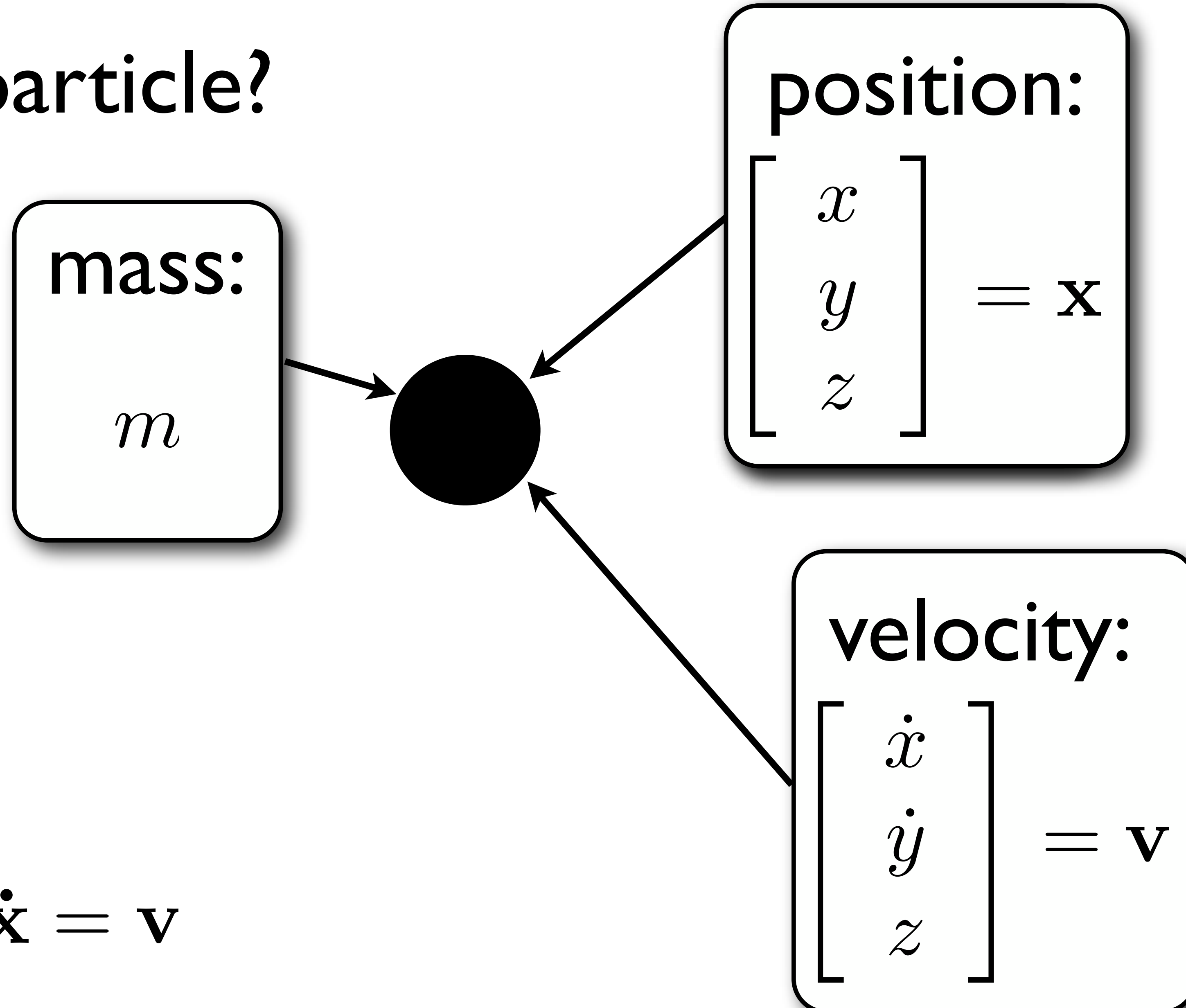
Particle Dynamics

from Zoran Popović

What is a particle?



- *Inertia*: $\dot{\mathbf{x}} = \mathbf{v}$
- $\therefore$ we can't change a particles velocity, only it's *acceleration*
  - *FORCES!*

Newtonian particle

- Differential equations: f=ma
- Forces depend on:
- Position, velocity, time

$$\ddot{x} = \frac{f(x, \dot{x})}{m}$$

Second order equations

$$\ddot{x} = \frac{f(x, \dot{x})}{m}$$

Has 2nd derivatives

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= \frac{f(x, \dot{x})}{m} \end{aligned}$$

Add a new variable v to get a pair of coupled 1st order equations

Phase space

$$\begin{bmatrix} x \\ v \end{bmatrix}$$

Concatenate x and v to make a 6-vector: position in phase space

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix}$$

Velocity on Phase space:

Another 6-vector

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ f/m \end{bmatrix}$$

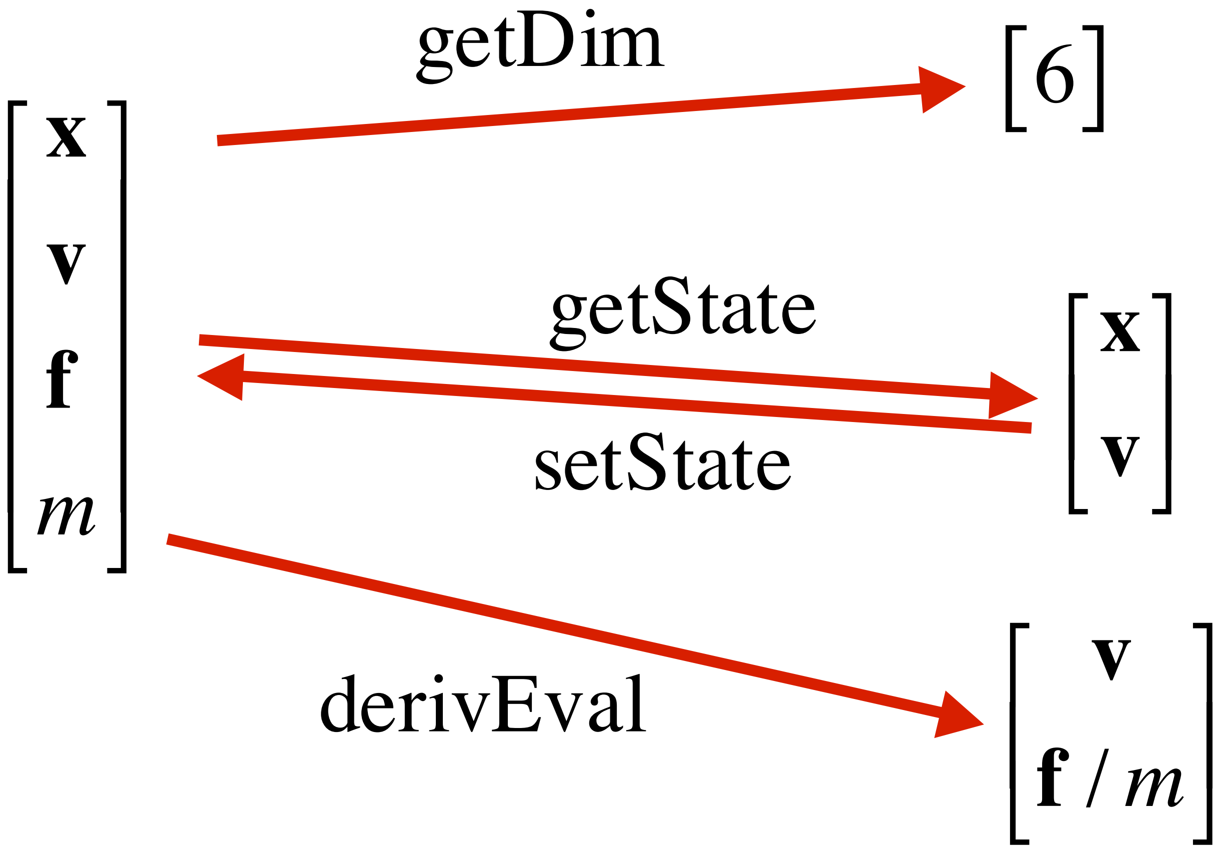
A vanilla 1st-order differential equation

Particle structure

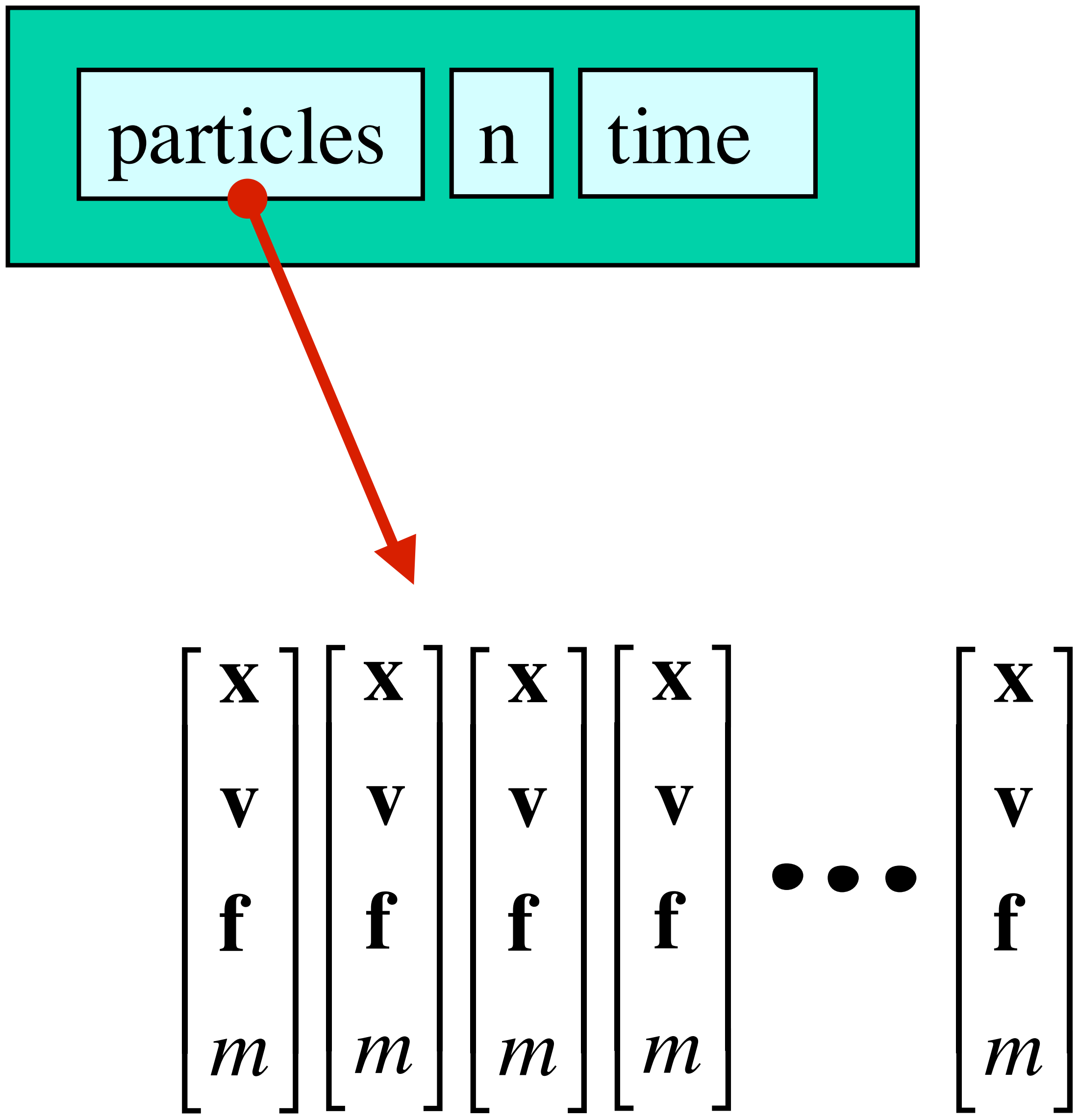
Position in phase space

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{v} \\ \mathbf{f} \\ m \end{bmatrix} \begin{array}{l} \leftarrow \text{position} \\ \leftarrow \text{velocity} \\ \leftarrow \text{force accumulator} \\ \leftarrow \text{mass} \end{array}$$

