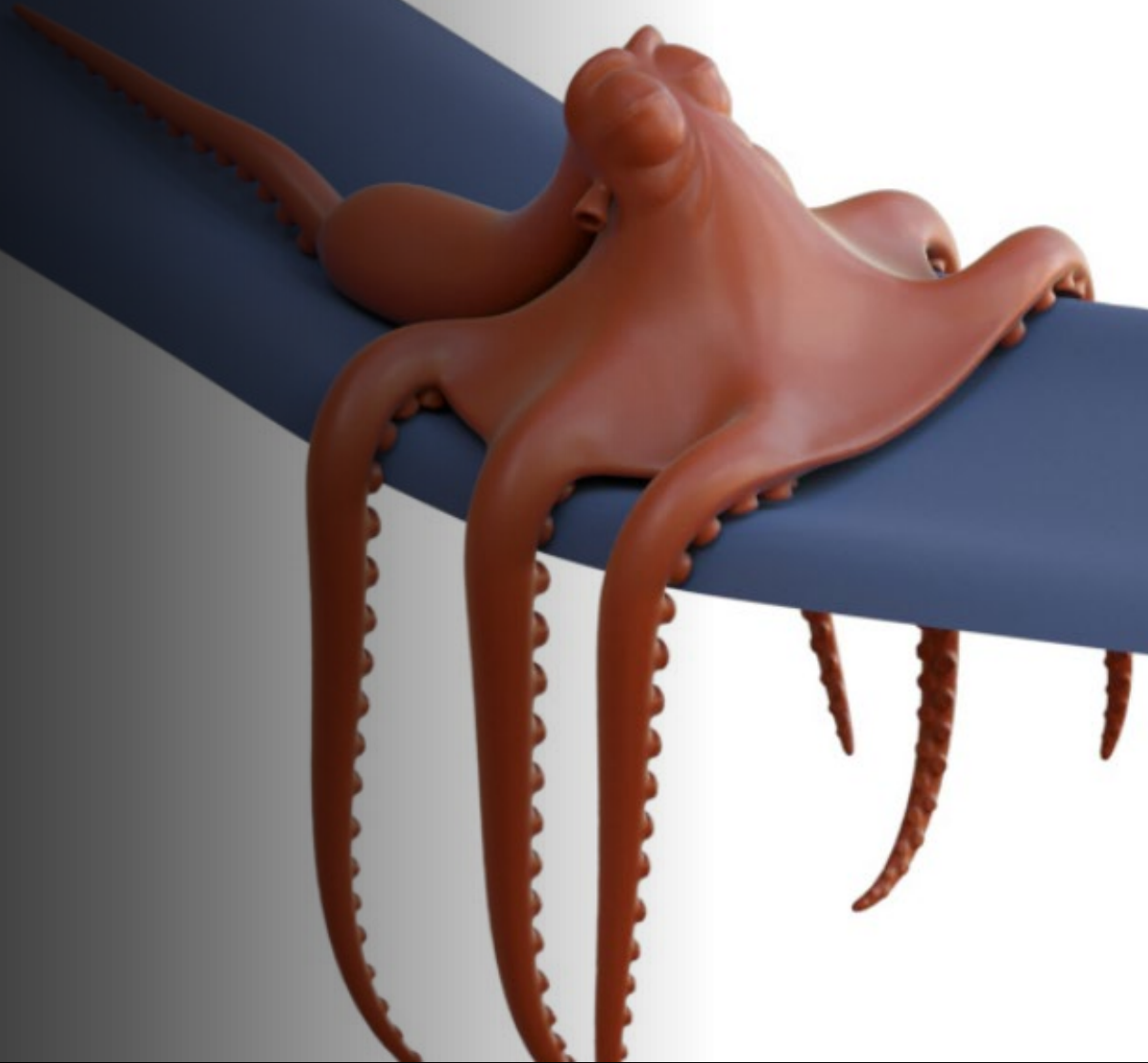


# Deformable Simulation

---

Spatial and Temporal Discretization



# Outline today

- The temporal integration
  - Explicit
  - Implicit
  - Symplectic
- The spatial integration
  - Mass-Spring system
  - The finite element method

# Equations of motion (general cases)

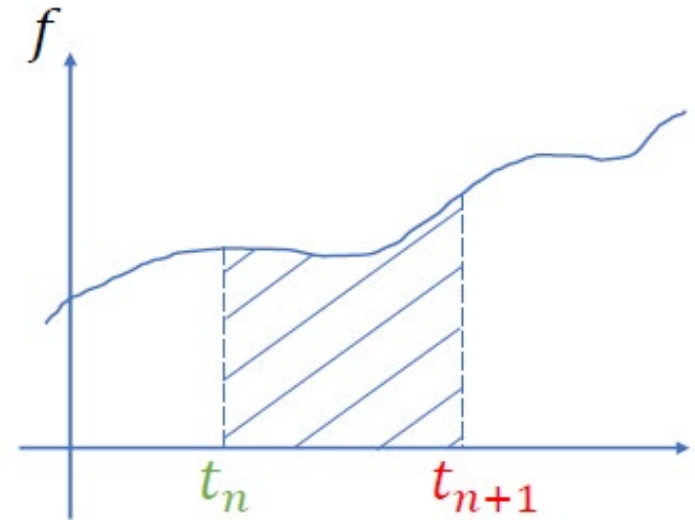
$$M\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

$$\dot{\mathbf{x}} = \mathbf{v} \quad \dot{\mathbf{v}} = \mathbf{a} = M^{-1}\mathbf{f}$$

$$\mathbf{x}(t_n + h) = \mathbf{x}(t_n) + \int_0^h \mathbf{v}(t_n + t) dt$$

$$\mathbf{v}(t_n + h) = \mathbf{v}(t_n) + \int_0^h M^{-1}\mathbf{f}(t_n + t) dt$$

