Particle Systems & Time Integration
Part 2

Second Order Motion

F = ma

or

$$\ddot{x} = F(x, \dot{x}, t) / m$$

Second Order Motion

- If particle state is just position (and color, size, …) then 1st order motion
  - No inertia
  - Good for very light particles that stay suspended : smoke, dust…

- But most often, want inertia: Newtonian physics
  - State includes velocity, ODE specifies accelerations
  - Can then do parabolic arcs due to gravity, etc.

What's New?

- If $\mathbf{q}=(x,v)$, this is just a special form of 1st order: $d\mathbf{q}/dt=(v,a)=\mathbf{v}(\mathbf{q},t)$

Example

- Orbital motion due to gravitational force

$$F_{gravity} = -GmM_0 \frac{x - x_0}{\left|x - x_0\right|^3}$$

- Let x_0 be a fixed point (e.g. the Sun) with coordinates (0,0)
- For simplicity, approximate as:

$$\ddot{x} = -kx$$

$\ddot{x} = -kx$:1D Version

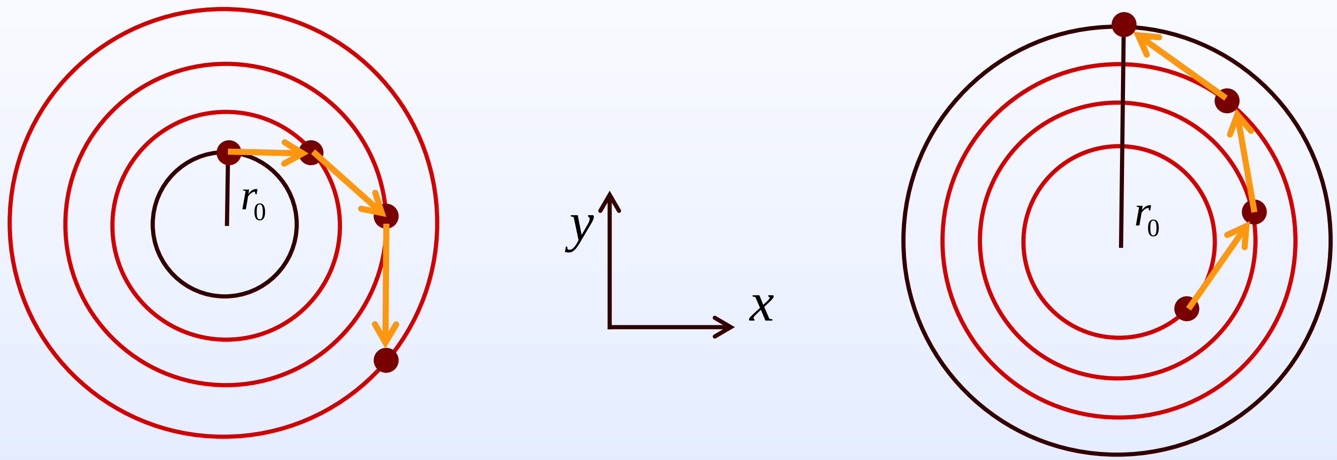
- Simple harmonic oscillator
  - What is the corresponding 1st order ODE?
  - How can you tell it creates oscillatory motion?
  - How do we analyze stability and time stepping restrictions for higher-dimensional ODEs?

$$\dot{q} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} q$$

$\ddot{x} = -kx$:2D Version

- ~ orbital motion
- What do FE and BE look like?

Forward Euler vs Backward Euler



Forward **Euler**

$$q_{n+1} = q_n + h\dot{q}(t_n, q_n)$$

Backward **Euler**

$$q_{n+1} = q_n + h\dot{q}(t_{n+1}, q_{n+1})$$

What's New?

- If **q**=(x,v), this is just a special form of 1st order: d**q**/dt=(v,a)=**v**(**q**,t)
- But since we know the special structure, can we take advantage of it?
  (i.e. better time integration algorithms)
  - More stability for less cost?
  - Handle position and velocity differently to better control error?

