Elastic models

Spring structure
Numerical solution of ODE
Cloth simulation

Numerical solution of ODE

General formulation

Consider relation given a system of **first order** differential equation

Mechanical systems are often expressed as

- *single equation of second order in* p
- *system of first order in* $u = (p, v)$

In general, we can write

$$u'(t) = \mathcal{F}(u(t), t)$$

If $\mathcal{F}$ is an affine function in u

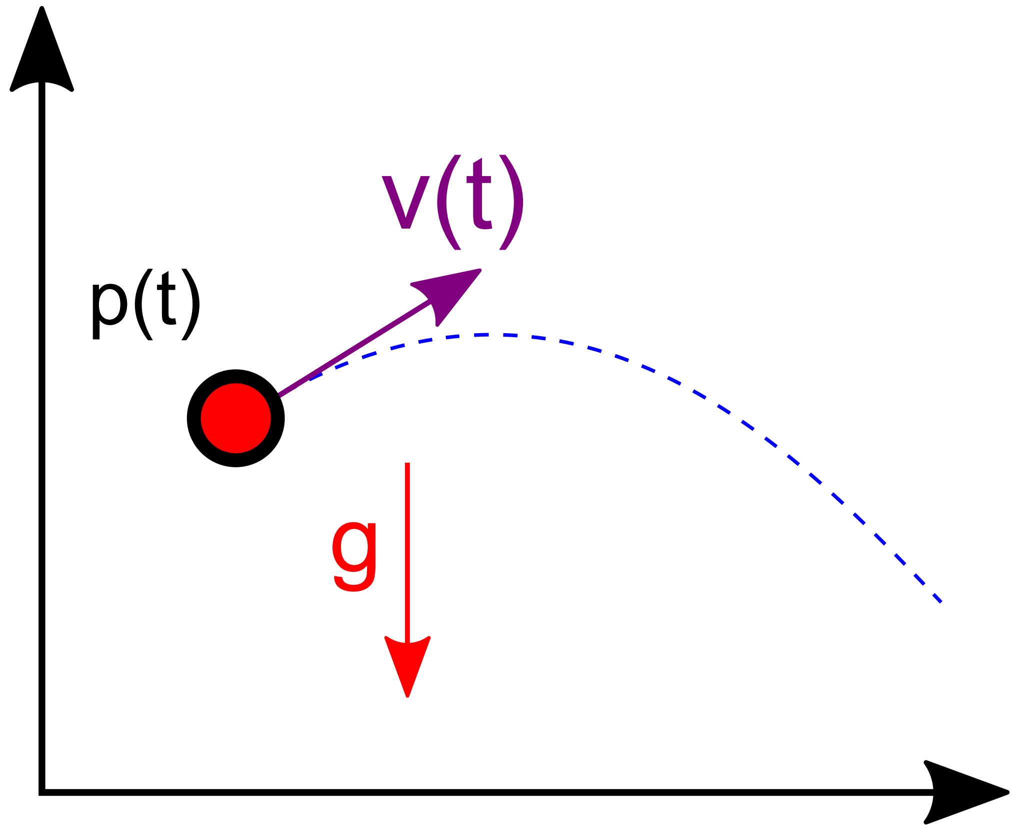
$$u'(t) = \mathrm{A}(t)\, u(t) + b(t)$$

When A is constant through time

$$u'(t) = \mathrm{A}\, u(t) + b(t)$$

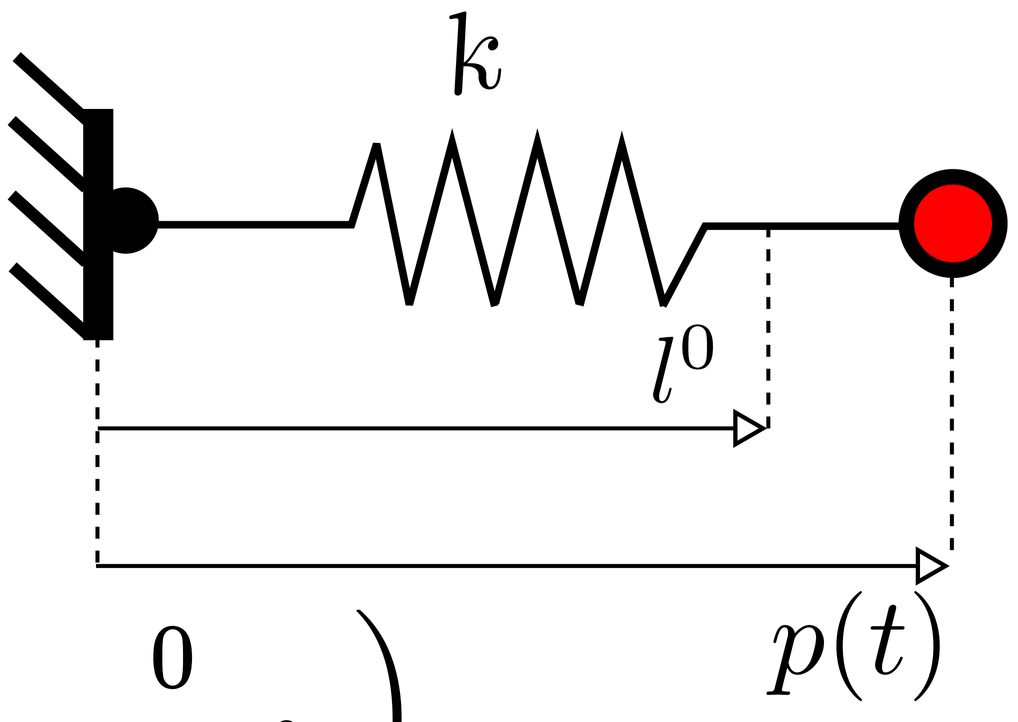
Example: Free fall under gravity

- Force $F(t) = m\, g$
- Second order differential equation: $p''(t) = g$
- First order system $\underbrace{\begin{pmatrix} p \\ v \end{pmatrix}'(t)}_{u'(t)} = \underbrace{\begin{pmatrix} v(t) \\ g \end{pmatrix}}_{\mathcal{F}(u,t)}$
- Linear system $\underbrace{\begin{pmatrix} p \\ v \end{pmatrix}'(t)}_{u'(t)} = \underbrace{\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}}_{\text{A}} \underbrace{\begin{pmatrix} p \\ v \end{pmatrix}(t)}_{u(t)} + \underbrace{\begin{pmatrix} 0 \\ g \end{pmatrix}}_{b(t)}$
- Exact solution known: $p(t) = \frac{1}{2} g\, t^2 + v_0\, t + p_0$



Note: variables are vectors - matrix can be expressed by block, or per components.

