Computational Astrophysics

7. Particle methods

Troels Haugbølle

Niels Bohr Institute
University of Copenhagen

Overview for this week

Tuesday: Particle methods

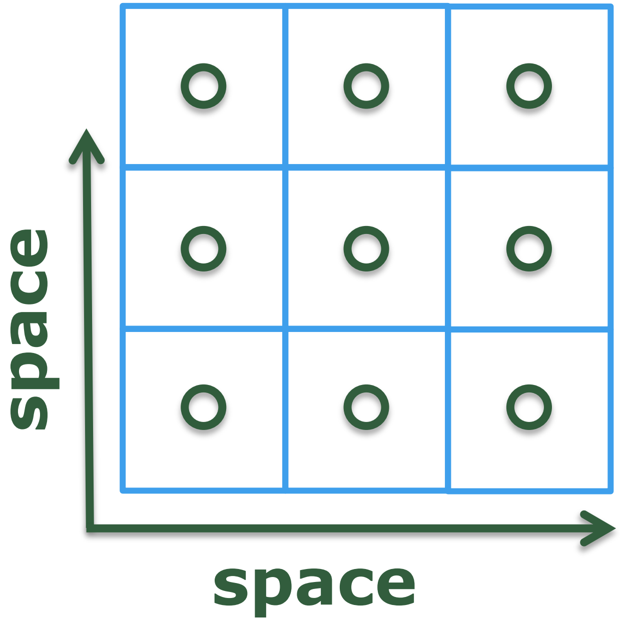
- ❑ Different particle systems
- ❑ Point particles and few-body systems
- ❑ Time integration and symplectic integrators

Thursday: Introduction to exam projects

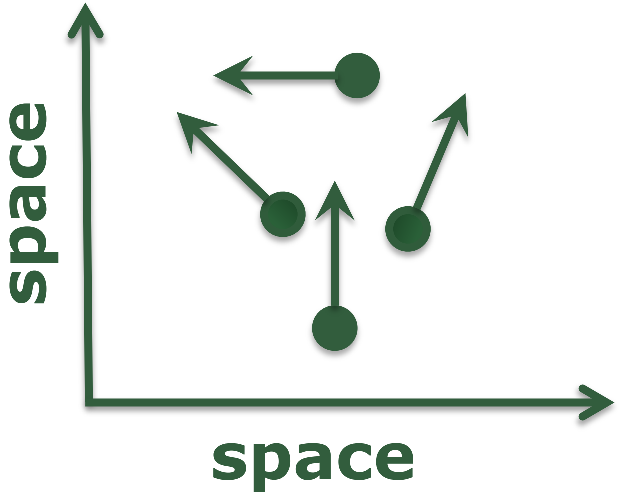
Physical systems described by particle methods

- **Few-body systems:** We have a system of a few particles – e.g. planets – and would like to integrate it over many, many dynamical times
  - Crucial to have a stable and precise integration technique
  - Deal with close encounters
  - Hard to parallelize because of few bodies
- **Test particle systems**
  - Particles represent a sub-dominant species, but we would like to know what is happening
  - Calculate external force, integrate movement
  - Used for diagnostics e.g. to study sub-dominant particle acceleration
- **Many-body systems:** Dark matter, collisionless plasmas, dust, …
  - Dynamics is determined through the back-reaction of the constituents
  - Often collisionless and cannot easily be described through thermodynamic quantities like *temperature* or *moments*

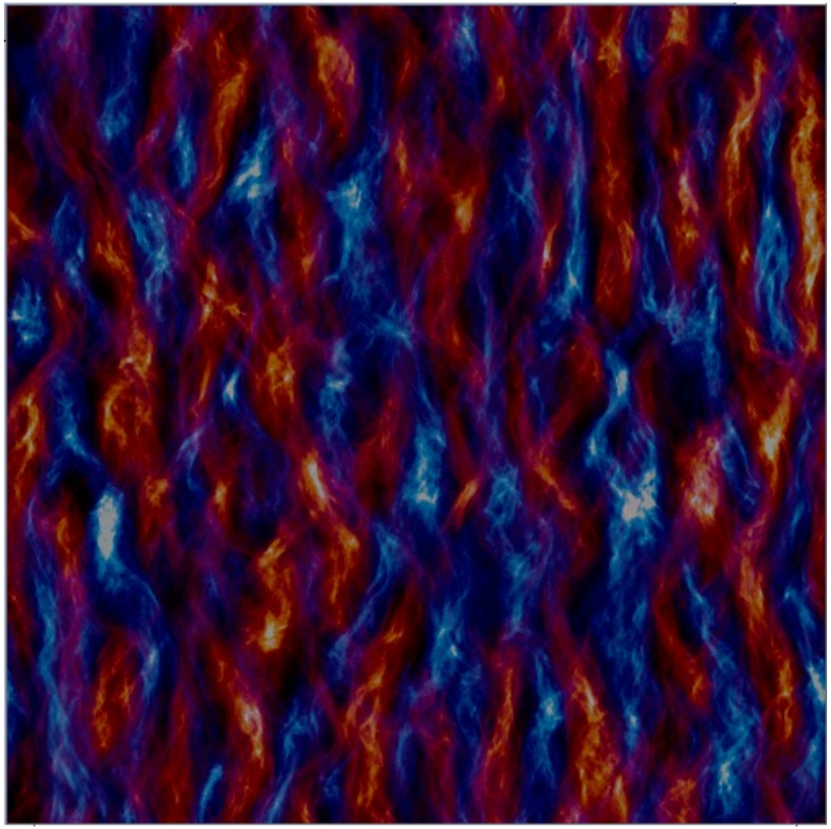
Why use particle methods and not grids?

