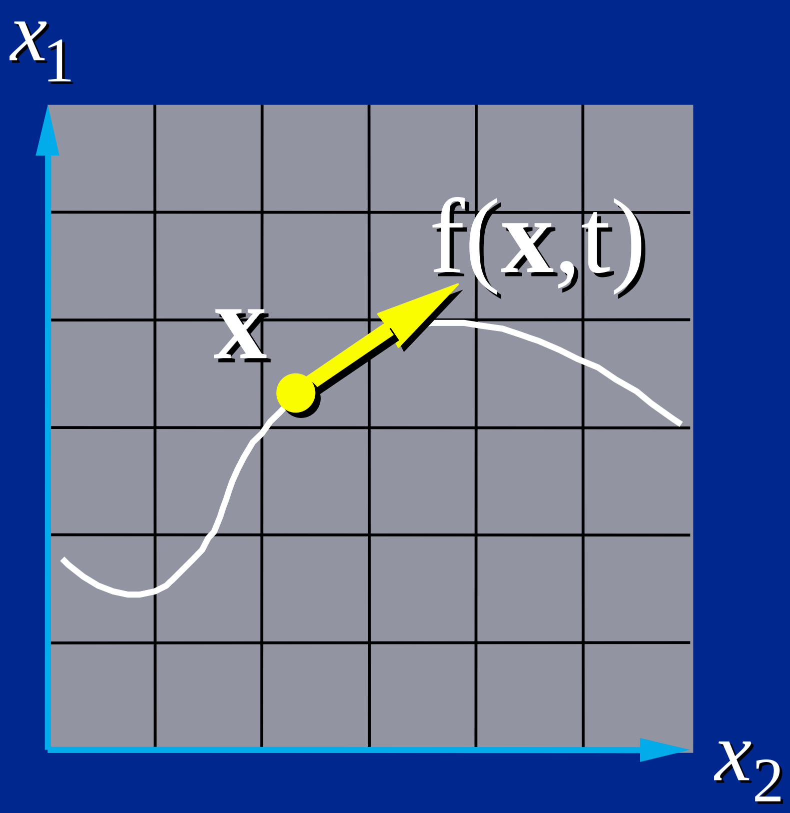
Differential Equation Basics

Andrew Witkin

PIXAR
ANIMATION STUDIOS

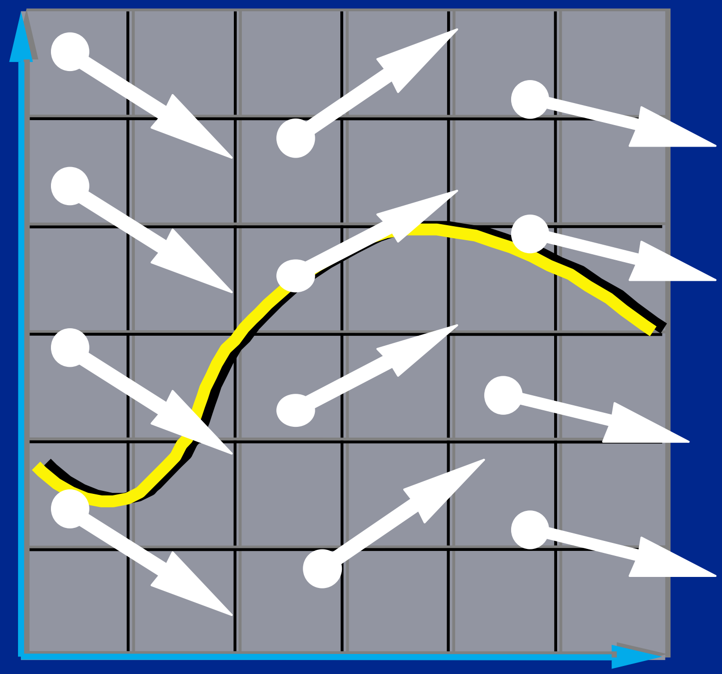
A Canonical Differential Equation



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

- **x**(t): a moving point.
- **f**(**x**,t): **x**'s velocity.

