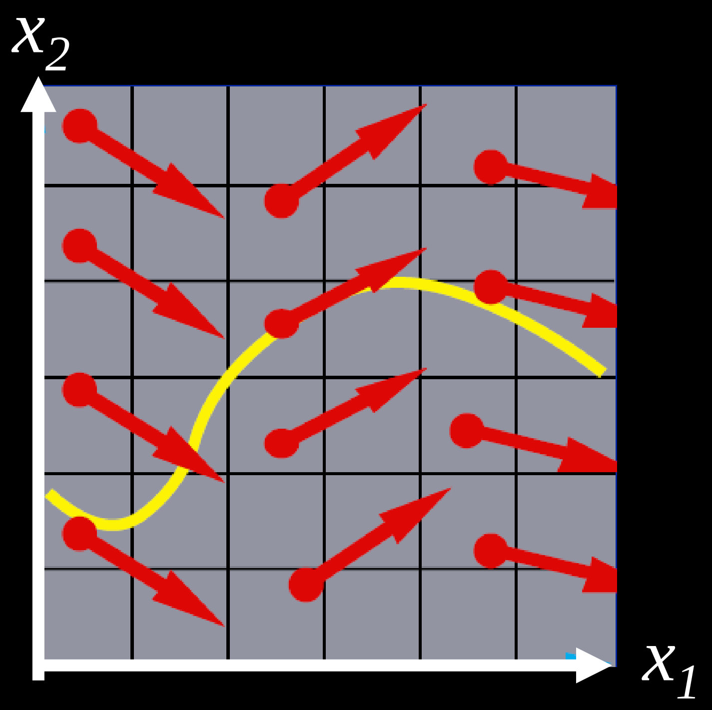
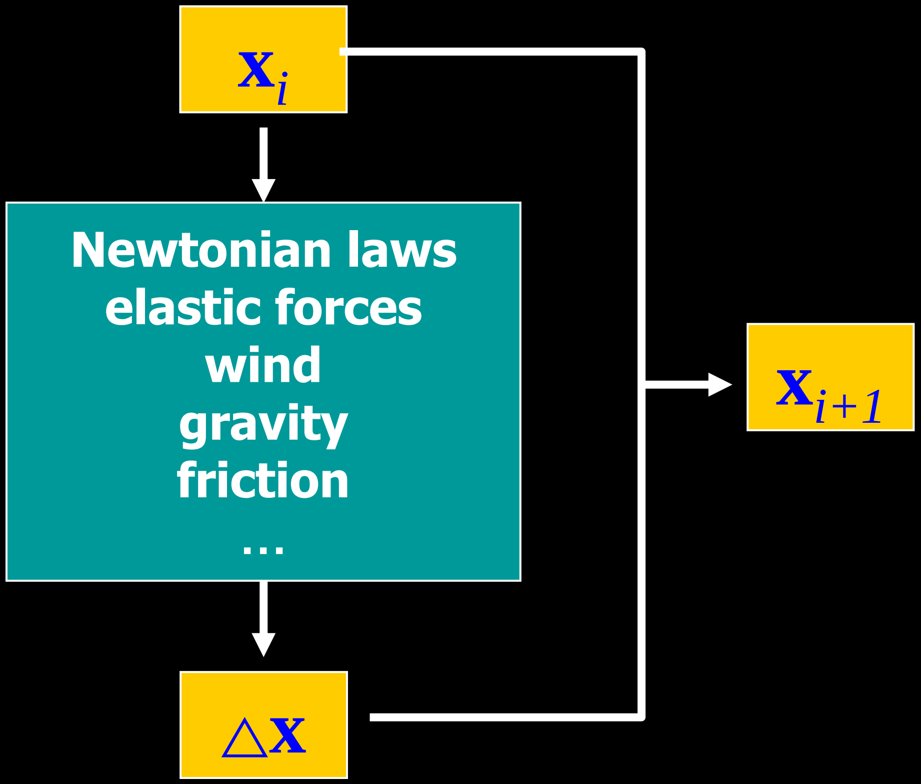
Differential Equation Basics

