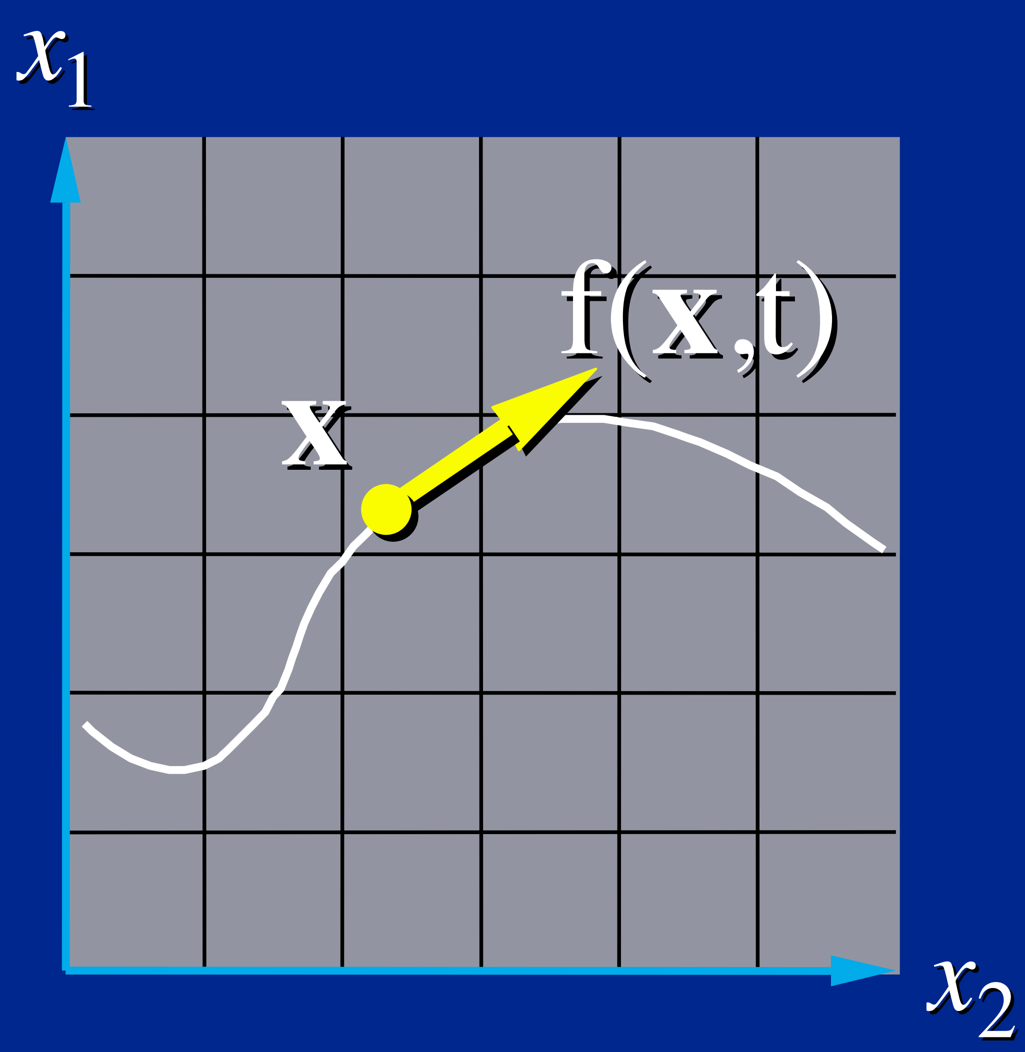
CS260
Lecture 2: Differential Equation Basics

Differential Equation Basics

Andrew Witkin

PIXAR
ANIMATION STUDIOS

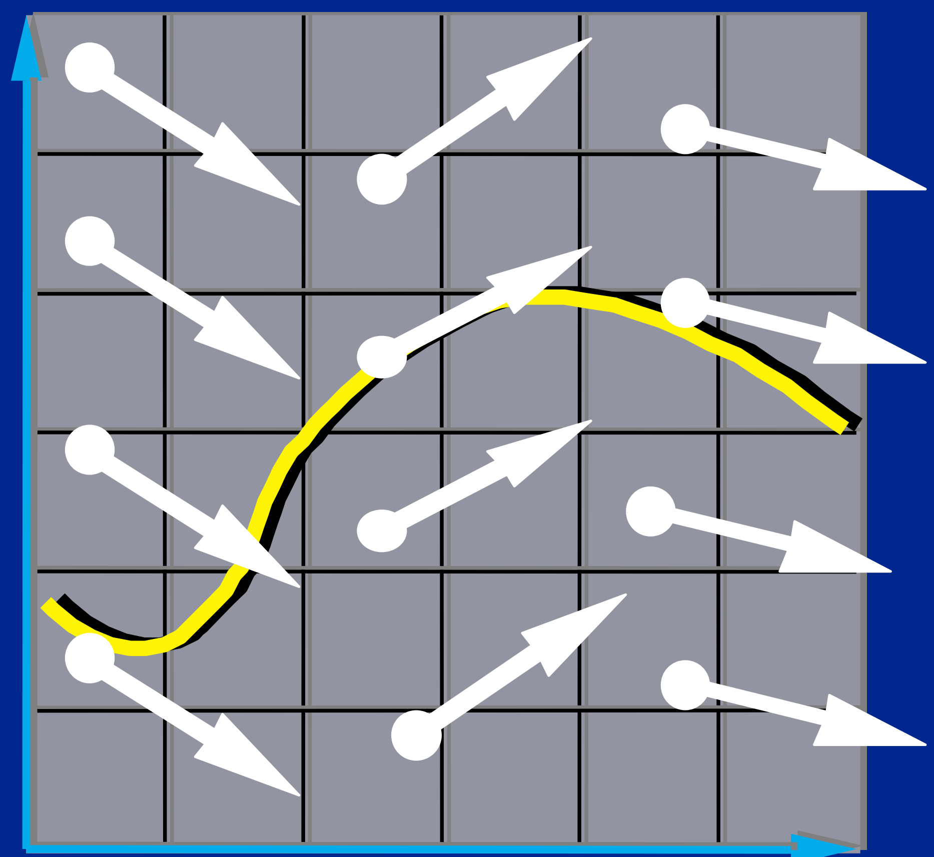
A Canonical Differential Equation



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

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

Vector Field



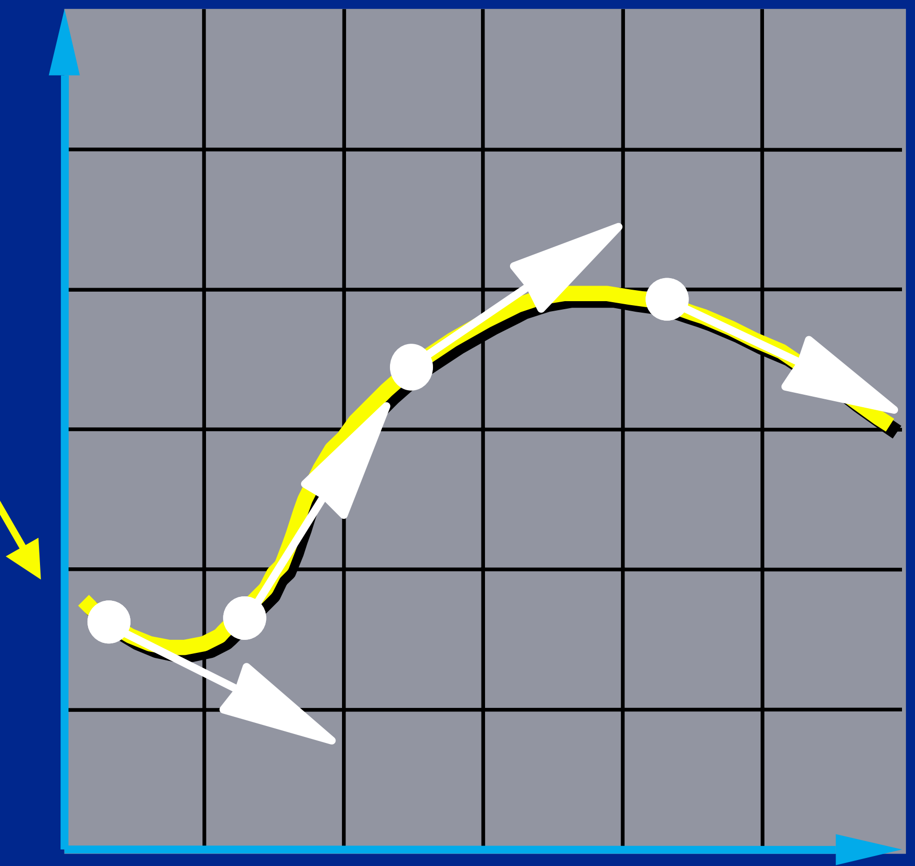
The differential equation

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

defines a vector field over x.