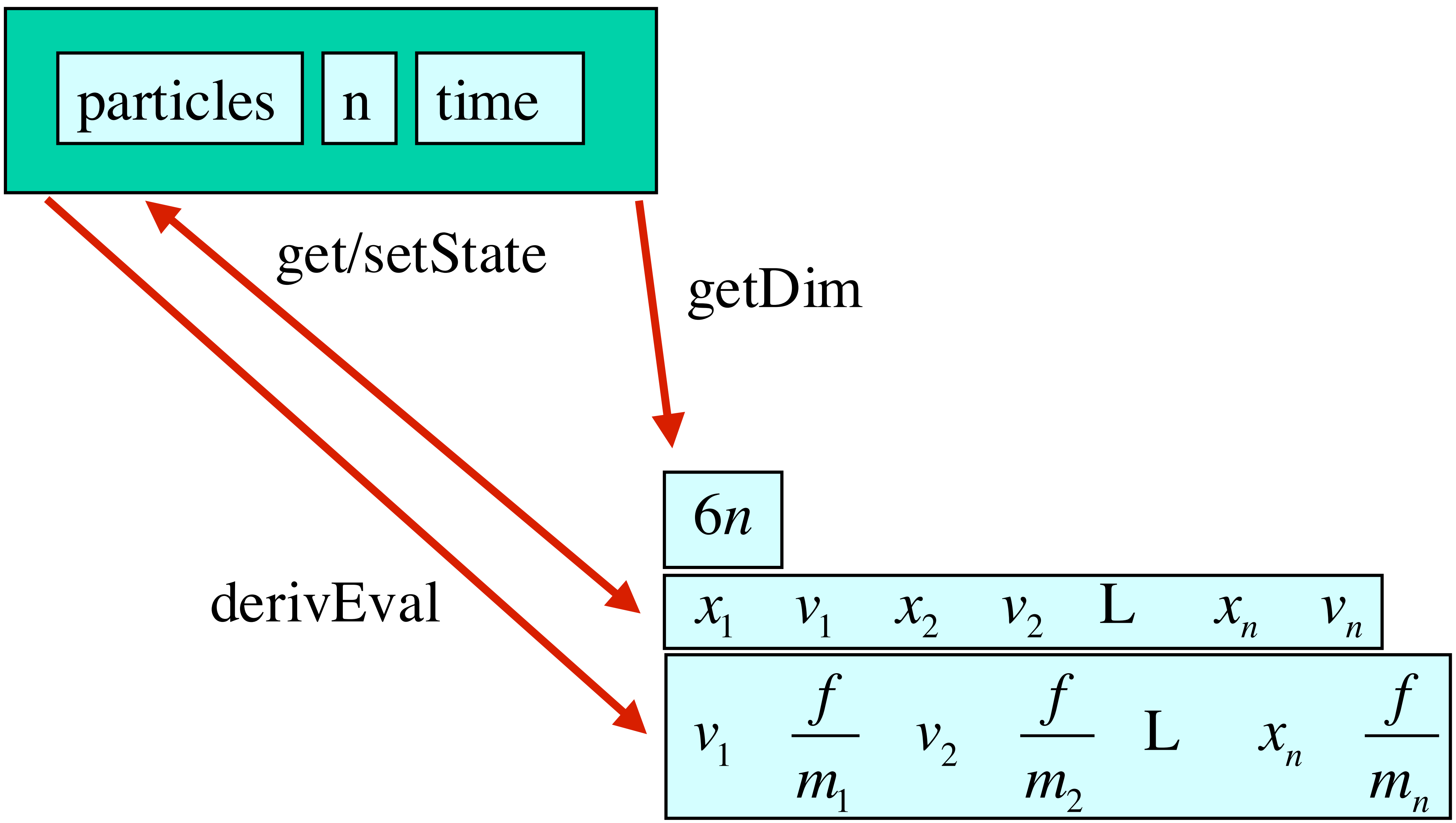
Solver interface



Particle systems



Solver interface



Differential equation solver

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ f/m \end{bmatrix}$$

Euler method: $x(t+h) = x(t) + h \cdot \dot{x}(t)$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta t \cdot \dot{x}$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \Delta t \cdot \dot{v}$$

Gets very unstable for large Δt

Higher order solvers perform better: (e.g. Runge-Kutta)

derivEval loop

1. Clear forces
   – Loop over particles, zero force accumulators
2. Calculate forces
   – Sum all forces into accumulators
3. Gather
   – Loop over particles, copying v and f/m into destination array

Forces

- Constant (gravity)
- Position/time dependent (force fields)
- Velocity-dependent (drag)
- N-ary (springs)