$h = t_{n+1} - t_n$  is the time-step size



We don't know how to integrate this quantity  
We don't know anything after  $t_n$

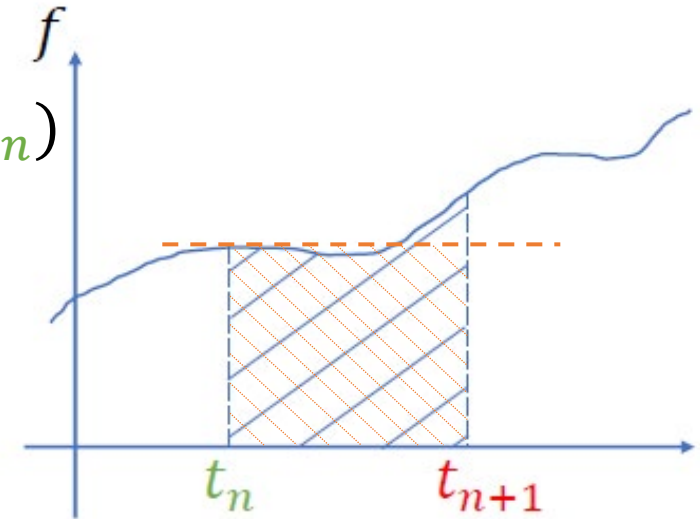
# Temporal integration (Explicit)

$$\mathbf{x}_{n+1} = \mathbf{x}(t_n) + \int_0^h \mathbf{v}(t_n + t) dt \approx \mathbf{x}_n + h\mathbf{v}_n$$

$$\mathbf{v}_{n+1} = \mathbf{v}(t_n) + \int_0^h M^{-1}f(t_n + t) dt = \mathbf{v}_n + hM^{-1}f(\mathbf{x}_n)$$

**Accuracy:** 1<sup>st</sup>-order accurate

$$\begin{aligned} \int_0^h \mathbf{v}(t_n + t) dt &= h\mathbf{v}(t_n + 0) + \frac{h^2}{2} \mathbf{v}'(t_n + 0) + \dots \\ &= h\mathbf{v}_n + O(h^2) \end{aligned}$$



# Temporal integration (Explicit)

Stability:

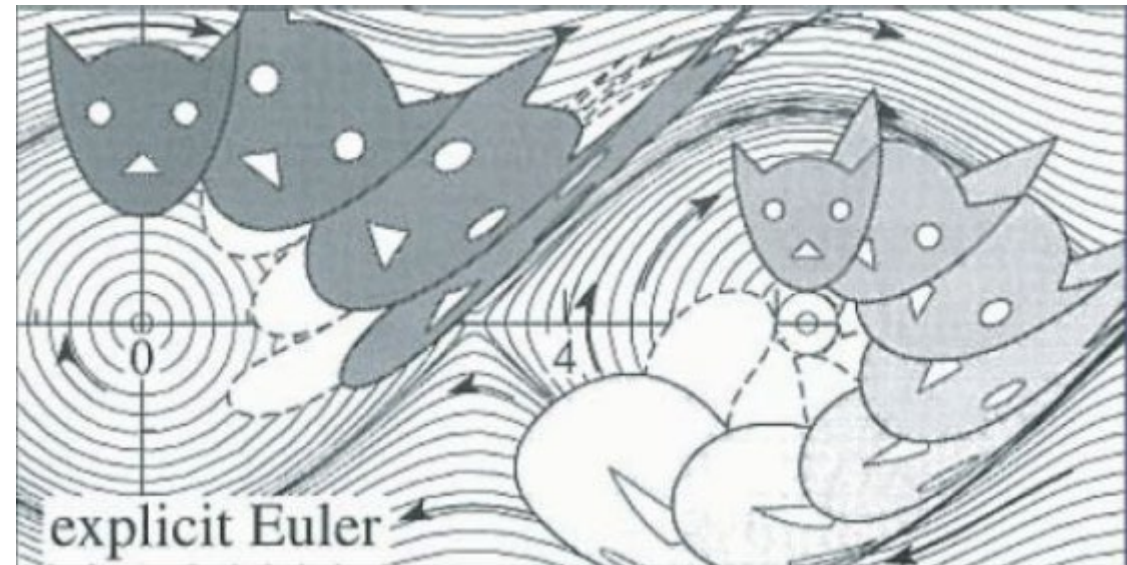
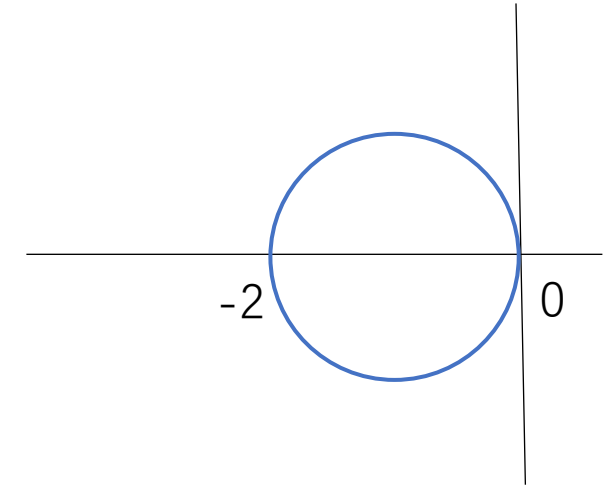
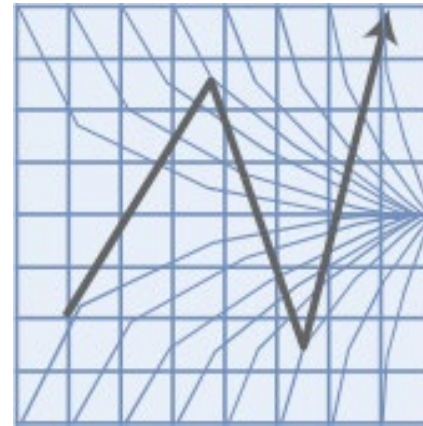
$$\frac{d}{dt}x = \lambda x$$