Vector Field

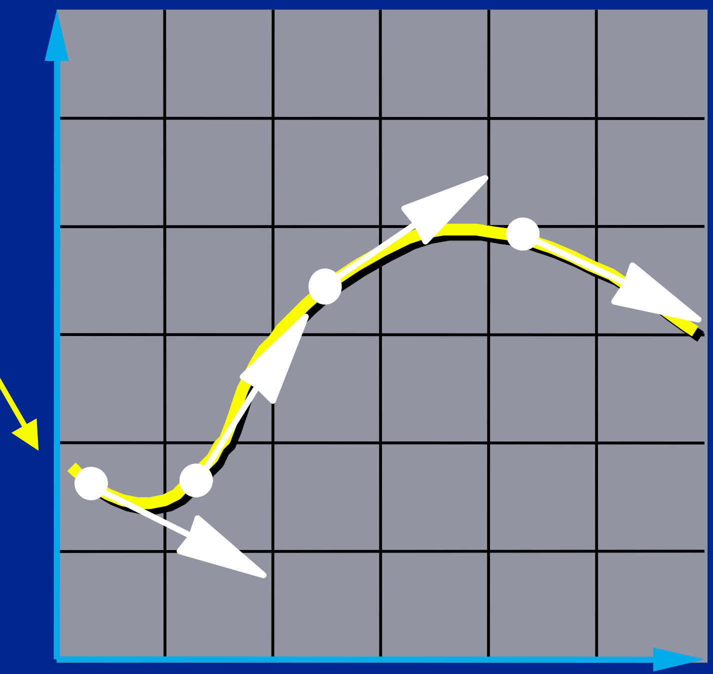


The differential equation

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

defines a vector field over x.

Integral Curves

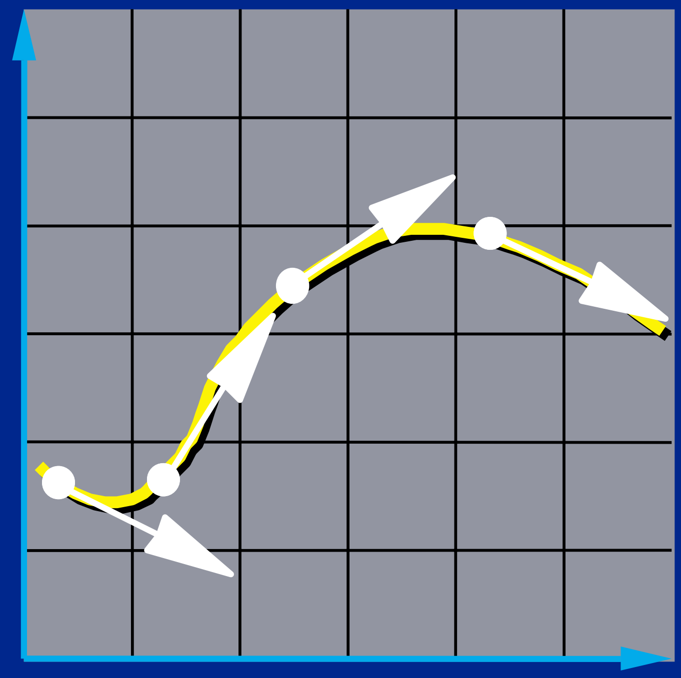


Start Here

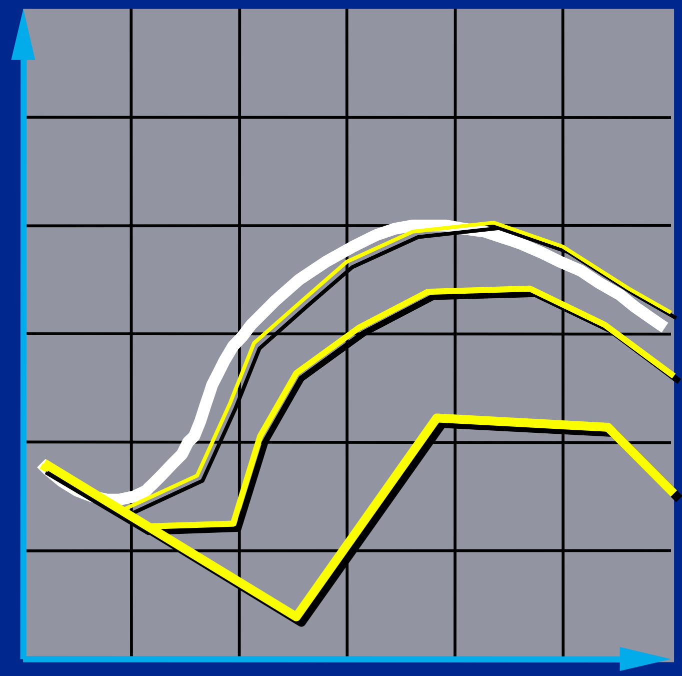
Pick any starting point, and follow the vectors.

Initial Value Problems

Given the starting point, follow the integral curve.