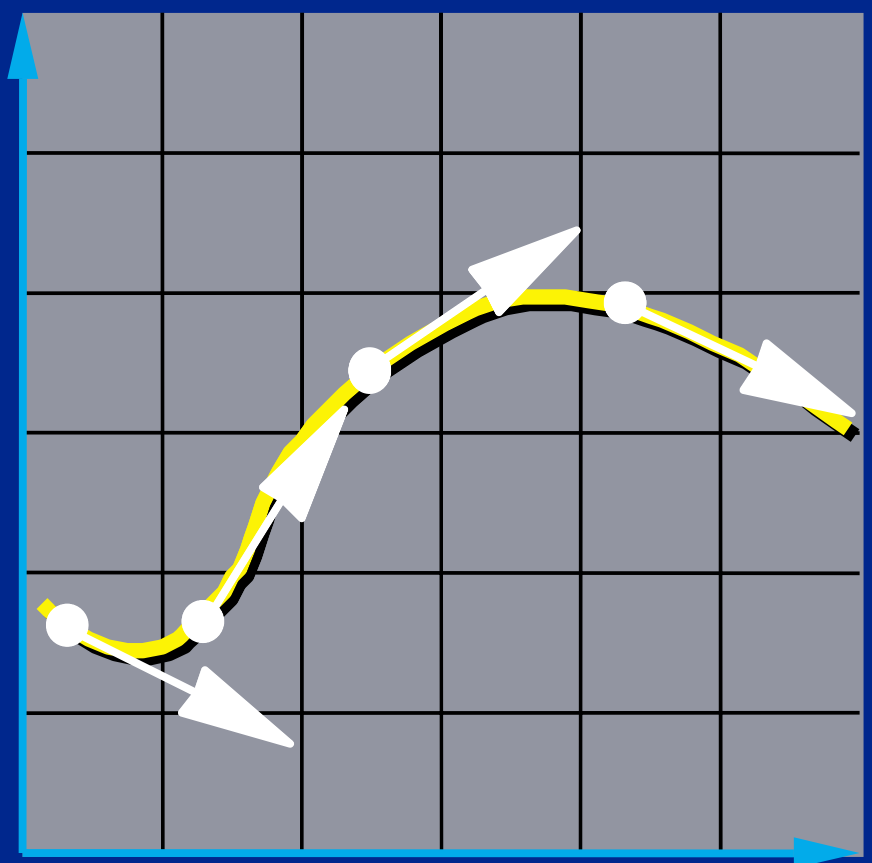
Integral Curves

Start Here



Pick any starting point, and follow the vectors.

Initial Value Problems



Given the starting point, follow the integral curve.

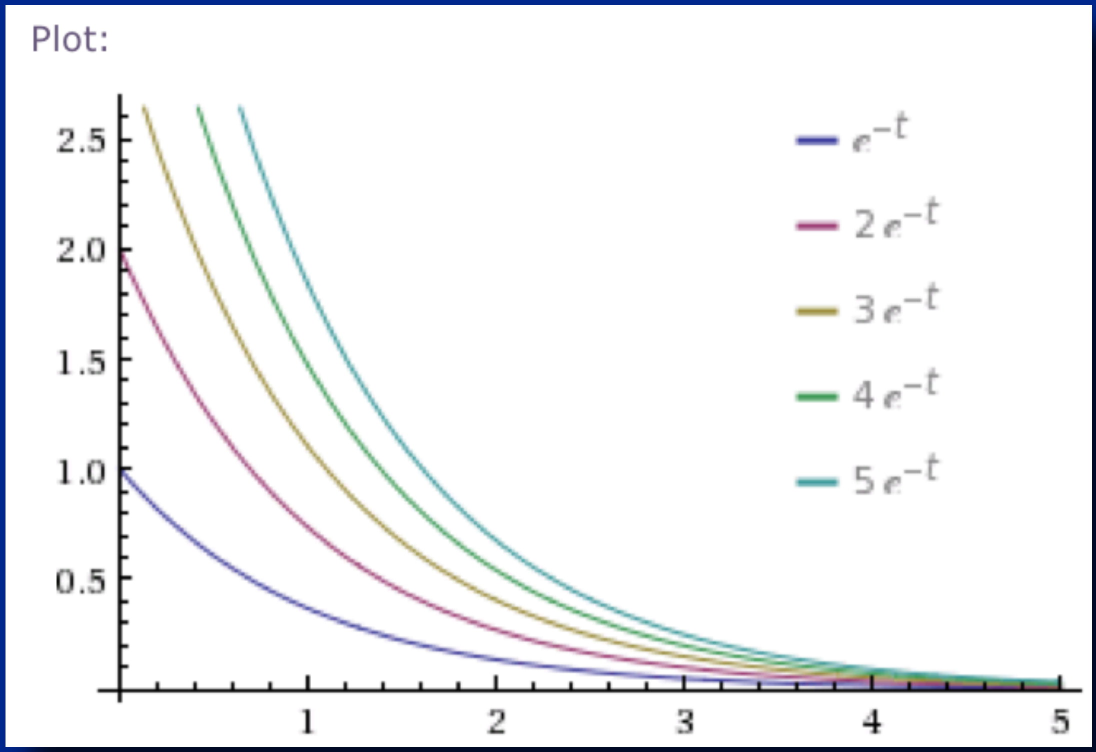
$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, t) \\ \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases} \Rightarrow \ \mathbf{x}(t), t \geq t_0$$

Closed Form Solutions

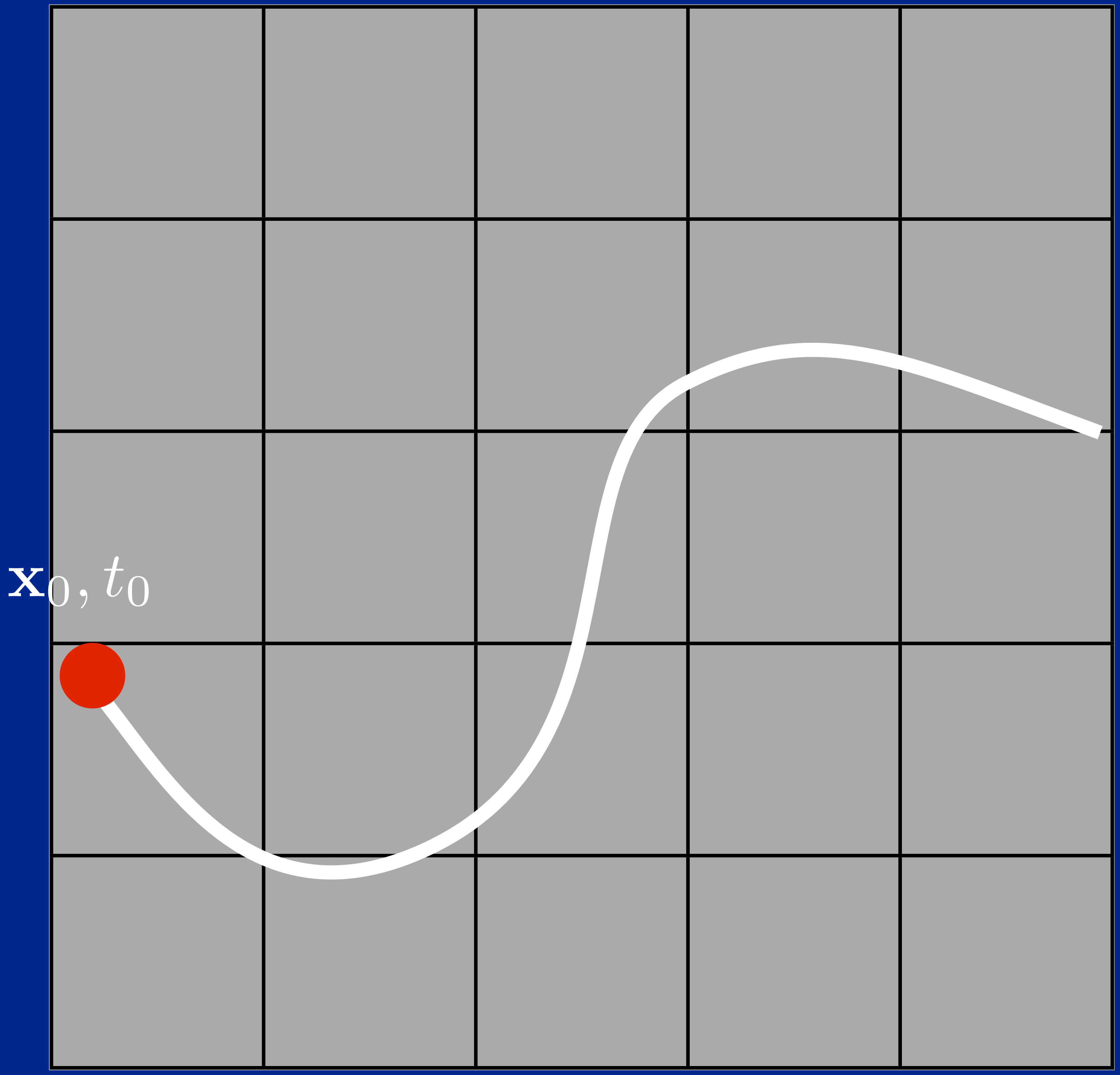
Some simpler IVPs have closed form solutions

$$\begin{cases} \dot{\mathbf{x}}(t) = -k\mathbf{x}(t) \\ \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases}$$