$$x_{n+1} = x_n + h\lambda x_n$$

$$x_n = (1 + h\lambda)^n x_0$$

Condition of stability:

$$|1 + h\lambda| \leq 1, \quad h \leq \frac{2}{|\lambda|}$$



Explicit Euler is **extremely fast**, but it will also increase the system energy gradually. **Not recommended in almost all situations.**

# Temporal integration (Implicit)

$$\mathbf{x}_{n+1} = \mathbf{x}(t_n) + \int_0^h \mathbf{v}(t_n + t) dt \approx \mathbf{x}_n + h\mathbf{v}_{n+1}$$

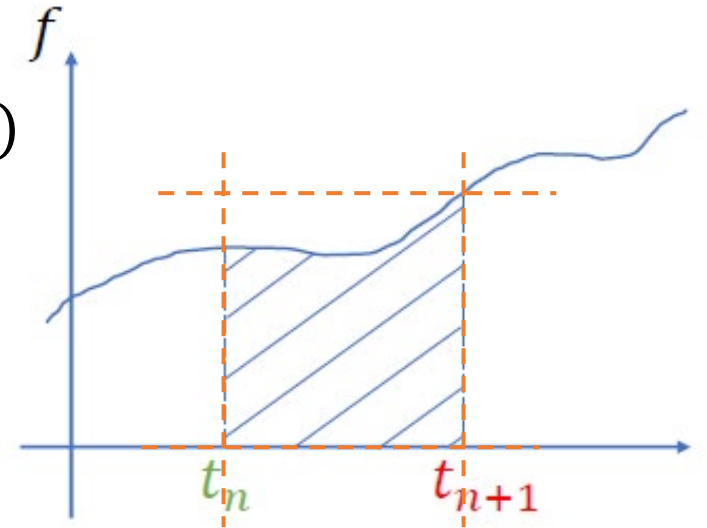
$$\mathbf{v}_{n+1} = \mathbf{v}(t_n) + \int_0^h M^{-1}f(t_n + t) dt \approx \mathbf{v}_n + hM^{-1}f(\mathbf{x}_{n+1})$$

$$g(\mathbf{x}) = \frac{1}{2h^2} (\mathbf{x}_{n+1} - \mathbf{y}_n)^T M (\mathbf{x}_{n+1} - \mathbf{y}_n) + E(\mathbf{x}_{n+1})$$

**Accuracy:** 1<sup>st</sup>-order accurate

$$\begin{aligned} \int_0^h \mathbf{v}(t_n + t) dt &= h\mathbf{v}(t_n + h) - \frac{h^2}{2} \mathbf{v}'(t_n + h) - \dots \\ &= h\mathbf{v}_{n+1} + O(h^2) \end{aligned}$$

Expand where  $\Delta t = h$



# Temporal integration (Implicit)

Stability:

$$\frac{d}{dt}x = \lambda x$$

$$x_{n+1} = x_n + h\lambda x_{n+1}$$

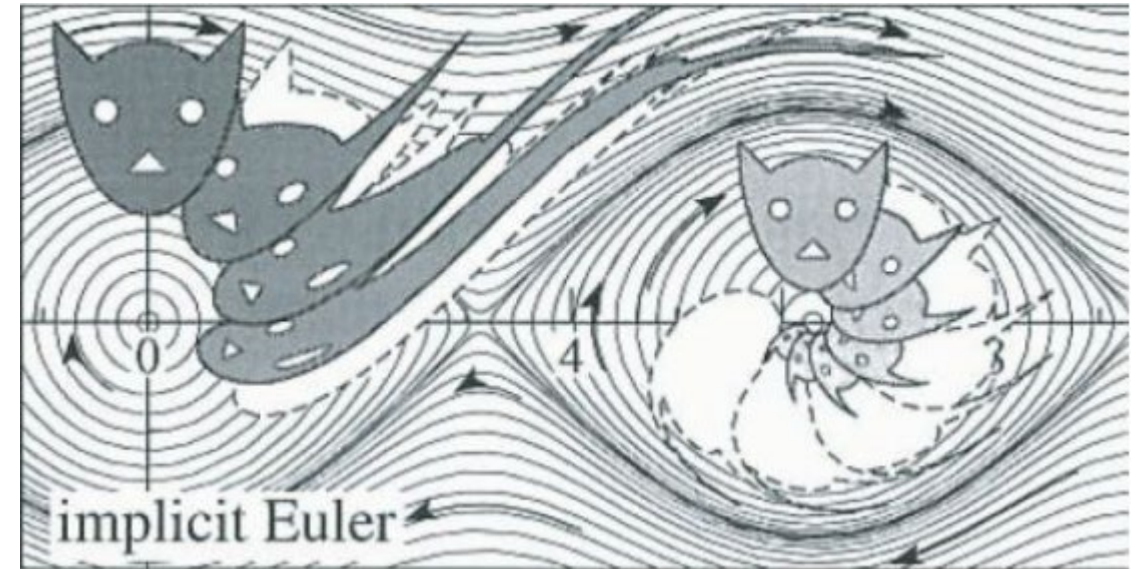
$$x_n = (1 - h\lambda)^{-n} x_0$$

Condition of stability:

$$|1 - h\lambda| \geq 1$$

Implicit Euler integrals are unconditionally stable!

Implicit Euler is often **expensive** due to the nonlinear optimization, it is often stable for **large time steps** and is widely used in performance centric applications.  
(game / design / animation)



# Temporal integration (Symplectic)

$$\begin{aligned} \mathbf{v}_{n+1} &= \mathbf{v}(t_n) + \int_0^h M^{-1} f(t_n + t) dt \approx \mathbf{v}_n + h M^{-1} f(\mathbf{x}_n) \\ \mathbf{x}_{n+1} &= \mathbf{x}(t_n) + \int_0^h \mathbf{v}(t_n + t) dt \approx \mathbf{x}_n + h \mathbf{v}_{n+1} \end{aligned}$$

**Accuracy:** 2<sup>nd</sup>-order accurate

$$\mathbf{v}_{n+0.5} \approx \mathbf{v}_{n-0.5} + h M^{-1} f(\mathbf{x}_n)$$

$$\mathbf{x}_{n+1} \approx \mathbf{x}_n + h \mathbf{v}_{n+0.5}$$

Midpoint

$$\begin{aligned} \int_0^h \mathbf{v}(t_n + t) dt &= \int_0^{h/2} \mathbf{v}(t_n + t) dt + \int_{h/2}^h \mathbf{v}(t_n + t) dt \\ &= \frac{h}{2} \mathbf{v}(h/2) - \frac{h^2}{2} \mathbf{v}'\left(\frac{h}{2}\right) + O(h^3) + \\ &\quad \frac{h}{2} \mathbf{v}(h/2) + \frac{h^2}{2} \mathbf{v}'\left(\frac{h}{2}\right) + O(h^3) = h \mathbf{v}_{n+0.5} + O(h^3) \end{aligned}$$



# Temporal integration (Symplectic)

Stability:

$$\frac{d}{dt}x = \lambda x$$

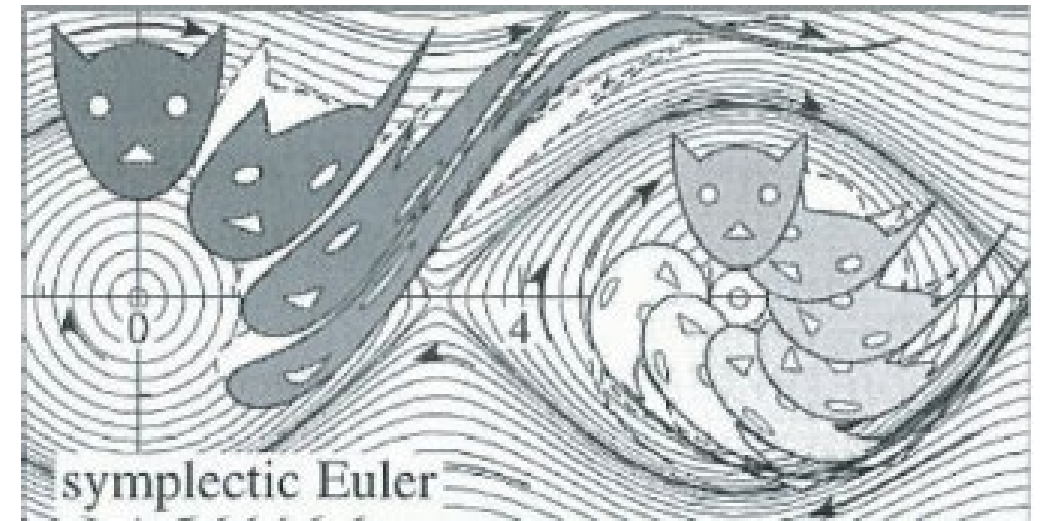
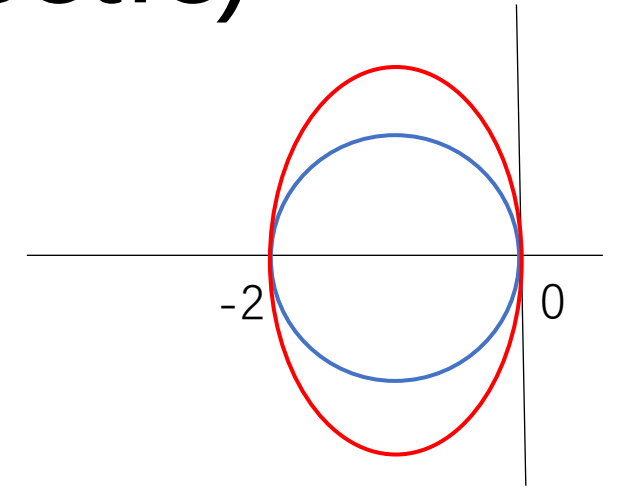
$$\begin{aligned}x_{n+1} &= x_n + h\lambda x_{n+0.5} = x_n + h\lambda(x_n + 0.5h\lambda x_n) \\ &= x_n(1 + h\lambda + 0.5(h\lambda)^2)\end{aligned}$$