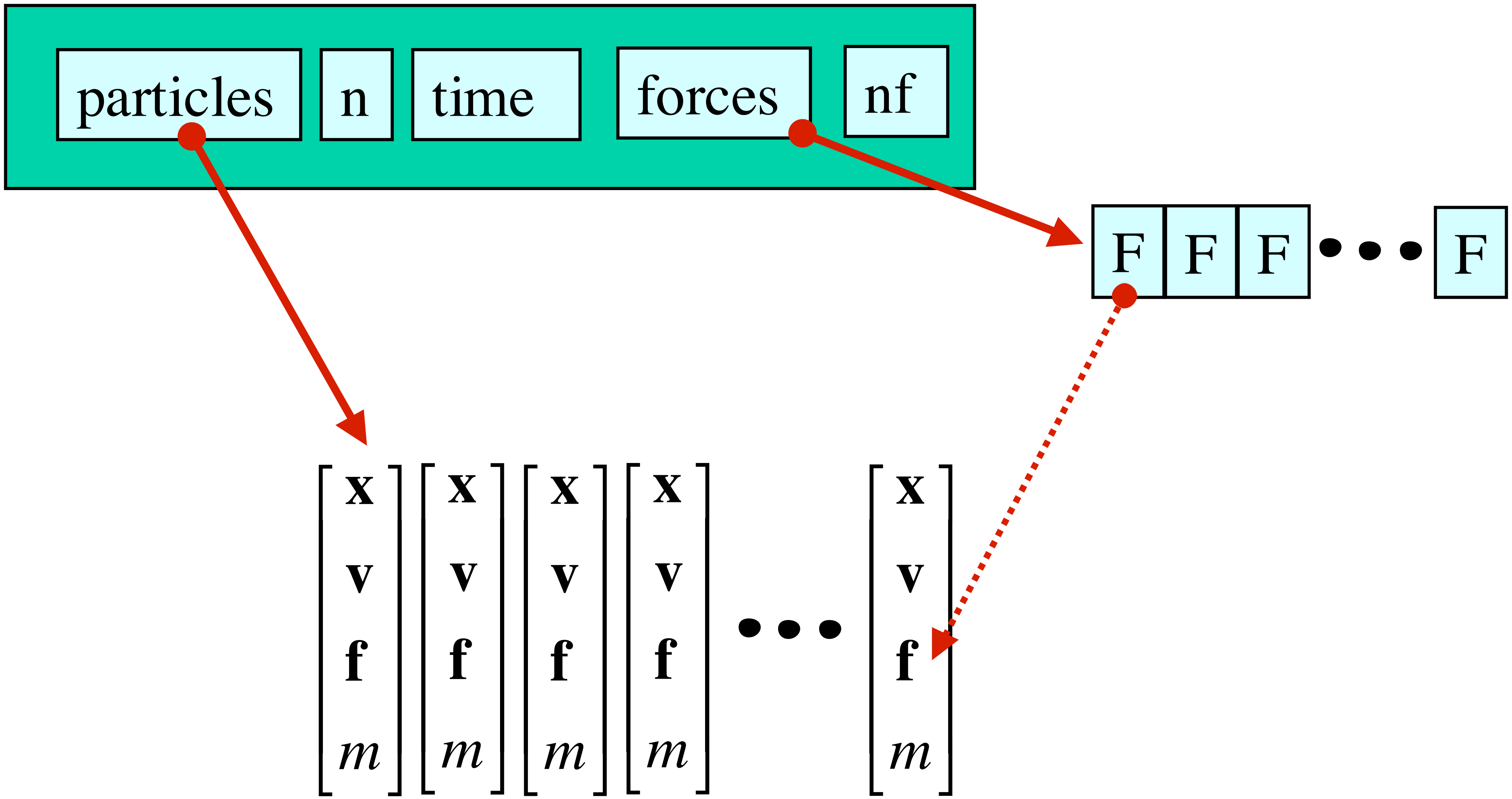
Force structures

Force objects are black boxes that point to the particles they influence, and add in their contribution into the force accumulator.

Global force calculation:

- Loop, invoking force objects

Particle systems with forces



Gravity

Force law:

$\mathbf{f}_{grav} = m\mathbf{G}$

```
p->f += p->m * F->G
```

Viscous drag

Force law:

$$\mathbf{f}_{drag} = -k_{drag}\mathbf{v}$$

```
p->f -= F->k * p->v
```

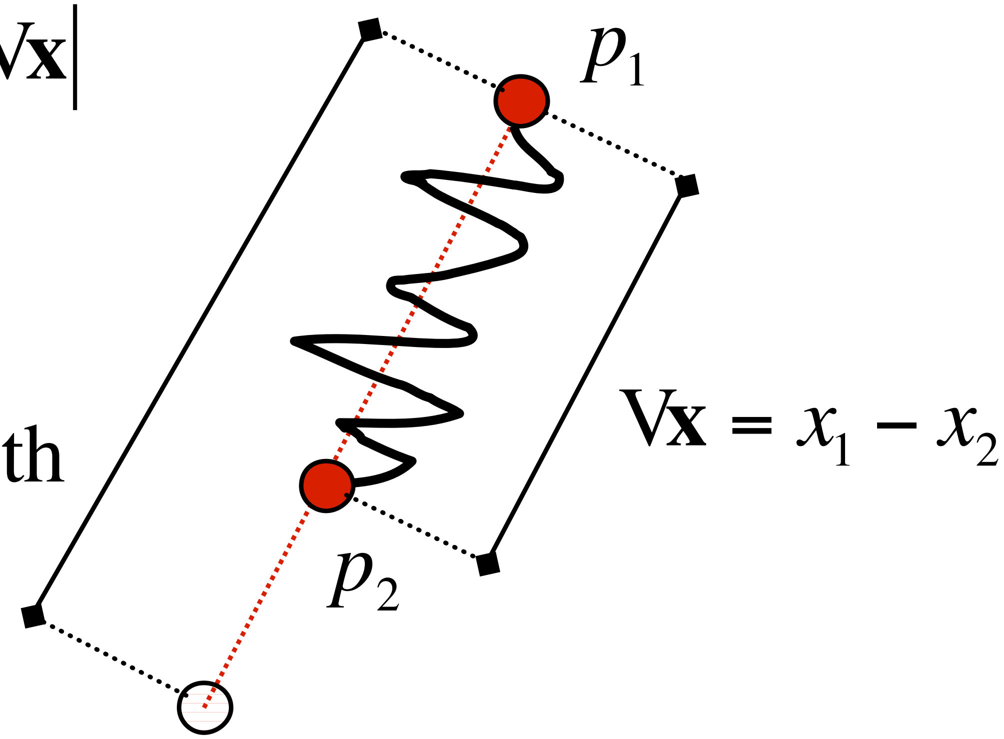
Damped spring

Force law:

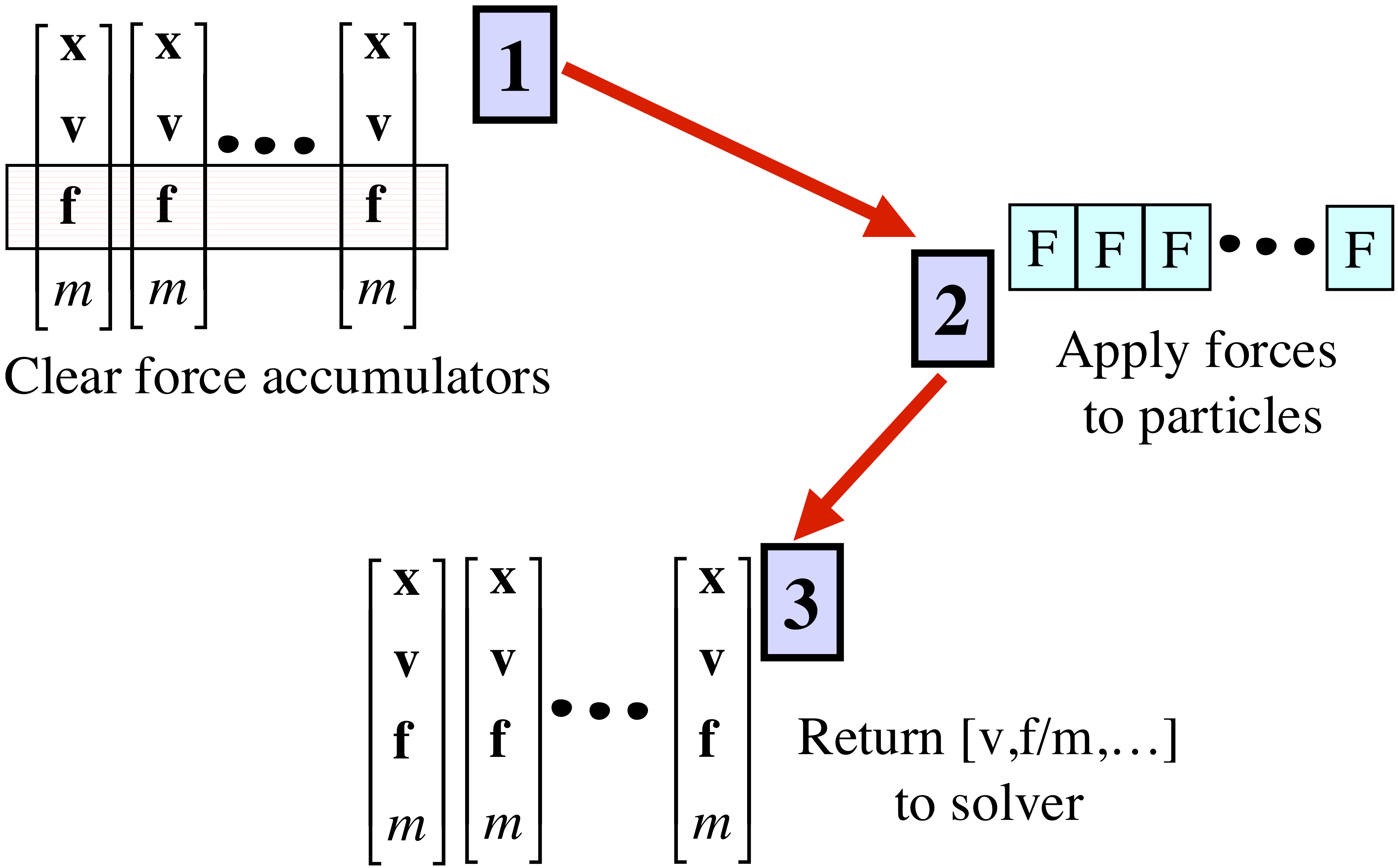
$$\mathbf{f}_1 = -\left[k_s(|V\mathbf{x}| - \mathbf{r}) + k_d\left(\frac{V\mathbf{v}V\mathbf{x}}{|V\mathbf{x}|}\right)\right]\frac{V\mathbf{x}}{|V\mathbf{x}|}$$