Example: 1D spring (/Harmonic oscillator)



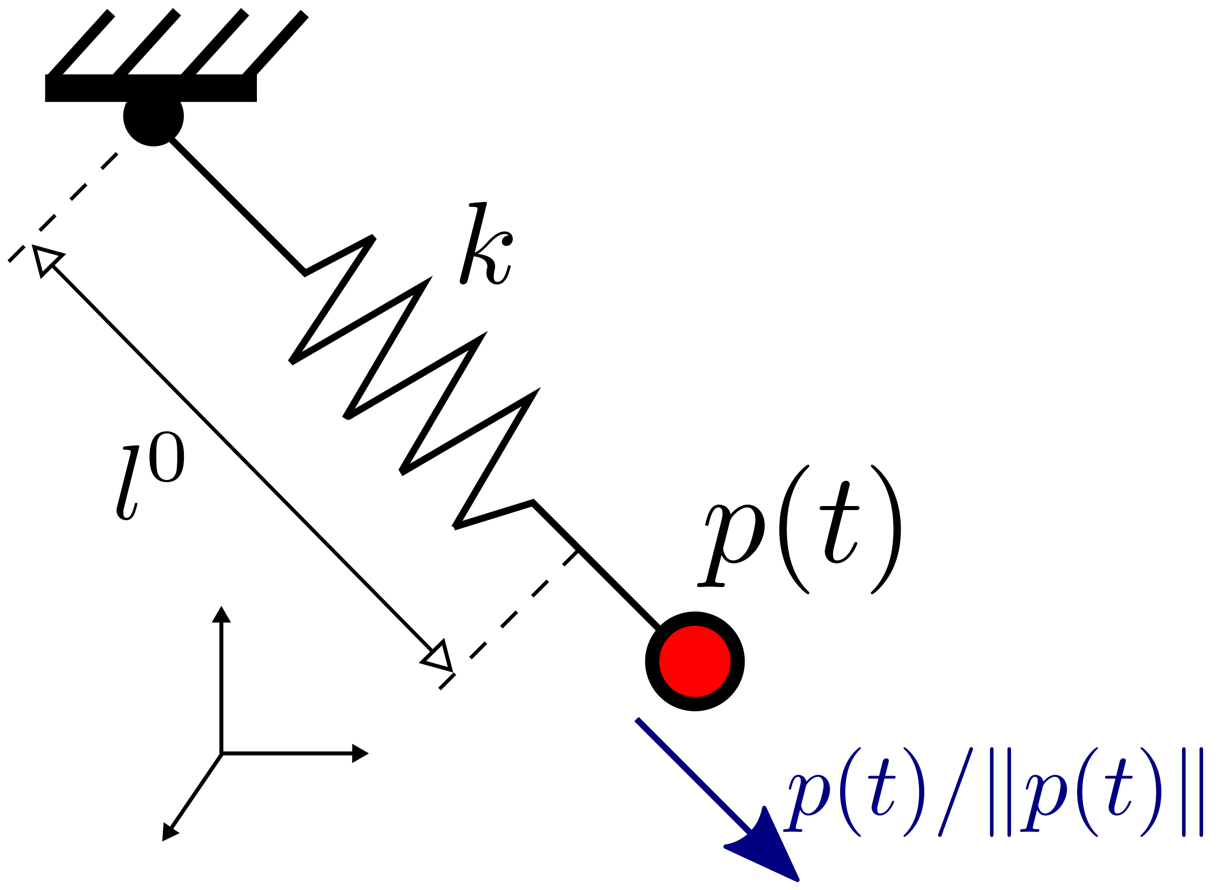
- Force $F(t) = -k\,(p(t) - l^0)$, k spring stiffness, l^0 rest length
- Second order differential equation: $m\,p''(t) + k\,p(t) = k\,l^0$
- First order system $\underbrace{\begin{pmatrix} p \\ v \end{pmatrix}'(t)}_{u'(t)} = \underbrace{\begin{pmatrix} v(t) \\ -k/m\,(p(t) - l^0) \end{pmatrix}}_{\mathcal{F}(u,t)}$
- Linear system $\underbrace{\begin{pmatrix} p \\ v \end{pmatrix}'(t)}_{u'(t)} = \underbrace{\begin{pmatrix} 0 & 1 \\ -k/m & 0 \end{pmatrix}}_{\mathrm{A}} \underbrace{\begin{pmatrix} p \\ v \end{pmatrix}(t)}_{u(t)} + \underbrace{\begin{pmatrix} 0 \\ k/m\,l^0 \end{pmatrix}}_{b}$

- Exact solution known: $p(t) = A\,\sin(\omega\,t + \varphi) + l^0$

$$\omega = \sqrt{k/m},\, A^2 = (p^0 - l^0)^2 + \left(\frac{v_0}{\omega}\right)^2,\, \tan(\varphi) = \frac{p^0 - l^0}{v^0/\omega}$$

Example: 3D mass spring

- Force $F(t) = m\,g - k\,(\|p(t)\| - l^0)\,\frac{p(t)}{\|p(t)\|}$, k spring stiffness, l^0 rest length
- Second order differential equation: $m\,p''(t) = mg - k\,(\|p(t)\| - l^0)\,\frac{p(t)}{\|p(t)\|}$
- First order system $\underbrace{\begin{pmatrix} p \\ v \end{pmatrix}'(t)}_{u'(t)} = \underbrace{\begin{pmatrix} v(t) \\ g - k/m\,(\|p(t)\| - l^0)\,\frac{p(t)}{\|p(t)\|} \end{pmatrix}}_{\mathcal{F}(u,t)}$
- Not linear
- No simple explicit solution



Numerical solution

1st order Explicit Euler

Naive numerical scheme: Approximation of the derivative

$$\frac{u^{k+1} - u^k}{h} = \mathcal{F}(u^k, t^k)$$

$$\Rightarrow u^{k+1} = u^k + h\,\mathcal{F}(u^k, t^k)$$

In the linear case

$$u^{k+1} = (\mathrm{I} + h\,\mathrm{A})\;u^k + h\,b^k$$

Pro : very easy to implement

Is u^k a good approximation of the true solution $\tilde{u}(t^k)$?

For a single particle with p, v variables:

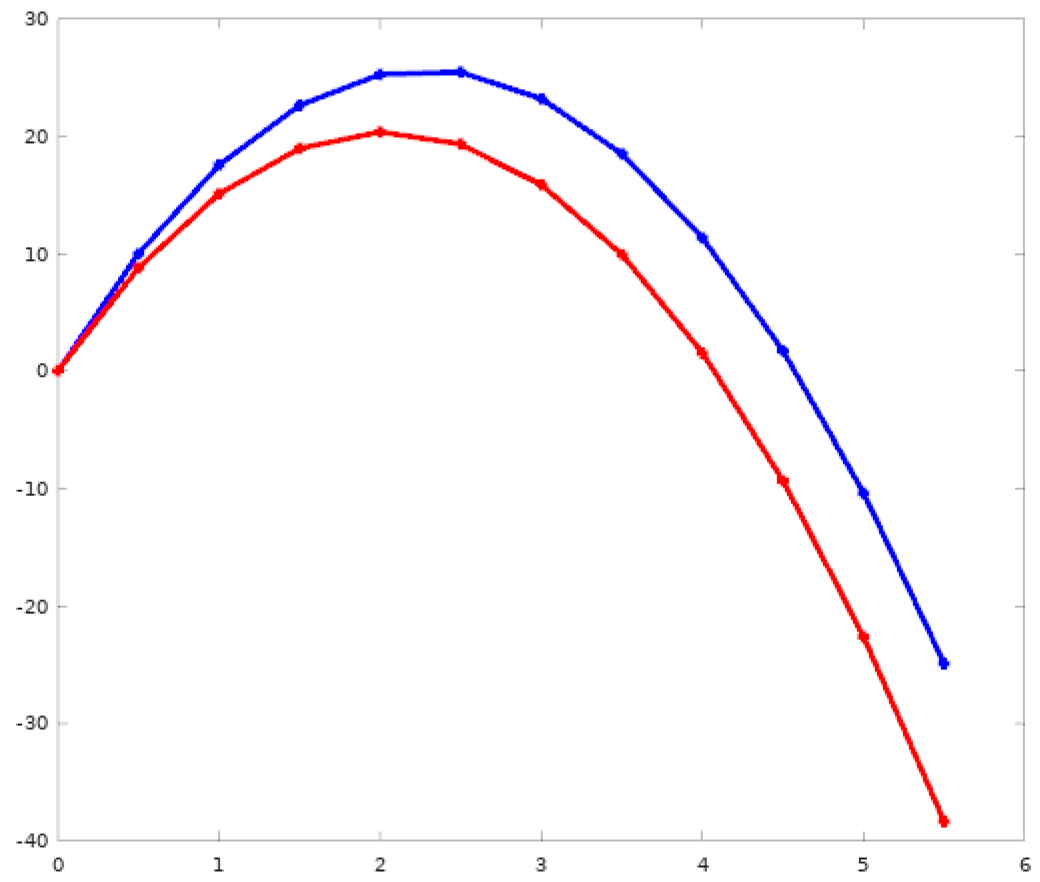
$$\begin{cases} v^{k+1} = v^k + h\,F(u^k, t^k)/m \\ p^{k+1} = p^k + h\,v^k \end{cases}$$