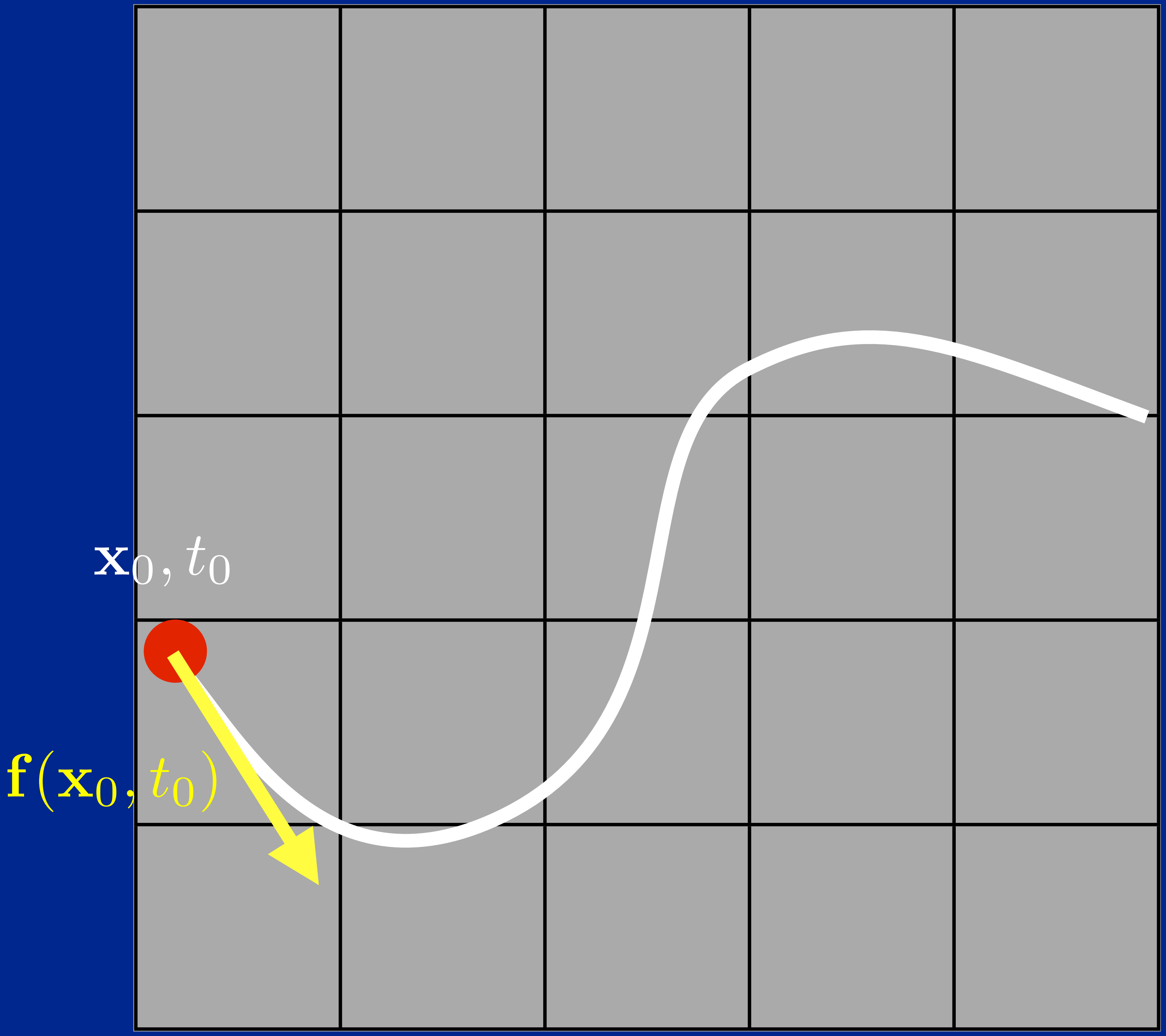
$$\Rightarrow \ \mathbf{x}(t) = \mathbf{x}_0 e^{-k(t-t_0)}, t \geq t_0$$



