Time Integration Techniques

Explicit Euler

$$y_{n+1} = y_n + f(y_n)\Delta t$$

Only stable for sufficiently small Δt. First order accurate, not time symmetric.

Implicit Euler

$$y_{n+1} = y_n + f(y_{n+1})\Delta t$$

Very stable, still first order accurate and not time-symmetric.

Implicit midpoint Euler

$$y_{n+1} = y_n + f\left(\frac{y_n + y_{n+1}}{2}\right)\Delta t$$

Symmetrized variant of implicit/explicit Euler. Second order accurate, time-symmetric.

Runge-Kutta-Methods

Idea: Predict unknown y_{n+1} by an Euler step:

$$y_{n+1} = y_n + \frac{f(y_n) + f(y_{n+1})}{2}\Delta t$$

$$\Rightarrow \quad k_1 = f(y_n, t_n)$$

$$k_2 = f(y_n + k_1 \Delta t, t_{n+1})$$

$$y_{n+1} = \frac{k_1 + k_2}{2}\Delta t + y_n$$

Second order accurate

4th order:

$$k_1 = f(y_n, t_n)$$

$$k_2 = f\left(y_n + k_1\frac{\Delta t}{2}, t_n + \frac{\Delta t}{2}\right)$$

$$k_3 = f\left(y_n + k_2\frac{\Delta t}{2}, t_n + \frac{\Delta t}{2}\right)$$

$$k_4 = f(y_n + k_3 \Delta t, t_n + \Delta t)$$

$$y_{n+1} = y_n + \left(\frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}\right)\Delta t + \mathcal{O}(\Delta t^5)$$

Leapfrog

For second order differential equations of the type $\ddot{x} = f(x)$,

$V \equiv \dot{x}$

Then

$$v_{n+1/2} = v_n + f(x_n)\frac{\Delta t}{2}$$

$$x_{n+1} = x_n + v_{n+1/2}\,\Delta t$$

$$v_{n+1} = v_{n+1/2} + f(x_{n+1})\frac{\Delta t}{2}$$

Time-symmetric, symplectic, second-order accurate.

Superior long-term stability: the leapfrog generates the exact solution of a modified Hamiltonian $H_{leap} = H + H_{err}$, $H_{err} \propto \frac{\Delta t^2}{12}$

It respects phase-space conservation and the long term energy error stays bounded and reasonably small.

Collisionless N-body Dynamics

The state of an N-particle ensemble at time t can be specified by the exact distribution function:

$$F(\vec{r}, \vec{v}, t) = \sum_i^N \delta(\vec{r} - \vec{r}_i(t))\,(\vec{v} - \vec{v}_i(t))$$

which effectively gives the number density of particles in phase space point $(\vec{r}, \vec{v})$. Now let $\rho(\vec{r}_1, \dots, \vec{r}_N, \vec{v}_1, \dots, \vec{v}_N, t)\, d^3\vec{r}_1 \dots d^3\vec{r}_N\, d^3v_1 \dots d^3v_N$ be the probability that the system is in the given state at time t. Then a reduced statistical description is given by ensemble averaging:

$$f_1(\vec{r}, \vec{v}, t) = \langle F(\vec{r}, \vec{v}, t) \rangle = \int F \cdot \rho \, d\vec{r}_1 \dots d\vec{r}_N\, d\vec{v}_1 \dots d\vec{v}_N$$

f_1 gives the mean number of particles in a phase space volume around $(\vec{r}, \vec{v})$. Using Liouville's theorem:

$$\frac{\partial f_1}{\partial t} + \vec{v}_1 \cdot \frac{\partial f_1}{\partial \vec{x}_1} + \underbrace{\frac{\vec{F}_1^{\,ext}}{m_1} \cdot \frac{\partial f_1}{\partial \vec{v}_1}}_{\text{external force term}} + \underbrace{\int \frac{\vec{F}_{12}}{m_1} \frac{\partial f_2}{\partial \vec{v}_1}\, d\vec{x}_2\, d\vec{v}_2}_{\text{interparticle force term}}$$

The evolution equation for f_1 involves a term in f_{1+1}. Some approximations are required to close the chain of equations.