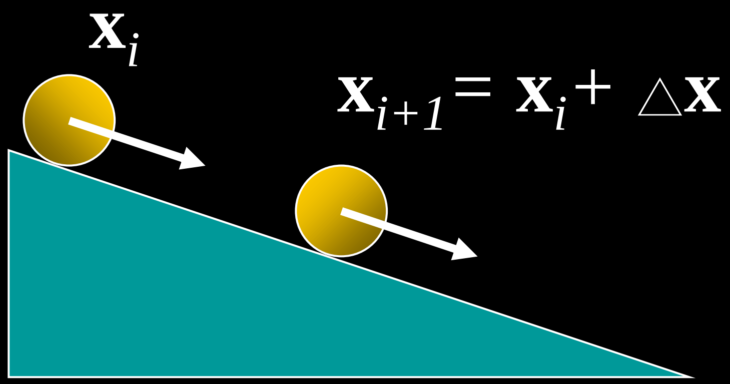


Witkin and Baraff's course notes in SIGGRAPH'01: Physically-based modeling
http://www.cs.cmu.edu/~baraff/sigcourse/

Physics-based Simulation

- A procedure that generates a sequence of the states of a system based on physics laws





Outline

- Differential equation basics (this class)
- Implicit methods (next class)

Differential Equations

- Differential equation describes the relation between an unknown function and its derivatives
- Solving a differential equation is to find a function that satisfies the relation
- Numerical solution of differential equations is based on finite-dimensional approximation

Ordinary Differential Equations

- Ordinary differential equation (ODE)
  - All derivatives are with respect to single independent variable, usually representing time

Known function

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}(t)) = f(\mathbf{x}, t)$$

Time derivative of the unknown function

Unknown function that evaluates the state given time

$\mathbf{x}(t_0)$: state vector at time t_0

*We'll show that a higher ODE can be transformed into this 1st order system soon!

Higher-Order ODEs

- Order of ODE determined by highest-order derivative of solution function appearing in ODE

- Equations with higher derivatives can be transformed into equivalent first-order system

Higher-Order ODEs (cont.)

- Given *k*-th order ODE

$$\frac{d^{(k)}y}{dt} = f(y^{(k-1)}, y^{(k-2)}, \ldots, y', y, t)$$

- Define

$$x_1(t) = y$$

$$x_2(t) = y'$$

$$x_3(t) = y''$$

$$x_k(t) = y^{(k-1)}$$

- Original ODE equivalent to first order system

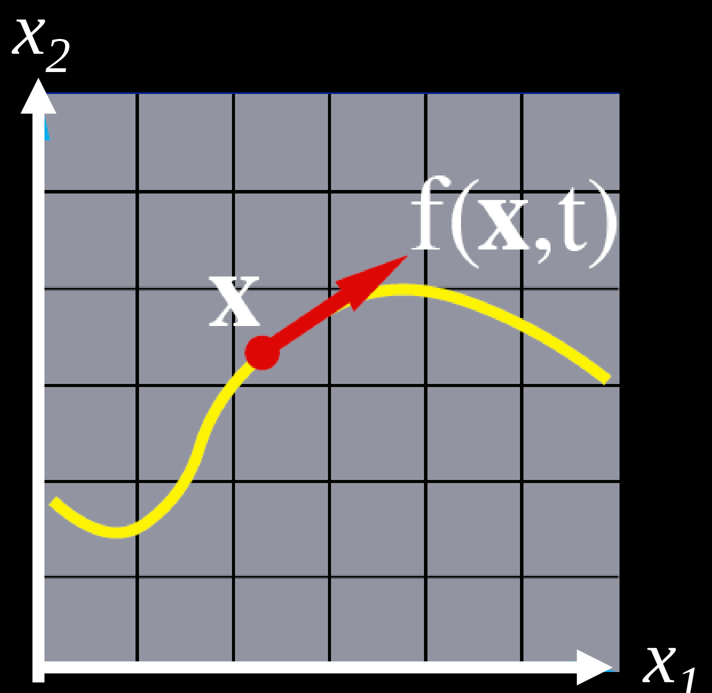
$$\begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_{k-1} \\ x'_k \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_k \\ f(y^{(k-1)}, \cdots, y', y, t) \end{bmatrix}$$