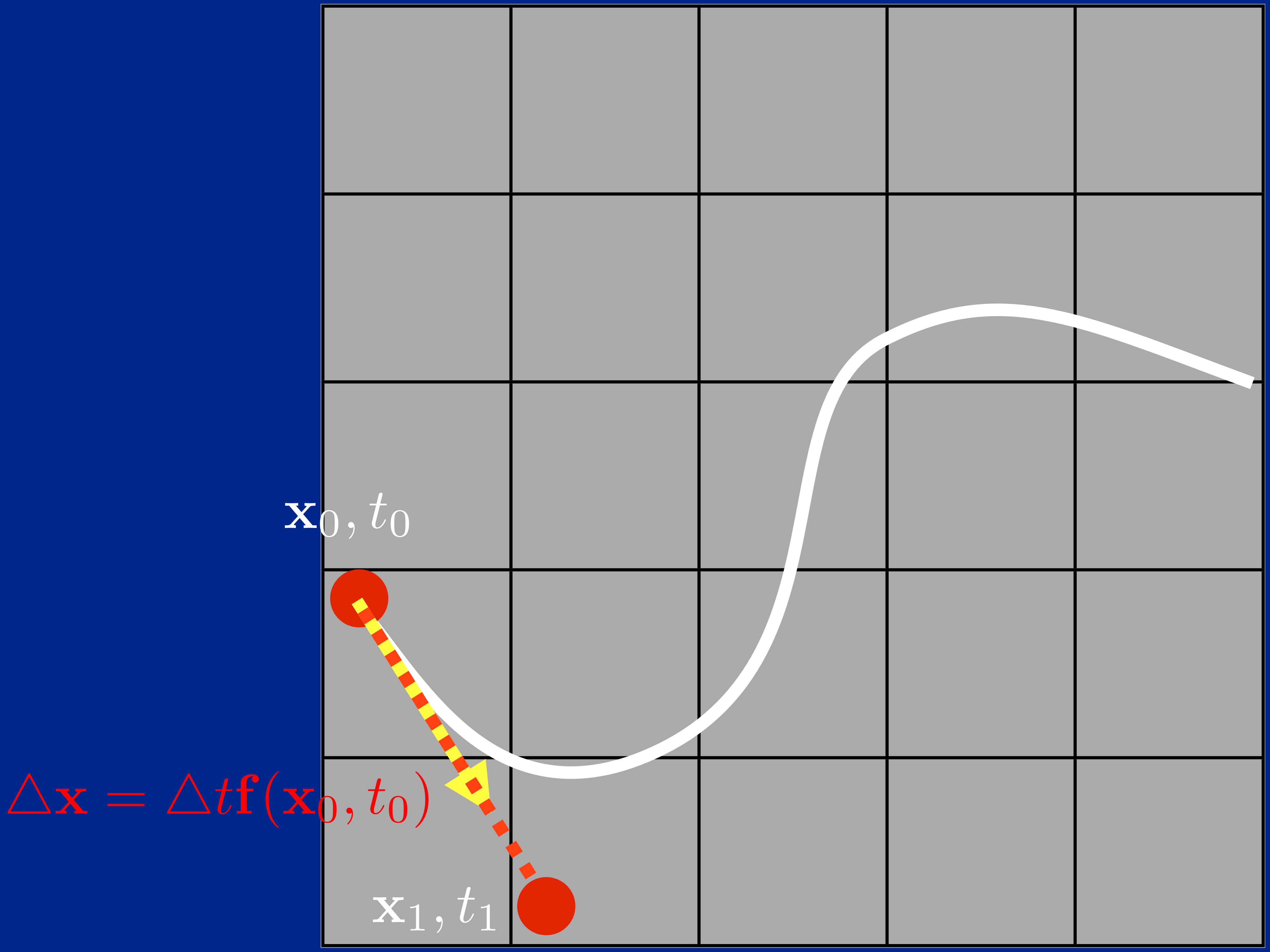
Numerical Solutions



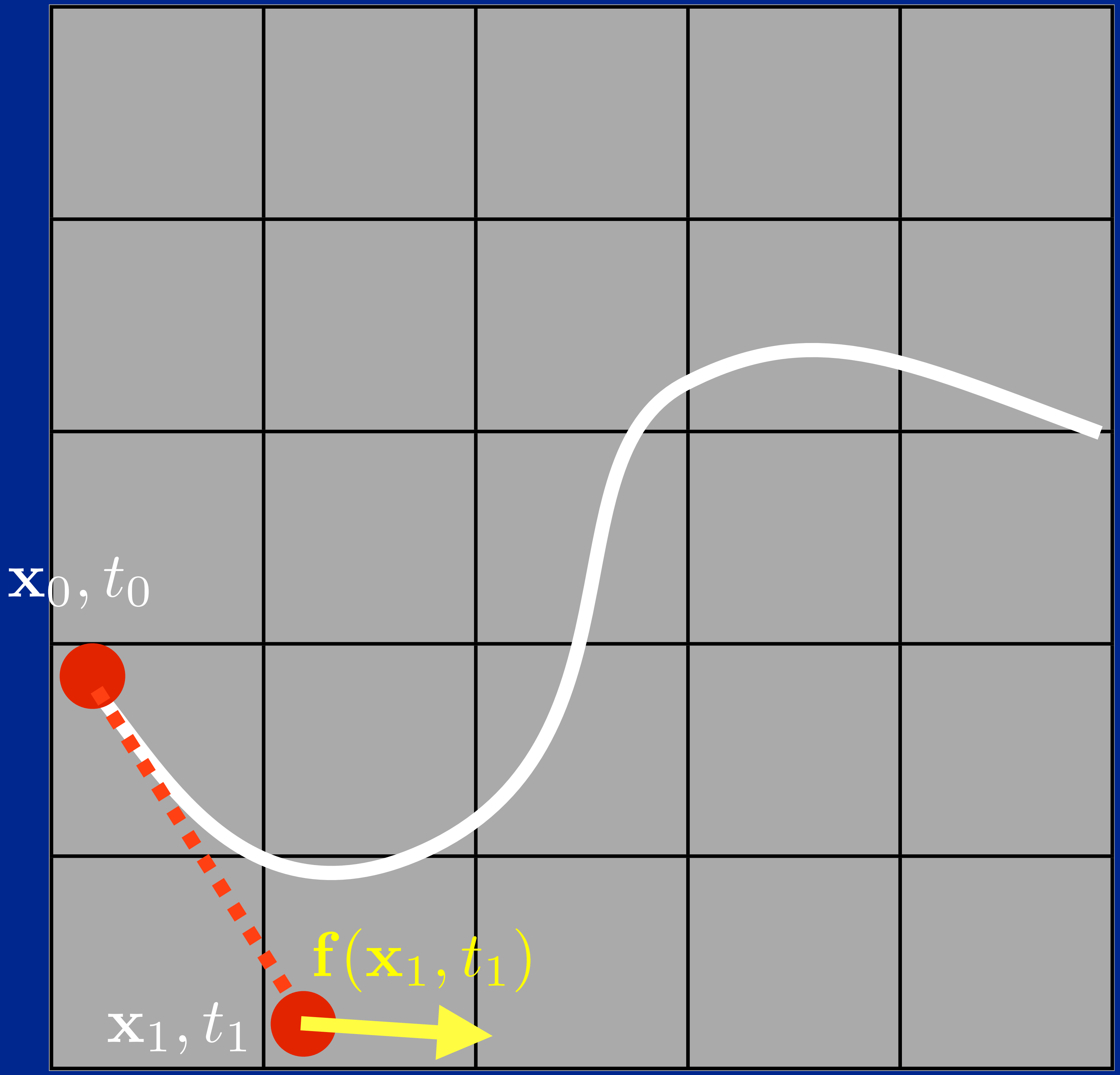
Numerical Solutions



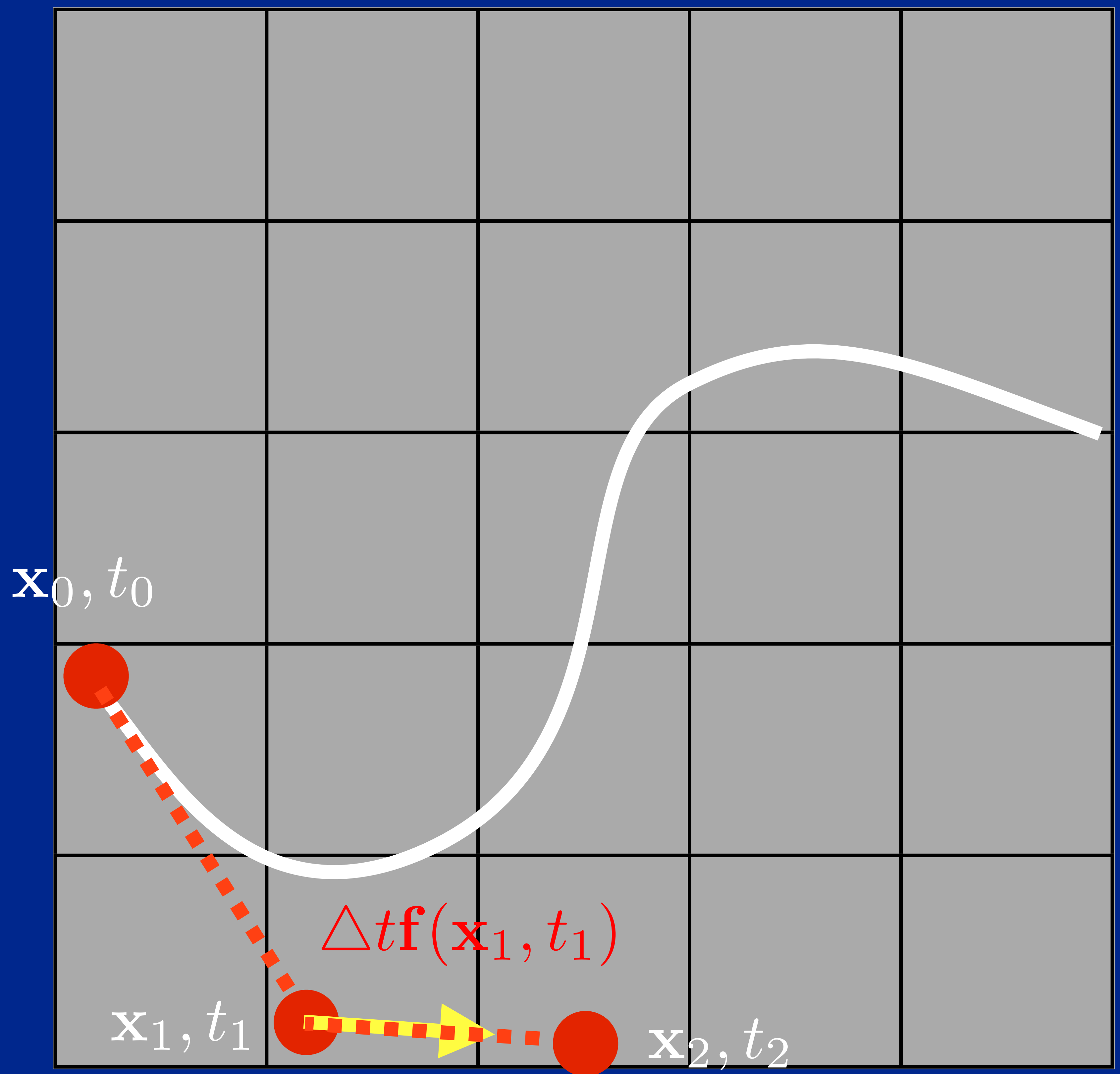
Numerical Solutions



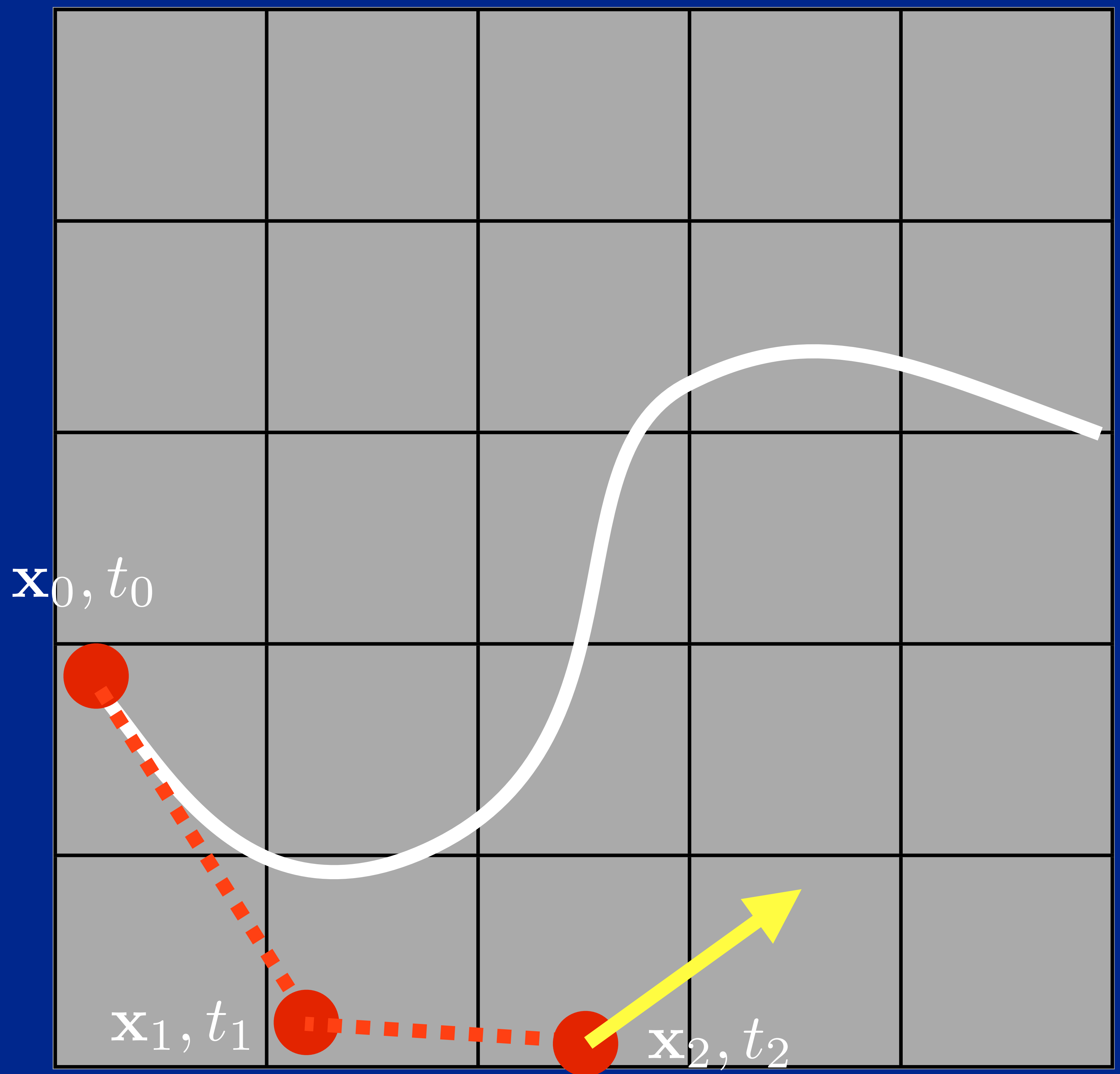
Numerical Solutions



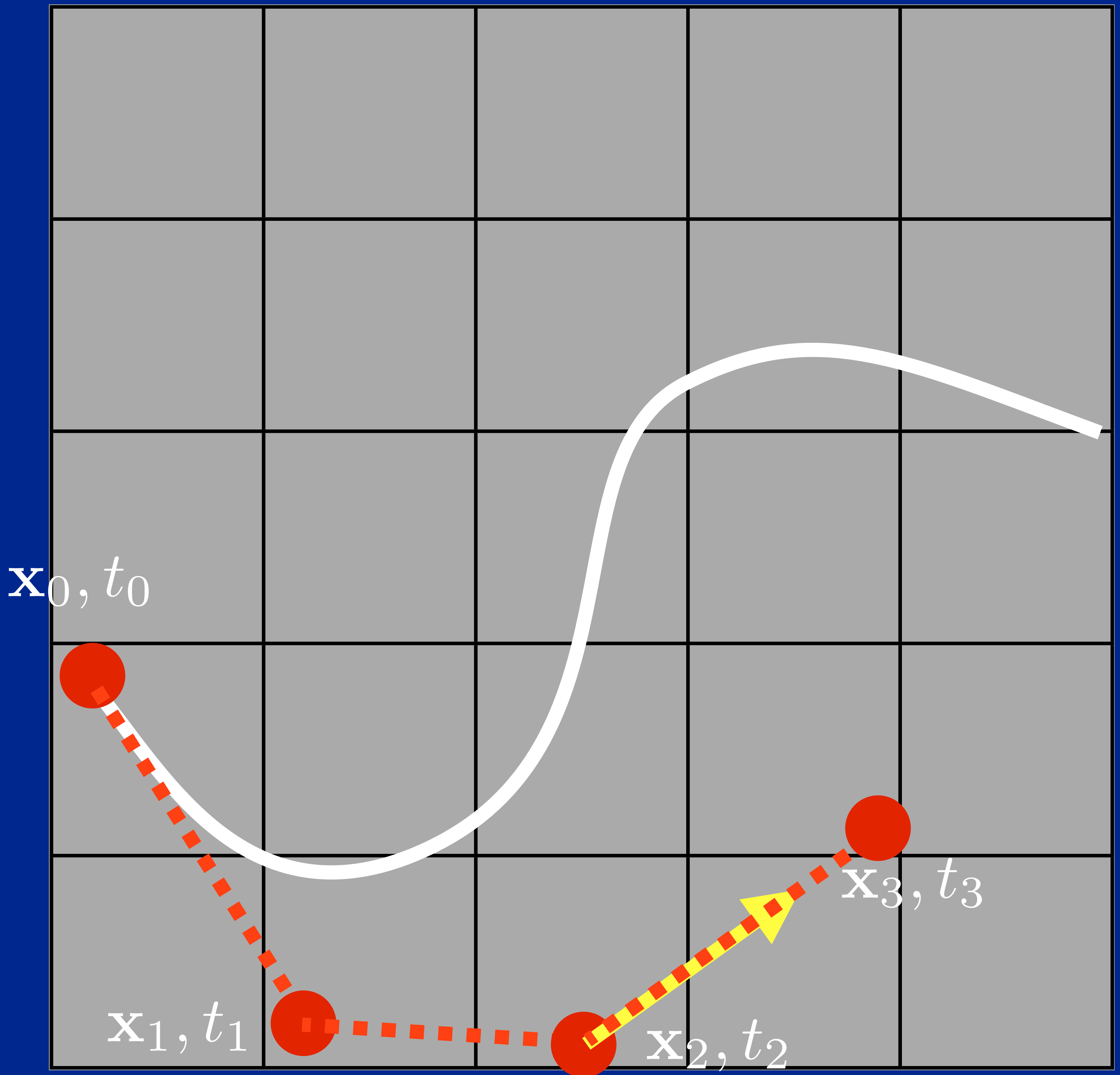
Numerical Solutions



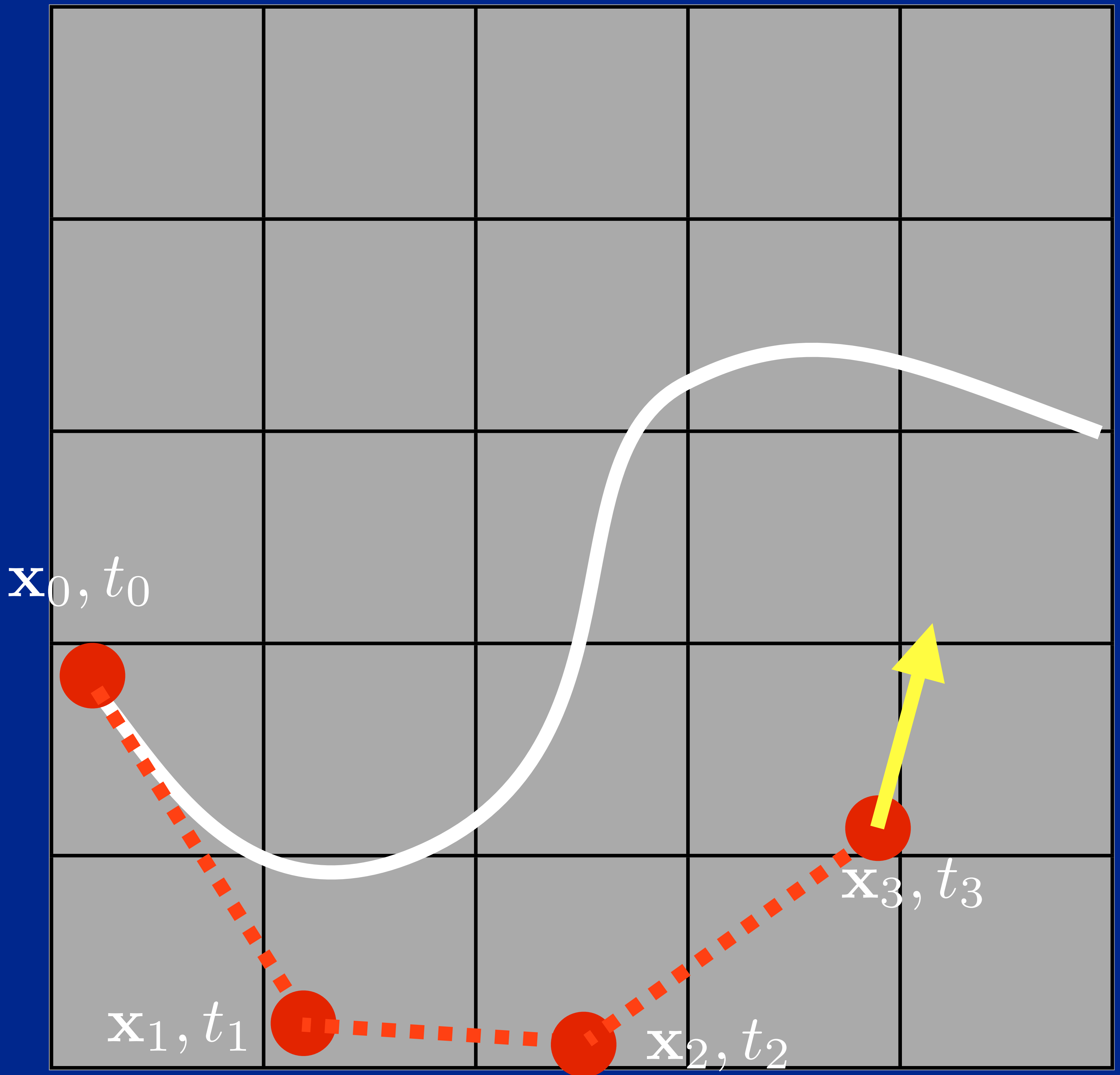
Numerical Solutions