Simplest closure relation: Assume uncorrelated systems.

$$f_2(\vec{r}, \vec{r}', \vec{v}, \vec{v}', t) = f_1(\vec{r}, \vec{v}, t)\, f_1(\vec{r}', \vec{v}', t)$$

meaning that a particle at $(\vec{r}, \vec{v})$ is completely unaffected by a particle at $(\vec{r}', \vec{v}')$. The dynamical equations follow from the notion of Probability conservation: $P = \int_\Gamma d^6\vec{w}\ \vec{w} = \text{const}$. That gives the Vlassov/collisionless Boltzmann equation:

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \vec{v}\frac{\partial f}{\partial \vec{r}} + \vec{a}\frac{\partial f}{\partial \vec{v}} = 0$$

The acceleration can't be due to another single particle, but collective effects (gravitational field produced by the whole system) are still allowed. The gravitational field of the mass density is given by the Poisson equation:

$$\nabla^2 \phi(\vec{r}) = 4\pi G \rho(\vec{r}, t); \qquad \rho(\vec{r}) = m n(\vec{r}) = m \int f(\vec{r}, \vec{v}, t)\, d\vec{v}$$

$$a = -\nabla \phi$$

When can a system be considered collisionless? When $t_{age} \ll t_{relax}$:

t_{age} = time of interest; $t_{relax} \sim \frac{N t_{cross}}{8\pi \ln(N/2)}$; $t_{cross} \sim R/v$

For weakly correlated systems: Fluid approximation

N-Body Techniques

Particle-Mesh (PM) Methods

Uses auxiliary mesh. Construct density field ρ on a suitable mesh, then compute ϕ using $\Delta\phi = 4\pi G\rho$ and forces from $\vec{F} = -\nabla\phi$ on the mesh. Lastly, compute force on particle from force on mesh.

Acquiring ρ: Associate each particle a shape to spread the mass of particles across cells (NGP, CIC, TSC)
Introduces second discretisation: first being the use of superparticles (discretises physics), mesh discretises the physical model.

Tree-Codes

Hierarchical multipole methods. Use multipole expansion for distant groups of particles instead of summing up the forces from all individual particles.
Hierarchical grouping of particles, for each group, the multipole moments are pre-computed. Tree codes are geometrically highly flexible and adjust to any clustering state the particles have.

Hybrid Schemes

P^3M: Use PP (direct summation) for short range, PM for long range force terms. Problem: CPU time dominated by PP-part for highly clustered configurations.

AP^3M: Use only coarsest grid for long range force calculation. PM is used for finer grids, and PP for finest. Not more accurate than P^3M, but faster.

Tree PM: large scale forces calculated by PM, short scales by trees. Ideal for cosmological simulations, since it avoids problem of introducing periodic boundaries. Exploit natural periodicity of the long-range part performed by a FFT.

Poisson Solvers

Iterative Methods

Turn non-linear PDE $\Delta\phi = 4\pi G\rho$ into a set of linear equations that can be solved on the mesh: $\left(\frac{\partial^2\phi}{\partial x^2}\right)_i \simeq \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2} + \mathcal{O}(h^2)$

$= 4\pi G\rho_i$

This is a problem of the form $A\vec{x} = \vec{b}$ with $\vec{x} = (\phi_i)$, $\vec{b} = h^2 4\pi G\rho_i$.

But: $A\vec{x} = \vec{b}$ is $\mathcal{O}(N^3)$ → Split up matrix A in smaller matrices using repeated iterations after an initial guess until convergence is reached. Convergence is achieved when the absolute or relative difference $|x_i^{(n)} - x_i^{(n-1)}| <$ some parameter, where (n) is the iteration counter.