Visualizing Solution of ODE

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = f(\mathbf{x}, t)$$

$\mathbf{x}(t)$: a moving point

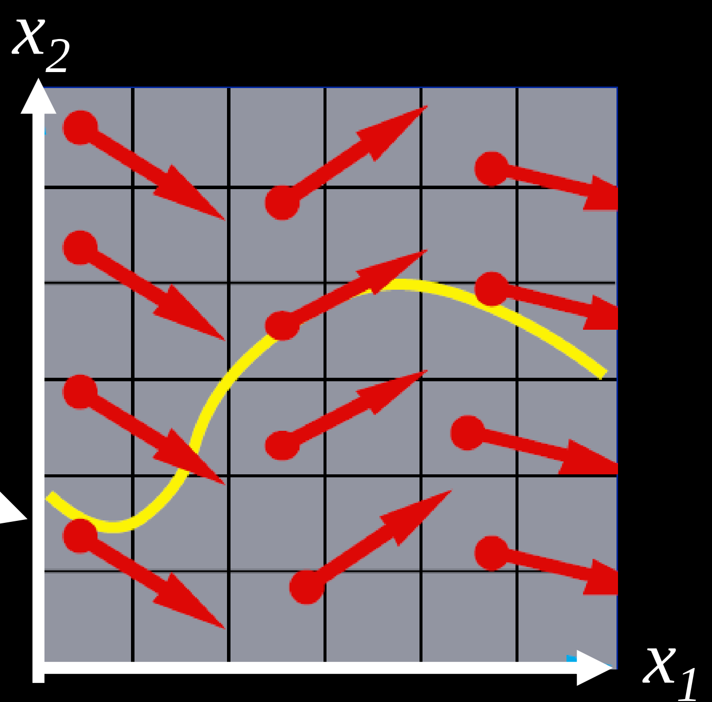
$f(\mathbf{x}, t)$: $\mathbf{x}$'s velocities



Vector Field

- The differential equation $\dot{\mathbf{x}} = f(\mathbf{x}, t)$ defines a vector field over **x**

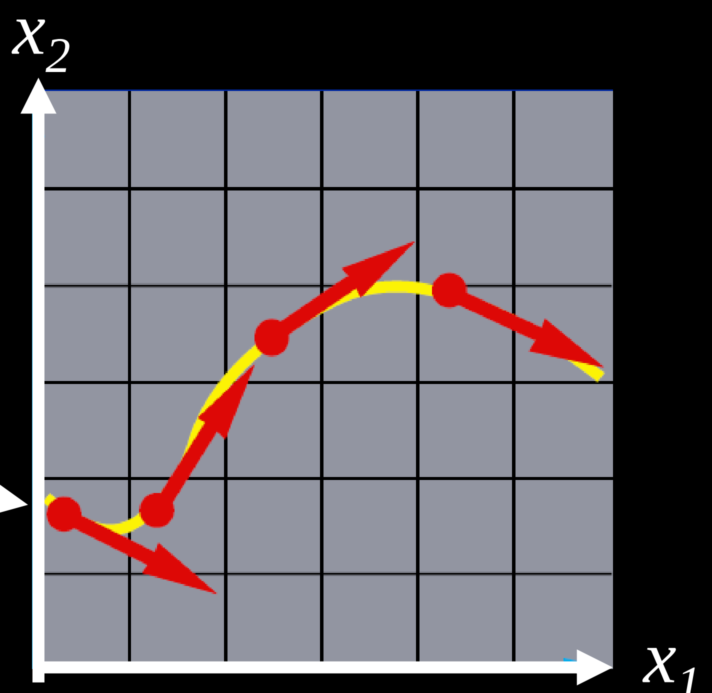
Think of this vector field as the sea, and the velocity of current at different places and time is defined by $f(\mathbf{x},t)$