$$x_n = (1 + h\lambda + 0.5(h\lambda)^2)^n x_0$$

Condition of stability:

$$h\lambda = a + ib, \quad |1 + h\lambda + 0.5(h\lambda)^2| \leq 1$$

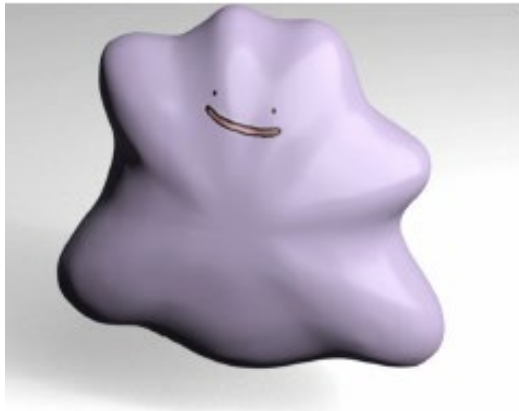
$$\left| \begin{bmatrix} 1 + a + \frac{a^2 - b^2}{2} \\ b + ab \end{bmatrix} \right| \leq 1$$



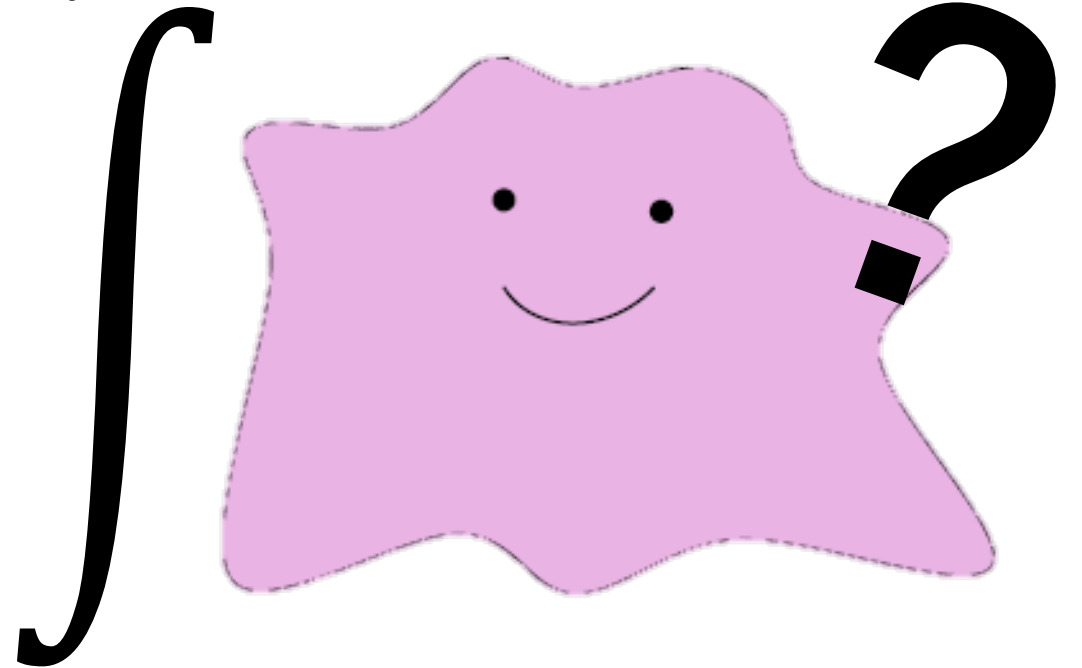
Symplectic Euler is as **fast** as forward Euler. It has been widely used in **accuracy centric applications** (astronomy simulation / molecular dynamics etc).

# Spatial integration

- For conservative forces (as most of the elastic forces are), the energy definition is all we need for their simulations.
- A deformable object is a **continuum** body



$E =$



How to describe the **deformation**?  
How to describe the **elastic energy**?

