms of orbital elements: semimajor axis a , eccentricity e , inclination i , argument of periapsis ω , longitude of ascending node Ω , and true anomaly ν . Of these, only ν changes with time in a purely Keplerian orbit, and one may write down equations for the slow variation of the 5 other variables by averaging the perturbations over the fast angle ν .

These equations of perturbation theory can then be integrated with a much longer timestep ($\gg T_{\text{orbit}}$); e.g. Laskar [1986, 1989] used equations with $\sim 150\,000$ terms and a timestep $\Delta t = 500$ yr (compared to $\Delta t \sim$ a few days for a direct integration).

A similar approach is used for stars orbiting a supermassive black hole (N -ring systems, Touma et al. 2009, Kocsis & Tremaine 2015).

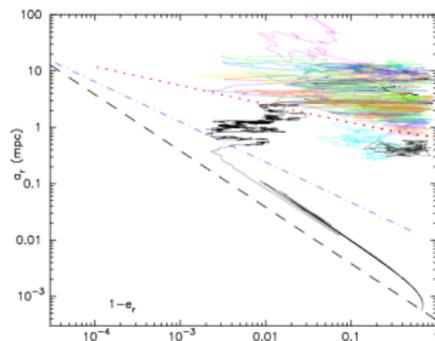
Long-term stability of Solar System



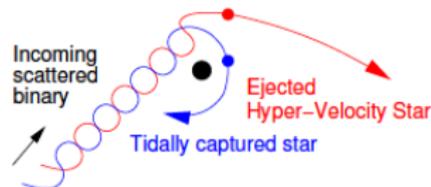
[from Laskar 1994]

Collisional evolution of galactic nuclei

- ▶ Loss-cone phenomena:
capture of stars by the central black hole.
- ▶ Resonant relaxation:
near-Keplerian stellar orbits remain aligned for $T \gg T_{\text{dyn}} \implies$ mutual torques lead to a faster exchange of angular momentum (but not energy).
- ▶ Relativistic effects:
precession of orbit \implies destroys coherence (more important for eccentric orbits);
gravitational-wave emission \implies
inspiral or plunge into the central black hole.
- ▶ Binary stars:
tidal disruption of a binary [Hills 1988] \implies
deposit stars on tightly-bound orbits,
ejects hypervelocity stars.



[from Merritt+ 2011]

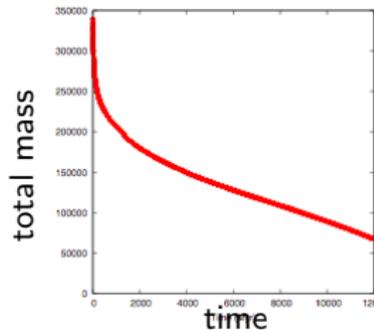
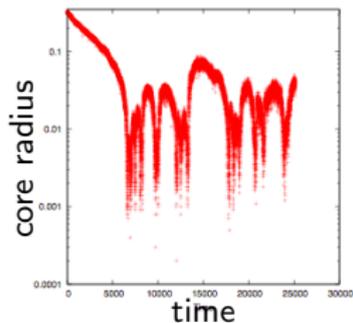
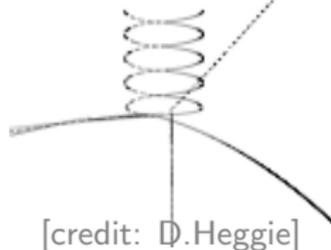


Long-term evolution of globular clusters

Most important physical ingredients:

- ▶ Two-body relaxation \implies energy flow, core collapse.
- ▶ Stellar evolution (mass loss due to stellar winds, formation of compact objects, mass transfer in tight binary systems).
- ▶ Three- and four-body interactions (binary+single star, binary+binary) \implies energy source (hard binaries), arrest the core collapse.
- ▶ Mass segregation (heavy objects sink to the center) \implies formation of supermassive stars or sub-systems of stellar black holes?
- ▶ Tidal stripping in the external Galactic field (evaporation and ejection, and eventually total dissolution).

three-body binary formation



Lecture 3. Collisionless systems

- ▶ Force softening
- ▶ Gravity solvers
- ▶ Cosmological simulations
- ▶ Galaxy formation and evolution

Mean-field dynamics

Solve the system of equations

$$\frac{\partial f}{\partial t} + \mathbf{v} \frac{\partial f}{\partial \mathbf{x}} - \frac{\partial \Phi}{\partial \mathbf{x}} \frac{\partial f}{\partial \mathbf{v}} = 0 \quad (\text{Collisionless Boltzmann eqn})$$

$$\nabla^2 \Phi(\mathbf{x}) = 4\pi G \rho(\mathbf{x}) \quad (\text{Poisson eqn})$$

$$\rho(\mathbf{x}) = \iiint d^3v f(\mathbf{x}, \mathbf{v}).$$

N -body approach: sample $f(\mathbf{x}, \mathbf{v})$ by N particles and let them move

in the potential
$$\Phi(\mathbf{x}) = -G \iiint d^3x' \frac{\rho(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|} = -G \sum_{k=1}^N \frac{m_k}{|\mathbf{x} - \mathbf{x}_k|}.$$

Problem 1 – discreteness noise in estimating $\hat{\rho}$ from particles.

Problem 2 – large-angle scattering in close encounters.

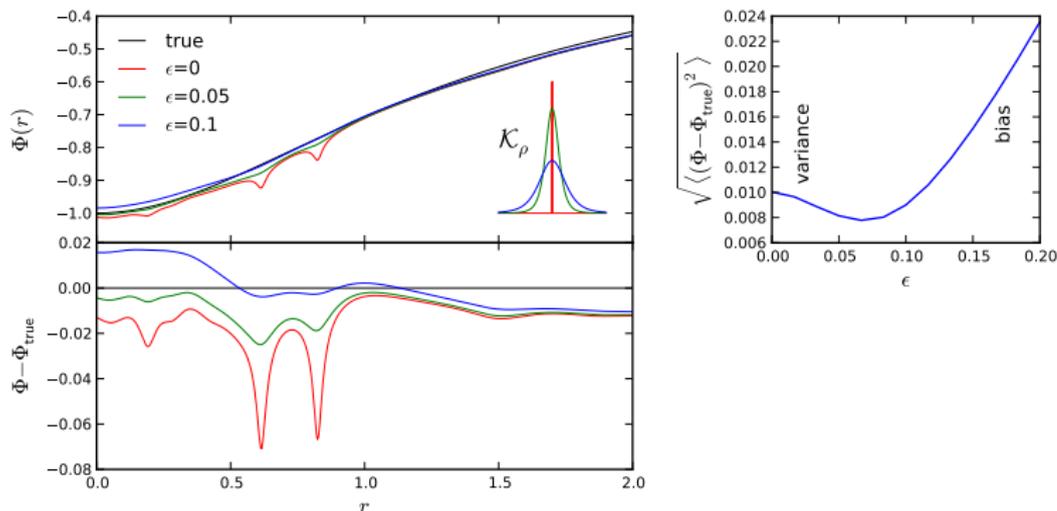
Force softening: motivation

Potential estimation: $\hat{\Phi}(\mathbf{x}) = \sum_{k=0}^N m_k \mathcal{K}_{\Phi}(|\mathbf{x} - \mathbf{x}_k|)$,

equivalent to the density estimation with kernel $\mathcal{K}_{\rho} \equiv \frac{1}{4\pi} \nabla^2 \mathcal{K}_{\Phi}$.

Unsoftened gravity: $\mathcal{K}_{\Phi}(r) = -1/r$, $\mathcal{K}_{\rho}(r) = \delta_{3d}(r)$;

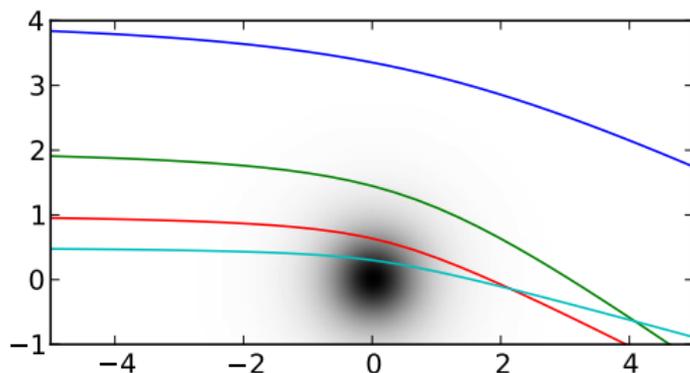
typically $\mathcal{K}_{\Phi}(r; \epsilon) = \mathcal{K}(r/\epsilon)$, and the choice of softening length ϵ determines the balance between bias and variance:



Force softening: suppression of close encounters

- ▶ Large-angle deflections should not occur in a collisionless system:

$$\epsilon \gtrsim b_{90} \equiv Gm/v^2$$



- ▶ Softening simplifies the time integration:
no singularity in force, minimum timestep $\Delta t \lesssim \epsilon/v$.
- ▶ However, two-body relaxation is not substantially suppressed:
Coulomb logarithm $\ln \Lambda = \ln(r_{\max}/\epsilon) \lesssim \ln(r_{\max}/b_{90})$.