Integral Curves

- Pick a starting point, and follow the vectors

Release a ball at any starting point and let it drift following the current. The trajectory swept out by the ball is an integral curve

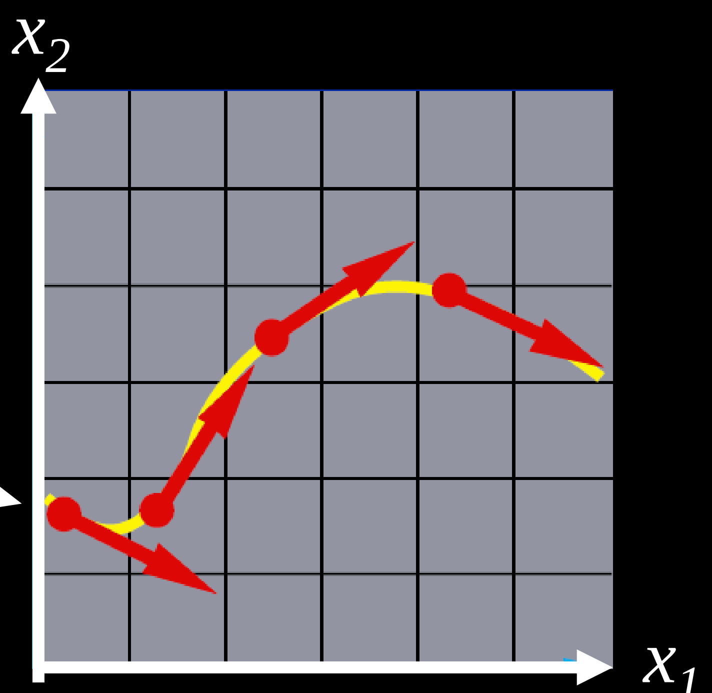


Initial Value Problem

$$\text{Given } \dot{\mathbf{x}} = f(\mathbf{x}, t) \text{ and } \mathbf{x}_0 = \mathbf{x}(t_0) \text{, find } \mathbf{x}(t)$$

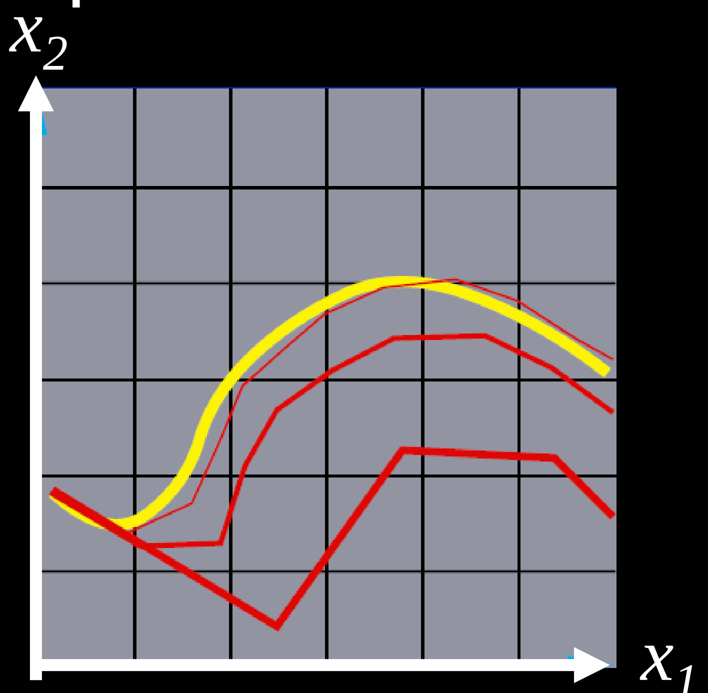
- Given the starting point, follow the integral curve

Where the ball is carried depends on where we initially drop it, but once dropped, all future motion is determined by $f(\mathbf{x},t)$