Linear Analysis

- Approximate acceleration:

$$a(x,v) \approx a_0 + \underbrace{\frac{\partial a}{\partial x} x}_{-Kx} + \underbrace{\frac{\partial a}{\partial v} v}_{-Dv}$$

- Split up analysis into different cases
  - Which term dominates the problem you are trying to solve?

Three Test Equations

- Constant acceleration (e.g. gravity): $a(x,v,t)=a_0$
  - Want exact (2nd order accurate) position

$$v(t) = v_0 + a_0 t$$
$$x(t) = x_0 + v_0 t + \tfrac{1}{2} a_0 t^2$$

- Position dependence (e.g. spring force): $a(x,v,t)=-Kx$
  - Want stability but low or zero damping
  - Look at imaginary axis
- Velocity dependence (e.g. damping): $a(x,v,t)=-Dv$
  - Want stability, monotone decay
  - Look at negative real axis

Explicit methods from before

- Forward Euler
  - Constant acceleration: bad (1st order)
  - Position dependence: very bad (unconditionally unstable)
  - Velocity dependence: ok (conditionally monotone/stable)
- RK3 and RK4
  - Constant acceleration: great (high order)
  - Position dependence: ok (conditionally stable, but damps out oscillation)
  - Velocity dependence: ok (conditionally monotone/stable)

Implicit methods from before

- Backward Euler
  - Constant acceleration: bad (1st order)
  - Position dependence: ok (stable, but damps)
  - Velocity dependence: great (monotone)
- Trapezoidal Rule
  - Constant acceleration: great (2nd order)
  - Position dependence: great (stable, no damping)
  - Velocity dependence: good (stable but only conditionally monotone)

Specialized 2nd Order Methods

- This is again a big subject
- Again look at explicit methods, implicit methods
- Also can treat position and velocity dependence differently:
  mixed implicit-explicit methods

Symplectic Euler

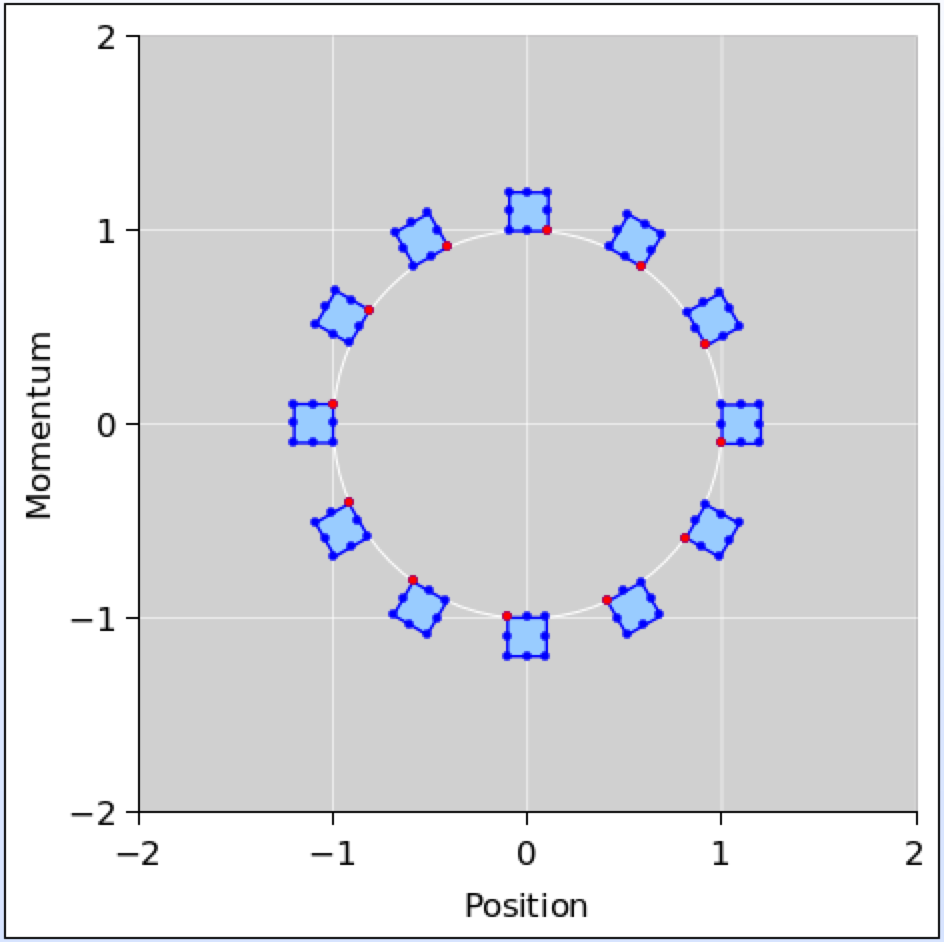
- Like Forward Euler, but updated velocity used for position