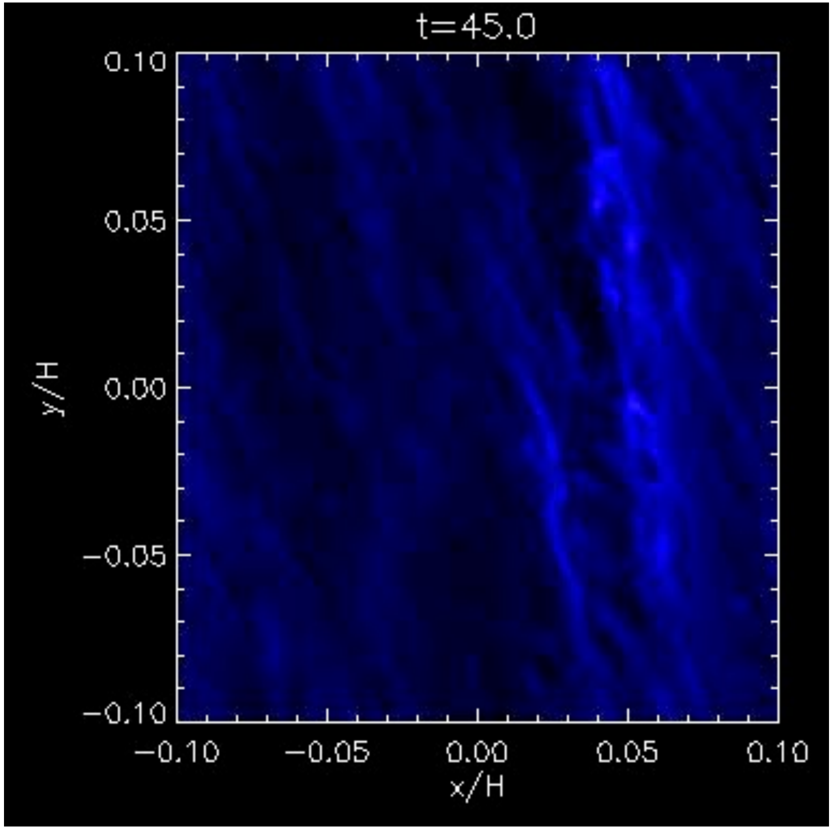


vs



Many-body systems





[Plasma streaming instability – TH] [Disk streaming instability – Johansen]

How to integrate particle orbits?

- ❑ **Few-body and test particle systems:**
  - ❑ Particles can be viewed as zero-dimensional point objects
  - ❑ Calculate force, integrate movement
  - ❑ Then this is simply Newton's second law

$$\frac{\partial \mathbf{x}_i}{\partial t} = \mathbf{v}_i, \qquad \frac{\partial m_i \mathbf{v}_i}{\partial t} = \mathbf{F}(\mathbf{x}_i, \mathbf{v}_i)$$

  - ❑ But! We need to take care that orbits are stable, and energy conserved for long-term evolution
  - ❑ Many programs exist, see e.g. "nbody" codes by Sverre Aarseth
- ❑ **Many-body systems:**
  - ❑ More complicated because we cannot integrate orbits for all particles. May also backreact on rest of system (DM, plasmas, dust).

Time integration methods

❑ Euler integration (explicit):

$$x(t+\Delta t) = x(t) + v(t)\Delta t\,, \qquad v(t+\Delta t) = v(t) + a(t)\Delta t$$

- ❑ Simple to code, right hand side only depend on known quantities
- ❑ But only first order accurate
- ❑ ***Never use***, even if you can afford a lot of integration steps

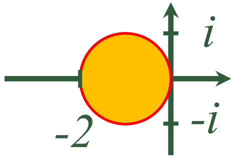
❑ Euler integration (implicit):

$$x(t+\Delta t) = x(t) + v(t+\Delta t)\Delta t\,, \qquad v(t+\Delta t) = v(t) + a(t+\Delta t)\Delta t$$

- ❑ Excellent stability
- ❑ Works even for very stiff systems
- ❑ But implicit in t+Δt

Example: stability for exponential test problem:

$$y'(x) = \lambda\, y(x), y(0) = 1. \quad \text{Assume } Re(\lambda) < 0 \Rightarrow \lim_{x\to\infty} y(x) = 0$$



- ❑ **Explicit Euler method:**
- ❑ Solution with Explicit Euler integration:

$$y_{n+1} = y_n + \Delta x\, \lambda\, y_n. \;\; \text{Solution:}\, y_n = (1 + \Delta x\, \lambda)^n$$

- ❑ Solution bounded if $|1 + \Delta x\, \lambda| < 1$. For real λ: $\Delta x < 2\, /\, (-\lambda)$ (yellow sphere).