# Mass-spring system

- Aggregate the volume mass to the vertices
- Link the mass vertices with springs

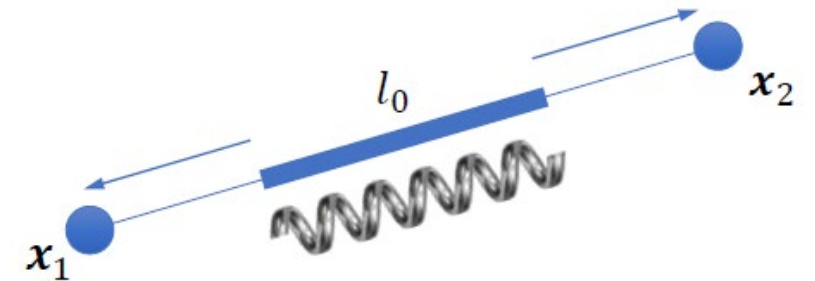
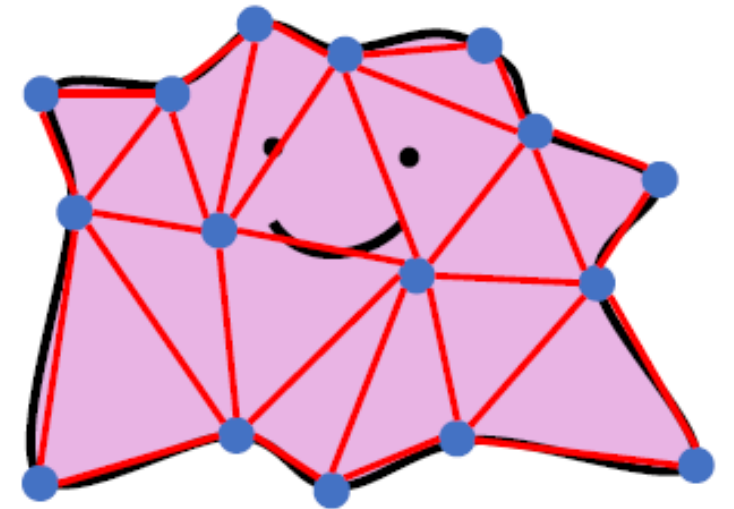
- How to describe the deformation?

$$\text{"Deformation"} = l - l_0$$

- How to describe the elastic energy?

$$\text{Hooke's Law: } E(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{2}k(l - l_0)^2$$

$$E = \int \text{[Diagram of a mesh]} = \sum \text{[Diagram of a spring]}$$



# Mass-spring system

- Mass-spring systems are particularly useful in:

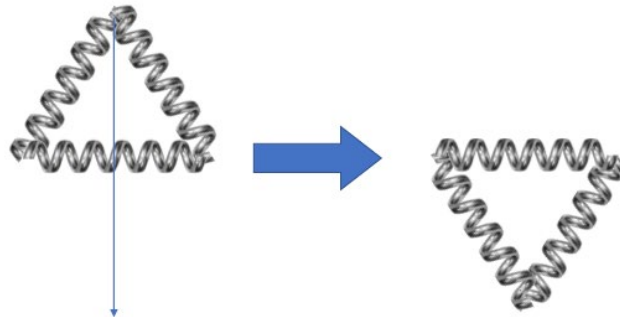


Cloth Sim  
Dinev et al. 2018



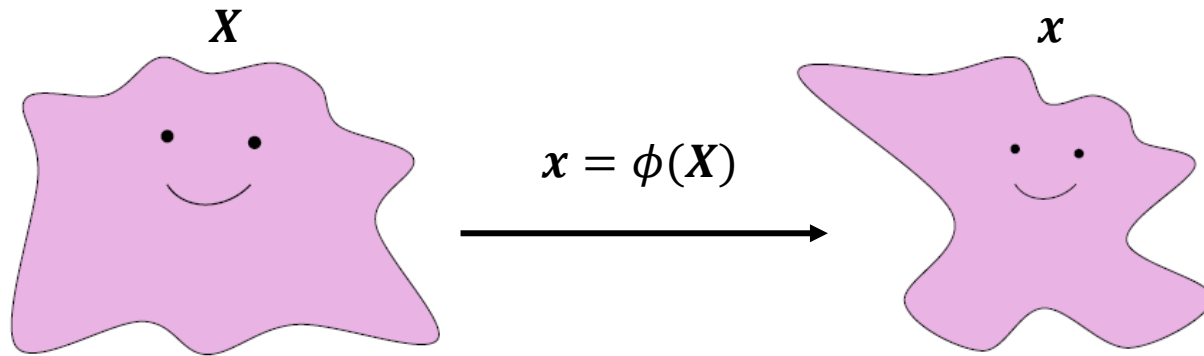
Hair Sim  
Selle et al. 2018

- Mass-spring systems are NOT the best choices when simulating continuum area/volume



# Linear finite element method (FEM)

- How to describe the deformation?



$$\phi(\mathbf{X}) \approx \underbrace{\frac{\partial \phi}{\partial \mathbf{X}}}_{\mathbf{F}} \mathbf{X} + \underbrace{\left( \phi(\mathbf{X}^*) - \frac{\partial \phi}{\partial \mathbf{X}} \mathbf{X}^* \right)}_{\mathbf{t}}$$
$$\mathbf{x} \approx \mathbf{F}\mathbf{X} + \mathbf{t}$$