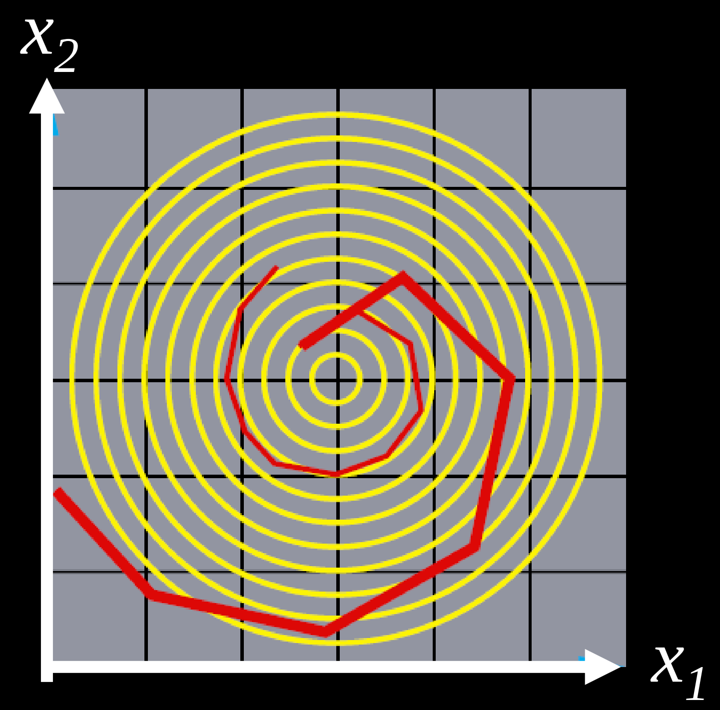
Numerical Solution of ODEs

- Instead of true integral curve, numerical solution follow a polygonal path
- Each leg is obtained by evaluating the derivative at discrete time steps
- Bigger steps, bigger errors

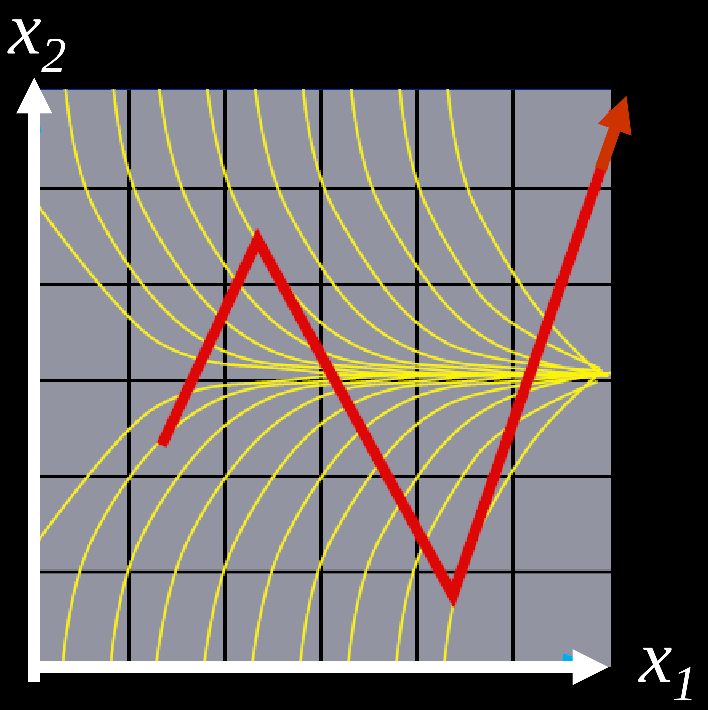


Issue I: Inaccuracy

- Error turns $\mathbf{x}(t)$ from a circle into the spiral of your choice!



