Numerical Integration of Ordinary Differential Equations

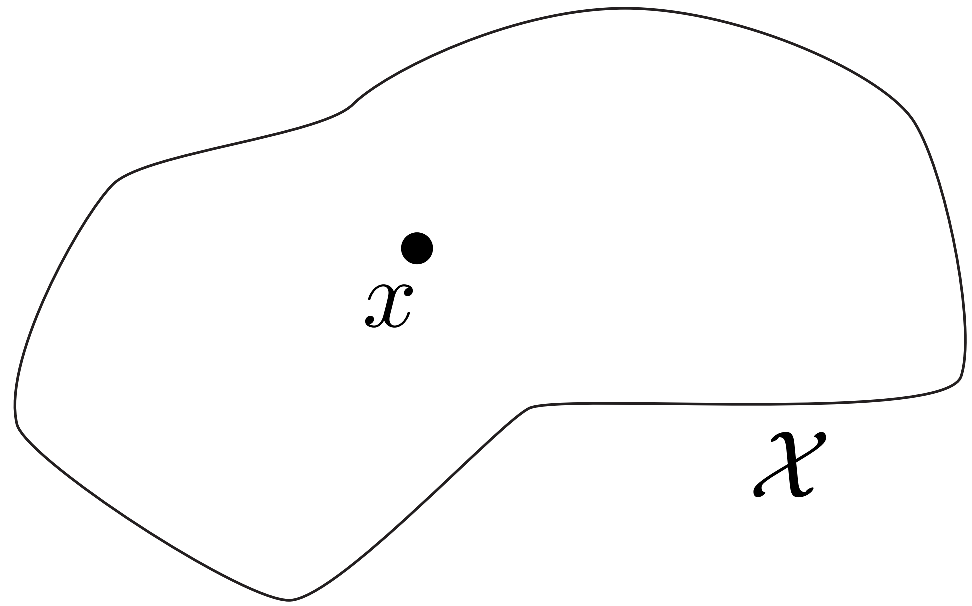
Reading:
NDAC Secs. 2.8 and 6.1

Numerical Integration ...

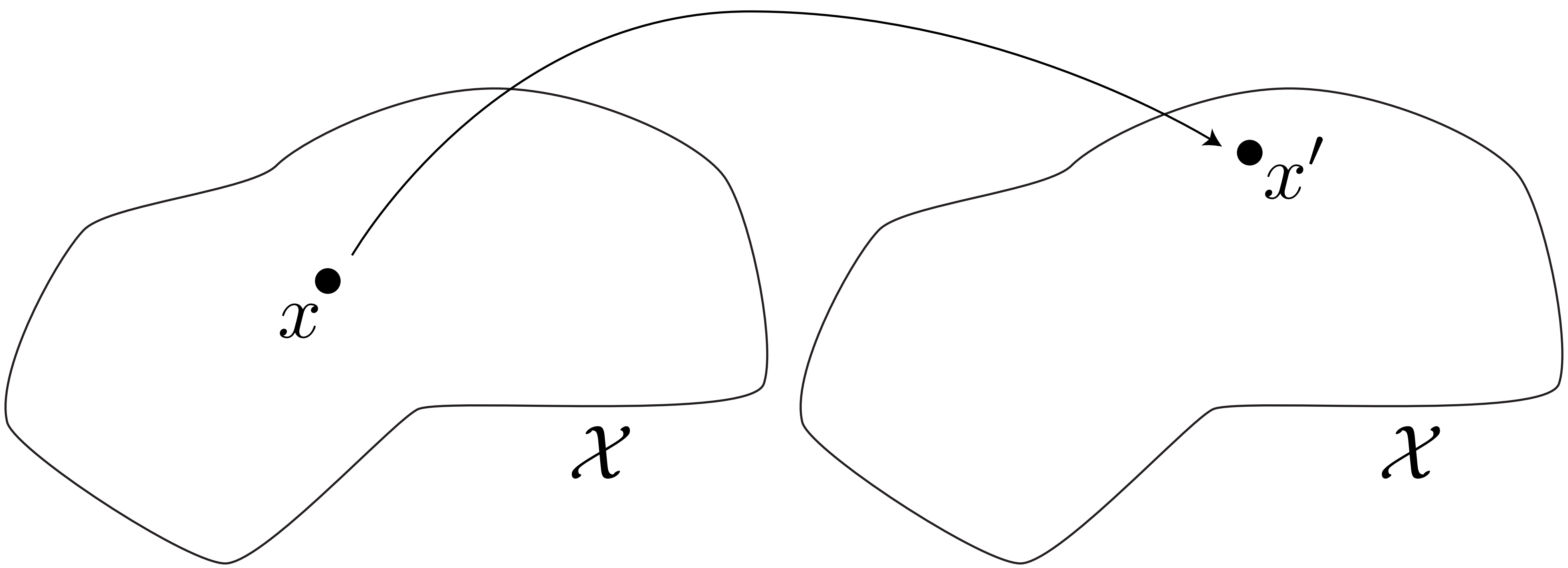
Dynamical System: $\{\mathcal{X}, \mathcal{T}\}$