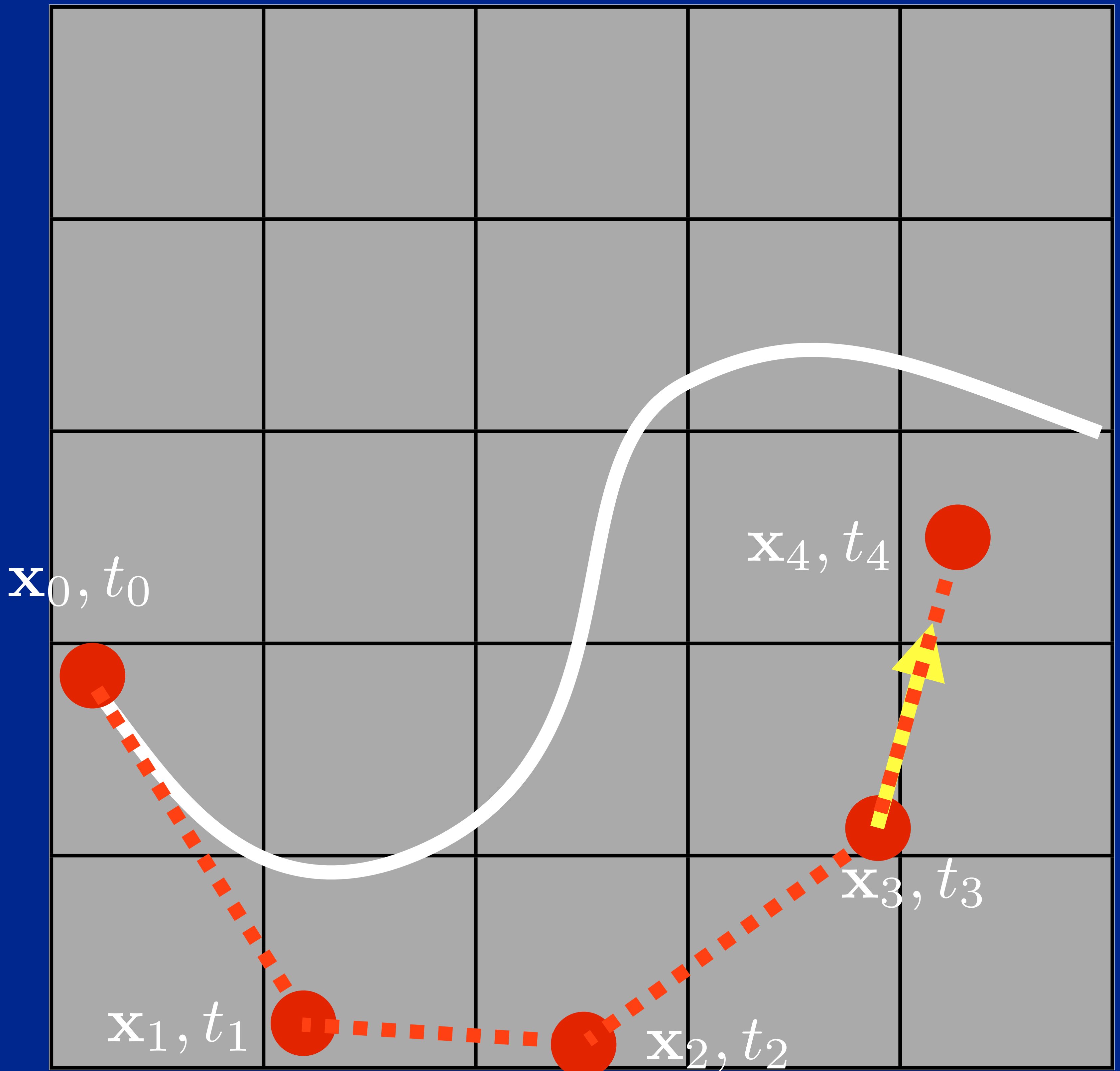
Numerical Solutions



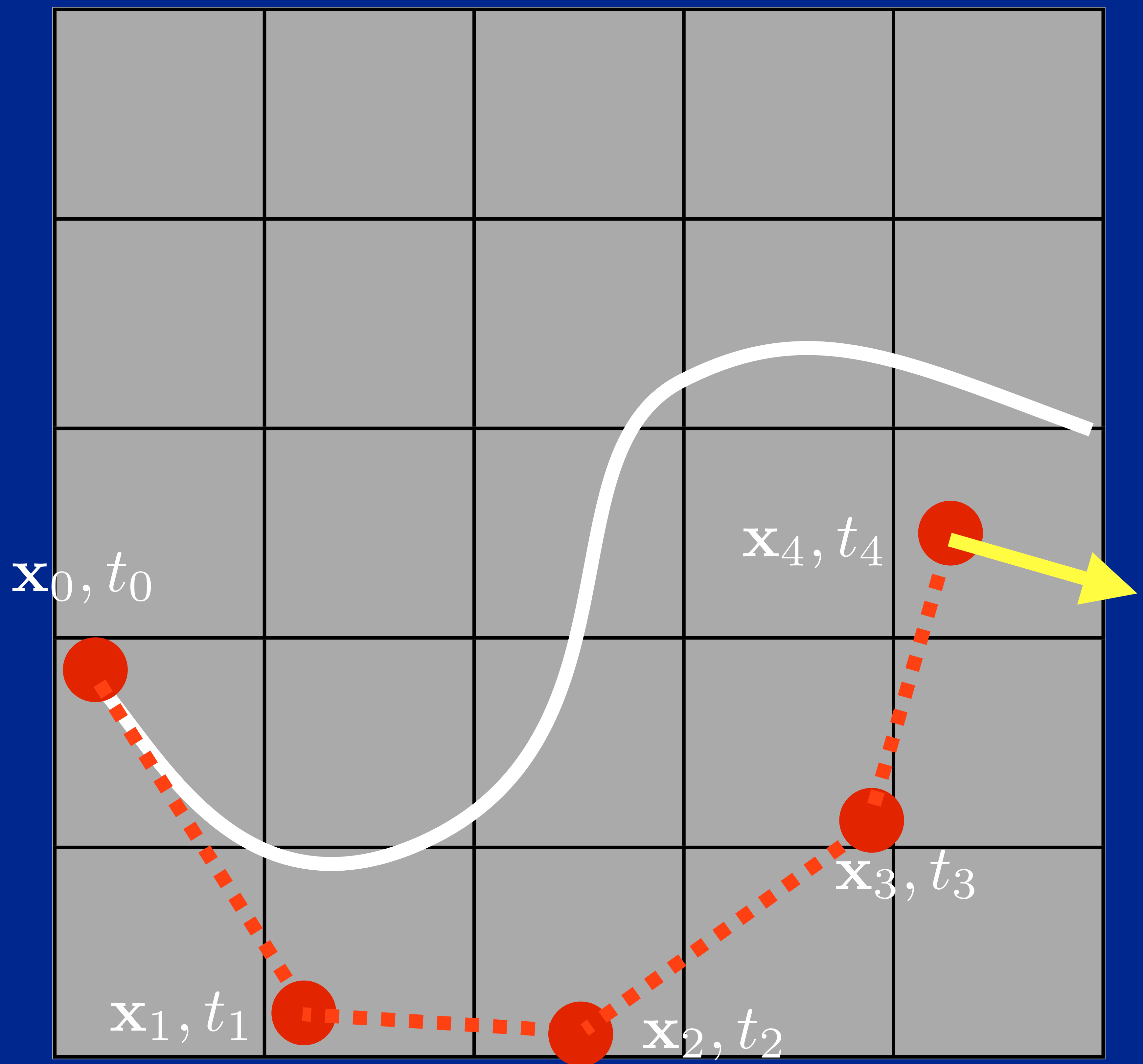
Numerical Solutions



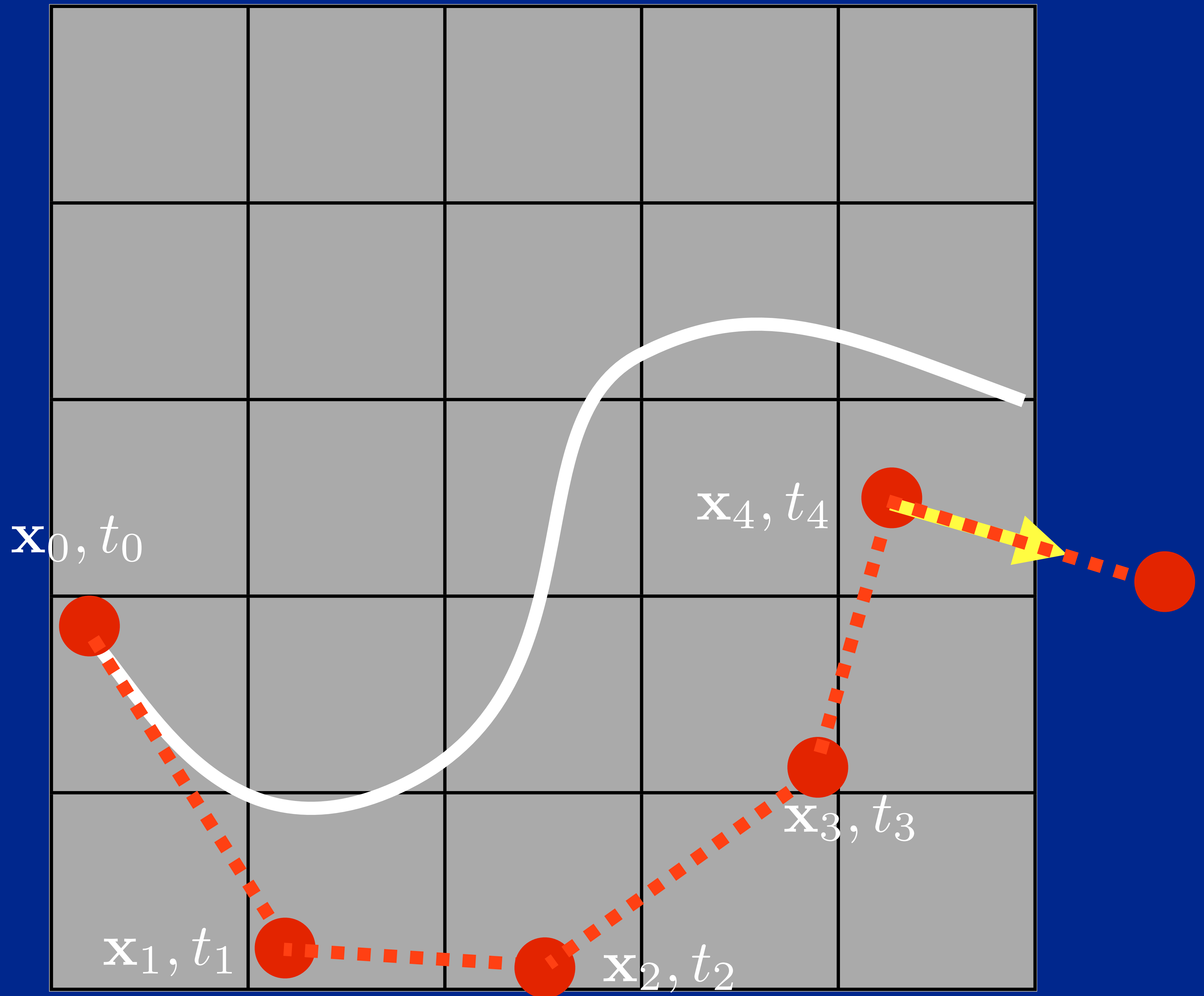
Numerical Solutions



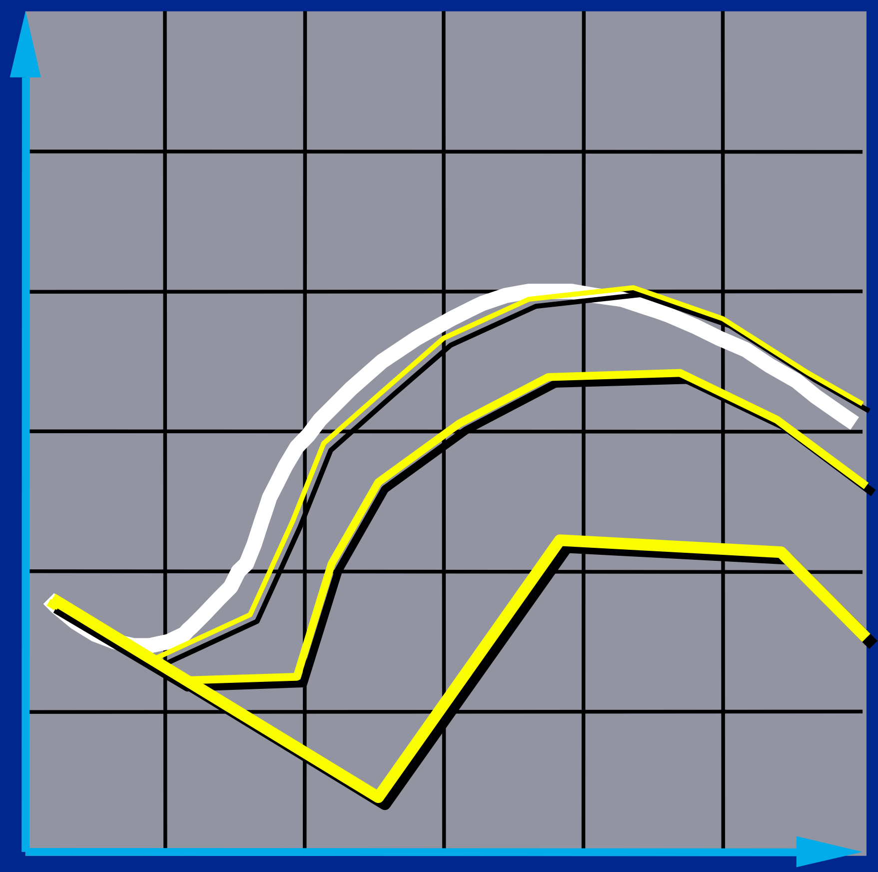
Numerical Solutions



Numerical Solutions



Euler's Method



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

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