Euler's Method



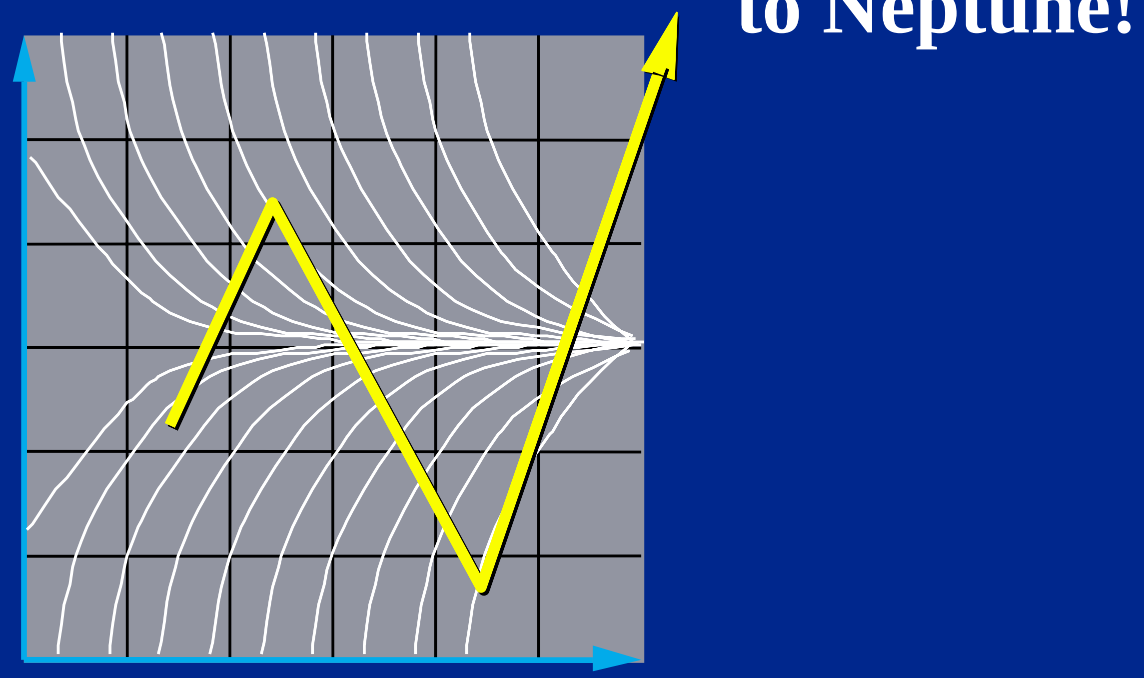
- Simplest numerical solution method
- Discrete time steps
- Bigger steps, bigger errors.

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

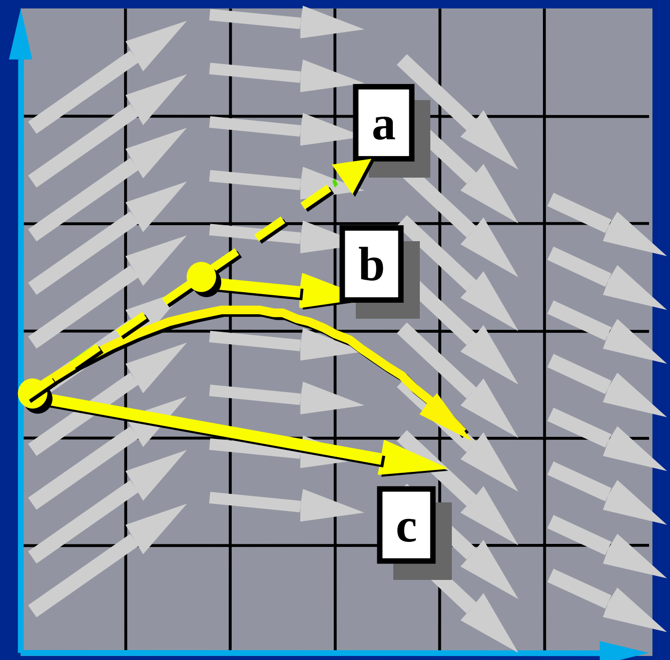
Problem I: Inaccuracy

Error turns x(t) from a circle into the spiral of your choice.