Explicit Euler - study case

Free fall under gravity

- True solution $\tilde{u}(k\,h) = p_0 + (k\,h)v_0 + \frac{(k\,h)^2}{2}\,g$
- Numerical scheme: $\begin{pmatrix} p \\ v \end{pmatrix}^{k+1} = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p \\ v \end{pmatrix}^{k} + \begin{pmatrix} 0 \\ h\,g \end{pmatrix}$
- Numerical solution: $p^k = p_0 + k\,h\,v_0 + \frac{k(k-1)}{2}\,h^2\,g$

$\Rightarrow$ **Not exact** : Error $e^k = |u^k - \tilde{u}(k\,h)| = \frac{g}{2} k\,h^2$

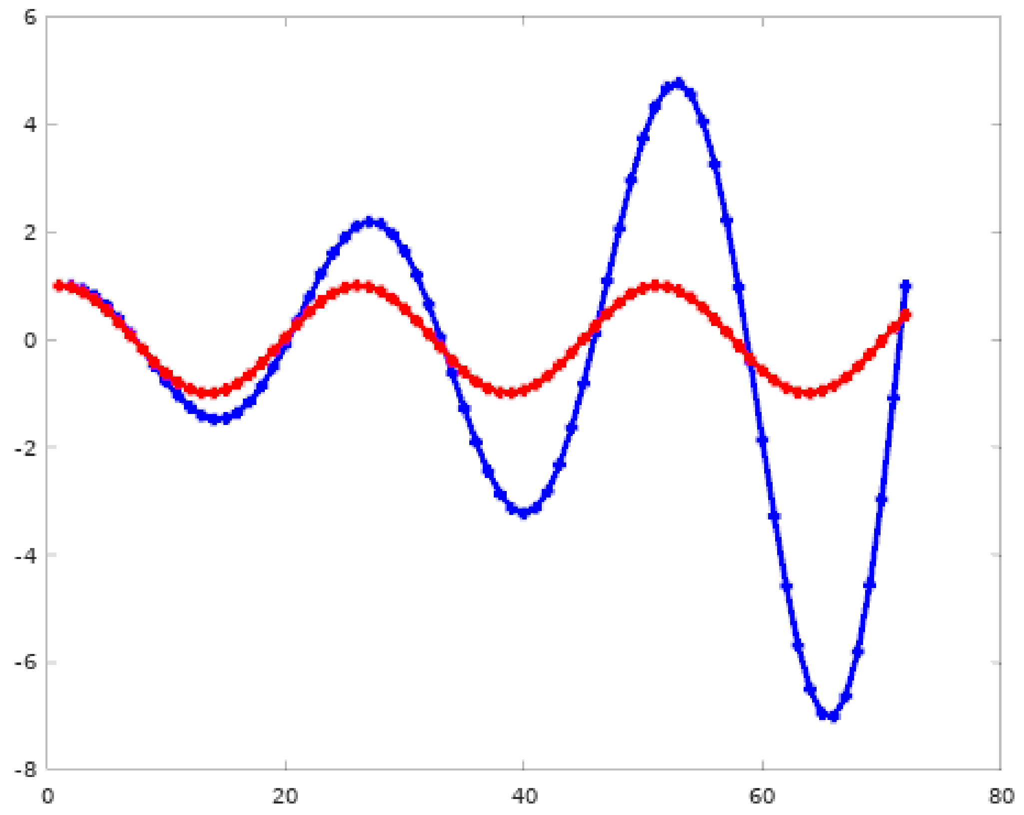


red: true solution
blue: numerical solution

Explicit Euler - study case

1D spring

- True solution: permanent oscillation
- Numerical scheme: $\begin{pmatrix} p \\ v \end{pmatrix}^{k+1} = \begin{pmatrix} 1 & h \\ -K/m\,h & 1 \end{pmatrix} \begin{pmatrix} p \\ v \end{pmatrix}^{k} + \begin{pmatrix} 0 \\ K/m\,h\,l^0 \end{pmatrix}$



red: true solution

red: numerical solution

- Numerical solution diverge to ∞
- Worse than bad accuracy for Graphics

Explicit Euler - study case

1D spring: Analysis of the system energy

- Energy $E = \frac{1}{2} m v^2 + \frac{K}{2}(p - l^0)^2$

$$E^{k+1} = \frac{1}{2} m \left(-\frac{K}{m} \Delta t \left(p^k - l^0\right) + v^k \right)^2 + \frac{1}{2} K \left(p^k + \Delta t\, v^k - l^0\right)^2$$

$$E^{k+1} = \underbrace{\frac{1}{2} m\, (v^k)^2 + \frac{1}{2} K (p^k - l^0)^2}_{E^k} + \frac{1}{2} \left[\underbrace{\frac{K^2}{m} (\Delta t)^2 (p^k - l^0)^2}_{>0} - \underbrace{2\, K \Delta t \left(p^k - l^0\right)\, v^k + 2 K \Delta t \,(p^k - l^0)\, v^k}_{=0} + \underbrace{K (\Delta t)^2\, (v^k)^2}_{>0} \right]$$