Force softening: choice of kernel

Softening kernels for density:

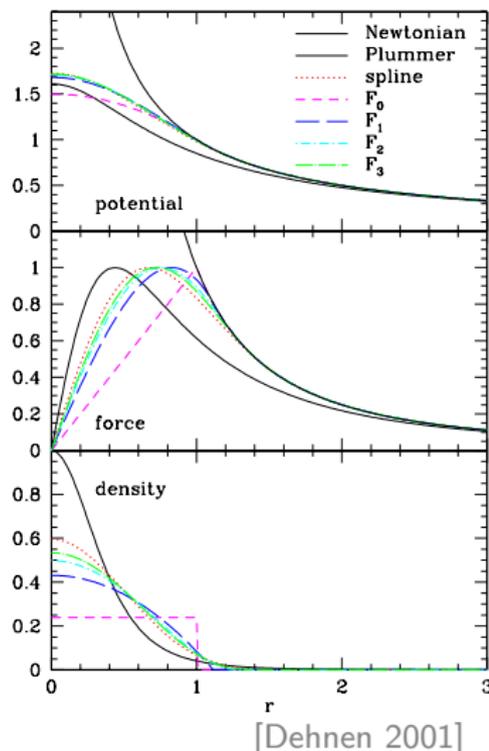
$$\mathcal{K}_\rho(r) = \epsilon^{-3} \mathcal{K}(x), \quad x \equiv r/\epsilon:$$

- ▶ Plummer: $\mathcal{K} = \frac{3}{4\pi}(1+x^2)^{-5/2}$
most common, but not the best choice.

- ▶ Spline:
$$\begin{cases} \frac{1}{4\pi}(4 - 6x^2 + 3x^3) & x \leq 1 \\ \frac{1}{4\pi}(2 - x)^3 & 1 \leq x \leq 2 \\ 0 & x > 2 \end{cases}$$

- ▶ Uniform-density sphere:
$$\begin{cases} \frac{3}{4\pi} & x \leq 1 \\ 0 & x > 1 \end{cases}$$

- ▶ Epanechnikov:
$$\begin{cases} \frac{15}{8\pi}(1-x^2) & x \leq 1 \\ 0 & x > 1 \end{cases}$$



Force softening: choice of softening length

- ▶ Suppression of strong deflections: $\epsilon \gtrsim b_{90} \equiv Gm/v^2$.
- ▶ Tradeoff between bias $B \equiv \langle \hat{\Phi}(\mathbf{x}) \rangle - \Phi_{\text{true}}(\mathbf{x})$ and variance $V \equiv \langle \hat{\Phi}^2(\mathbf{x}) \rangle - \langle \hat{\Phi}(\mathbf{x}) \rangle^2$:
 $B(\mathbf{x}) \propto \epsilon^2, \quad V(\mathbf{x}) \propto [V_{\epsilon=0} - \kappa\epsilon] N^{-1}$.

For equal-mass particles and fixed ϵ , optimal softening for force estimation $\epsilon \propto N^{-\alpha}$, $\alpha \sim 0.1 \div 0.3$

[e.g., Merritt 1996, Athanassoula+ 2000, Dehnen 2001].

For density estimation (e.g., in SPH), typically $\epsilon \propto n^{-1/3} \equiv (\rho/m)^{-1/3}$ (mean interparticle distance).

- ▶ Adaptive ϵ requires additional effort to maintain energy conservation [e.g., Price & Monaghan 2007].
- ▶ In grid-based gravity solvers, $\epsilon_{\text{eff}} \sim$ cell size.
- ▶ Force resolution (ϵ) and mass resolution ($m \propto N^{-1}$) should be compatible!

Force calculation

We do not know the true force field $\mathbf{F}(\mathbf{x})$, but only its estimate $\hat{\mathbf{F}} \equiv \nabla \hat{\Phi}(\mathbf{x}) \implies$ unavoidable error in force estimation.

May use approximate methods for force computation with complexity better than N^2 , if the approximation error is smaller than the estimation error:

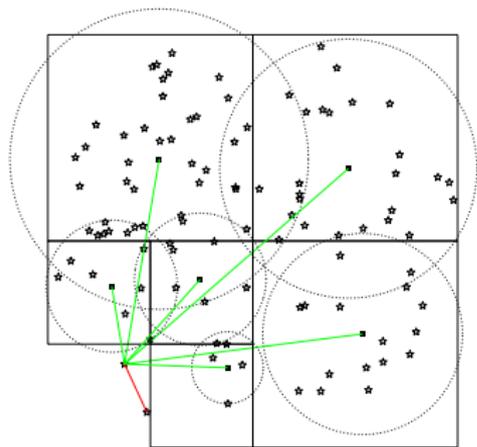
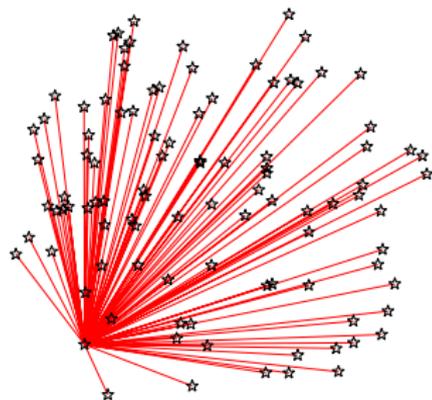
- ▶ Tree code and fast multipole method (FMM);
- ▶ Grid-based potential solvers (particle–mesh, PM);
- ▶ Combination of tree and PM;
- ▶ Basis-set expansion.

Tree code

Idea: approximate the force from a group of distant particles with a single effective interaction.

Method: group particles into a hierarchical tree-like structure

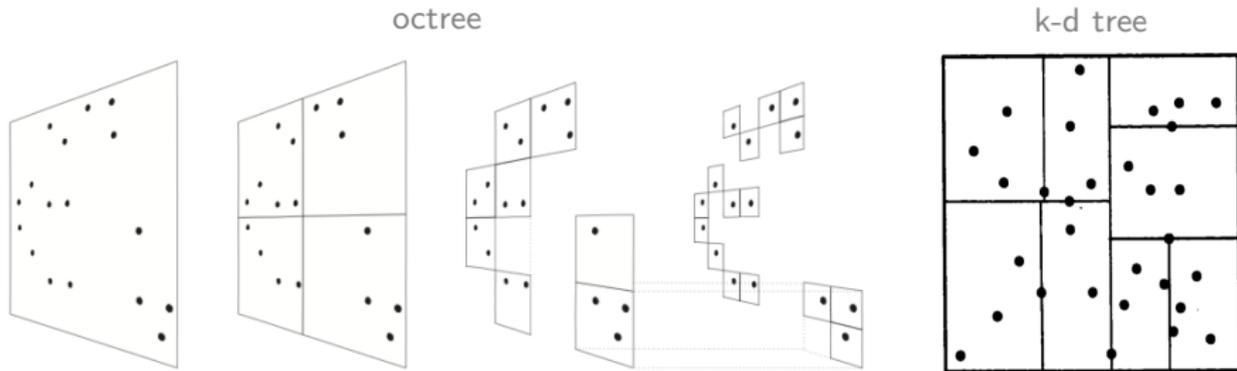
[Barnes & Hut 1986].



[from Dehnen & Read 2011]

Tree construction

1. Hierarchical top-down division of the entire volume into cells:
 - ▶ octree – each cell is split into 8 equal-size subcells.
 - ▶ binary k-d tree – each cell is split into 2 along alternating coordinates at the median.



2. Bottom-up tree scan to determine the properties of cells (center-of-mass, multipole moments, etc).

Tree traversal

For each *sink* particle, we scan the entire tree top-down, starting from the root cell:

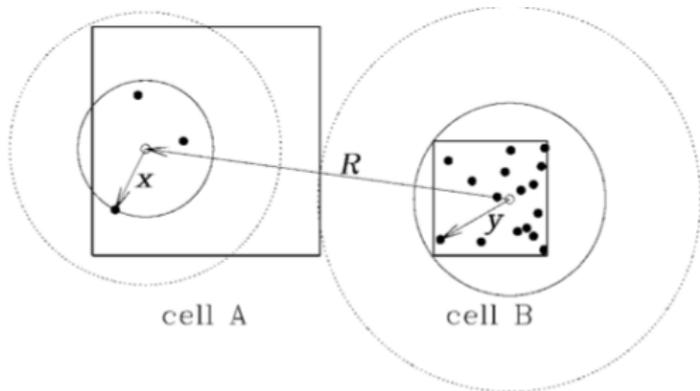
- ▶ If the current cell is *sufficiently far*, the force from the entire cell is computed (using a point-mass approximation or several multipole terms).
- ▶ Otherwise, the cell is *opened* and the force from its daughter cells (or individual *source* particles) are computed recursively.