Jacobi-Iteration

Decompose: $A = D - (L + U)$ [Diagonal − (lower + upper)] part of matrix

$\Rightarrow \vec{x} = D^{-1}\vec{b} + D^{-1}(L+U)\vec{x}$

$\Rightarrow$ Define $\vec{x}^{(n+1)} = D^{-1}\vec{b} + D^{-1}(L+U)\vec{x}^{(n)}$

Gauss-Seidel

Jacobi computes value for all cells. Gauss-Seidel: Use updated values as soon as they become available. Again from $A\vec{x} = \vec{b}$, $A = D - (L+U)$

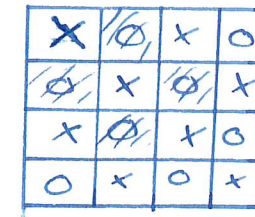
$\Rightarrow \vec{x} = (D-L)^{-1}U\vec{x} + (D-L)^{-1}\vec{b}$

$\Rightarrow$ Use $\vec{x}^{(n+1)} = (D-L)^{-1}U\vec{x}^{(n)} + (D-L)^{-1}\vec{b}$.

$\Rightarrow \vec{x}^{(n+1)} = D^{-1}U\vec{x}^{(n)} + D^{-1}L\vec{x}^{(n+1)} + D^{-1}\vec{b}$

$\Rightarrow$ Start computing highest row ($i=0$), where no lower elements enter the equation. For the second, then only the first is needed.

Red - Black - Ordering



Compromise between Jacobi and Gauss-Seidel.

Remember Poisson: $\frac{\Phi_{i+1} - 2\Phi_i + \Phi_{i-1}}{h^2} = 4\pi G \rho_i$

⇒ Only need cell itself and 1 neighbour

⇒ Update in a chess-like pattern fashion to converge faster.

Multigrid Technique

Speed up iterative techniques by computing initial guess on a coarser grid, where it takes less time to compute. (better)

Define Prolongation: Map coarse-to-fine

Define Restriction: Map fine-to-coarse

Coarse grid correction scheme:

- Carry out 1 relaxation step on h (Gauss-Seidel/Jacobi)
- Compute residual $\vec{r}^{(h)} = \vec{b}^{(h)} - A^{(h)} \vec{x}^{(h)}_{current}$
- Restrict residual to higher mesh: $\vec{r}^{(2h)}$
- Solve $A^{(2h)} \vec{e}^{(2h)} = \vec{r}^{(2h)}$ on coarser mesh; Error vector $\vec{e}^{(2h)}$ has initial guess $\vec{e}^{(0)} = 0$.
- Prolong obtained $\vec{e}^{(2h)}$ to finer mesh, correct current solution: $\vec{x}_{new} = \vec{x}_{current} + e$
- Repeat until converged

Solving $A\vec{e} = \vec{r}$ can be done recursively on coarser and coarser grids. (V-cycle). At the maximally coarse grid, one solves the problem exactly.

Obtaining A^{2h}:

- Direct approximation: Discretise Poisson directly on coarse grid
- Galerkin approximation: $A^{(2h)} = I_h^{2h} A^{(h)} I_{2h}^{h}$

Fourier Techniques

Knit potential $\phi = -\int G \frac{\rho(x')\,dx'}{|\vec{x}-\vec{x}'|} = \int g(\vec{x}-\vec{x}')\,\rho(\vec{x}')\,dx' = g * \rho$

as convolution.