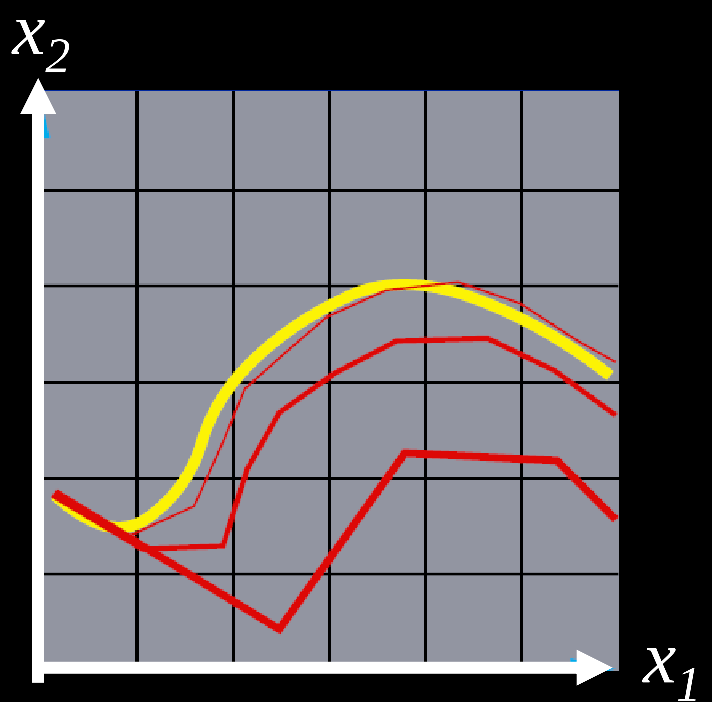
Issue II: Instability



Euler's Method

- Simplest numerical solution method
- Bigger time steps, bigger errors

$$\mathbf{x}(t+h) = \mathbf{x}(t) + h \cdot f(\mathbf{x}, t)$$



Euler's Method (cont.)

Given $\dot{\mathbf{x}} = f(\mathbf{x}, t)$ and $\mathbf{x}_0 = \mathbf{x}(t_0)$, find $\mathbf{x}(t)$

- Solves ODE using one-term Taylor-series

$$\mathbf{x}(t_0 + h) = \mathbf{x}(t_0) + h\dot{\mathbf{x}}(t_0) + \frac{h^2}{2}\ddot{\mathbf{x}}(\xi)$$

O(h^2):
2nd order accurate

$$\dot{\mathbf{x}} = f(\mathbf{x}, t)$$
$$\mathbf{x}_0 = \mathbf{x}(t_0)$$

$$\mathbf{x}(t + h) = \mathbf{x}(t) + h \cdot f(\mathbf{x}, t)$$

$$\longrightarrow \mathbf{x}_{n+1} = \mathbf{x}_n + h\dot{\mathbf{x}}_n$$

Drawbacks of Euler's Method

- Inaccuracy
- Inefficiency
  - Need to use small time-steps to avoid divergence
  - Example: http://heath.cs.illinois.edu/iem/ode/eulrmthd/
- Improvement using the midpoint method
  - Slope at midpoint is used

The Midpoint Method

- Compute an Euler step

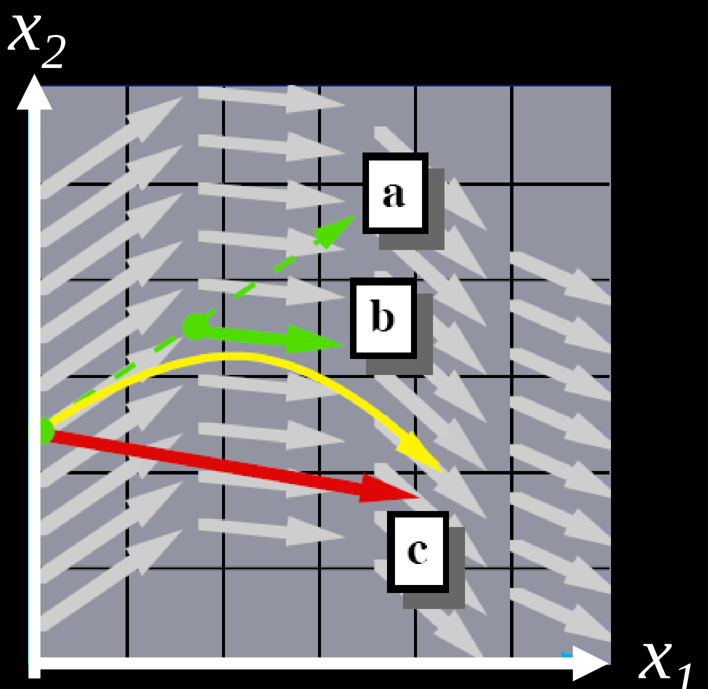
$$\Delta\mathbf{x} = h \cdot f(\mathbf{x}(t_0))$$