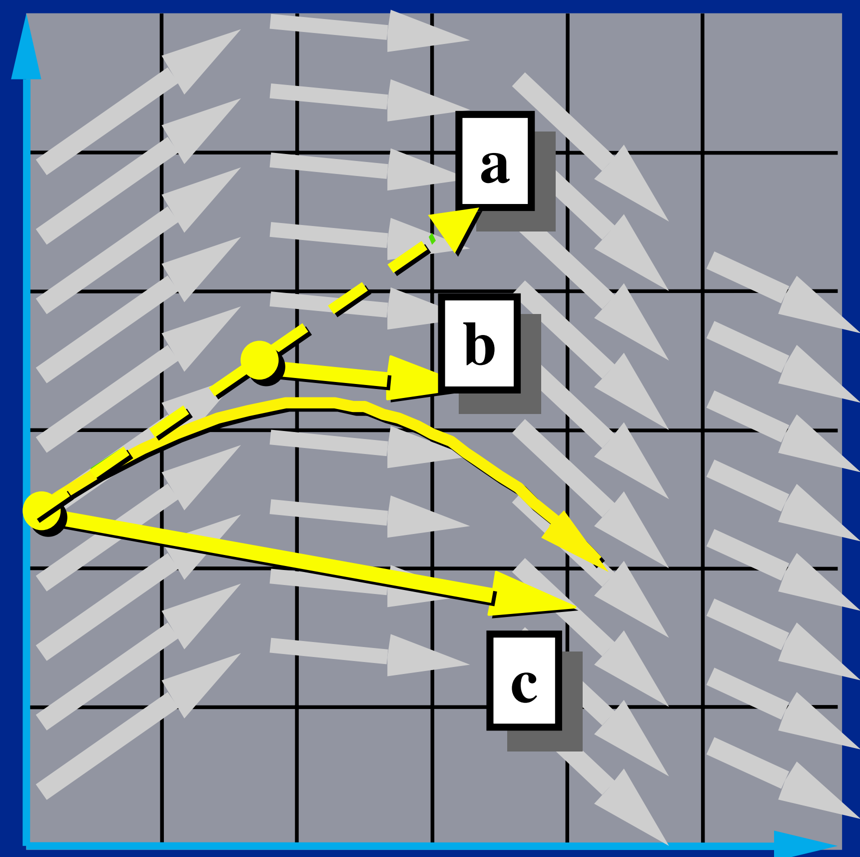
Efficiency

$$\text{cost} = \frac{\text{cost}}{\text{step}} * \#\ \text{steps}$$

Problem I: Inaccuracy

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

The Midpoint Method



a. Compute an Euler step

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

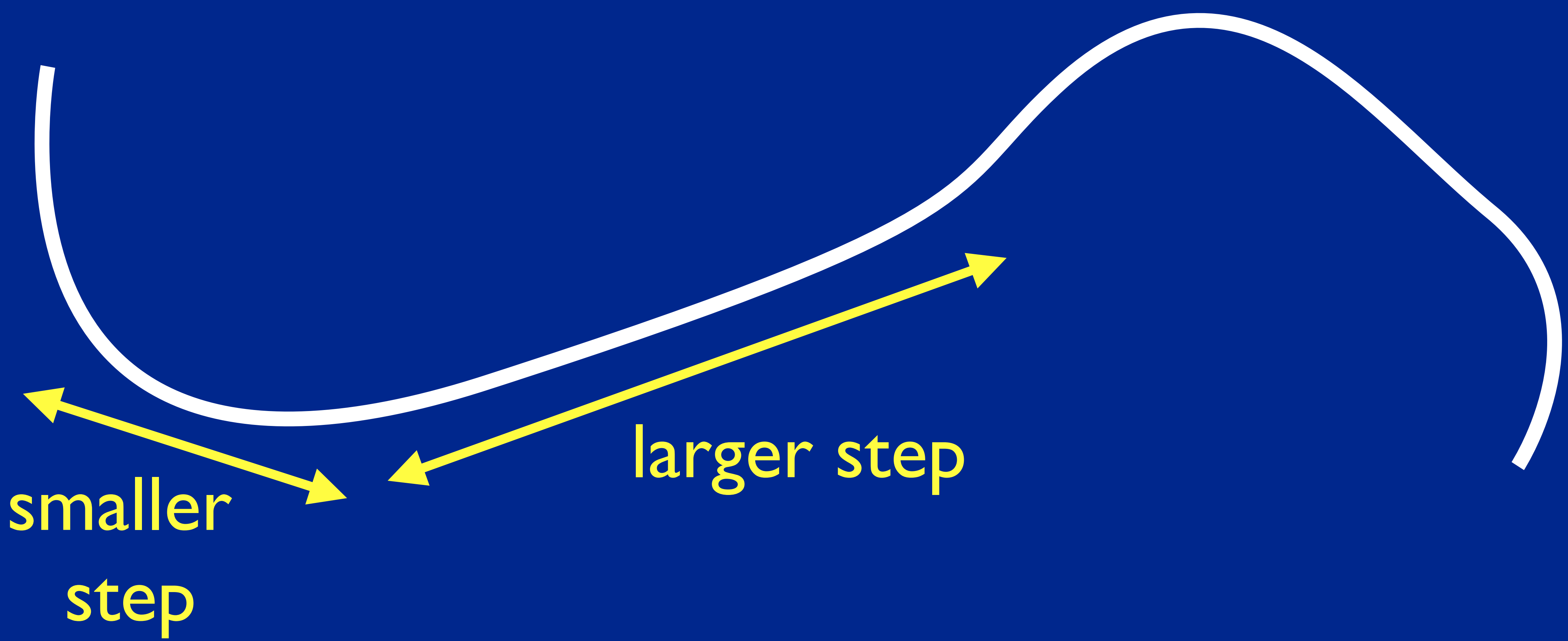
b. Evaluate f at the midpoint

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