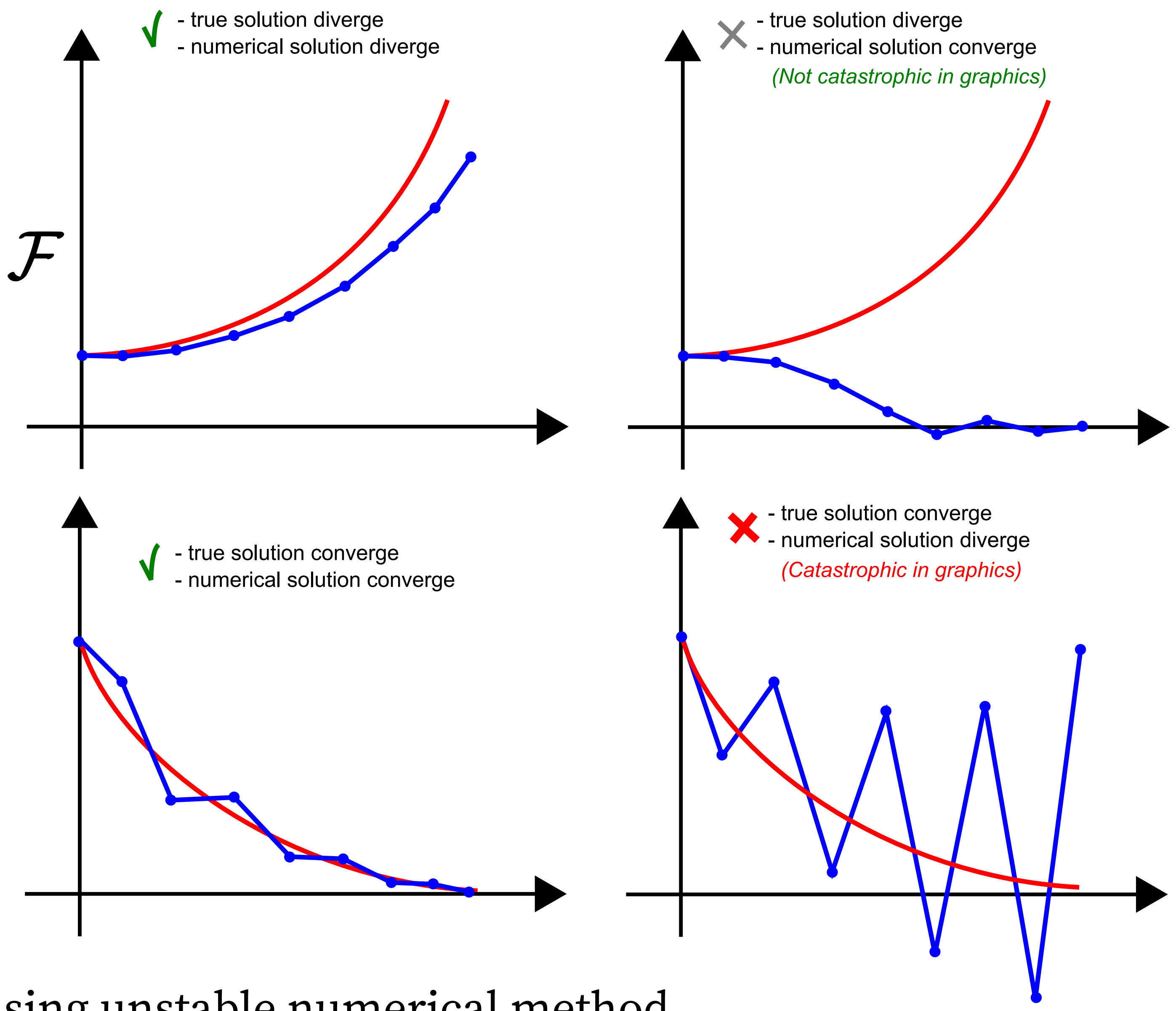
$$E^{k+1} = E^k + \epsilon\, (\Delta t)^2, \epsilon > 0$$

$\Rightarrow$ gain of energy

$\Rightarrow$ divergence

Stability of a numerical method - general definition

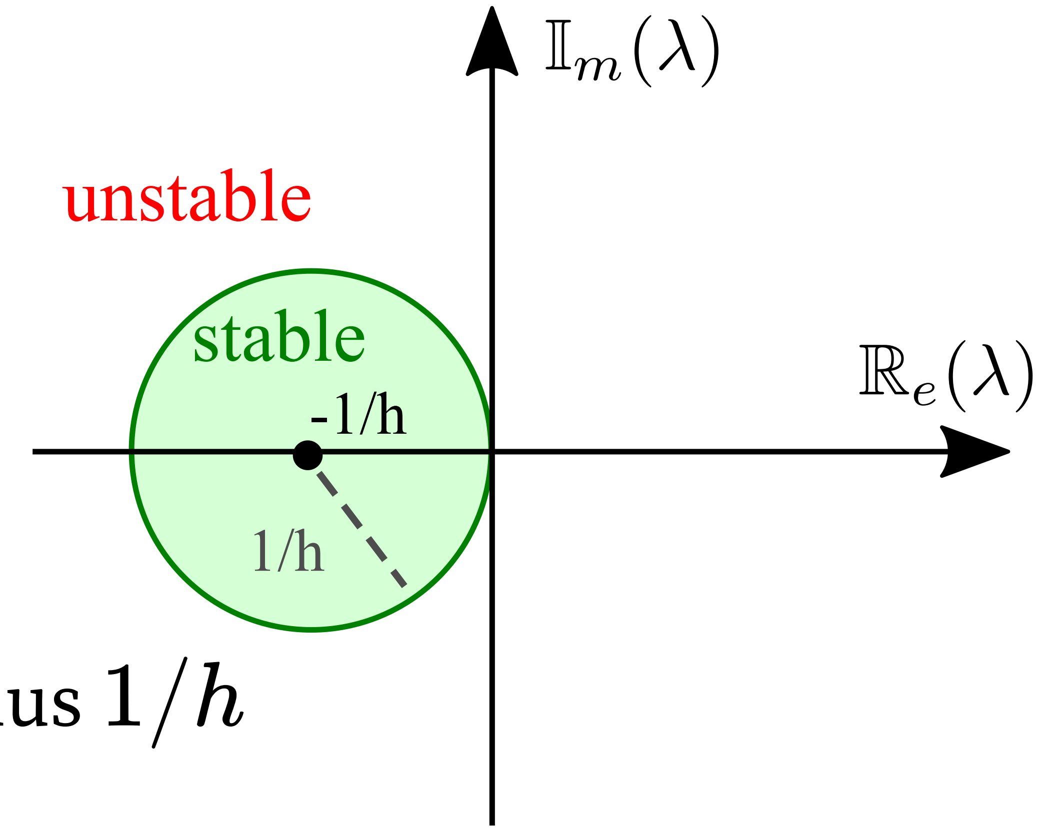
- Classical stability of a method studied on $u'(t) = \lambda\, u(t), \lambda \in \mathbb{C}$.
  - The true solution $\tilde{u}(t) = \exp(\lambda\, t)$ converge if $\mathbb{R}_e(\lambda) \leq 0$.
  - For linear system, λ refers to eigenvalues of A.
  - For non linear system, λ refers to eigenvalues of the Jacobian of $\mathcal{F}$



- A numerical method is *unconditionnaly stable* if $\mathbb{R}_e(\lambda) \leq 0$
  $\Rightarrow$ stable discrete solution.
- Otherwise, it is conditionnally stable/unstable.
- Region of stability:
  Set of conditions on λ such that the discrete solution doesn't diverge.

Rem.
- A numerical solution can diverge even when the true ODE solution converge when using unstable numerical method.
- Converseley, a numerical solution can converge even when the true ODE solution diverge when using stable numerical method.
- Stability $! =$ Accuracy.

Stability analysis of explicit Euler



$u'(t) = \lambda\, u(t)$

$\Rightarrow u^{k+1} = u^k + \lambda\, u^k$ using explicit Euler

$\Rightarrow u^{k+1} = (1 + \lambda\, h)\, u^k$

$\Rightarrow$ Stable if $|1 + \lambda\, h| \leq 1$ (conditionnal stability)

$|1/h + \lambda| \leq 1/h$: Interior of a disc centered on $(-1/h, 0)$ with radius $1/h$

Rem. For 1D elastic spring

$\lambda = \pm i\sqrt{K/m}$

$\Rightarrow |1 + i\sqrt{K/m}h| = \sqrt{1 + K/m\, h^2} > 1$

$\Rightarrow$ Explicit euler is always unstable on the elastic spring problem.