$$\mathbf{f}_2 = -\mathbf{f}_1$$

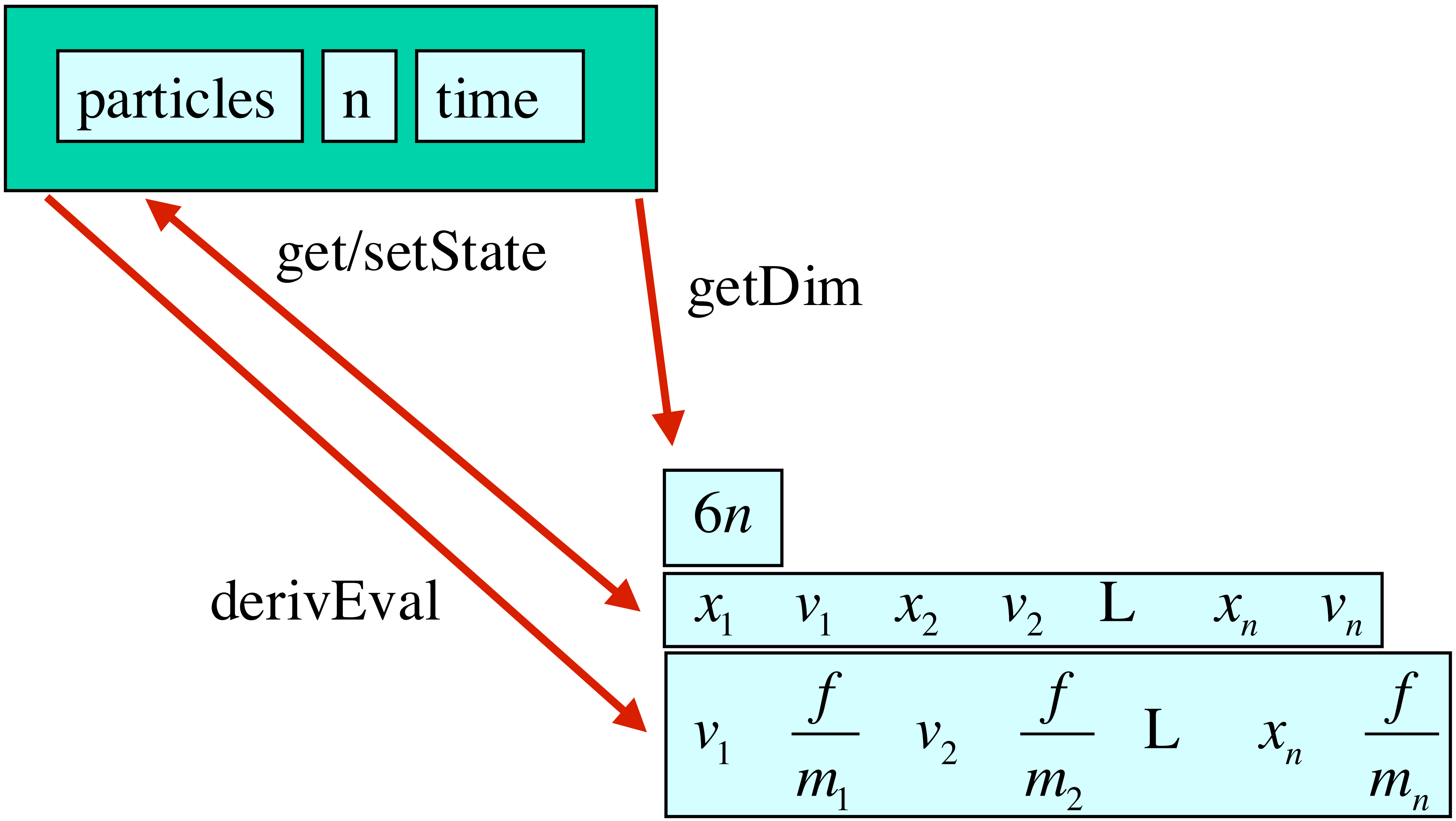
$\mathbf{r}$ = rest length



derivEval Loop



Solver interface



Differential equation solver

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ f/m \end{bmatrix}$$

Euler method:

$$\begin{bmatrix} x_1^{i+1} \\ v_1^{i+1} \\ \vdots \\ x_n^{i+1} \\ v_n^{i+1} \end{bmatrix} = \begin{bmatrix} x_1^{i} \\ v_1^{i} \\ \vdots \\ x_n^{i} \\ v_n^{i} \end{bmatrix} + \Delta t \begin{bmatrix} v_1^{i} \\ f_1^{i}/m_1 \\ \vdots \\ v_n^{i} \\ f_n^{i}/m_n \end{bmatrix}$$

Differential equation solver

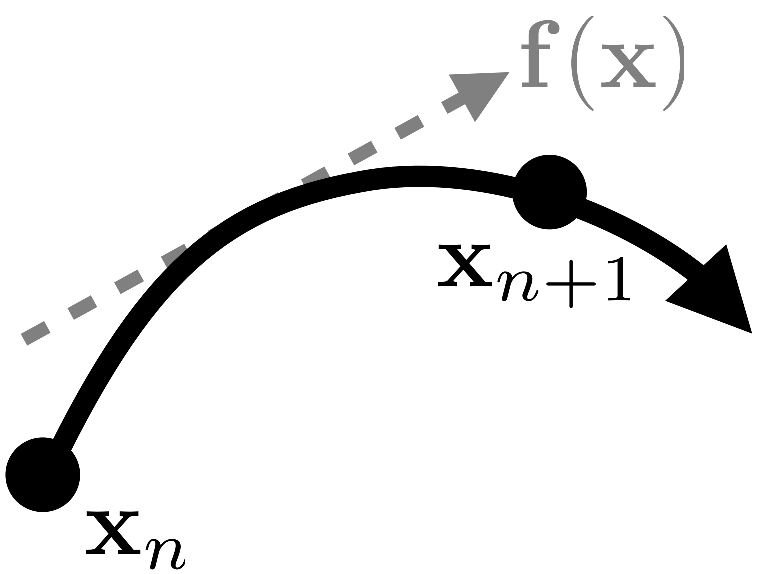
$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ f/m \end{bmatrix}$$

In general:

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

- $\mathbf{x}$ could be a single particle
- $\mathbf{x}$ could be a whole particle system
- ...*or more!*

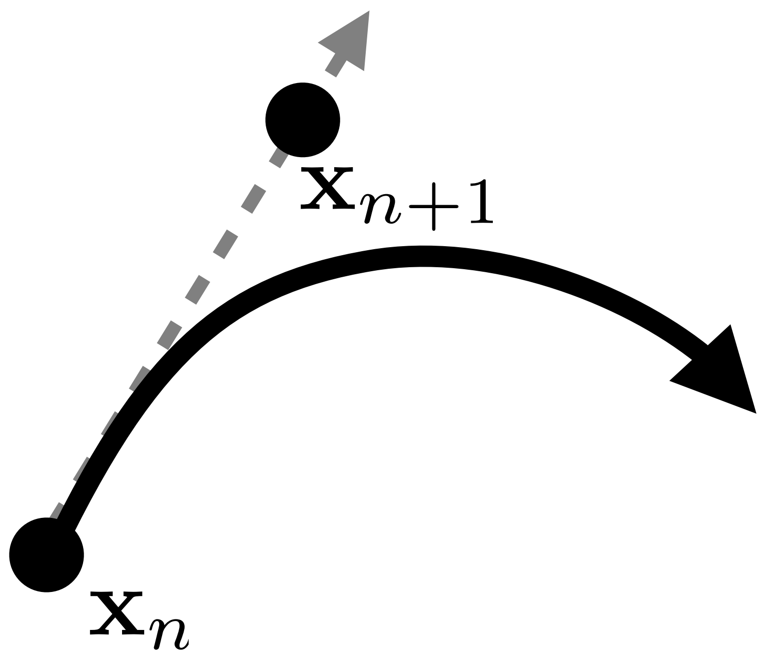
$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$



$$\mathbf{x}_n = \mathbf{x}(n\Delta t)$$

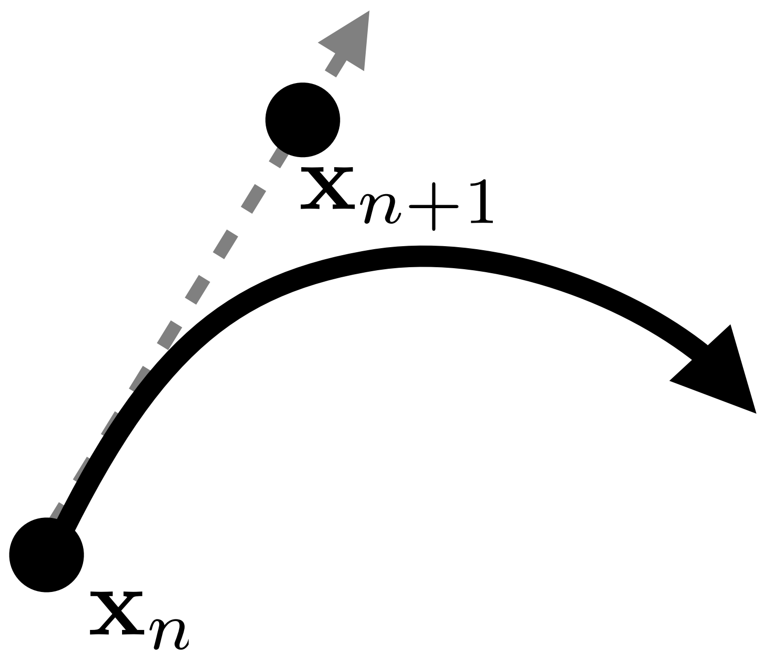
- Explicit
- Implicit
- 2nd-Order Runge-Kutta
- 4th-Order Runge-Kutta
- Symplectic