Problem II: Instability



The Midpoint Method



a. Compute an Euler step

$$\Delta \mathbf{x} = \Delta t \, \mathbf{f}(\mathbf{x}, t)$$

b. Evaluate f at the midpoint

$$\mathbf{f}_{\mathrm{mid}} = \mathbf{f}\left(\frac{\mathbf{x} + \Delta \mathbf{x}}{2}, \frac{t + \Delta t}{2}\right)$$

c. Take a step using the midpoint value

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \, \mathbf{f}_{\mathrm{mid}}$$

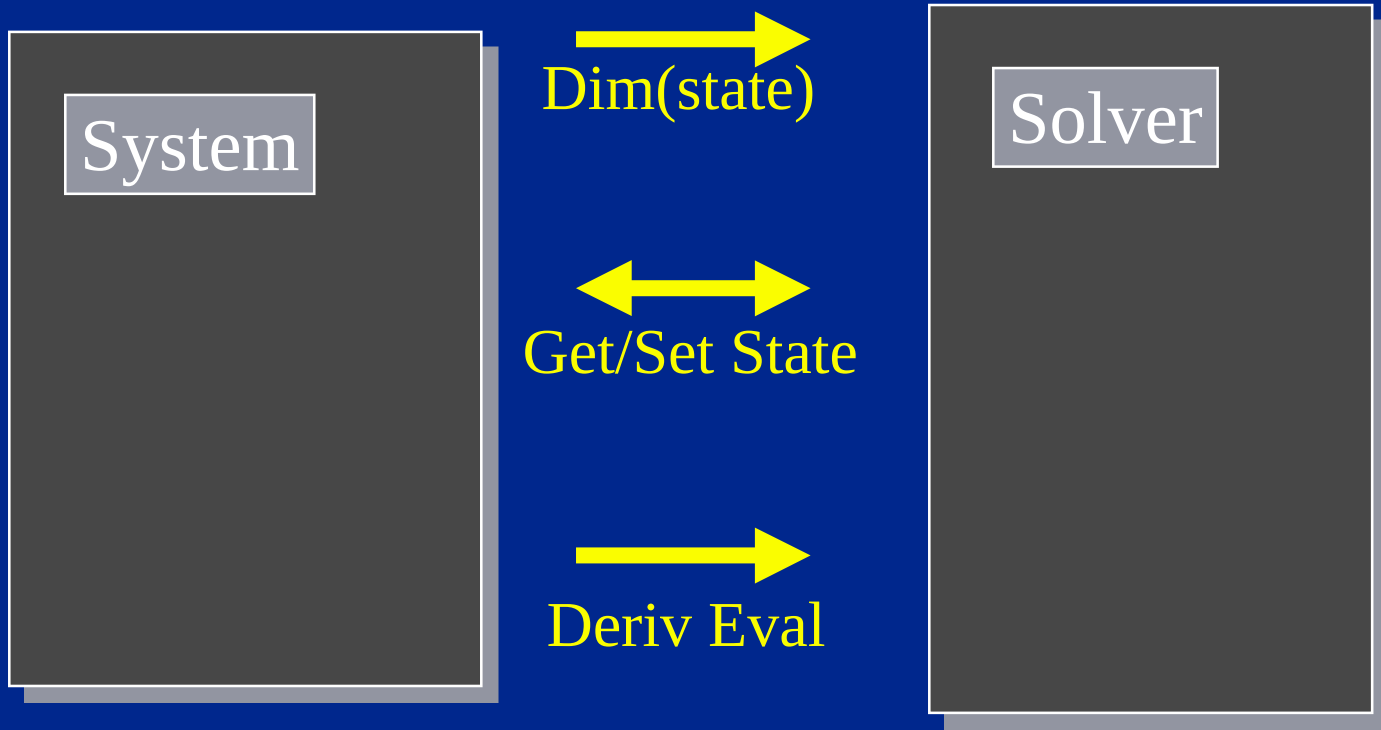
More methods…

- Euler's method is *1st Order*.
- The midpoint method is *2nd Order*.
- Just the tip of the iceberg. See *Numerical Recipes* for more.
- Helpful hints:
  - *Don't* use Euler's method (you will anyway.)
  - *Do* use adaptive step size.

Modular Implementation

- Generic operations:
  - Get dim(x)
  - Get/set x and t
  - Deriv Eval at current (x,t)
- Write solvers in terms of these.
  - Re-usable solver code.
  - Simplifies model implementation.

Solver Interface



A Code Fragment

```
void eulerStep(Sys sys, float h) {
   float t = getTime(sys);
   vector<float> x0, deltaX;

   t = getTime(sys);
   x0 = getState(sys);
   deltaX = derivEval(sys,x0, t);
   setState(sys, x0 + h*deltaX, t+h);
}
```