Numerical integration of ODE

General formulation: $u'(t) = \mathcal{F}(u,t), u(t) = (p(t), v(t))$.

Explicit Euler

$u^{k+1} = u^k + \Delta t\, \mathcal{F}(u^k, t^k)$

(+) Easy to implement

(-) Worst scheme in all cases (divergence, low accuracy)

Explicit Runge-Kutta

$u^{k+1} = u^k + \Delta t \sum_j \alpha_j k_j$

(+) Good **accuracy**

(+) Efficient to apply

(+/-) Stability OK for non-stiff problem, diverge on stiff problem

(-) Artificial damping for constant energy system

Implicit methods

$u^{k+1} = u^k + \Delta t\, \mathcal{F}(u^{k+1}, t^{k+1})$

(+) Good to deal with **stiff problem** - very stable

(-) Add numerical damping (converge even if solution oscillates)

(-) Hard/computationally costly to apply on non linear problem

Symplectic integrator

$v^{k+1} = v^k + \Delta t\, F^k/m$

$p^{k+1} = p^k + \Delta t\, v^{k+1}$

(+) Handle well constant energy system, preserves energy (Hamiltonian systems)

(+) Simple and efficient to implement

(-) Less accurate than RK

(-) Diverge on stiff problem

Introduction to symplectic methods

Standard approaches trade-off

- Explicit methods: (+) Simple to compute, (-) limited stability
- Implicit methods: (-) Hard to compute (especially on non linear functions), (+) very stable
- Oscillatory systems are not easy to model
  (-) Numerical solution either diverge or converge.

Symplectic approach

- *Remark*: Mechanical systems have position and velocity variables
  - Derivative of position is linear w/r velocity
  - Derivative of velocity is more complex (forces - non linear)

$\Rightarrow$ *General idea*: separate treatment of velocity and position

Semi-implicit:

- Implicit scheme for position p^{k+1} (linear part)
- Explicit scheme for velocity v^{k+1} (non linear part)

$\Rightarrow$ In practice: use velocity v^{k+1} to evaluate p^{k+1}.

Pro

(+) As simple as explicit method to implement
(+) Improved stability
(+) Well adapted to oscillatory systems

Semi implicit method

Simplest semi-implicit method: **Semi-implicit Euler / Verlet**

General case

$$v^{k+1} = v^k + h\,\mathcal{F}_v(p^k, v^k, t^k)$$
$$p^{k+1} = p^k + h\,\mathcal{F}_p(p^k, v^{k+1}, t^k)$$

Application to classical mechanical cases

$p'(t) = v(t), m\,v'(t) = F(p, t)$

$$\Rightarrow \begin{cases} v^{k+1} = v^k + h\,F(p^k, t^k)/m \\ p^{k+1} = p^k + h\,v^{k+1} \end{cases}$$

(+) Trivially easy to convert explicit Euler to semi-implicit Euler

Expressed using positions only

$$p^{k+1} = p^k + h(v^k + h\,F(p^k, t^k))$$
$$p^{k+1} = p^k + h\left(\frac{p^k - p^{k-1}}{h} + h\,F(p^k, t^k)/m\right)$$
$$\Rightarrow p^{k+1} = 2\,p^k - p^{k-1} + h^2\,F(p, t)/m$$

1st order accurate (like explicit/implicit Euler) in position and speed.

Stability on oscillatory system

1D spring system: $F(p^k, v^k, t^k) = -K\,p^k$
$p^{k+1} = 2\,p^k - p^{k-1} - h^2\,K/m\,p^k$
$\Rightarrow p^{k+1} = (2 - h^2\,K/m)\,p^k - p^{k-1}$
Stable and permanent oscillation when $h < \frac{2}{\sqrt{K/m}} = \frac{2}{\omega}$

Q. How can we demonstrate it ?

Note: $h < 2/\omega$ only valid for one 1D spring. Coupled 3D springs may require smaller value of h.

Use of semi-implicit

Ex. Elastic Spring: $F(p) = -K(p - L_0)$

Explicit Euler

```
for(int k=0; k<N; ++k) {
    p = p + h*v;
    v = v + h * F/m;
}
```

(-) Always diverges for elastic problem

Semi-Implicit Euler

```
for(int k=0; k<N; ++k) {
    v = v + h * F/m;
    p = p + h*v;
}
```

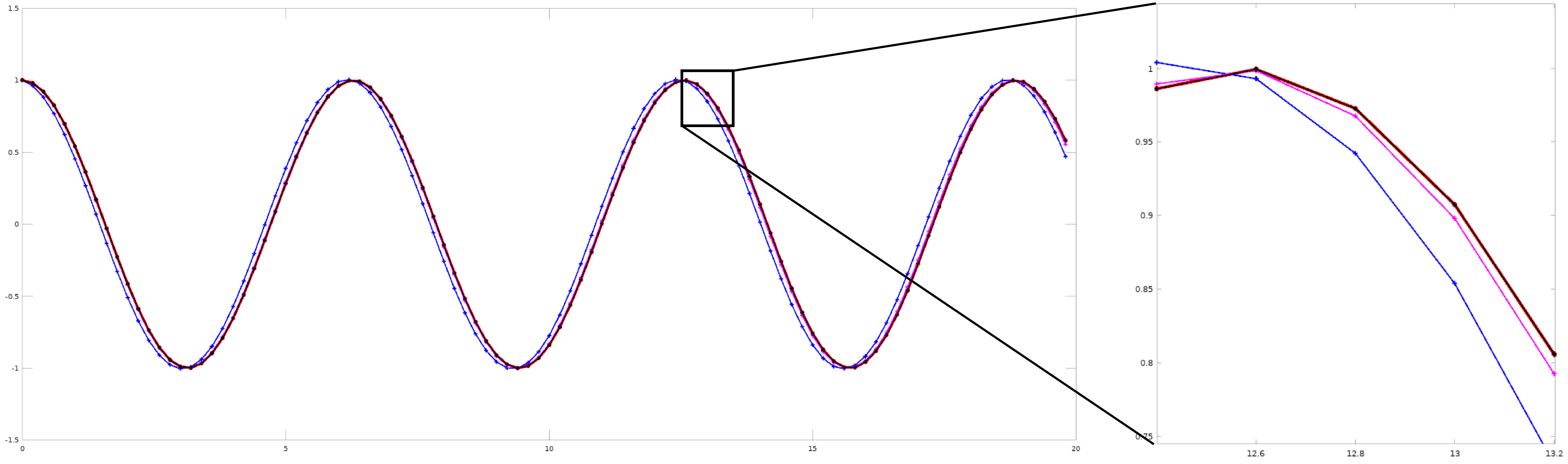
(+) Permanent oscillation for sufficiently small h.

$\Rightarrow$ Extremly simple change !