- Evaluate f at the midpoint

$$f_{mid} = f\left(\mathbf{x}(t_0) + \frac{\Delta\mathbf{x}}{2}\right)$$

- Take a step using the

$$\mathbf{x}(t_0 + h) = \mathbf{x}(t_0) + h \cdot f_{mid}$$



Accuracy of the Midpoint Method is O(h^3)

- Solves ODE using two-term Taylor-series

$$\mathbf{x}(t_0 + h) = \mathbf{x}(t_0) + h\dot{\mathbf{x}}(t_0) + \frac{h^2}{2}\ddot{\mathbf{x}}(t_0) + \frac{h^3}{3}\dddot{\mathbf{x}}(\xi)$$

$$\ddot{\mathbf{x}} = \frac{d}{dt}\dot{\mathbf{x}} = \frac{d}{dt} f(\mathbf{x}(t)) = \frac{df(\mathbf{x})}{d\mathbf{x}}\frac{d\mathbf{x}}{dt} = f'(\mathbf{x}) f(\mathbf{x})$$

- Approximating *f'* by Taylor-series

$$\Delta\mathbf{x} = \frac{h}{2} f(\mathbf{x}_0)$$

$$f(\mathbf{x}_0 + \Delta\mathbf{x}) \approx f(\mathbf{x}_0) + \Delta\mathbf{x} f'(\mathbf{x}_0)$$

$$f(\mathbf{x}_0 + \frac{h}{2} f(\mathbf{x}_0)) \approx f(\mathbf{x}_0) + \frac{h}{2} f(\mathbf{x}_0) f'(\mathbf{x}_0) = f(\mathbf{x}_0) + \frac{h}{2}\ddot{\mathbf{x}}_0$$

Accuracy of the Midpoint Method (cont.)

$$f(\mathbf{x}_0 + \frac{h}{2} f(\mathbf{x}_0)) \approx f(\mathbf{x}_0) + \frac{h}{2}\ddot{\mathbf{x}}_0$$

$$\frac{h^2}{2}\ddot{\mathbf{x}}_0 \approx h\left[f(\mathbf{x}_0 + \frac{h}{2} f(\mathbf{x}_0)) - f(\mathbf{x}_0) \right]$$

$$\mathbf{x}(t_0 + h) = \mathbf{x}(t_0) + h\dot{\mathbf{x}}(t_0) + \frac{h^2}{2}\ddot{\mathbf{x}}(t_0) + \frac{h^3}{3}\dddot{\mathbf{x}}(\xi)$$

$$= \mathbf{x}(t_0) + hf(\mathbf{x}_0) + h\left[f(\mathbf{x}_0 + \frac{h}{2} f(\mathbf{x}_0)) - f(\mathbf{x}_0) \right] + O(h^3)$$

$$= \mathbf{x}(t_0) + hf(\mathbf{x}_0 + \frac{h}{2} f(\mathbf{x}_0)) + O(h^3)$$

Runge-Kutta 4th Order Method

$$\mathbf{x}(t_0 + h) = \mathbf{x}(t_0) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) + O(h^5)$$