- ❑ **Implicit Euler integration:**
- ❑ Solution with Implicit Euler integration:

$$y_{n+1} = y_n + \Delta x\, \lambda\, y_{n+1}. \;\; \text{Solution:}\, y_n = \frac{1}{(1 - \Delta x\, \lambda)^n}$$

- ❑ If $Re(\lambda) < 0$ this is bounded for all values of Δx . Method is said to be absolutely stable. Notice: stability says nothing about the precision of the solution at arbitrarily large time step.

Time integration methods

❑ Midpoint integration:

$$x(t+\Delta t) = x(t) + v(t+\tfrac{\Delta t}{2})\Delta t\,, \qquad v(t+\Delta t) = v(t) + a(t+\tfrac{\Delta t}{2})\Delta t$$

- ❑ Second order accurate
- ❑ Time symmetric (symplectic)
- ❑ *But still implicit*

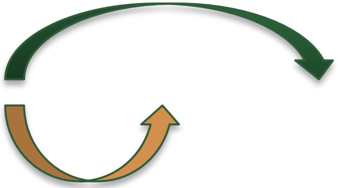
❑ (Explicit) Runge-Kutta integration:

- ❑ Use a predictor-corrector strategy, where we take a guess at future values, and use that *combined* with old values to construct a higher order result.
- ❑ Second order example (*"explicit mid-point rule"*)

$$x^* = x(t) + v(t)\Delta t\,, \quad v^* = v(t) + a(t)\Delta t$$

$$x(t+\Delta t) = x(t) + \tfrac{1}{2}(v(t) + v^*)\Delta t\,,$$

$$v(t+\Delta t) = v(t) + \tfrac{1}{2}(a(t) + a(x^*, v^*))\Delta t$$



Time integration methods

- ❑ **Midpoint integration:**

$$x(t+\Delta t) = x(t) + v(t+\tfrac{\Delta t}{2})\Delta t\,, \qquad v(t+\Delta t) = v(t) + a(t+\tfrac{\Delta t}{2})\Delta t$$

  - ❑ Second order accurate
  - ❑ Time symmetric (symplectic)
  - ❑ But still implicit

- ❑ **(Explicit) Runge-Kutta integration:**
  - ❑ Use a predictor-corrector strategy, where we take a guess at future values, and use that *combined* with old values to construct a higher order result.
  - ❑ R-K integrators come in many variants, and for example the 4th order version is a classic integrator, in particular with adaptive time-stepping
  - ❑ Another useful option for massive systems, like millions of particles, or grid cells are the so-called low-storage R-K schemes (Williamson 1980)
  - ❑ BUT! R-K integrators are not time-symmetric, and therefore do not a priori conserve quantities like e.g. the energy in a system

Time integration methods

❑ Midpoint integration:

$$x(t+\Delta t) = x(t) + v(t+\tfrac{\Delta t}{2})\Delta t\,, \qquad v(t+\Delta t) = v(t) + a(t+\tfrac{\Delta t}{2})\Delta t$$

- ❑ Second order accurate
- ❑ Time symmetric (symplectic)
- ❑ *But still implicit*

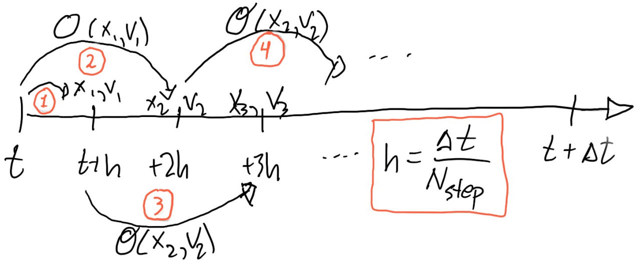
❑ Modified Euler integration

- ❑ Use a predictor-corrector strategy, where we step forward half a time-step to make an approximate second order accurate step (very similar to Runge-Kutta):

$$x^* = x(t) + \tfrac{1}{2}\mathrm{v(t)}\Delta t, \qquad v^* = v(t) + \tfrac{1}{2}a(x(t))\Delta \mathrm{t}$$

$$x(t+\Delta t) = x(t) + v^*\Delta t, \qquad v(t+\Delta t) = v(t) + \ a(x^*)\Delta \mathrm{t}$$