Big gain in stability

Comparison between approaches

Small $h = 0.2/\omega$

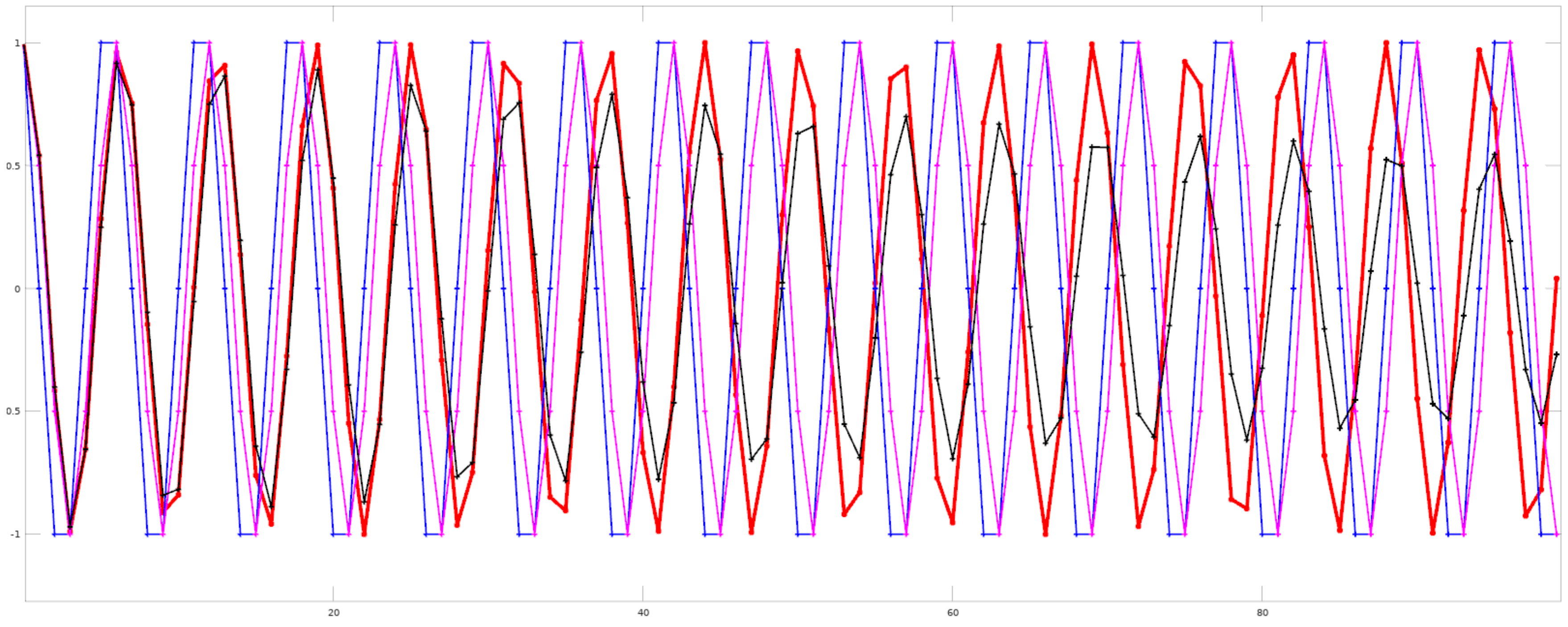


True solution , *Semi-implicit Euler* , *Velocity Verlet* , *Runge-Kutta RK4*

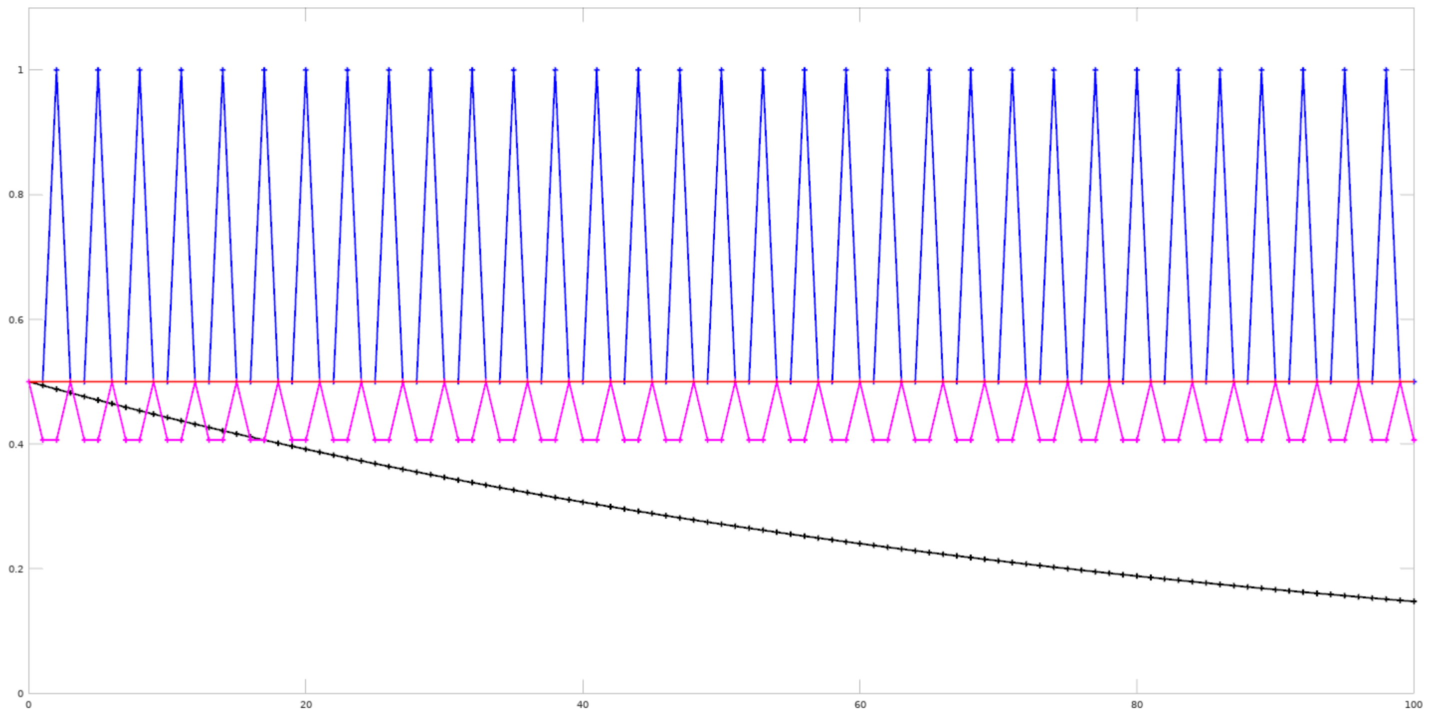
- RK4 - best behavior (undistinguishable from true solution)

Comparison between approaches

Larger $h = 1.0/\omega$



Temporal evolution of p^k



Temporal evolution of energy $E = 1/2m\,(v^k)^2 + 1/2K\,(p^k)^2$

True solution , *Semi-implicit Euler* , *Velocity Verlet* , *Runge-Kutta RK4*

- RK4 loose energy and p^k converge toward l^0
- Symplectic integrator keep oscillating

Material model

Elasticity: Shape goes back toward its original rest position when external forces are removed.

-Purely elastic models don't loose energy when deformed (potential $\leftrightarrow$ kinetic)

Plasticity: Opposite of elasticity. Plastic material don't come back to their original shape (/change their rest position during deformation).