The criterion for opening a cell is typically based on geometry: $d < \theta L$, where $d \equiv |\mathbf{x}_{\text{point}} - \mathbf{x}_{\text{cell}}|$ is the distance to the center of mass of the cell, its geometric center, or the center of multipole expansion, L is the cell size, and $\theta \leq 1$ is the *opening angle*.

The total number of force evaluations per cell is $\propto \log N$; computational cost and accuracy depend on θ and the order of multipole expansion.

Fast multipole method [Greengard & Rokhlin 1987]

Idea: use cell-cell interactions: if two cells are *well separated*, employ multipole expansion for both *source* and *sink* cells, and evaluate the mutual force between all particles at once.



[Dehnen 2000]

Advantages:

- ▶ Symmetry in force evaluation (Newton's 3rd law);
- ▶ Computational cost scales as $\mathcal{O}(N)$ or even better;
- ▶ By adjusting the order of expansion, may achieve very high accuracy ($\sim 10^{-6}$, comparable to hardware-accelerated direct-summation [Dehnen 2014]).

Drawback: high algorithmic complexity, more difficult to parallelize.

Grid methods

Idea: solve the Poisson equation $\nabla\Phi(\mathbf{x}) = 4\pi G \rho(\mathbf{x})$
by discretizing Φ and ρ on a regular [multi-level] 3d grid
(CBE is still solved by the method of characteristics, using particles).

Particle-mesh (PM) – particles move under forces represented on a mesh.

Two most common approaches:

1. [Fast] Fourier transform (FFT):

the Poisson equation in the Fourier domain is simply

$$\mathbf{k}^2\Phi(\mathbf{k}) = 4\pi G \rho(\mathbf{k}).$$

Most suitable for periodic boundary conditions, but may be used with vacuum boundary conditions by doubling the size of the k grid in each dimension.

2. Multigrid relaxation:

solve the linear matrix equation for the finite-difference Laplacian.

Poisson equation on a grid

Finite-difference Poisson equation in 1d:

$$\frac{\Phi_{i-1} - 2\Phi_i + \Phi_{i+1}}{h^2} = 4\pi\rho_i, \quad i = 1..N_{\text{grid}} - 1,$$

i.e., a system of linear equations with a tridiagonal matrix – solved in $\mathcal{O}(N_{\text{grid}})$ by forward- and back-substitution.

In more than one dimension this is not so trivial:

e.g., in 2d:
$$\frac{\Phi_{i-1,j} + \Phi_{i+1,j} + \Phi_{i,j-1} + \Phi_{i,j+1} - 4\Phi_{i,j}}{h^2} = 4\pi\rho_{i,j}$$

a sparse matrix of size $N \equiv N_{\text{grid}}^2$ and bandwidth N_{grid} – could be solved in $\mathcal{O}(N N_{\text{grid}})$ operations with a direct method (e.g., sparse LU decomposition), but this is impractical (too large memory consumption).

An alternative to direct solution of a system of linear equations is the iterative approach.

Relaxation approach for solving the Poisson equation

Introduce fictitious time and evolve $\frac{\partial \Phi}{\partial t} = \nabla^2 \Phi - 4\pi \rho$.

Jacobi's method: iterate $\Phi^{(n)} \rightarrow \Phi^{(n+1)}$ as

$$\Phi_{i,j}^{(n+1)} = \frac{1}{4} [\Phi_{i-1,j}^{(n)} + \Phi_{i+1,j}^{(n)} + \Phi_{i,j-1}^{(n)} + \Phi_{i,j+1}^{(n)} - 4\pi h^2 \rho_{i,j}].$$

Converges in $\mathcal{O}(N)$ iterations, each one requires $\mathcal{O}(N)$ operations.

Gauss-Seidel method – use $\Phi_{i-1,j}^{(n+1)}, \Phi_{i,j-1}^{(n+1)}$ – twice as fast.

Successive over-relaxation: use also $\Phi_{i,j}^{(n)}$ in the RHS, or

$$\Phi_{i,j}^{(n+1)} = w \Phi_{i,j}^{(n+1)}_{\text{Jacobi}} + (1-w) \Phi_{i,j}^{(n)} \quad \text{with } 1 < w < 2.$$

Converges in $\mathcal{O}(N_{\text{grid}})$ only if the over-relaxation parameter is close to the optimal value $w \sim 2/(1 + \pi/N_{\text{grid}})$, but is less robust otherwise.

Multigrid relaxation

The speed of convergence for relaxation methods is limited by the need for the “signal” to propagate across the grid (N_{grid} iterations).

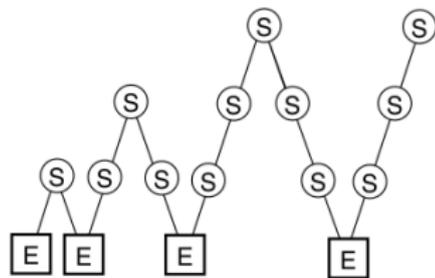
One may accelerate it by using a hierarchy of nested grids.

Algorithm for a two-level grid:

1. Compute the residual $\nabla^2\Phi^{(n)} - 4\pi\rho$ on the fine grid.
2. *Restrict* it to the coarse grid.
3. Solve for the correction on the coarse grid.
4. Interpolate it back to the fine grid and update $\Phi^{(n+1)}$.

This cycle is recursively applied down to the coarsest grid, in which the correction is found exactly.

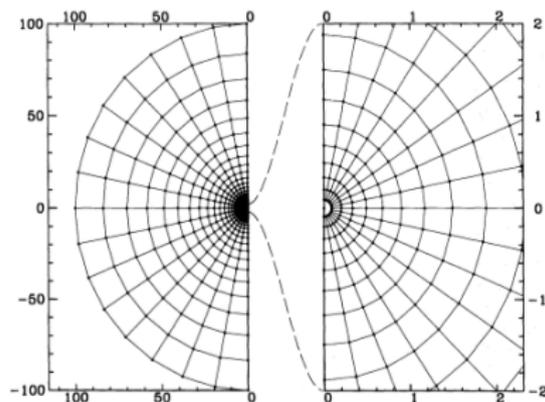
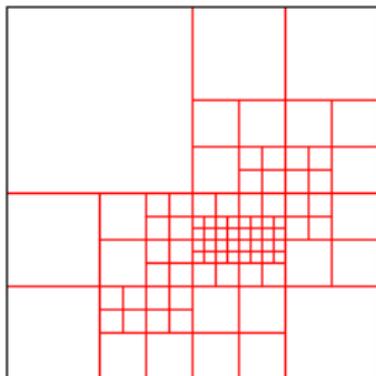
The method has complexity $\mathcal{O}(N)$ and is well-suited for AMR grids [Ricker 2008].



More on grid methods

Uniformly-spaced grids are not efficient for strongly clustered density distributions. Possible solutions to increase the resolution:

- ▶ Adaptive mesh refinement.
- ▶ Separation of total force into long-range (on the grid) and short-range parts.
- ▶ Non-cartesian grids.



Hybrid P³M (TreePM) algorithms

Idea: use the grid-based Poisson solver for long-range (particle–mesh) interactions, and particle–particle interactions on a shorter (subgrid) scale.

- ▶ Modify the long-range potential in the Fourier space:

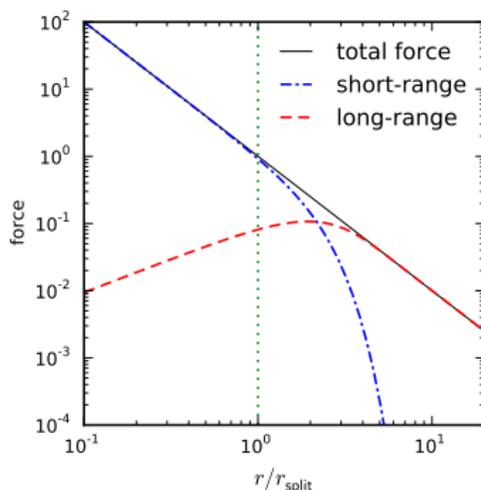
$$\Phi_{\text{long}}(\mathbf{k}) = 4\pi\rho(\mathbf{k})\mathbf{k}^{-2} \exp(-\mathbf{k}^2 r_{\text{split}}^2).$$

- ▶ Add the short-range potential:

$$\Phi_{\text{short}}(\mathbf{x}) = -\sum_i \frac{Gm_i}{|\mathbf{x}-\mathbf{x}_i|} \left[1 - \operatorname{erf}\left(\frac{|\mathbf{x}-\mathbf{x}_i|}{2r_{\text{split}}}\right) \right].$$

The short-range force is computed with the tree method, but using only the particles within $\sim 5r_{\text{split}}$, where the transition radius $r_{\text{split}} \simeq h$ (cell size).