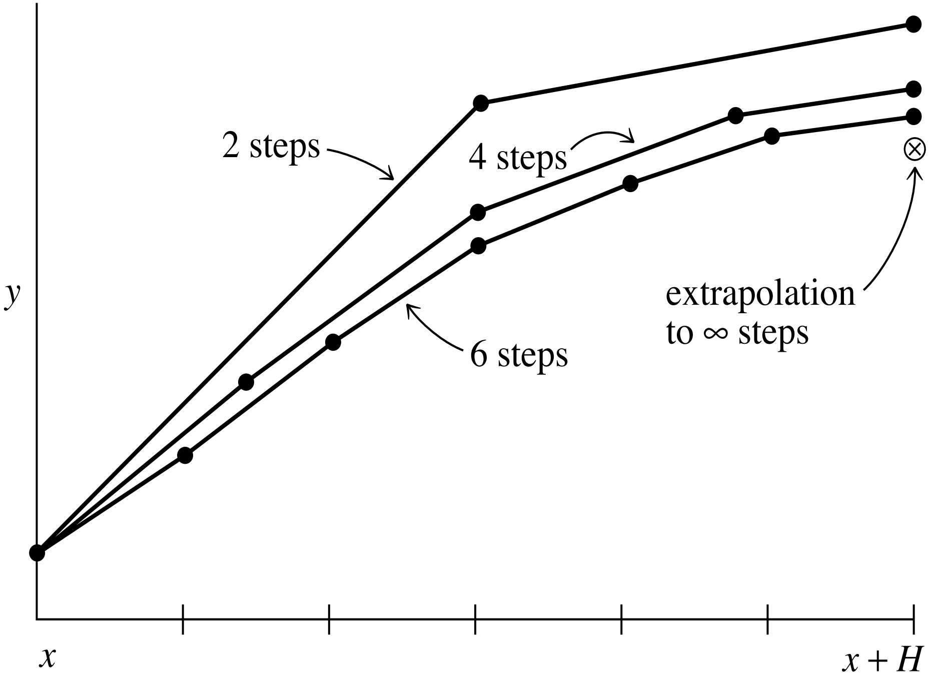
- ❑ This can be generalized to the modified midpoint rule, that does the same over an interval (saving derivative evaluations)



Time integration methods

❑ Bulirsch-Stoer integration:

- ❑ Use modified midpoint integration with different stepsizes: N_{step} = [2,4,6,8,...,16]
- ❑ *Richardson extrapolation:* use polynomial or rational interpolation to fit the function value as a function of step size and predict function value for *h=0*
- ❑ Compare different values of N_{step} to estimate error. If *error > tolerance* then reduce stepsize *H* and retry
- ❑ See chapter 3.4 and Numerical Recipes chapter 16 (http://numerical.recipes)
- ❑ Very useful method for integrating at high precision. In particular, for smooth, non-stiff, problems.



Time integration methods

- □ **Midpoint integration:**

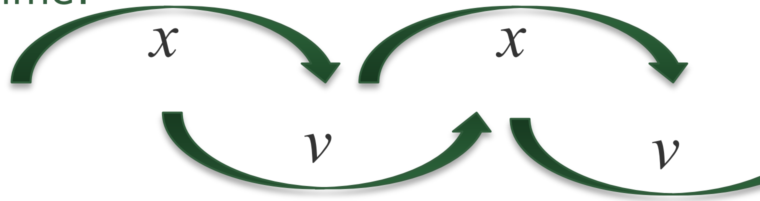
$$x(t+\Delta t) = x(t) + v(t+\tfrac{\Delta t}{2})\Delta t\,, \qquad v(t+\Delta t) = v(t) + a(t+\tfrac{\Delta t}{2})\Delta t$$

  - □ Second order accurate
  - □ Time symmetric (symplectic)
  - □ But still implicit

- □ **Leap frog integration:**
  - □ Observe how update in mid-point depends on going one step forward by knowing values half-way through. Stagger quantities in time:

$$x(t+\Delta t) = x(t) + v(t+\tfrac{\Delta t}{2})\Delta t\,,$$

$$v(t+\tfrac{3}{2}\Delta t) = v(t+\tfrac{\Delta t}{2}) + a(t+\Delta t)\Delta t$$



  - □ Identical to midpoint integrator, but now explicit, if we assume acceleration is due to a conservative force, and only depends on position

Leap Frog Integration

- ❑ We can reformulate the Leap Frog with "Kick" and "Drift" operators

Drift: D(Δt): $x_i \rightarrow x_i + v_i \, \Delta t$

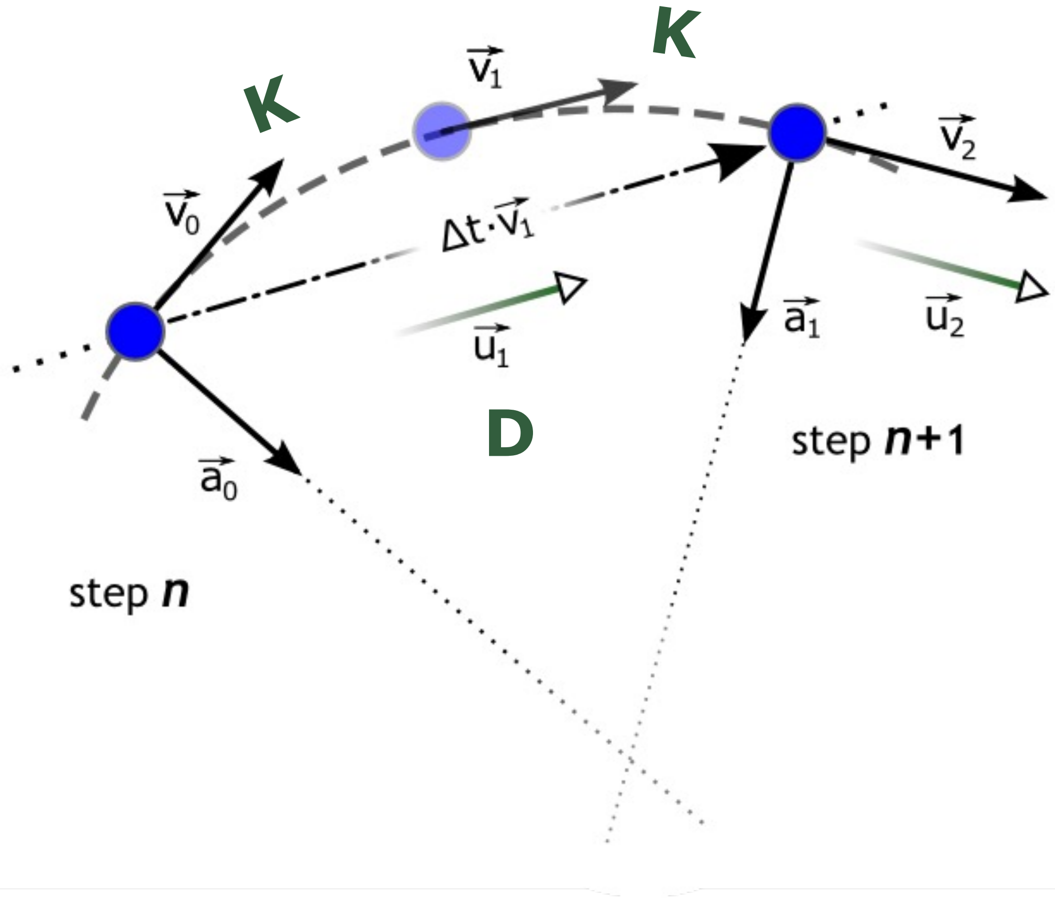
Kick: K(Δt): $v_i \rightarrow v_i + a(x_i) \, \Delta t$