Use Fourier transforms: $\mathcal{F}[g*\rho] = \mathcal{F}[g]\cdot\mathcal{F}[\rho]$

$\Rightarrow \hat{\phi} = \hat{g}\cdot\hat{\rho} = -\frac{4\pi G}{k^2}\hat{\rho}$ $\Rightarrow$ Derivation reduces to multiplication

$\Rightarrow$ Compute $\hat{\rho}$, then $\hat{\phi}$; then transform back: $\phi = \mathcal{F}[\hat{\phi}]$

On the mesh we need to use discrete Fourier transforms.
Watch out for aliases! $k_{Nyquist} = k_{max} = \frac{N}{2}\frac{2\pi}{L}$

The DFT can be sped up with the FFT, where terms are split into parts, which are re-used. Reduces $\mathcal{O}(N^2) \to \mathcal{O}(N \log_2 N)$

The DFT is intrinsically periodic. For non-periodic problems use zero-padding trick:

- Arrange mesh such that the source distribution is only in one square/oct, the rest of the density field = 0.
- Set up real-space Green's function $g_{ij} = \frac{-G}{|\vec{x}_i - \vec{x}_j|}$, the function for the whole mesh is mirrored.
- Carry out real-space convolution $\phi_{\vec{p}} = \sum_{\vec{n}} g_{\vec{p}-\vec{n}}\,\rho_{\vec{n}}$, both ρ and g are treated as periodic fields. Is still correct over the source region.
- Use FFT to solve convolution quickly.

Numerical Hydrodynamics

Introduction

- Eulerian approach:
  static frame of reference, from which flow is observed.
  ↳ grid-based codes with fixed meshes
- Lagrangian approach
  frame of reference comoving with flow.
  ↳ particle-based codes with macroparticle tracers.

Euler equations:

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \vec{v}) = 0 \qquad \frac{d\rho}{dt} + \rho \nabla\cdot\vec{v} = 0$$

$$\frac{\partial(\rho\vec{v})}{\partial t} + \nabla(\rho\vec{v}\vec{v}^T + P) = 0 \qquad \frac{d\vec{v}}{dt} + \frac{\nabla P}{\rho} = 0$$

$$\frac{\partial(\rho e)}{\partial t} + \nabla[(\rho e + P)\vec{v}] = 0 \qquad \frac{de}{dt} + \frac{P}{\rho}\nabla\cdot\vec{v} = 0$$

$e = u + \vec{v}^2/2$ total energy per unit mass,
u = thermal energy per unit mass

Navier-Stokes:

$$\frac{\partial \rho}{\partial t} + \nabla(\rho\vec{v}) = 0$$

$$\frac{\partial}{\partial t}(\rho\vec{v}) + \nabla(\rho\vec{v}\vec{v}^T + P) = \nabla\Pi$$

$$\frac{\partial}{\partial t}(\rho e) + \nabla[(\rho e + P)\vec{v}] = \nabla(\Pi\vec{v})$$

$$\Pi = \eta\left[\nabla\vec{v} + (\nabla\vec{v})^T - \frac{2}{3}(\nabla\cdot\vec{v})\mathbb{1}\right] + \zeta(\nabla\cdot\vec{v})\mathbb{1}$$

= viscous stress tensor (material property)

η: shear viscosity (traceless part of Π)
ζ: bulk viscosity (diagonal part of Π)

Navier-Stokes include internal stresses, which dissipate relative motions of the fluid into heat. In many astrophysical situations, the densities and therefore the internal friction are very low, making the Euler equations a suitable approximation.

We also need an equation of state. For an ideal gas:

$P = (\gamma - 1)\, \rho u$ with $\gamma = c_p/c_v = 5/3$ for monoatomic gas.

Smoothed Particle Hydrodynamics

Approximate continuum dynamics of fluids using particles.

Has excellent conservation properties, no advection errors and is fully Galilean invariant.

Particles = interpolation points to obtain values of hydrodyn. variables at the location of the particle. Obtained using convolution integrals to get smooth estimate.

$$\underbrace{F_S(\vec{r})}_{\text{interpolated } F} = \int \underbrace{F(\vec{r}')}_{\text{fluid field}} \underbrace{W(\vec{r}-\vec{r}', h)}_{\text{kernel}}\, dr'$$

h: smoothing length.