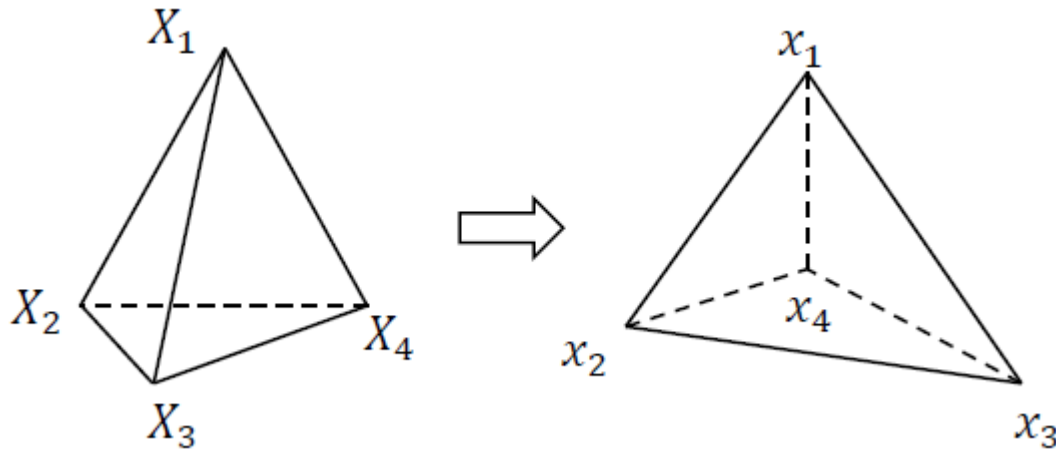
- Translation:  $\phi(\mathbf{X}) = \mathbf{X} + \mathbf{t}$
- Rotation:  $\phi(\mathbf{X}) = \mathbf{R}\mathbf{X}$
- Scaling:  $\phi(\mathbf{X}) = \mathbf{S}\mathbf{X}$

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{F} = \frac{\partial \phi}{\partial \mathbf{X}} = \mathbf{I}$$
$$\mathbf{F} = \frac{\partial \phi}{\partial \mathbf{X}} = \mathbf{R}$$
$$\mathbf{F} = \frac{\partial \phi}{\partial \mathbf{X}} = \mathbf{S}$$

# Linear finite element method (FEM)

- How to describe the elastic energy?
- Define:  $\Psi(\mathbf{x}) = \Psi(F)$  is an energy density function at  $\mathbf{x} = \phi(\mathbf{X})$ , so  $E = \sum w_e \Psi(F_e)$
- Is it appropriate to do so?



$$\begin{aligned} x_1 &= FX_1 + t \\ x_2 &= FX_2 + t \\ x_3 &= FX_3 + t \\ x_4 &= FX_4 + t \end{aligned}$$

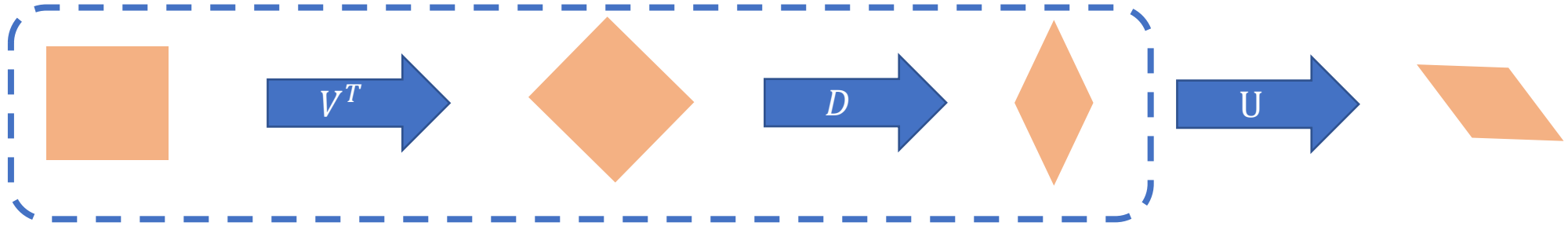
$$F = D_s D_m^{-1}$$

$$\underbrace{[x_1 - x_4 \quad x_2 - x_4 \quad x_3 - x_4]}_{D_s} = F \underbrace{[X_1 - X_4 \quad X_2 - X_4 \quad X_3 - X_4]}_{D_m}$$

$F$  is related to deformation,  
But it contains **rotation**.

# Linear finite element method (FEM)

Ideally, we need a tensor to describe shape deformation only. Recall that SVD gives  $F = UDV^T$ , where only  $V^T$  and  $D$  are relevant to deformation.



So we get rid of  $U$  as:

$$\epsilon_G(F) = \frac{1}{2}(F^T F - I) = \frac{1}{2}(VD^2V^T - I) \text{ Green strain}$$

Strain: Descriptor of severity of deformation

- $\epsilon(I) = 0$
- $\epsilon(F) = \epsilon(RF)$

# Linear finite element method (FEM)

- St. Venant-Kirchhoff model (StVK):

- Strain:  $\epsilon_{stvk}(F) = \frac{1}{2}(F^T F - I)$
- Energy density:  $\Psi(\epsilon_{stvk}) = \mu \|\epsilon_{stvk}(F)\|_F^2 + \frac{\lambda}{2} \text{tr}(\epsilon_{stvk}(F))^2$
- $P = \frac{\partial \Psi}{\partial F} = F[2\mu \epsilon_{stvk} + \lambda \text{tr}(\epsilon_{stvk})I]$  **1<sup>st</sup>-Piola-Kirchhoff stress tensor**