Advantages: more accurate long-range force than with a pure tree code, and better force resolution than with a grid code.



Mass assignment

Assign the discretized density values $\rho(\mathbf{x}_{ijk})$, $\mathbf{x}_{ijk} \equiv \{x_i, y_j, z_k\}$:
the mass of a particle located at \mathbf{x}_p is distributed in space
according to a kernel $S(\mathbf{x} - \mathbf{x}_p)$:

| name | 1d kernel $S(x)$ | # of cells | force properties |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|--------------|----------------------------------|
|  nearest grid point (NGP) | $\delta(x)$ | 1 | piecewise-constant |
|  clouds in cells (CIC) | $\Pi(x) \equiv \begin{cases} 1 & \text{if } x < \frac{h}{2} \\ 0 & \text{otherwise} \end{cases}$ | 8 (in 3d) | continuous (piecewise-linear) |
|  triangular-shaped clouds (TSC) | $\Pi(x - \frac{h}{2}) \Pi(x + \frac{h}{2})$ | 27 | continuously differentiable |

NB: the force on each particle is obtained by interpolation between forces at grid nodes; the order of force and density interpolations must match to ensure momentum conservation.

Basis-set expansion for Poisson equation

Idea: use series representation for ρ and Φ .

$$\rho(\mathbf{x}) = \sum_{\alpha} A_{\alpha} \rho_{\alpha}(\mathbf{x}), \quad \Phi(\mathbf{x}) = \sum_{\alpha} A_{\alpha} \Phi_{\alpha}(\mathbf{x}), \quad \nabla^2 \Phi_{\alpha} = 4\pi \rho_{\alpha}.$$

Example in 1d: Fourier decomposition, $\rho_k = e^{ikx}$, $\Phi_k = k^{-2} e^{ikx}$.

Most easily formulated with bi-orthogonal basis functions:

$$\int d\mathbf{x} \rho_{\alpha}(\mathbf{x}) \Phi_{\beta}(\mathbf{x}) \propto \delta_{\alpha\beta}.$$

Algorithm:

1. Compute the coefficients of density expansion

$$A_{\alpha} = \int d\mathbf{x} \rho(\mathbf{x}) \Phi_{\alpha}(\mathbf{x}) = \sum_{i=1}^{N_{\text{body}}} m_i \Phi_{\alpha}(\mathbf{x}_i) \quad \text{for } \forall \alpha.$$

2. Obtain the force at each particle's location

$$\ddot{\mathbf{x}}_i = \sum_{\alpha} A_{\alpha} \nabla \Phi_{\alpha}(\mathbf{x}_i) \quad \text{for } i = 1..N_{\text{body}}.$$

Basis-set expansion: spherical harmonics

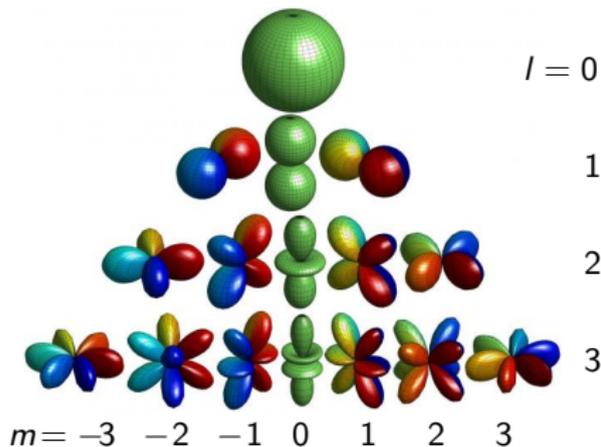
The standard choice for 3d systems:

$$\rho_\alpha \equiv \rho_{nlm} = B_{nl}(r) Y_l^m(\theta, \phi),$$

$$Y_l^m \equiv P_l^m(\cos \theta) e^{im\phi},$$

$$n = 0..n_{\max}, \quad l = 0..l_{\max}, \quad m = -l..l,$$

where P_l^m are Legendre polynomials, and B_{nl} are typically some other orthogonal polynomials in scaled r .



Used in the self-consistent field method [e.g., Hernquist & Ostriker 1992].

Advantages (and drawbacks):

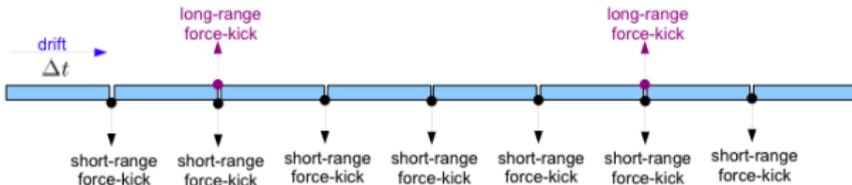
- ▶ Fast – $\mathcal{O}(N_{\text{body}} N_{\text{basis}})$ operations, easily parallelized.
- ▶ Smooths out small-scale fluctuations (effective softening length is large).
- ▶ Moderately suppresses two-body relaxation (factor of few).
- ▶ May enforce a given kind of symmetry (retain only a subset of terms).
- ▶ Suitable only for isolated systems with well-defined center.

Equations of motion

Integration method: almost always leap-frog.

Large dynamic range \implies need to use adaptive (typically block) timesteps (e.g., $\Delta t = [\eta\epsilon/|\mathbf{a}|]^{1/2}$, where ϵ is the softening length, and η is the accuracy parameter).

Block timesteps are well adapted for AMR grids;
force splitting in the TreePM method is used to separate long- (slow) and short-range (fast) variations:



Cosmology: comoving frame

In cosmological simulations the dynamics is followed in *comoving* coordinates \mathbf{x} , such that the physical distance and velocity are $\mathbf{r} = a(t) \mathbf{x}$, $\mathbf{v} = a(t) \dot{\mathbf{x}}$, and $a(t) \propto (1+z)^{-1}$ is the scale factor (determined by the background cosmological model – not simulated).

The equations of motion are

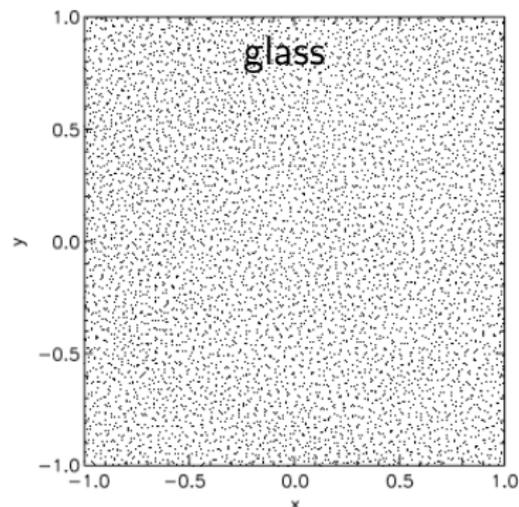
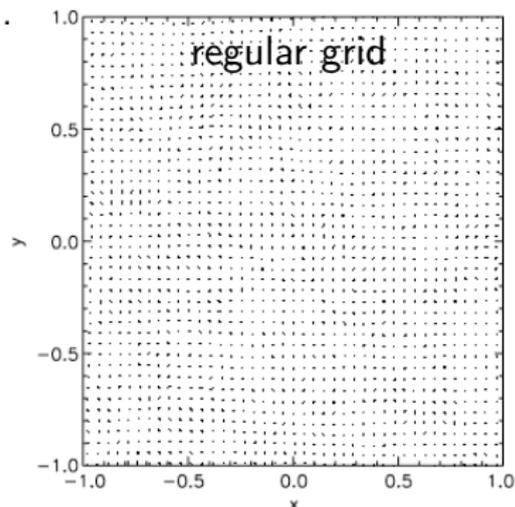
$$\frac{d\mathbf{v}}{dt} + H(t)\mathbf{v} = -\frac{1}{a(t)}\nabla\varphi(\mathbf{x}, t), \quad \text{where } H(t) \equiv \frac{\dot{a}}{a} \text{ is the Hubble}$$

parameter, and φ is the peculiar gravitational potential satisfying $\nabla^2\varphi(\mathbf{x}, t) = 4\pi G a(t)^2 [\rho(\mathbf{x}, t) - \bar{\rho}(t)]$.

The peculiar potential is computed using periodic boundary conditions (trivial with FFT, and employs Ewalds summation approach with tree-code, but not with TreePM).