c. Take a step using the midpoint value

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

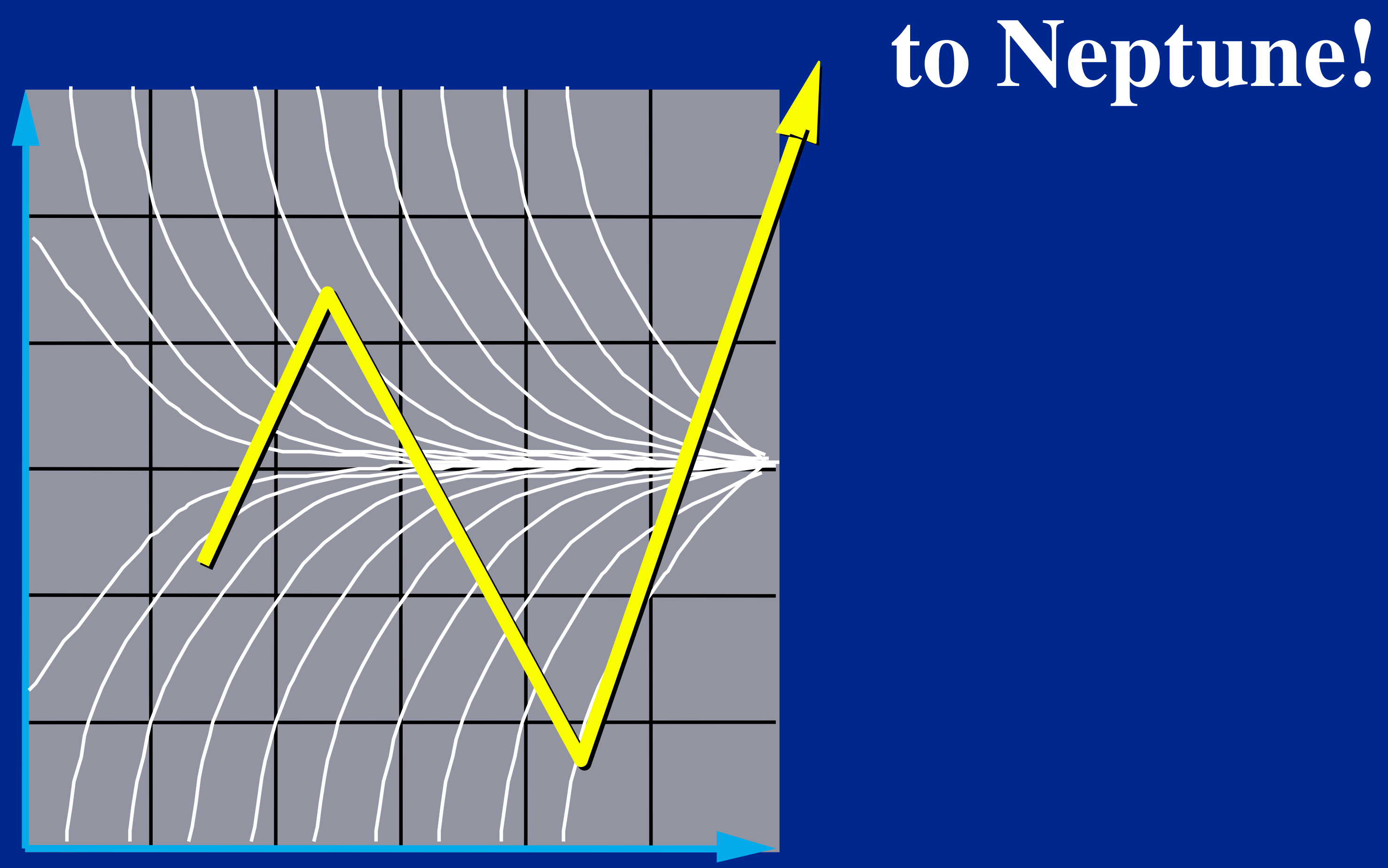
Adaptive Time Stepping



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.

Problem II: Instability



Brochu, Batty, Bridson 2010

As unresolved surface features accumulate, they can cause instability.

Brochu, Batty, Bridson 2010