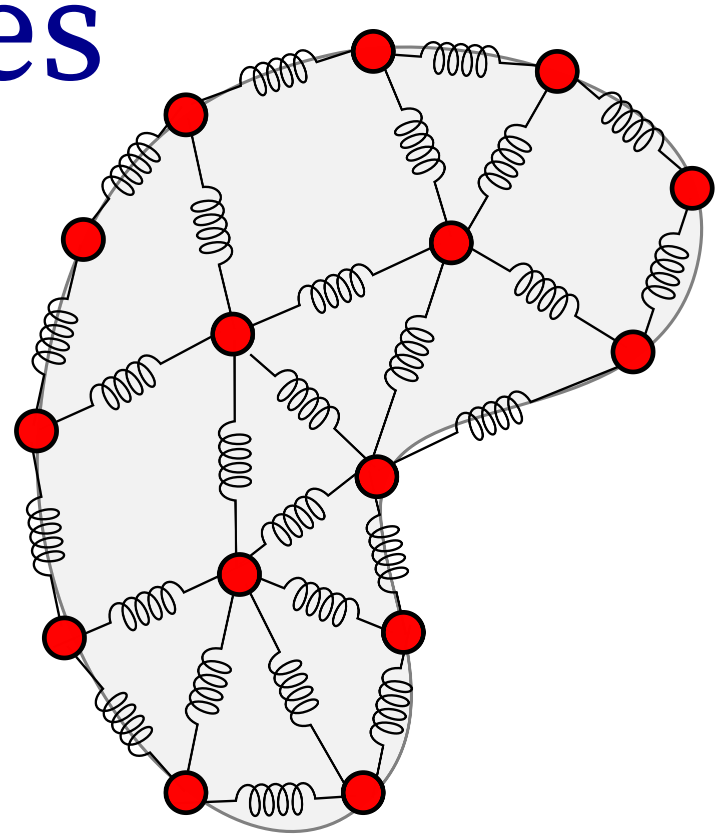
- Ductile material - can allow large amount of plastic deformation without breaking (plastic)
- Brittle - Opposite (glass, ceramics)

Viscosity: Resistance to flow (usually for fluid, ex. honey)

In reality

- *Elasto-plastic materials*: Allow elastic behavior for small deformation, and plastic at larger one.
- *Visco-elastic materials*: Elastic properties with delay.

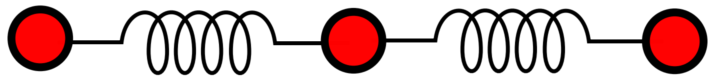
Modeling elastic shapes with particles



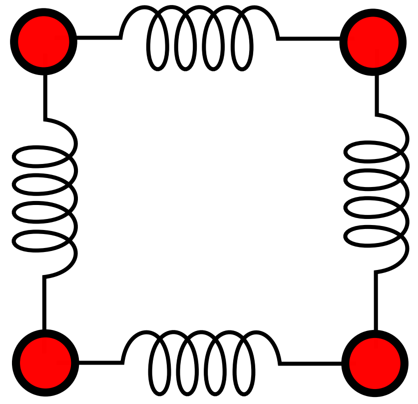
Spring mass systems

- Particles (position, velocity, mass): samples on shape
- Springs : link closed-by particles in the reference shape

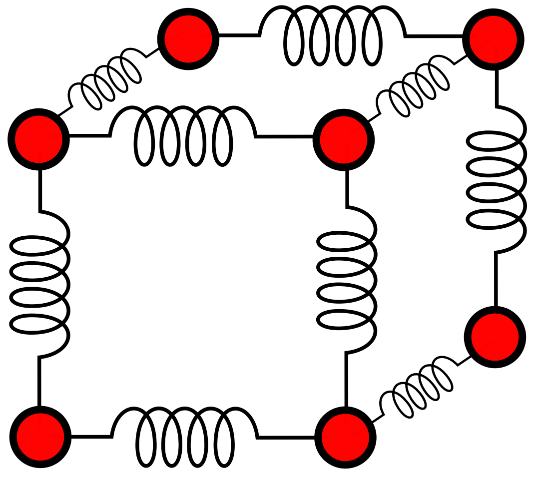
1D curve structure



2D surface structure



3D volume structure

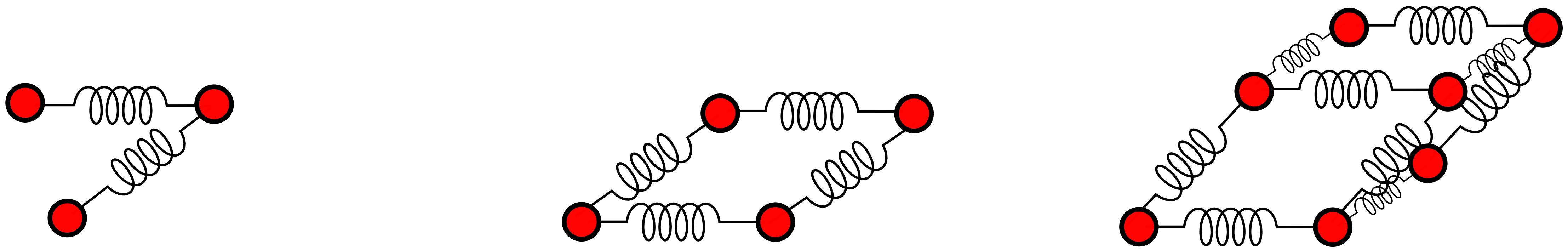


Spring structure

How to model spring connectivity ?

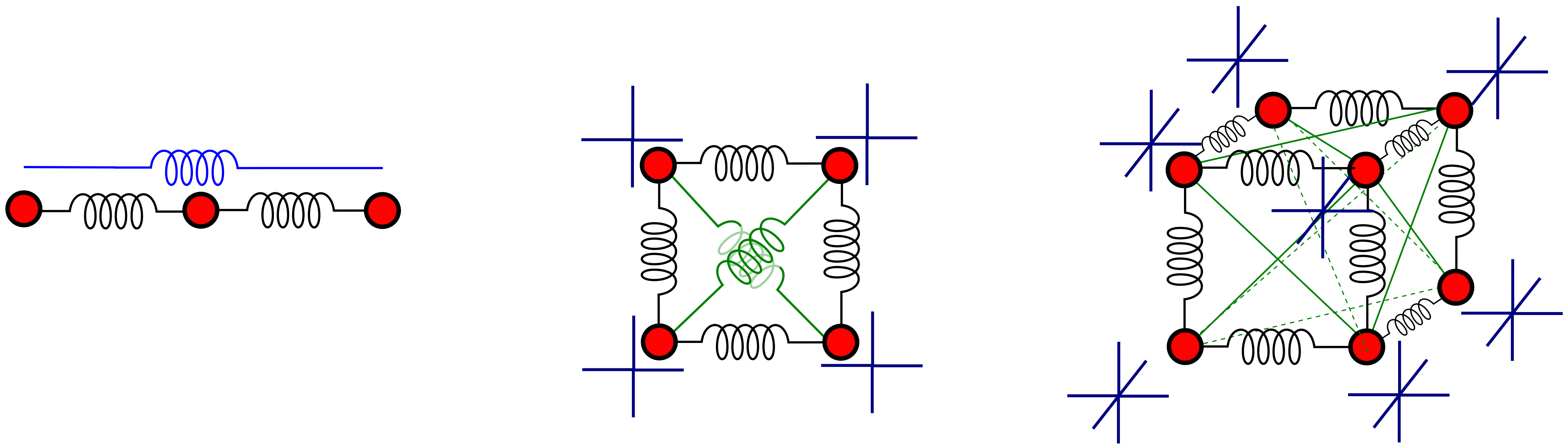
- **Structural springs**: 1-ring neighbors springs ($\simeq$ mesh edges)