Cosmology: initial conditions

1. Set up a nearly uniform particle distribution at high redshift ($z_0 \gtrsim 100$).



2. Generate a realization of random potential fluctuations $\varphi(\mathbf{x})$.
3. Add a linear perturbation to \mathbf{x} and \mathbf{v} (Zeldovich approximation):
$$\delta\mathbf{x} = -D(z_0) \nabla\varphi(\mathbf{x}), \quad \delta\mathbf{v} = \dot{D}(z_0) \nabla\varphi(\mathbf{x}),$$
where D is the amplitude of the growing mode at the initial redshift.

Cosmology: zoom simulations

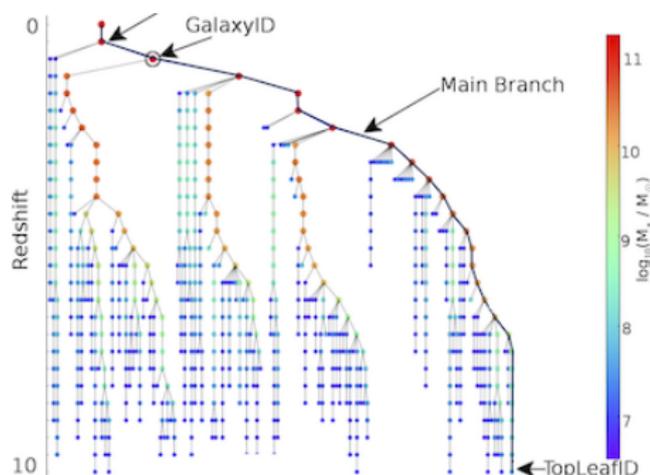
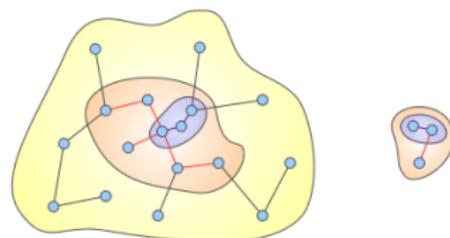
Idea: simulate individual galaxies or clusters in a wider cosmological context.

1. Take a final snapshot ($z = 0$) of a large-scale simulation with coarse resolution.
2. Identify particles that ended up in the region of interest.
3. Track them back to the initial redshift ($z \sim 100$).
4. Refine the entire region covered by these particles (using the same realization of spectrum of perturbations above the scale of the coarse simulation, but augmenting it with higher- k perturbations at smaller scales).
5. Perform the zoom-in simulation of the region of interest sampled with lower-mass particles and higher resolution, enclosed by some fraction of the volume of the coarse simulation.

Cosmology: halo finders

Goal: find bound structures in cosmological simulations.

- ▶ Locate overdensity regions ($\rho > \rho_{\text{threshold}}$).
- ▶ Link them together (friends-of-friends).
- ▶ Remove unbound particles ($v > v_{\text{escape}}$).
- ▶ Construct hierarchy of nested haloes.
- ▶ Build merger trees.



Lecture 4. Miscellaneous topics

- ▶ Computational aspects
- ▶ Unconventional simulation approaches
- ▶ Extra physics beyond pure gravity
- ▶ Summary

Parallelization

Performance-limiting factors:

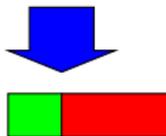
- ▶ Computational cost
- ▶ Memory requirements
- ▶ Communication

Amdahl's law:

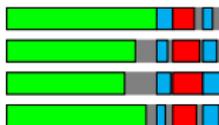
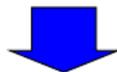
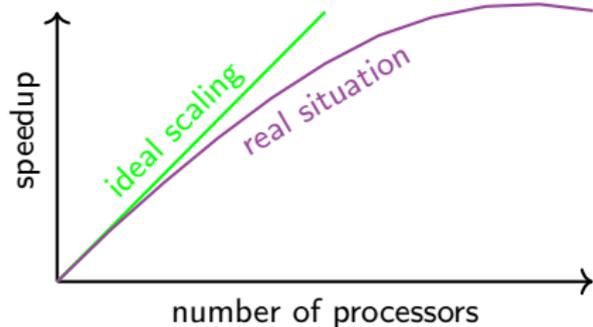
overall speedup cannot exceed $1/(1 - p)$,
where p is the fraction of time spent
in the part of code that can't be improved.



10x speed up task A



Communication and synchronization overhead:



communication

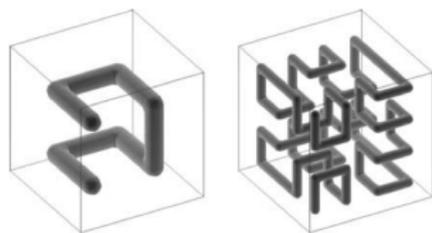
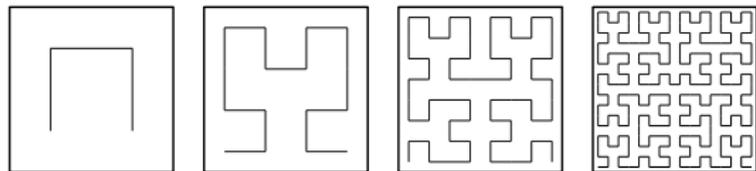
idle wait

Domain decomposition

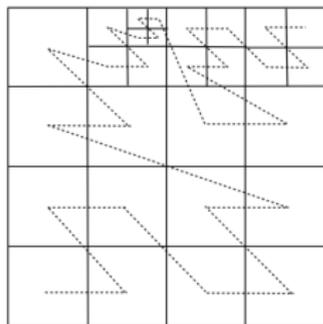
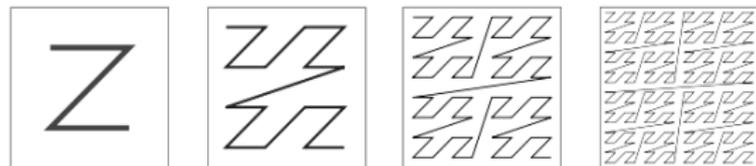
Task: Need to split the workload into approximately equal chunks, while minimizing the communication cost (maintain locality).

Solution: Space-filling curves.

Hilbert



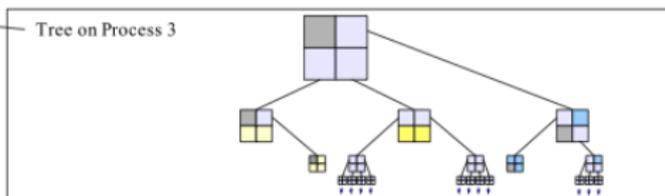
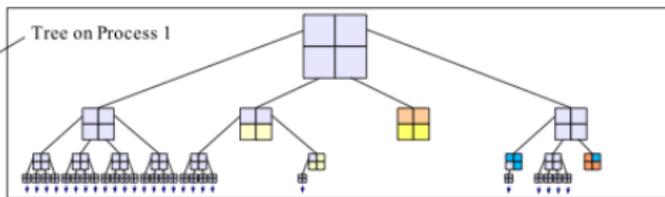
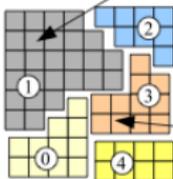
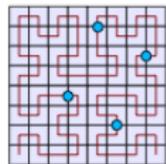
Morton



Domain decomposition for tree codes

- ▶ Sort particles according to their location along the space-filling curve.
- ▶ Split particles between processors in equal amounts.
- ▶ Each processor constructs and walks the tree for “local” particles:
 - ▶ Construct the tree down from the root cell, creating only non-empty “local” cells;
 - ▶ Obtain properties of topmost non-local cells (“pseudo-particles”) from other processors;
 - ▶ Find the forces for all local particles by tree-walk;
 - ▶ If a non-local node needs to be opened, it is added to the “to-do” list;
- ▶ The lists of requested interactions are exchanged between processors, and a second tree-walk to find forces from non-local particles is performed.