Convergence

consistency
+
stability

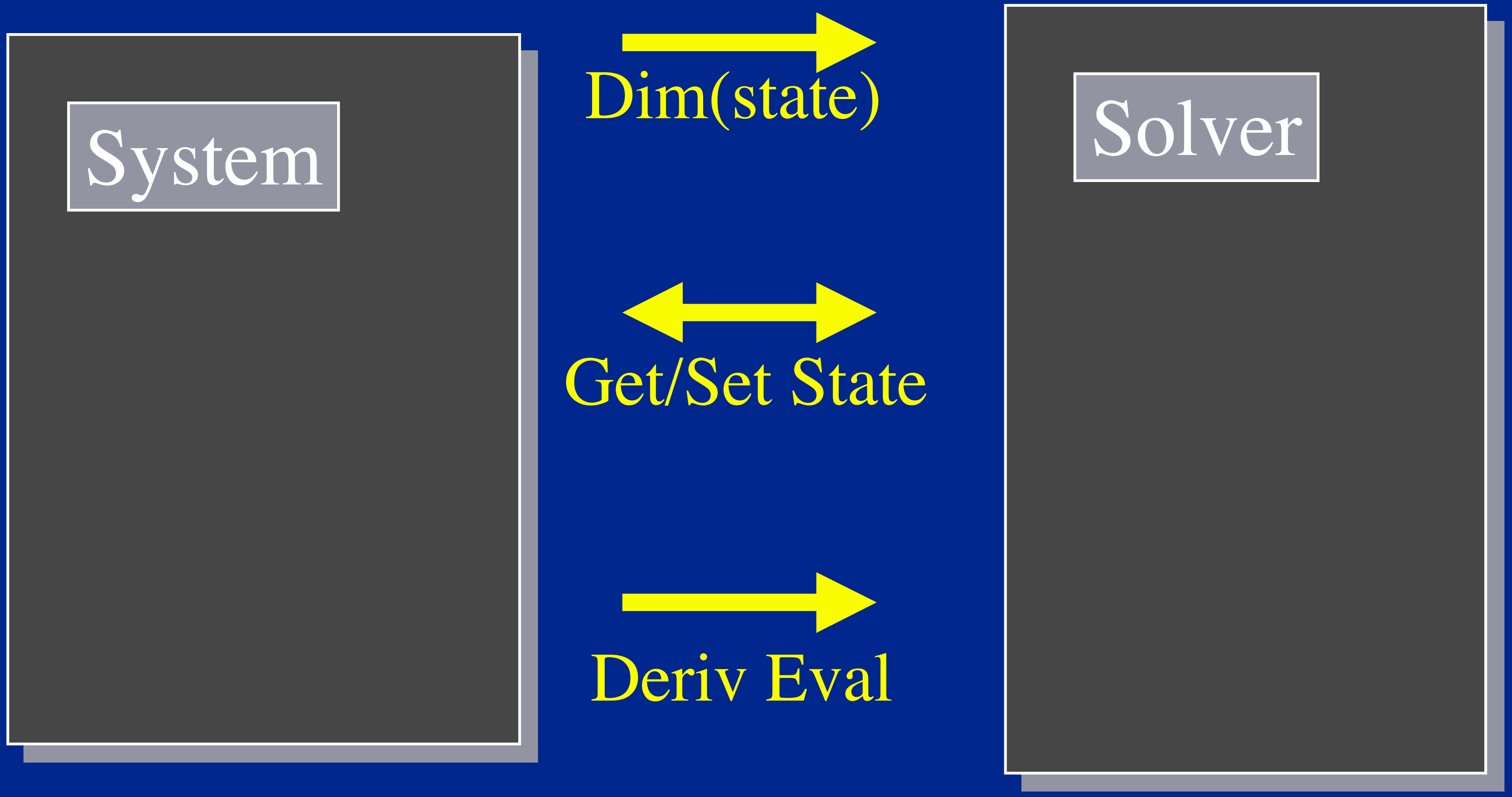
$\longrightarrow$ convergence

see also: Dahlquist equivalence theorem,
and Lax equivalence theorem

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 dt)
{
    float t;
    vector<float> x, dx;

    t = getTime(sys);
    x = getState(sys);
    dx = dt * derivEval(sys, x, t);
    setState(sys, x + dx, t + dt);
}
```