- ❑ Then the Leap Frog integrator over a full time interval can be written with two (identical) versions:

KDK: K(Δt/2) D(Δt) K(Δt/2)

DKD: D(Δt/2) K(Δt) D(Δt/2))

The Kick-Drift-Kick (leap-frog) algorithm:



Leap Frog Integration

- ❑ We can reformulate the Leap Frog with “Kick” and “Drift” operators

Drift: $D(\Delta t)$: $x_i \rightarrow x_i + v_i\,\Delta t$

Kick: $K(\Delta t)$: $v_i \rightarrow v_i + a(x_i)\,\Delta t$

- ❑ Then the Leap Frog integrator over a full time interval can be written with two (identical) versions:

KDK: $K(\Delta t/2)\ D(\Delta t)\ K(\Delta t/2)$

DKD: $D(\Delta t/2)\ K(\Delta t)\ D(\Delta t/2))$

- ❑ How does all these integrators perform for particle orbits?

Time Integration for an elliptic orbit

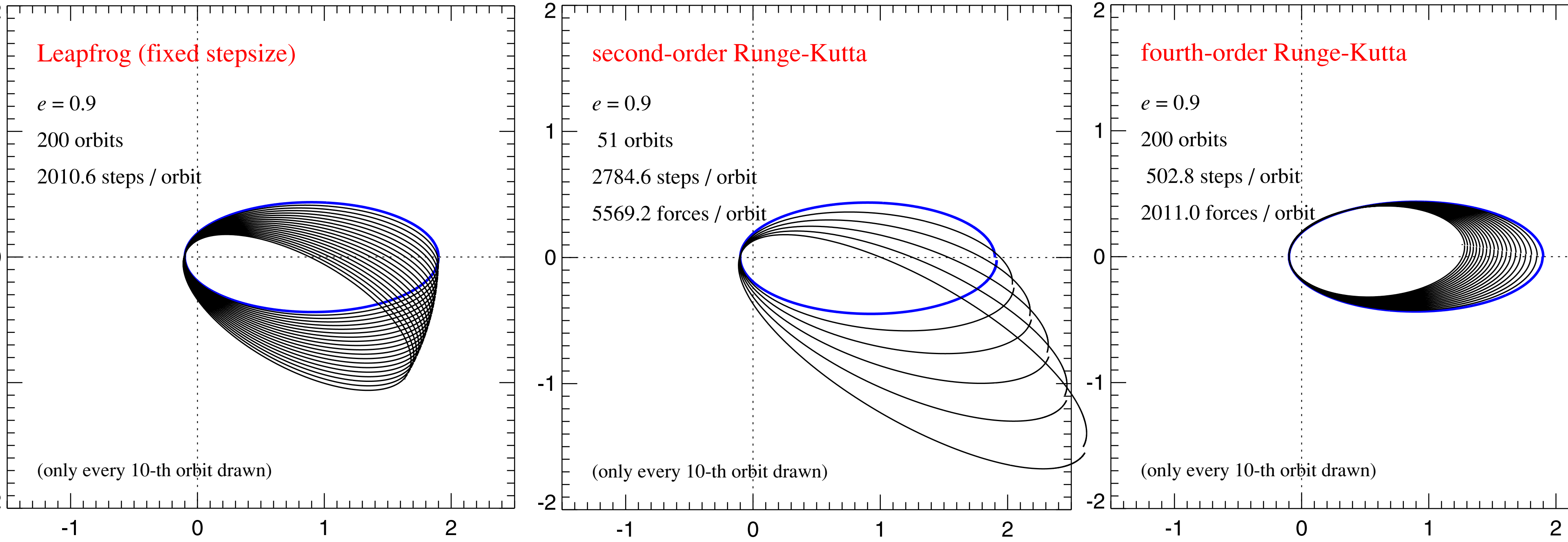
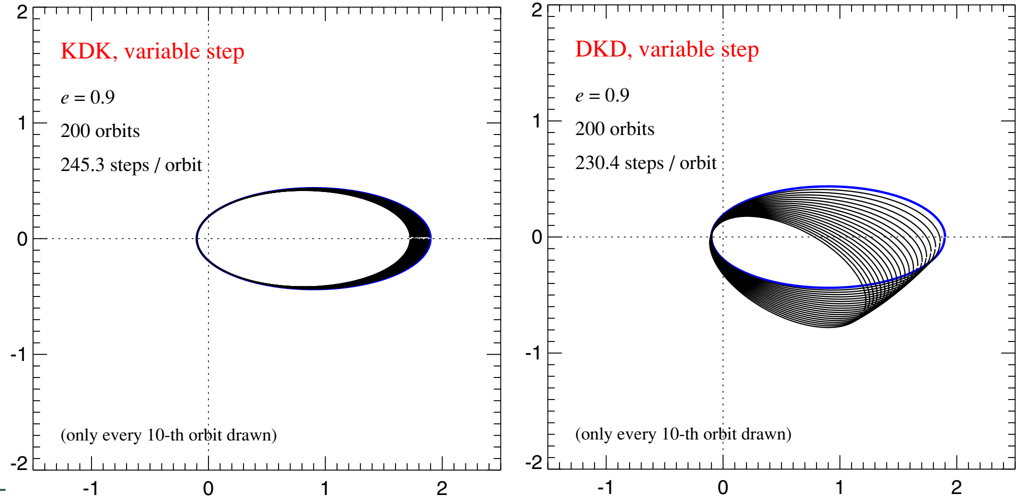