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



- Explicit
- Implicit
- 2nd-Order Runge-Kutta
- 4th-Order Runge-Kutta
- Symplectic

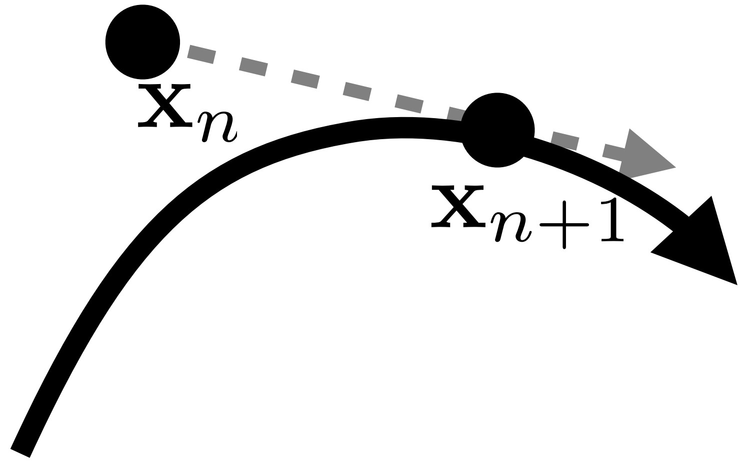
$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$



- Explicit

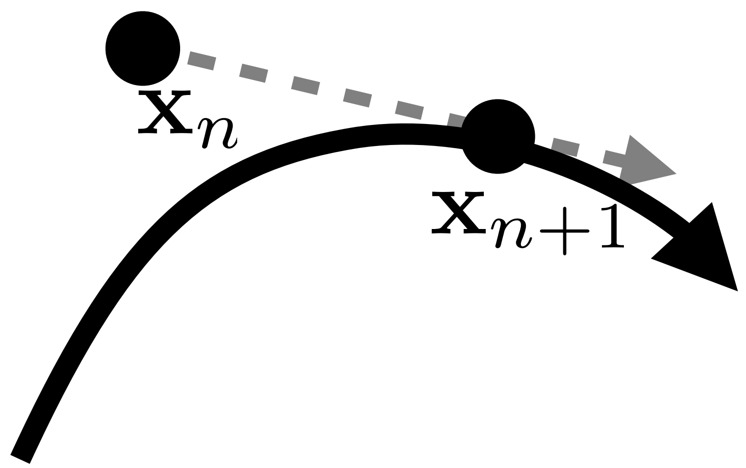
$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t \mathbf{f}(\mathbf{x}_n)$$

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



- Explicit
- Implicit
- 2nd-Order Runge-Kutta
- 4th-Order Runge-Kutta
- Symplectic

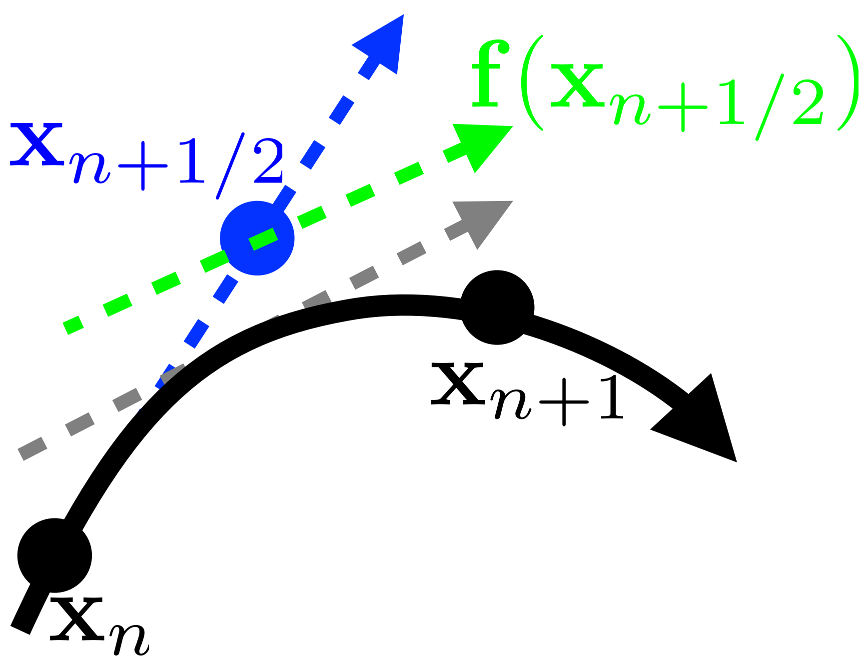
$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$