State Space: $\mathcal{X}$

State: $x \in \mathcal{X}$



Dynamic: $\mathcal{T} : \mathcal{X} \rightarrow \mathcal{X}$



Numerical Integration ...

Dynamical System ...

For example, continuous time ...

Ordinary differential equation: $\dot{\vec{x}} = \vec{F}(\vec{x})$ $\quad\left(\dot{} = \frac{d}{dt} \right)$

State: $\vec{x}(t) \in \mathbf{R}^n$ $\quad \vec{x} \; = \; (x_1, x_2, \ldots, x_n)$

Initial condition: $\vec{x}(0)$

Dynamic: $\vec{F} : \mathbf{R}^n \to \mathbf{R}^n$ $\quad \vec{F} \; = \; (f_1, f_2, \ldots, f_n)$

Dimension: n

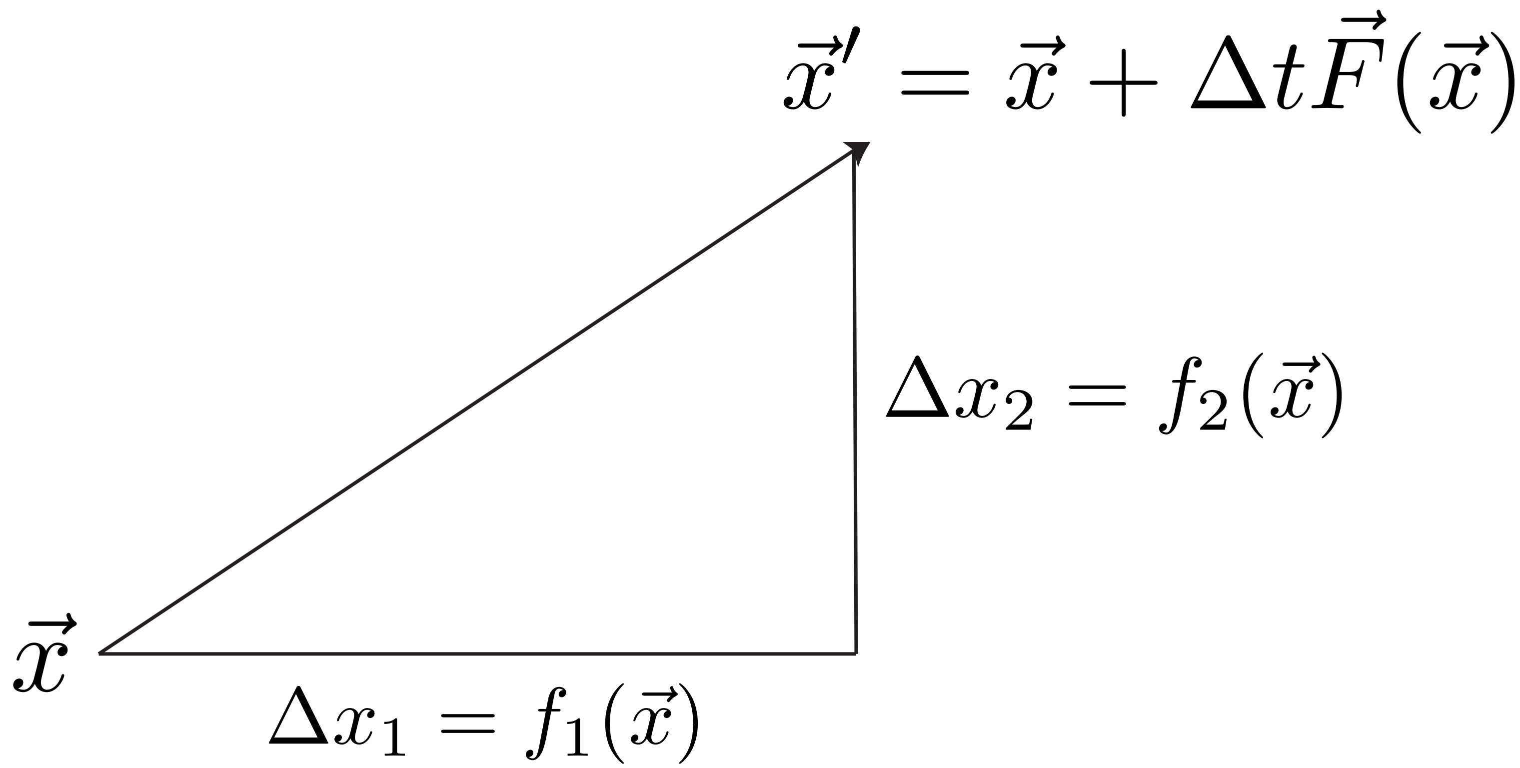
Numerical Integration ...