Domains are obtained by cutting the Peano-Hilbert curve into segments

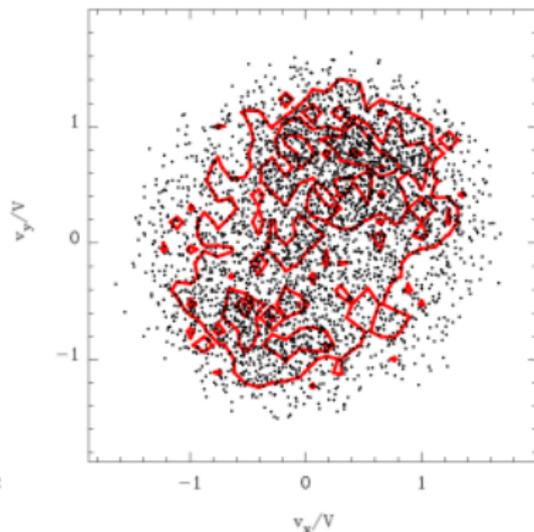
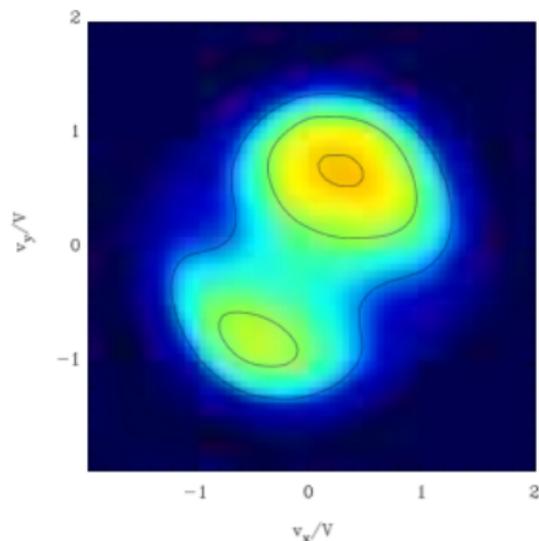


[from Springel 2005]

Direct solution of collisionless Boltzmann equation

Grid-based solution of the Vlasov–Poisson system – recently has been demonstrated in the 3d case [Yoshikawa+ 2012].

(Flux-conservative positive-definite Vlasov solver on a uniform 64^6 grid).



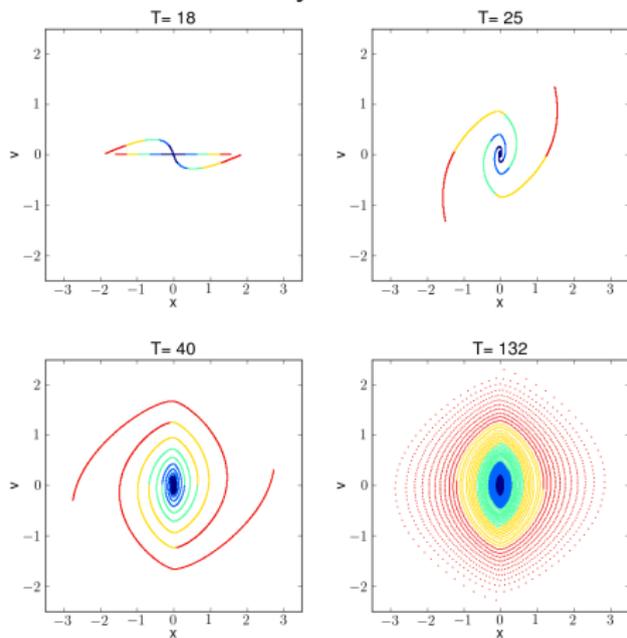
Challenges: dynamical range, development of fine-grained structure.

Vlasov–Poisson system in 1d

1d N -body system is “exactly” solvable (up to roundoff error):

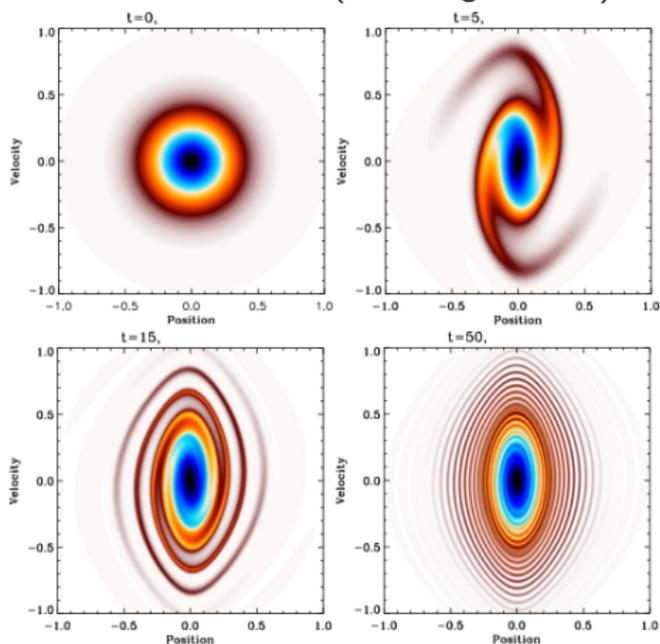
$$\Phi(x_i) = 2\pi G \sum_j |x_i - x_j| - \text{the force is constant between collisions.}$$

N-body simulation



[from Schulz+ 2012]

Vlasov–Poisson (waterbag method)

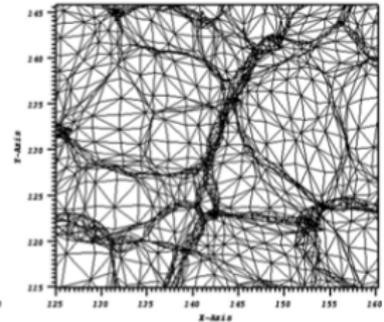
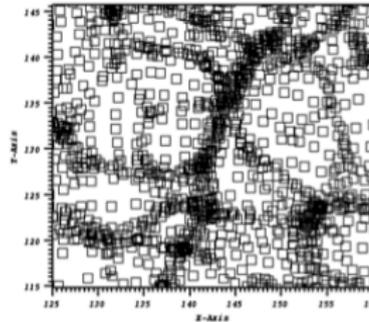
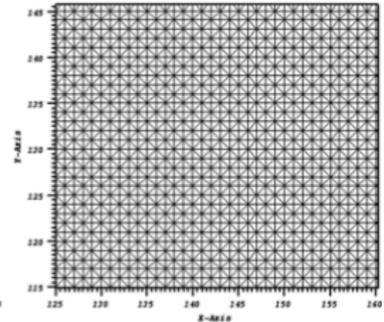
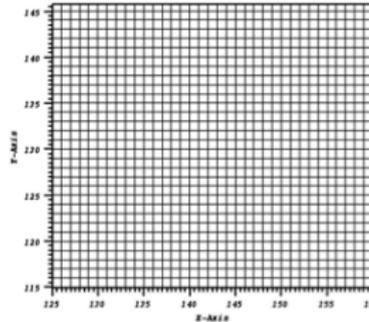
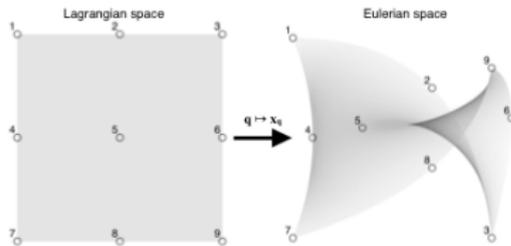


[from Colombi & Touma 2014]

Tessellation-based method for dark matter simulations

Idea: Cold dark matter is localized on a 3d sheet in 6d phase space; track the nonlinear evolution of this sheet in Lagrangian coordinates.

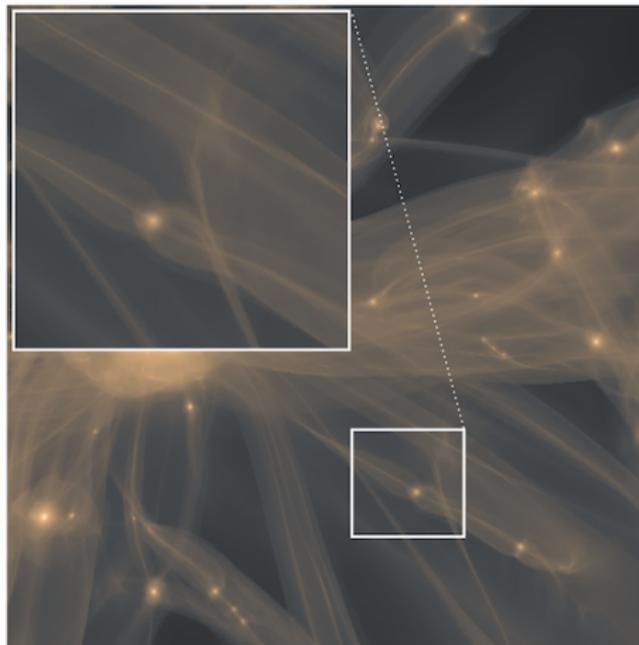
- ▶ Tessellate the initial dark matter distribution into polyhedra;
- ▶ Distribute the mass of each polyhedron over the space enclosed by its vertices;
- ▶ Compute the gravity and move the vertices as N -body particles.
- ▶ Refine twisted polyhedra.