Typical choice to define h: fix the number of neighbours to be included for calculation; in general, may be variable in space. This makes the estimate of any field automatically adaptive.

Basic approaches for adaptive kernels:

- scatter: particle receives contribution from other particles, whose smoothing volume scatter onto its location.
- gather: particle gathers contribution from all other particles that are within its smoothing volume $\propto h^3(\vec{r}_i)$. Computationally faster.

Both approaches are equivalent for a globally fixed h.

To get the SPH equations of motion, one can either recast the inviscid fluid equations (in Lagrangian form) in SPH form by replacing all individual fields and operators with their SPH version, or by the variation principle using the Lagrangian $L = \int \rho\left(\frac{1}{2}\vec{v}^2 - u\right) dV = \sum_i m_i \left(\frac{1}{2}\vec{v}^2 - u\right)$.

The Lagrangian assumes momentum, angular momentum and energy conservation.

Thermodynamics in SPH can both be expressed via entropy or internal energy as the fundamental variable.

The differential form of the equations used so far breaks down at sharp discontinuities such as shocks. Solution ansatzes:

- switch to integral form (conservative form)
- introduce a dissipation term (artificial viscosity)

All functions in SPH are affected by Poisson noise, namely the fact that the fluid is coarsely sampled by particles. ⇒ There cannot be really sharp discontinuities by construction. The shock profile is never sharp, ringing arises post-shock.

In shocks, kinetic energy is dissipated into heat, otherwise the shock will never end. The specific entropy of the gas always increases at a shock front ⇒ in the shock layer itself the gas dynamics can no longer be described as inviscid ⇒ The derived SPH equations can't properly describe a shock, they keep entropy strictly constant.

⇒ Introduce artificial viscosity; parametrised in terms of a friction force that damps relative motion of particles.

Problem: This is strictly for numerical reasons. Viscosity needs to be formulated in a way that it turns off away from discontinuities.

Solution: Make viscous tensor Π_{ij} symmetric and $\Pi_{ij} \neq 0$ for $|\vec{r}_i - \vec{v}_j| \cdot |\vec{r}_i - \vec{r}_j| < 0$, such that viscosity only turns on for particles that approach each other.

Eulerian Hydrodynamics Methods

Solve Eulerian form of fluid equations on meshes. There is no general solution method that works equally well for all types of problems.

- Finite difference methods:
  Differential operators → finite difference approximations
- Finite volume methods
  Construct fluxes for all quantities at cell faces, use integral form
- Finite element methods
  Solve PDEs on subvolumes of arbitrary shape. Solution is represented in terms of simple functions on the element, the PDE is transformed to algebraic problem for function coefficients.
- Spectral methods
  solution = linear combination of functions; transform PDE to algebraic equations. (→ Fourier Techniques)
- Method of lines
  Express all derivatives but one via finite differences. Remaining derivative then forms a set of ODEs. Often used for time-independent problems.

Advection Problem

Integrating fluid equations essentially means to advect the functions defining the fluid $(\rho, \vec{v}, P)$ through the grid.

1D advection equ: $\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0$

Analytical solution: $u(x,t) = q(x - vt)$

Simple update scheme: $u_i^{(n+1)} = u_i^{(n)} - v \frac{u_{i+1}^{(n)} - u_{i-1}^{(n)}}{2h} \Delta t$ is unstable: update formula includes terms from both upstream (u_{i-1}) and downstream (u_{i+1}). This is inconsistent with the way information flows.

Instead: $\frac{du_i}{dt} + v \frac{u_i - u_{i-1}}{h} = 0$ (upwind differencing)

The solution is not advected in a perfectly faithful way, but is significantly smoothed out through numerical diffusion: First order upwind scheme effectively solves the diffusion equation.

Courant-Friedrichs-Lewy timestep condition: $\Delta t \leq h/v$
Timestep cannot be larger than h/v to be physically self-consistent.