- Co-rotated linear model:

- Strain:  $\epsilon_c(F) = S - I$ , where  $F = RS$
- Energy density:  $\Psi(\epsilon_c) = \mu \|\epsilon_c\|_F^2 + \frac{\lambda}{2} \text{tr}(R^T F - I)^2$
- $P = \frac{\partial \Psi}{\partial F} = R[2\mu \epsilon_c + \lambda \text{tr}(\epsilon_c)I]$

Eventually we will need the gradient of  $\Psi$  to run simulations.

$$\frac{\partial \Psi}{\partial \mathbf{x}} = \frac{\partial F}{\partial \mathbf{x}} : \frac{\partial \Psi}{\partial F}$$

A  $(12 \times 1) \times (3 \times 3)$  tensor

A  $(3 \times 3)$  tensor (matrix)

A  $(12 \times 1)$  tensor (vector)

Note (matrix contraction):  $B:A = A:B = \sum_{i,j} A_{ij}B_{ij} = \sqrt{\text{tr}(A^T B)}$



# Linear finite element method (FEM)

- Chain rule in detail:  $\frac{\partial \Psi}{\partial x_j^i} = \frac{\partial F}{\partial x_j^i} : P$

- Let's compute  $\frac{\partial F}{\partial x_j^i}$  first:

Since  $F = D_s D_m^{-1}$ , so  $\frac{\partial F}{\partial x_j^i} = \frac{\partial D_s}{\partial x_j^i} D_m^{-1}$

$$\frac{\partial D_s}{\partial x_j^i} = \delta_i \delta_j^T, \text{ for } j = 1, 2, 3$$

$$\text{Thus: } \frac{\partial F}{\partial x_j^i} = \delta_i \delta_j^T D_m^{-1}$$

- $\frac{\partial F}{\partial x_j^i} : P = \delta_i \delta_j^T D_m^{-1} : P = \text{tr}(D_m^{-T} \delta_j \delta_i^T P)$   
 $= \text{tr}(\delta_i^T P D_m^{-T} \delta_j) = \delta_i^T P D_m^{-T} \delta_j = [P D_m^{-T}]_{ij}$

$$D_s = \begin{bmatrix} x_1 - x_4 & x_2 - x_4 & x_3 - x_4 \\ x_1^1 - x_4^1 & x_2^1 - x_4^1 & x_3^1 - x_4^1 \\ x_1^2 - x_4^2 & x_2^2 - x_4^2 & x_3^2 - x_4^2 \\ x_1^3 - x_4^3 & x_2^3 - x_4^3 & x_3^3 - x_4^3 \end{bmatrix}$$

$$\frac{\partial D_s}{\partial x_1^2} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T = \delta_2 \delta_1^T$$

- The energy gradient:  $\frac{\partial E_e}{\partial x_{e,j}} = w_e \frac{\partial \Psi_e}{\partial x_{e,j}}$  = the j-th col of  $P D_m^{-T}$  for j=1, 2, 3 in the e-th tetrahedron
  - For j = 4:  $\frac{\partial E_e}{\partial x_{e,4}} = w_e \frac{\partial \Psi_e}{\partial x_{e,4}} = -w_e \sum_{j=1}^3 \frac{\partial \Psi_e}{\partial x_{e,j}}$

The sum of all internal forces in a tetrahedron should be **zero**

Note (matrix contraction):  $B : A = A : B = \sum_{i,j} A_{ij} B_{ij} = \sqrt{\text{tr}(A^T B)}$

# Linear finite element method (FEM)

- Now we get the internal (elastic) force in a tetrahedron, while a particle often exists in multiple tetrahedra.
- At each time step, execute:
  - $\mathbf{f} \leftarrow \mathbf{0}$
  - For e-th tetrahedron:
    - $D_s = [x_1 - x_4 \quad x_2 - x_4 \quad x_3 - x_4]$
    - $F = D_s D_m$
    - $P \leftarrow P(F)$
    - $H \leftarrow -w_e P D_m^{-T}$ ,  $H = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3]$
    - $\mathbf{f}_1 += \mathbf{h}_1$ ,  $\mathbf{f}_2 += \mathbf{h}_2$ ,  $\mathbf{f}_3 += \mathbf{h}_3$
    - $\mathbf{f}_4 += -(\mathbf{h}_1 + \mathbf{h}_2 + \mathbf{h}_3)$
  - $\mathbf{v}_{new} = \mathbf{v} + hM^{-1}\mathbf{f}$
  - $\mathbf{x}_{new} = \mathbf{x} + h\mathbf{v}$



# Reference and Futher Readings

- Finite Element Method, Part I [SIGGRAPH 2012 Course] [[Link](#)]
- Dynamic Deformables: Implementation and Production Practicalities [SIGGRAH 2020 Course] [[Link](#)]
- TaichiCourse 01: Lecture 08 [[Link](#)]
- GAMES Course 103 – Lecture 7: Other Constrained Methods and Linear Finite Element Method I [[Link](#)]

All relevant files (ppt, pdf, image, code, video):

[https://github.com/LILKOTYO/Lab-Presentation/tree/master/20220725\\_Spatial\\_and\\_Temporal\\_Discretization](https://github.com/LILKOTYO/Lab-Presentation/tree/master/20220725_Spatial_and_Temporal_Discretization)