Figure 4. A Kepler problem of high eccentricity evolved with different simple time integration schemes, using an equal time-step in all cases. Even though the leapfrog and the second-order Runge–Kutta produce comparable errors in a single step, the long-term stability of the integration is very different. Even a computationally much more expensive fourth-order Runge–Kutta scheme, with a smaller error per step, performs dramatically worse than the leapfrog in this problem.

[Gadget-2, Springel]

Leap Frog Integration

- ❑ The reason why Leap Frog is so much better is because it respects the underlying dynamics of the system and conserves phase-space density
- ❑ Even though errors are introduced in each time-step, they tend to oscillate around the true value, instead of having secular trends
- ❑ This is only true for fixed time-steps
- ❑ For varying time-steps different parts of the drift and kick operators do not happen with the same length. Then KDK is superior to DKD due to less asymmetry in the force evaluation.



Higher order symplectic integrators exist. In particular Candy & Rozmus (1991) have made a fourth order version consisting of a number of sandwiched Leap-Frog steps. These are typically used when integrating planetary systems

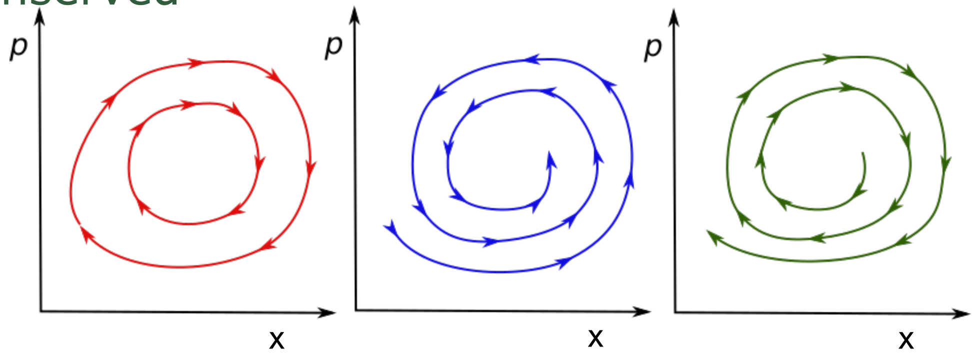
[Gadget-2, Springel]

Energy conservation and Symplectic Integrators

- ❑ Particle systems are described by Hamiltonian dynamics

$$\mathcal{H}(x,p) = \frac{p^2}{2m} + V(x), \qquad \frac{dx}{dt} = \frac{\partial\mathcal{H}}{\partial p} = \frac{p}{m}, \qquad \frac{dp}{dt} = -\frac{\partial\mathcal{H}}{\partial x} = -\frac{\partial V}{\partial x}$$

- ❑ Two (related) properties of such systems are that the energy ($\mathcal{H}$) is conserved and that phase space density along a characteristic (a particle orbit) is conserved



Energy conserved (left)
Energy dissipated (middle)
Energy gained (right)

- ❑ The problem with Euler integration (and the Runge-Kutta methods) is that they do not conserve phase space density and generally have systematic errors that leads to either gain or dissipation of energy.