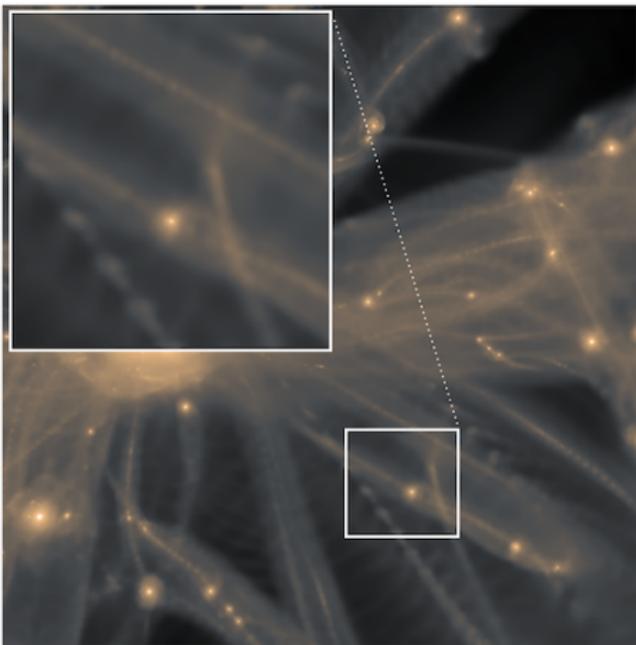
[Shandarin+ 2012; Hahn+ 2013, 2016; Sousbie & Colombi 2016]

Tessellation-based method for dark matter simulations

tessellation method



conventional N -body simulation



[from Kaehler+ 2012]

Advantage: better track of caustics at early stages of evolution.

Challenges: complicated mass assignment for Poisson solver, rapid increase of # of refined polyhedra with time.

Collisional systems – alternative approaches

The collisional Boltzmann equation:

$$\begin{aligned}\frac{df}{dt} &\equiv \frac{\partial f}{\partial t} + \mathbf{v} \frac{\partial f}{\partial \mathbf{x}} - \frac{\partial \Phi}{\partial \mathbf{x}} \frac{\partial f}{\partial \mathbf{v}} = \Gamma[f] = \\ &= \int d^6 \Delta \mathbf{w} [\Pi(\mathbf{w} - \Delta \mathbf{w}, \Delta \mathbf{w}) f(\mathbf{w} - \Delta \mathbf{w}) - \Pi(\mathbf{w}, \Delta \mathbf{w}) f(\mathbf{w})]\end{aligned}$$

$\Pi(\mathbf{w}, \Delta \mathbf{w})$ is the transition probability from \mathbf{w} to $\mathbf{w} + \Delta \mathbf{w}$
($\mathbf{w} \equiv \{\mathbf{x}, \mathbf{v}\}$).

In the limit of weak encounters, we may expand Γ to 2nd order and obtain the Fokker–Planck equation:

$$\Gamma[f] = -\frac{\partial}{\partial w_i} [\mathcal{D}_i(\mathbf{w}) f(\mathbf{w})] + \frac{\partial^2}{\partial w_i \partial w_j} [\mathcal{D}_{ij}(\mathbf{w}) f(\mathbf{w})]$$

$$\mathcal{D}_i \equiv \int d^6 \Delta \mathbf{w} \Pi(\mathbf{w}, \Delta \mathbf{w}) \Delta w_i$$

$$\mathcal{D}_{ij} \equiv \int d^6 \Delta \mathbf{w} \Pi(\mathbf{w}, \Delta \mathbf{w}) \Delta w_i \Delta w_j$$

drift and diffusion coefficients

Fokker–Planck equation

- ▶ Local approximation: $T_{\text{encounter}} \ll T_{\text{dyn}} \implies$
 - a) encounters affect only the velocity, not the position:
 $f(\mathbf{x}, \mathbf{v})$ changes due to $\mathcal{D}_{\mathbf{v}}(\mathbf{x}, \mathbf{v})$;
 - b) \mathcal{D} computed assuming uniform homogeneous background.
- ▶ Orbit-averaged approximation: $T_{\text{dyn}} \ll T_{\text{rel}} \implies$
use only the integrals of motion \mathcal{I} as arguments of both f and \mathcal{D} .

Most existing studies use the orbit-averaged approach with the classical integrals $\mathcal{I} = \{E\}$, $\{E, L\}$ or $\{E, L_z\}$, discretizing $f(\mathcal{I})$ and $\Phi(\mathbf{x})$ on a 1d or 2d grid.

Optional (needed for dynamical self-consistency):

recomputation of diffusion coefficients $\mathcal{D}_{\mathcal{I}}$ as integrals over $f(\mathcal{I})$;

update of $\Phi(\mathbf{x})$ – solution of the Poisson equation with

$$\rho(\mathbf{x}) \equiv \iiint f(\mathcal{I}(\mathbf{x}, \mathbf{v})) d^3v.$$

Monte Carlo method

A particle-based method for solving the Fokker–Planck equation:

- ▶ $f(\mathbf{x}, \mathbf{v})$ or $f(\mathcal{I})$ is discretized into particles.
- ▶ After each timestep, particle velocities (in the local approach) or integrals of motion (in the orbit-averaged approach) are perturbed according to the diffusion coefficients \mathcal{D} .
- ▶ The relaxation rate may be scaled to a different number of stars N_* than the number of particles N .
- ▶ In the local approach [Spitzer & Hart 1971, Vasiliev 2015], particle trajectories are computed numerically in the potential $\Phi(\mathbf{x})$, with a timestep $\tau \ll T_{\text{dyn}}$.
- ▶ In the orbit-averaged approach [Hénon 1971, Shapiro & Marchant 1980], particles are not localized, and the timestep is $T_{\text{dyn}} \ll \tau \ll T_{\text{rel}}$.
- ▶ Easy to include additional physics (stellar evolution, mass spectrum, binaries and few-body encounters, etc.)
- ▶ More expensive than grid-based Fokker–Planck methods, but much cheaper than direct N -body simulations (cost scales $\propto N$).

Hybrid methods

- ▶ Fokker–Planck + N -body [McMillan & Lightman 1984 – globular clusters, Glaschke et al. 2014 – protoplanetary discs].
- ▶ Monte Carlo + N -body [Rodriguez et al. 2015 – globular clusters].
- ▶ Monte Carlo + three-body scattering
[Vasiliev et al. 2015 – binary supermassive black holes].
- ▶ Self-consistent field + N -body [Quinlan & Hernquist 1997, Hemsendorf et al. 2002 – galactic nuclei].
- ▶ Tree-code + two-body regularization [McMillan & Aarseth 1993].
- ▶ “Bridge” coupling of two or more independent N -body codes [Fujii et al. 2007 – used in AMUSE].

Hydrodynamics – Two main approaches

Eulerian:

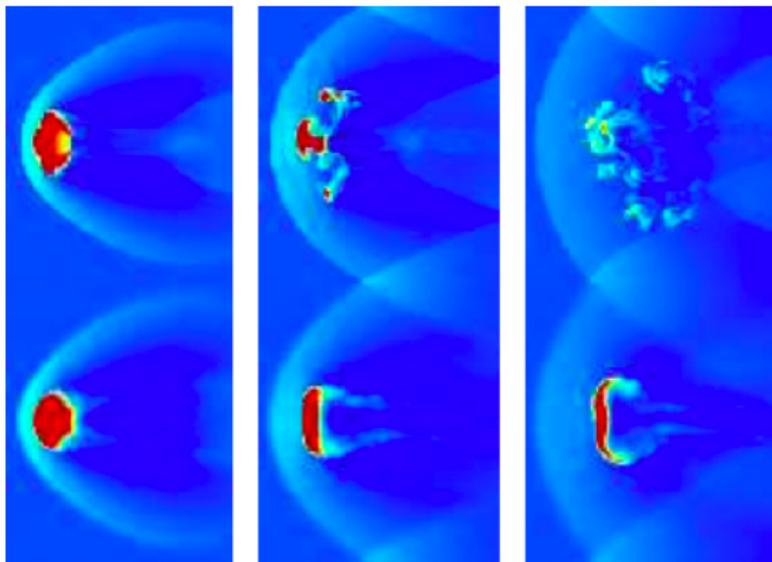
- ▶ dynamical quantities are discretized on a grid;
- ▶ naturally couple to grid-based gravity solvers;
- ▶ high-order finite-volume methods for solving HD PDEs;
- ▶ dynamical range increased by AMR;
- + accurate shock-capturing schemes;
- direction-dependent and velocity-dependent advection errors;
- numerical heating and diffusion, especially severe for cold supersonic flows.

Lagrangian (SPH):

- ▶ dynamical quantities are associated with particles;
- ▶ naturally couple to tree-code gravity solvers;
- + spatial resolution adapts to density;
- + invariant w.r.t. bulk velocity;
- noisy kernel estimates;
- poor shock resolution;
- “surface tension” at contact discontinuities.