$$\mathcal{X} = \mathbf{R}^2 \quad \vec{x} = (x_1, x_2)$$
$$\vec{F} = (f_1(\vec{x}), f_2(\vec{x}))$$

Geometric view of an ODE:

$$\frac{d\vec{x}}{dt} = \vec{F}(\vec{x})$$

$$\frac{d\vec{x}}{dt} \approx \frac{\Delta\vec{x}}{\Delta t} = \frac{\vec{x}' - \vec{x}}{\Delta t}$$



Each state $\vec{x}$ has a vector attached $\vec{F}(\vec{x})$

that says to what next state to go: $\vec{x}' = \vec{x} + \Delta t \cdot \vec{F}(\vec{x})$.

Numerical Integration ...

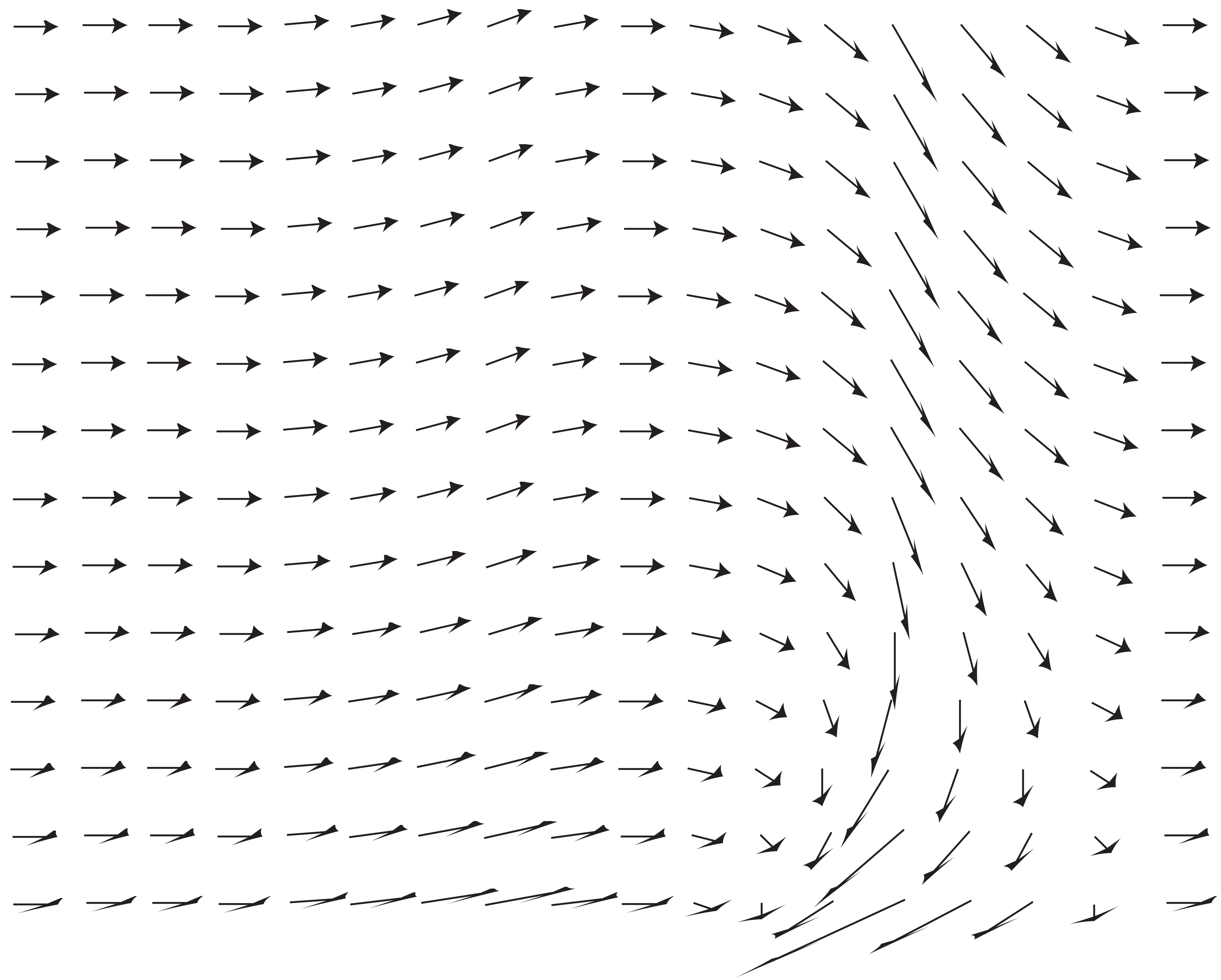
Vector field for an ODE (aka Phase Portrait)

$\mathcal{X} = \mathbf{R}^2$

A set of rules:

Each state has a vector attached

That says to what next state to go



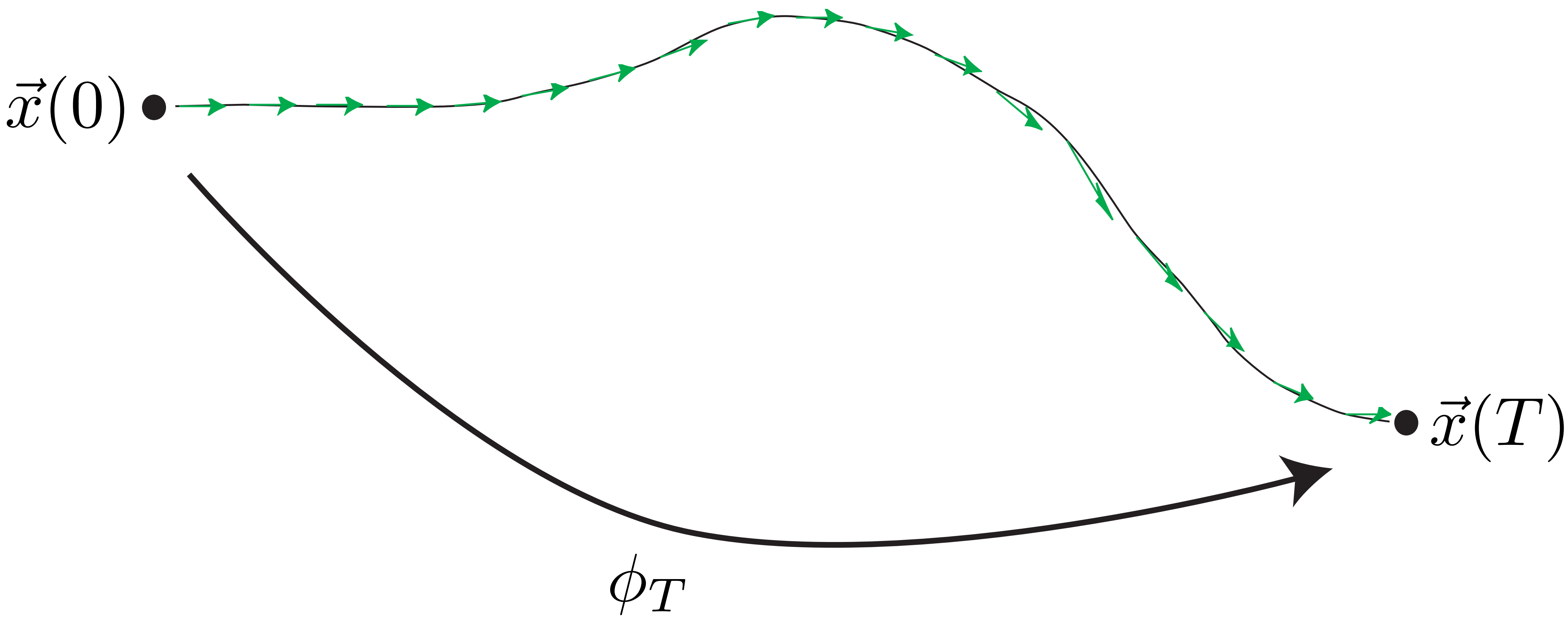
Numerical Integration ...