- Implicit

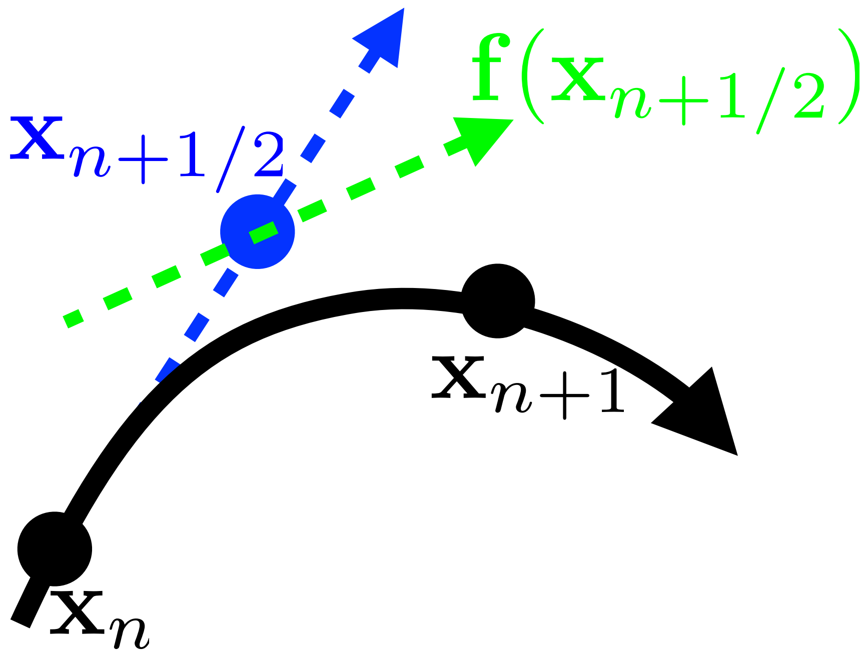
$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t \mathbf{f}(\mathbf{x}_{n+1})$$

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



- Explicit
- Implicit
- 2nd-Order Runge-Kutta
- 4th-Order Runge-Kutta
- Symplectic

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



- 2nd-Order Runge-Kutta

a. Compute an Euler step

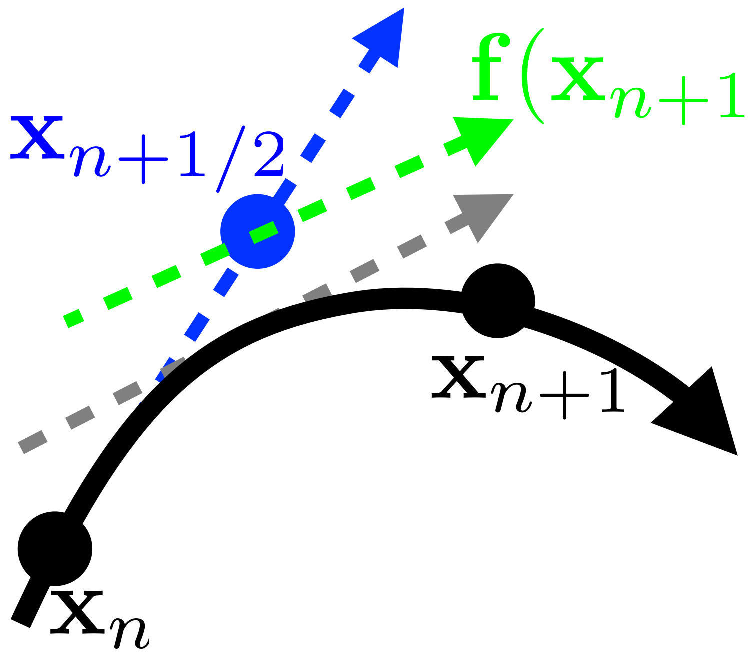
$$\Delta \mathbf{x} = \Delta t\, \mathbf{f}(\mathbf{x}, t)$$

b. Evaluate f at the midpoint

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

c. Take a step using the midpoint value

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



- Explicit
- Implicit
- 2nd-Order Runge-Kutta
- 4th-Order Runge-Kutta
- Symplectic

4th-Order Runge-Kutta

$$k_1 = hf(x_0, t_0)$$

$$k_2 = hf(x_0 + \frac{k_1}{2}, t_0 + \frac{h}{2})$$

$$k_3 = hf(x_0 + \frac{k_2}{2}, t_0 + \frac{h}{2})$$

$$k_4 = hf(x_0 + k_3, t_0 + h)$$

$$x(t_0 + h) = x_0 + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 + O(h^5)$$

q-Stage Runge-Kutta

General Form:

$$x(t_0 + h) = x_0 + h \sum_{i=1}^{q} w_i k_i$$

where:

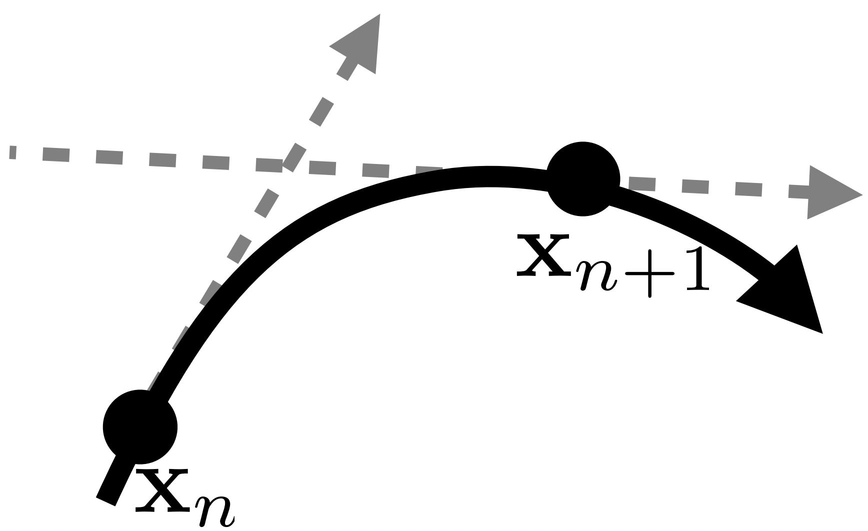
$$k_i = f\left(x_0 + h \sum_{j=1}^{i-1} \beta_{ij} k_j \right)$$

Find the constant that ensure accuracty $O(h^n)$.

Order vs. Stages

Order	1	2	3	4	5	6	7	8
Stages	1	2	3	4	6	7	9	11

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



- Symplectic
  - 2D system:

    $$\mathbf{x} = \left[\begin{array}{c} x \\ y \end{array} \right]$$

  - explicit in *x*
  - implicit in *y*

Question:

- *how could we generalize implicit integration to nonlinear functions?*

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t \mathbf{f}(\mathbf{x}_{n+1})$$

- ***bonus:*** can you construct a *time-reversible* integrator?
    - *where would you define the derivative?*