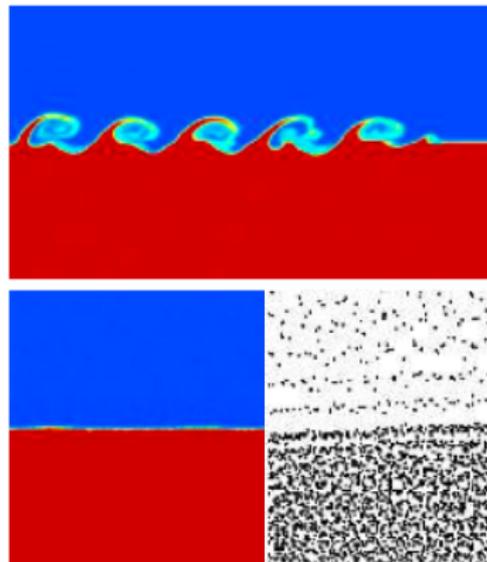
Grid vs. SPH

Eulerian method



Lagrangian method

Kelvin-Helmholtz instability

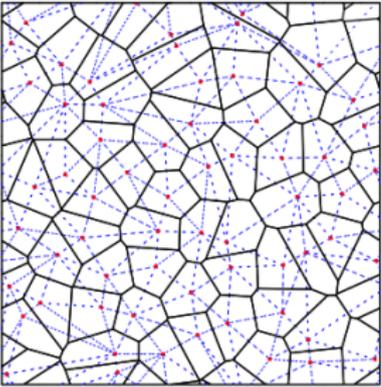


[from Agertz+ 2007]

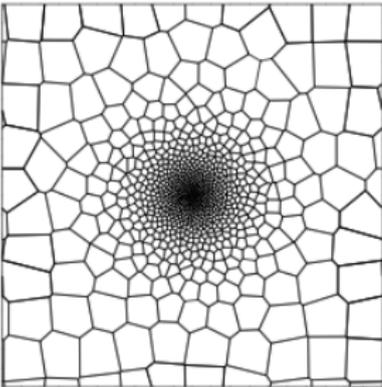
Moving-mesh hydrodynamics

Combine the advantages of grid- and SPH-based methods:

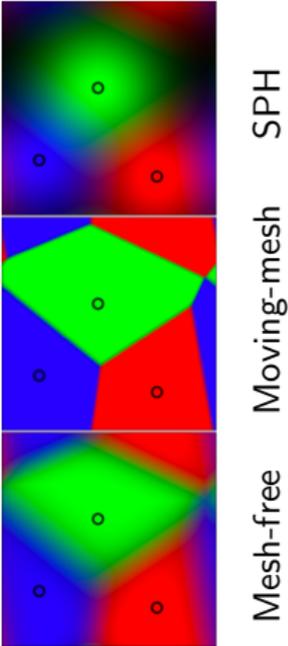
- ▶ Use Voronoi tessellation to maintain an unstructured mesh associated with moving particles.
- + High-accuracy grid-based Riemann solver.
- + Spatial adaptivity and Galileian invariance.



Voronoi and Delaunay tessellations

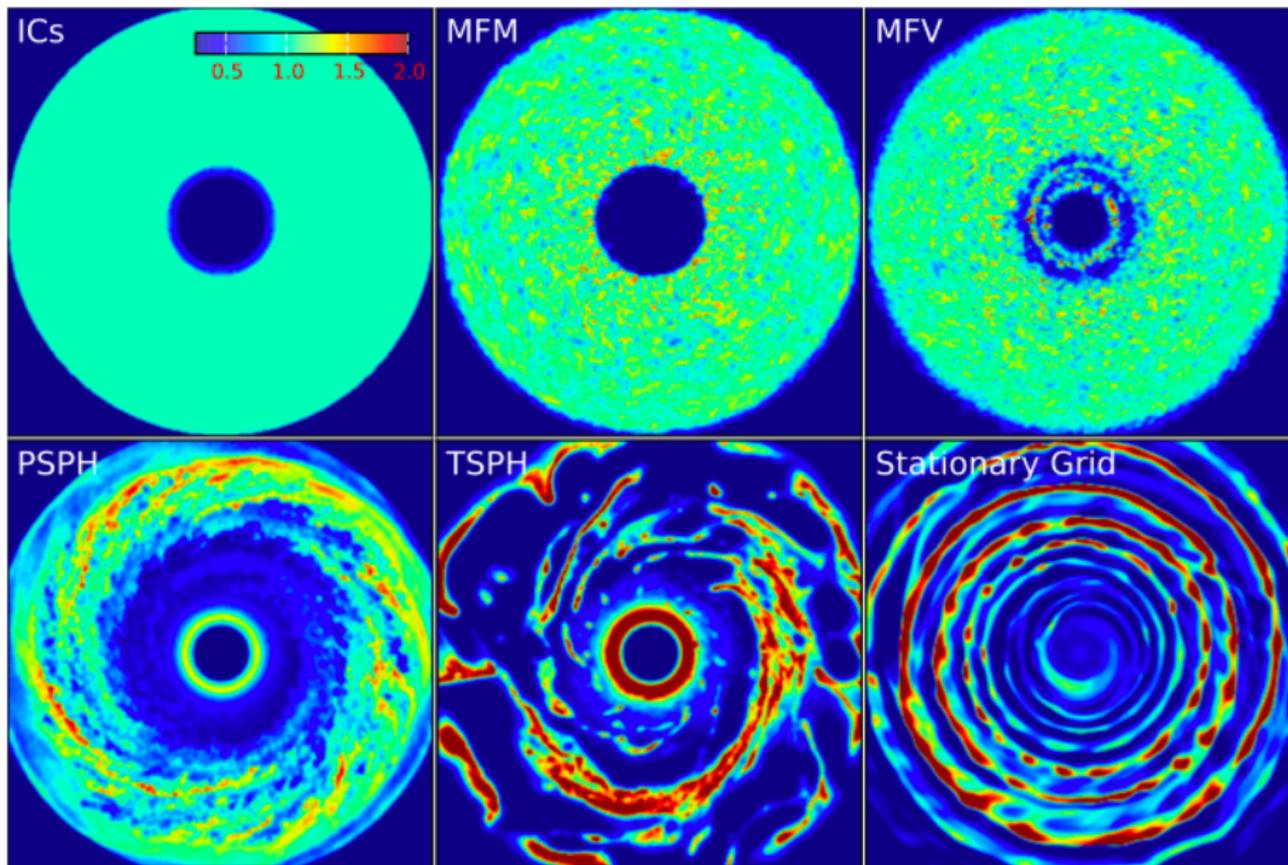


[from Springel 2010]



[from Hopkins 2014]

Example: a gaseous disk in a Kepler potential



[from Hopkins 2014]

Coupling strategies

- ▶ Stellar dynamics (gravity)
- ▶ Stellar evolution
- ▶ Hydrodynamics
- ▶ Radiative transfer
- ▶ ...

Various approaches for putting these ingredients together:

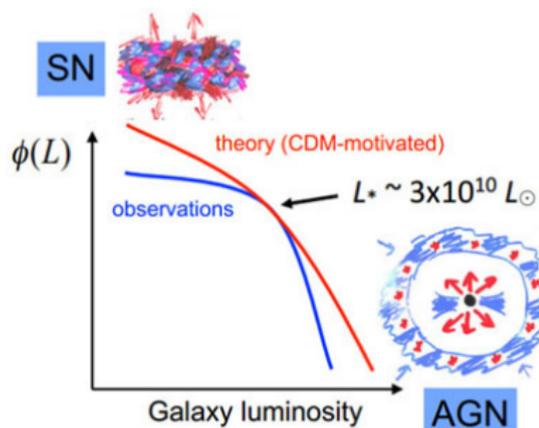
- ▶ Monolithic (“kitchen sink”) codes
(examples: `nbodyX` series and Monte Carlo codes for collisional systems, most cosmological codes for collisionless systems).
- ▶ Plugin architecture (examples: FLASH, HACC).
- ▶ Loose coupling between independent codes
(example: AMUSE project).
- ▶ Post-processing steps after pure gravity simulations
(examples: many DM-only cosmological simulations, semianalytic models of galaxy formation).

Sub-grid physics

- ▶ Multi-phase gas, chemical composition.
- ▶ Radiative cooling and heating, opacity, self-shielding.
- ▶ Star formation.
- ▶ Stellar evolution and mass loss.
- ▶ Energy injection from supernovae and stellar winds.
- ▶ Accretion and AGN feedback from supermassive black holes.

Calibration:

- ▶ Galaxy stellar mass function.
- ▶ Cosmic star formation rate evolution.
- ▶ Size–luminosity, Tully–Fisher and other galaxy relations.
- ▶ Relations between M_{BH} and galaxy properties.



[from Silk & Mamon 2012]

Summary

- ▶ Gravity rules the Universe and is a very simple phenomenon.
- ▶ Two classes of physical systems – collisional and collisionless, require different numerical methods.
- ▶ Numerical simulations are computationally challenging due to long-range nature of gravitational force and large dynamical range.
- ▶ Many different codes are publicly available and well-tested.
- ▶ Solving gravity is (mostly) a solved problem; extra physics beyond gravity is still in active development.

The end.