Time-T Flow:

$$\vec{x}(T) = \phi_T(\vec{x}(0)) = \int_0^T dt\ \dot{\vec{x}} = \int_0^T dt\ \vec{F}(\vec{x}(t))$$

The solution of the ODE, starting from some IC

Simply follow the arrows



Point: ODE is only instantaneous,

flow gives state for *any* time t.

Numerical Integration ...

Euler Method in 1D:

$$\dot{x} = f(x)$$

$$\frac{dx}{dt} \approx \frac{\Delta x}{\Delta t} = \frac{x_{n+1} - x_n}{\Delta t}$$

$$x(t_0 + \Delta t) \approx x_1 = x_0 + f(x_0)\Delta t$$

$$x_{n+1} = x_n + f(x_n)\Delta t$$

A discrete-time map!

Numerical Integration ...

Improved Euler Method in 1D:

$$\dot{x} = f(x)$$

A trial (Euler) step:

$$\widehat{x}_n = x_n + f(x_n)\Delta t$$

The resulting better estimate (averaged at t_n and t_{n+1}):

$$x_{n+1} = x_n + \frac{1}{2}\left[f(x_n) + f(\widehat{x}_n)\right]\Delta t$$

Fourth-order Runge-Kutta Method in 1D:

Intermediate estimates:

$$\begin{aligned} k_1 &= f(x_n)\Delta t \\ k_2 &= f(x_n + \tfrac{1}{2}k_1)\Delta t \\ k_3 &= f(x_n + \tfrac{1}{2}k_2)\Delta t \\ k_4 &= f(x_n + k_3)\Delta t \end{aligned}$$

Final estimate:

$$x_{n+1} = x_n + \frac{1}{6}\left[k_1 + 2k_2 + 2k_3 + k_4\right]$$

Good trade-off between accuracy and time-step size.

Numerical Integration ...

Runge-Kutta in nD:

$$\dot{\vec{x}} = \vec{f}(\vec{x}), \ x \in \mathbb{R}^n \qquad \vec{f} : \mathbb{R}^n \to \mathbb{R}^n$$

Intermediate estimates:

$$\begin{aligned}
\vec{k}_1 &= \vec{f}(\vec{x}_n)\Delta t \\
\vec{k}_2 &= \vec{f}(\vec{x}_n + \tfrac{1}{2}\vec{k}_1)\Delta t \\
\vec{k}_3 &= \vec{f}(\vec{x}_n + \tfrac{1}{2}\vec{k}_2)\Delta t \\
\vec{k}_4 &= \vec{f}(\vec{x}_n + \vec{k}_3)\Delta t
\end{aligned}$$

Final estimate:

$$\vec{x}_{n+1} = \vec{x}_n + \tfrac{1}{6}\left[\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4\right]$$