$$v_{n+1} = v_n + \Delta t\, a(x_n, v_n)$$
$$x_{n+1} = x_n + \Delta t\, v_{n+1}$$

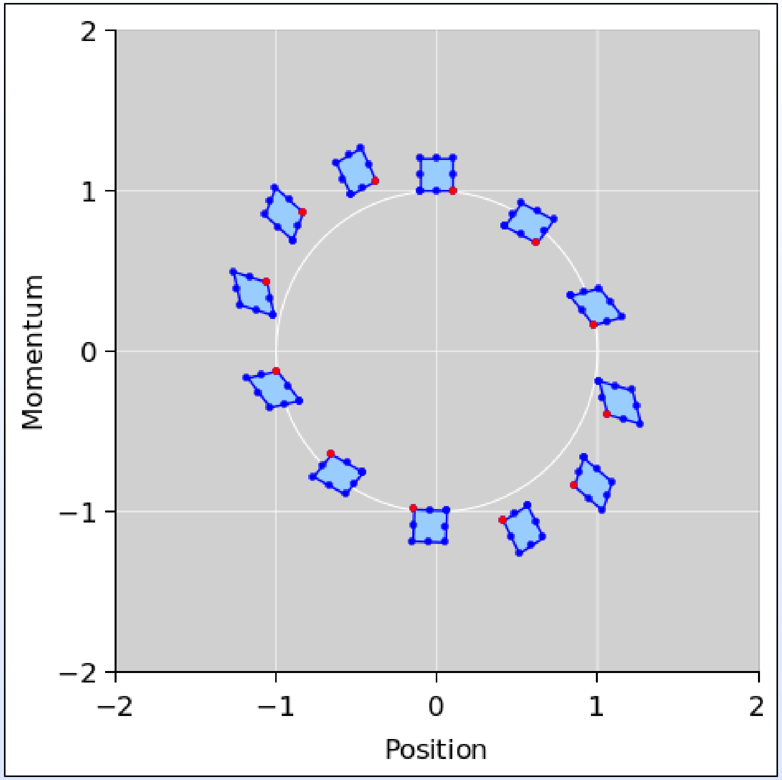
- Symplectic means that it preserves area in phase space (x,v).