Energy conservation and Symplectic Integrators

- Consider a simple Euler timestep of size τ for this system of equations:

$$x' = x + \tau\frac{p}{m}, \qquad p' = p - \tau\frac{\partial V}{\partial x}$$

- The new energy is

$$\mathcal{H}(x',p') = \frac{p'^2}{2m} + V(x') = \frac{p^2 - 2\tau p\frac{\partial V}{\partial x} + \tau^2\left|\frac{\partial V}{\partial x}\right|^2}{2m} + V\left(x + \tau\frac{p}{m}\right)$$

$$= \frac{p^2 - 2\tau p\frac{\partial V}{\partial x} + \tau^2\left|\frac{\partial V}{\partial x}\right|^2}{2m} + V(x) + \tau\frac{p}{m}\frac{\partial V}{\partial x} + \frac{1}{2}\tau^2\left(\frac{p}{m}\right)^2\frac{\partial^2 V}{\partial x^2} + \mathcal{O}(\tau^3)$$

$$= \mathcal{H}(x,p) + \frac{1}{2m}\tau^2\left(\left|\frac{\partial V}{\partial x}\right|^2 + \frac{p^2}{m}\frac{\partial^2 V}{\partial x^2}\right) + \mathcal{O}(\tau^3)$$

which is positive for a gravitational force in τ^2 and therefore in general the orbit (and energy) will grow indefinitely with time!

Energy conservation and Symplectic Integrators

- For the drift and kick operators we find that:

Drift: $x' = x + \tau \frac{p}{m}, \quad p' = p$

Kick: $x' = x, \qquad p' = p - \tau \frac{\partial V}{\partial x}$

- which are both symplectic operators. With a proper sandwiching (for example $D_{1/2}KD_{1/2}$) it gives a solution to the equation of motion that preserves the energy for a new Hamiltonian which is almost the right one
- It can be shown that the new $\mathcal{H}_{num}$ and the exact $\mathcal{H}$ are related as $\mathcal{H}_{num} = \mathcal{H} + \mathcal{H}_{err}$, where (e.g. Gadget-2 method paper)

$$\mathcal{H}_{err}(x', p') = \frac{\tau^2}{12}\left\{\left\{\mathcal{H}_{kin}, \mathcal{H}_{pot}\right\}, \mathcal{H}_{kin} + 2\mathcal{H}_{pot}\right\} + \mathcal{O}(\tau^4)$$

where $\mathcal{H}_{kin}, \mathcal{H}_{pot}$ are the kinetic and potential energy parts of the hamiltonian and { , } is a so-called Poisson bracket. Therefore as long as the timestep is constant energy is approximately conserved.

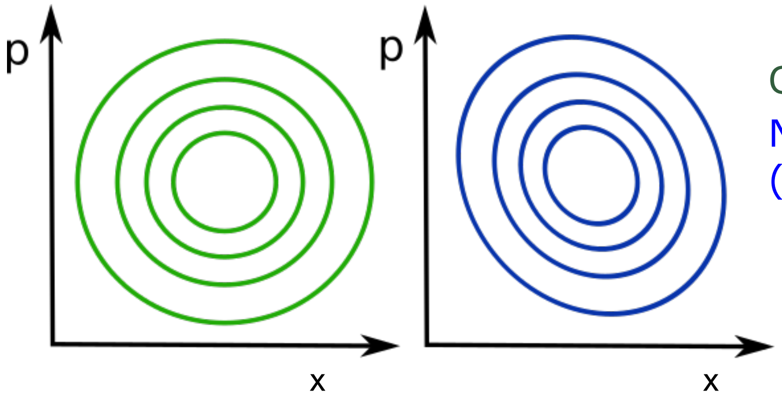
Energy conservation and Symplectic Integrators

- ❑ For the drift and kick operators we find that:

$$\text{Drift: } x' = x + \tau\frac{p}{m}, \quad p' = p$$

$$\text{Kick: } x' = x, \qquad p' = p - \tau\frac{\partial V}{\partial x}$$

- ❑ which are both symplectic operators. With a proper sandwiching (for example $D_{1/2}KD_{1/2}$) it gives a solution to the equation of motion that preserves the energy for a new Hamiltonian which is almost the right one



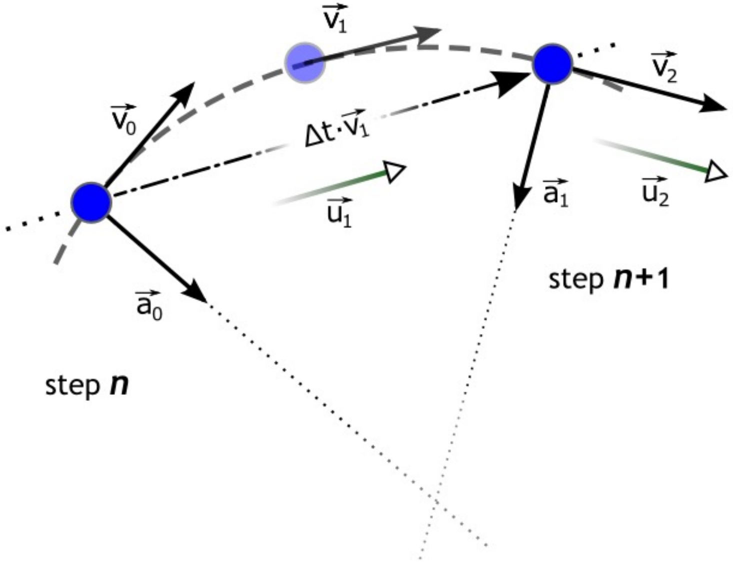
Original Hamiltonian (left)