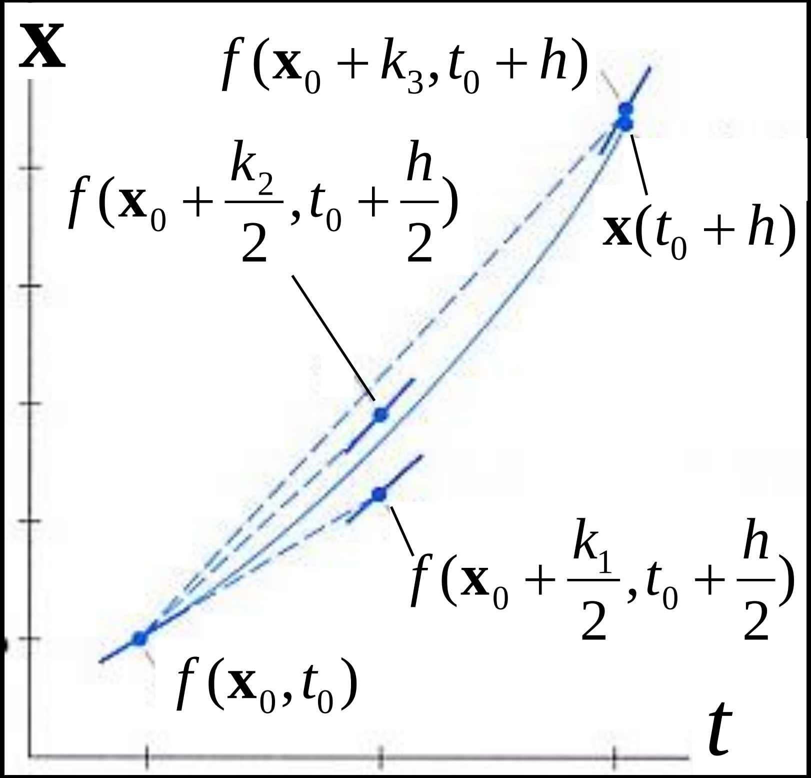
- Using a weighted average of slopes obtained at four points

$$k_1 = hf(\mathbf{x}_0, t_0)$$

$$k_2 = hf(\mathbf{x}_0 + \frac{k_1}{2}, t_0 + \frac{h}{2})$$

$$k_3 = hf(\mathbf{x}_0 + \frac{k_2}{2}, t_0 + \frac{h}{2})$$

$$k_4 = hf(\mathbf{x}_0 + k_3, t_0 + h)$$



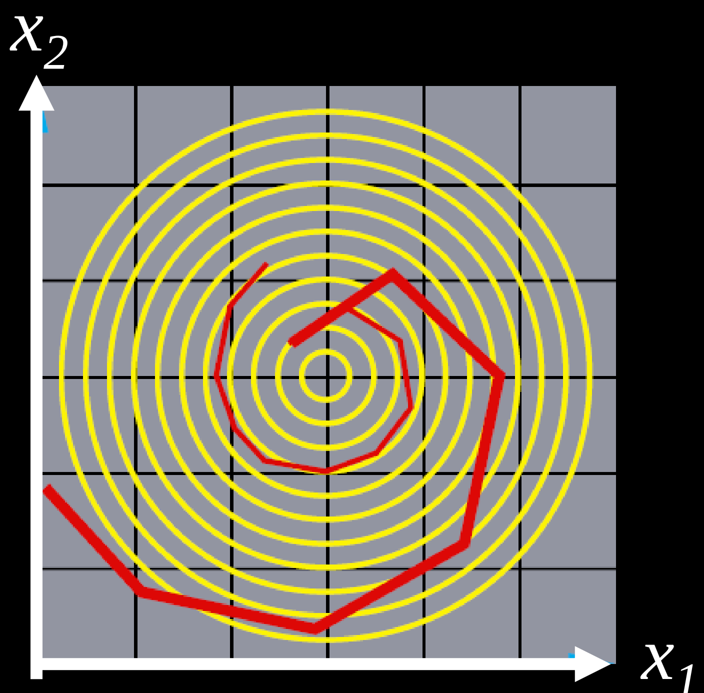
Adaptive Step Size

- Ideally, we want to choose h as large as possible, but not so large as to cause big error or instability

- We can vary *h* as we march forward in time

Take Home Message

- Don't use Euler's method
  - Inaccuracy
  - Inefficiency (or unstable)
- Do use adaptive step size or implicit method



- Read Witkin and Baraff's course notes in SIGGRAPH'01: Physics-based modeling