Phase-space Plot for harmonic oscillators

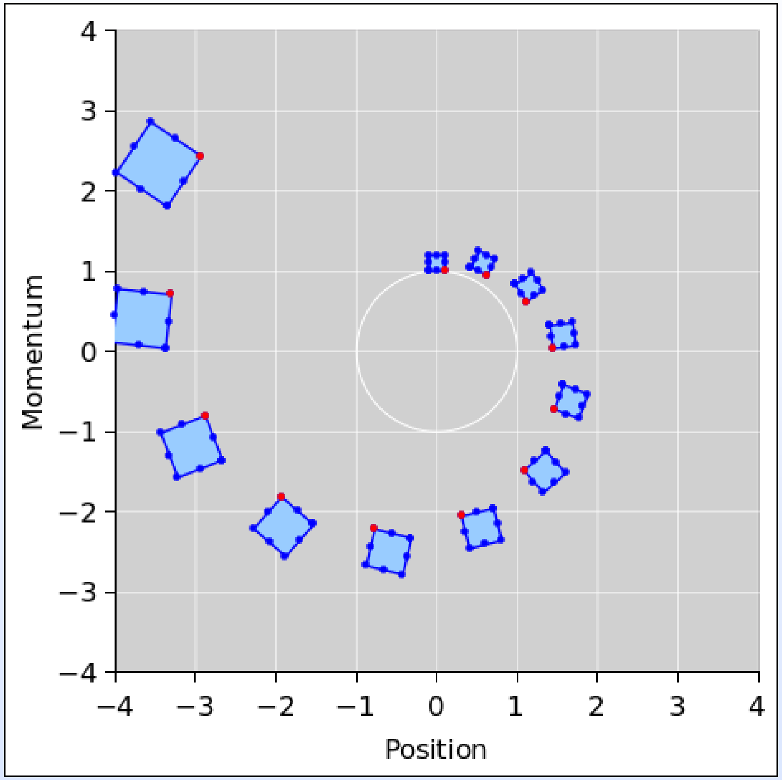
Look at time-evolution of ensemble of close-by oscillator configurations



Phase-space Plot for harmonic oscillators



***Symplectic* Euler**



***Forward* Euler**

Can show that SE preserves area in space phase exactly

SE approximately conserves energy – energy error remains bounded

Symplectic Euler performance

- Constant acceleration: not great
  - Velocity right, position off by O(Δt)
- Position dependence: good
  - Stability limit $\Delta t < \frac{2}{\sqrt{K}}$
  - No damping! (symplectic)
- Velocity dependence: ok
  - Monotone limit $\Delta t < 1/D$
  - Stability limit $\Delta t < 2/D$

Tweaking Symplectic Euler (Leapfrog integration)

- Stagger the velocity to improve x
- Start off with

$$v_{1/2} = v_0 + \tfrac{1}{2}\Delta t\, a\left(x_0, v_0\right)$$

- Then proceed with

$$v_{n+1/2} = v_{n-1/2} + \tfrac{1}{2}(t_{n+1} - t_{n-1})\, a\left(x_n, v_{n-1/2}\right)$$

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

- Second order accurate

Staggered Symplectic Euler

- Constant acceleration: great!
  - Position is exact now
- Other cases not effected
  - Was that magic?
  - Similar argumentation as for the midpoint method
- Only downside to staggering
  - At intermediate times, position and velocity not known together
  - May need to think a bit more about collisions and other interactions with outside algorithms…

Newmark Methods

- A general class of methods

$$x_{n+1} = x_n + \Delta t v_n + \tfrac{1}{2}\Delta t^2 \left[(1-2\beta) a_n + 2\beta a_{n+1} \right]$$
$$v_{n+1} = v_n + \Delta t \left[(1-\gamma) a_n + \gamma a_{n+1} \right]$$

- What happens when β=1/4, γ=1/2?
  - Trapezoidal Rule

Summary (2nd order)

- Depends a lot on the problem
- Explicit methods from last class are probably bad
- Symplectic Euler is a great fully explicit method
  - not any more difficult than Forward Euler!
  - Careful with time step size though
- Backward Euler is nice due to unconditional monotonicity
  - But only 1st order accurate
- Trapezoidal Rule is great for everything except damping with large time steps
  - 2nd order accurate, doesn’t damp pure oscillation/rotation

A lively example

Fireworks

- What kind of state variables and attributes should particles have?
- What type of forces need to be modeled?
- What numerical integration scheme should you use?
- Pseudo-code for particle manager?
  - When do particles get created, when do they die?