Numerical Hamiltonian (right)

Long-Term stability with time-variable integrators

- ❑ The reason why Leap Frog is so much better is because it respects the underlying dynamics of the system and conserves phase-space density
- ❑ Even though errors are introduced in each time-step, they tend to oscillate around the true value, instead of having secular trends
- ❑ This is only true for fixed time-steps
- ❑ For varying time-steps different parts of the drift and kick operators do not happen with the same length. Then KDK is superior to DKD due to less asymmetry in the force evaluation
- ❑ "symplectic integrators" respect the underlying Hamiltonian of the problem, but with a slight perturbation that depends on the size of the timestep.
- ❑ Changing time-step destroys the long-term stability, because the errors, and therefore Hamiltonian, change.
- ❑ One way of recovering the stability is by making the method time-reversible, then fluctuations in the error must be limited.

Time steps and Courant condition



- ❑ How do we set the time-step for a particle evolution?
  - ❑ Test with several time-step sizes and test for error?
    - ❑ Classical approach in numerical analysis – basically analyse convergence criteria
    - ❑ BUT expensive if have to test many step sizes for each particle
  - ❑ Look at curvature of orbit:
    - ❑ From acceleration and "radius" (s):

$$\mathrm{dt} = C_{Dt}\sqrt{s\ /\ a}$$

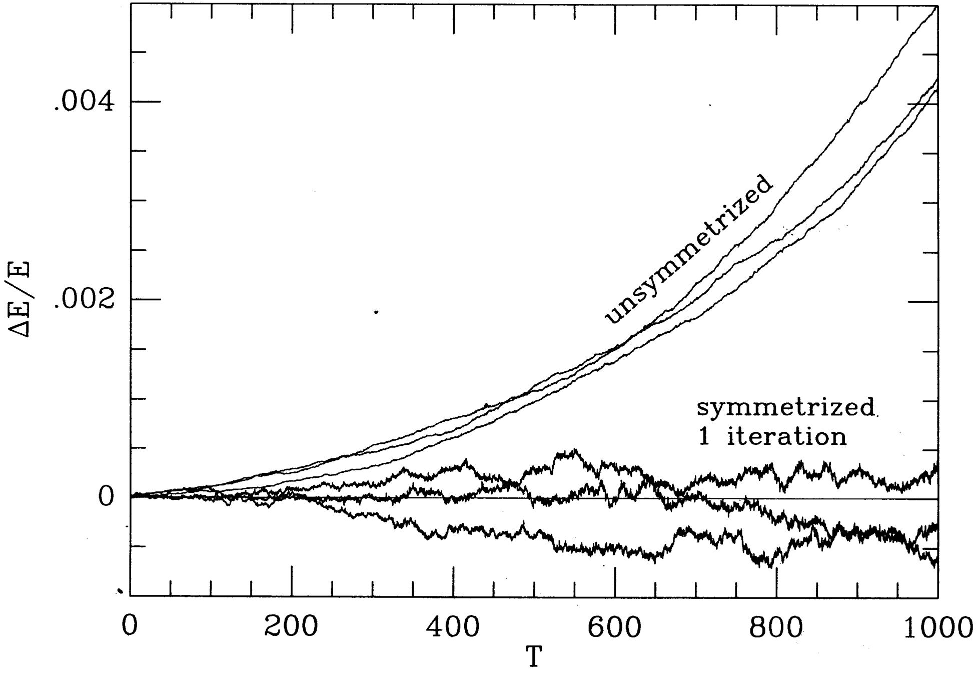
    - ❑ From acceleration and velocity

$$\mathrm{dt} = C_{Dt}\ \mathrm{v}\ /\ \mathrm{a}$$

    - ❑ Many others possible; see Power et al, MNRAS 338 (2003)

Long-Term stability with time-variable integrators

- ❑ One way of recovering the stability is by making the method time-reversible, then fluctuations in the error must be limited.
- ❑ Hut et al 95 showed that this worked. By iteratively using both start and end-point (“*symmetrized dt*”).
- ❑ In the exercise you will implement this as the “reflexive” option



Summary – Particles

- ❑ Particle methods are indispensable for
  - ❑ few-body systems that are well-described as point particles
  - ❑ many-body systems out-of-equilibrium
  - ❑ collisionless systems

- ❑ Particles will often undergo orbits in phase-space, and to conserve phase-space density, specific time integrators for the time evolution are needed.

- ❑ In particular, the Leap Frog (Kick-Drift-Kick) integrator and higher order symplectic integrators have proven successful in removing secular error terms, but this is challenging with adaptive time-steps.

- ❑ Bulirsch-Stoer method is a good alternative for few-body systems

Exercise 7.1: Particle dynamics

The goals of this exercise are to investigate

- ❑ how many steps per orbit are needed for good precision
- ❑ what kinds of errors are avoided with a symplectic method
- ❑ The impact of adaptive step-size

One can extend this by adding drag between the particles and place the particles in a rotating proto-planetary disk. This is a setup that is useful in planetary / disk models and could be explored as an exam project.