Riemann Problem and Solvers

= classic boundary value problem for a hyperbolic system, consisting of two piece-wise constant states ⊓⊔ that met at a plane at $t=0$.

For Euler equation: left and right state are $U_R = (\rho_R, v_R, P_R)$, $U_L = (\rho_L, v_L, P_R)$

→ Find states at $t > 0$.

Solution always contains characteristics for 3 self-similar waves: contact wave (= boundary) in middle, sandwiched by a shock and/or rarefaction wave (= not single characteristic, but rather a fan). These waves propagate with constant speed.

Finite Volume Solvers (Godunov)

Consider integral form of conservation law: $\frac{\partial \vec{U}}{\partial t} + \nabla \cdot \vec{F} = 0$

For Euler equation: $\vec{U} = (\rho, \rho\vec{v}, \rho e)$; $\vec{F} = (\rho\vec{v}, \rho\vec{v}\vec{v}^T + P, (\rho e + P)\vec{v})$

$e = u + \vec{v}^2/2$ = internal energy, u = thermal energy, $P = (\gamma - 1)\rho u$

The system to be solved is not continuous, but described as a finite set of cells, containing an average.

By integrating the conservation law over a cell and over a time interval:

$$\vec{U}_i^{(n+1)} = \vec{U}_i^{(n)} + \frac{\Delta t}{\Delta x}\left[F^*_{i-1/2} - F^*_{i+1/2} \right] \quad \text{with } \vec{F}^*_{i+1/2} = \vec{F}(\vec{x}_{i+1/2}, t)$$

Note that so far no approximations have been used! $\vec{F}^*_{i+1/2}$ given by solution of Riemann

REA - Scheme

Scheme based on Godunov.

1) Reconstruct: Construct fluxes at all interfaces
2) Evolve: Evolve reconstructed state forward in time
3) Average: Compute smooth average resulting from evolution: New U^{n+1}

Core of this method: Treat cell interfaces as Riemann problem.

Higher order accuracy: α-order accurate method converges to analytical result with $N^{-\alpha}$. Replace piece-wise constant approach by piecewise linear, then add slope limiters to counter overshoots and introduction of new extrema. Furthermore, evaluate fluxes at half timestep to become also second order accurate in time.

For even higher order methods, use reconstruction of cell interfaces with even higher order polynomes, and a larger stencil.

Multidimensional Godunov Schemes:

1) Unsplit schemes:

$$\partial_t \vec{U} + \partial_x \vec{F} + \partial_y \vec{G} + \partial_z \vec{H} = 0 \quad , \quad \vec{F}, \vec{G}, \vec{H} \text{ fluxes in } x, y, z \text{ direction.}$$

2) Split schemes:

Split problem in each direction:

$$\partial_t \vec{U} + \partial_x \vec{F} = 0; \quad \partial_t \vec{U} + \partial_y \vec{G} = 0; \quad \partial_t \vec{U} + \partial_z \vec{H} = 0$$

= augmented 1D problems.

Misc Notes for Exam

- Particle Mesh techniques introduces a second discretisation:
  First: using superparticles (discretises physics)
  Second: Mesh: discretises physical model

- Problems with static mesh methods:
  - Rigid structures have difficulty to handle strongly dynamical systems
  - You might lose resolution if all particles end up in one cell

- Smoothing out density field (CIC, TSC) loses some resolution, but adds stability and is numerically favourable

- Central difference formula introduces smaller error ($\mathcal{O}(h^2)$) than forward / backward differencing
  ↳ Lose accuracy by downwind methods
  To increase accuracy: Use larger stencil

- Evolution equation for f_s involves f_{s+1} term. Need closure.
- Weakly correlated systems: Fluid approximation
- $\langle v^2 \rangle \gg \frac{GM}{\varepsilon}$ to avoid bound pairs (would violate collisionless behaviour)