  (+) Limit elongation/contraction, (-) Allows shearing, and bending



$\Rightarrow$ Add extra springs connectivity

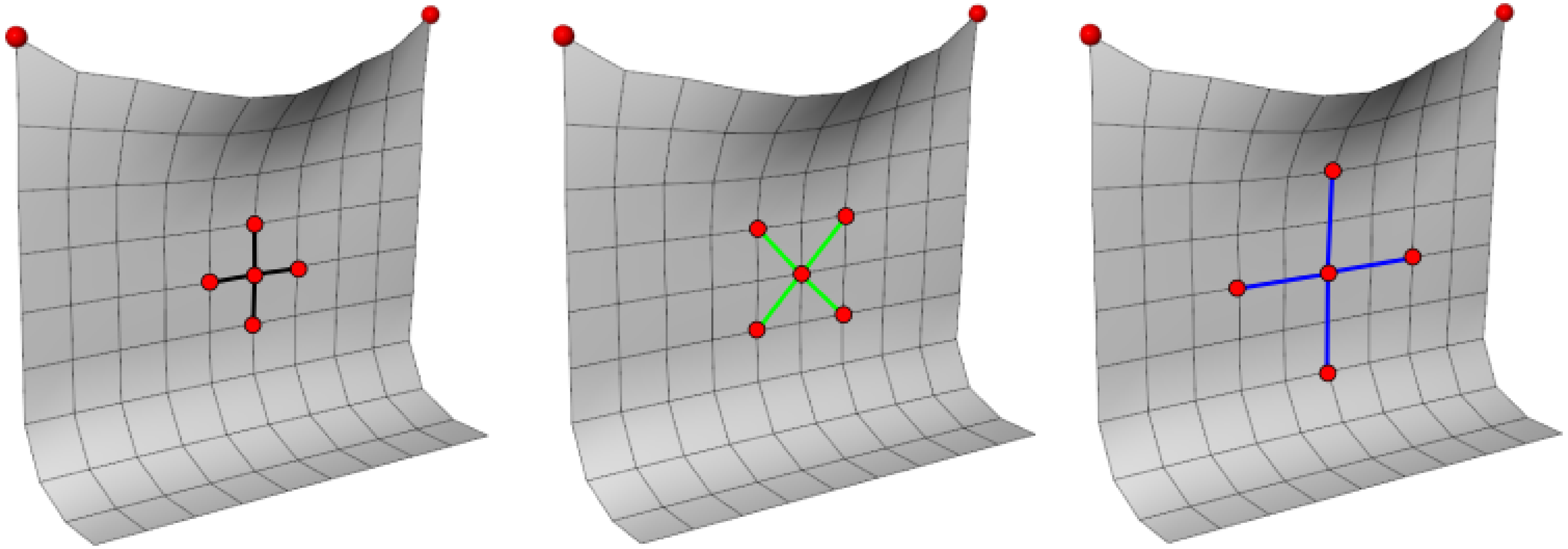
- **Shearing springs**: Diagonal links
- **Bending springs**: 2-ring neighborhood



Cloth Simulation

Mass-spring cloth simulation

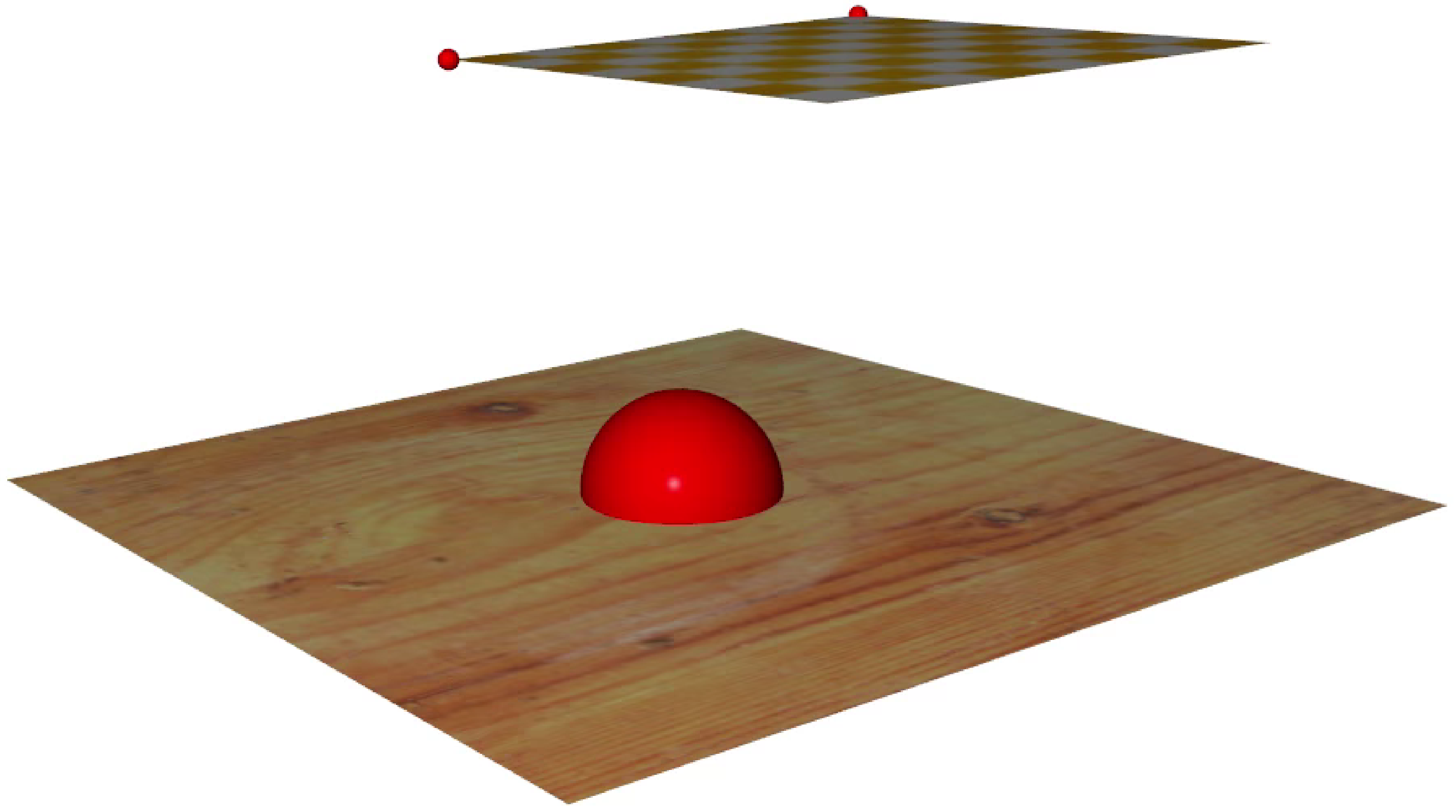
- Particles are sampled on a $N \times N$ grid.
  - Each particle has a mass m ($m_{cloth} = N^2\,m$)
- Set structural, shearing and bending springs.



Forces

- On each particle: gravity + drag + spring forces
- $F_i(p, v, t) = m_i\, g - \mu\, v_i(t) + \sum_{j \in \mathcal{V}_i} K_{ij}\ \left(\|p_j(t) - p_i(t)\| - L^0_{ij}\right) \dfrac{p_j(t) - p_i(t)}{\|p_j(t) - p_i(t)\|}$
  - $\mathcal{V}_i$: neighborhood of particle i
  - L^0_{ij}: rest length of spring ij

Associated ODE $\forall i, \begin{cases} p_i'(t) = v_i(t) \\ v_i'(t) = F_i(p, v, t)/m_i \end{cases}$



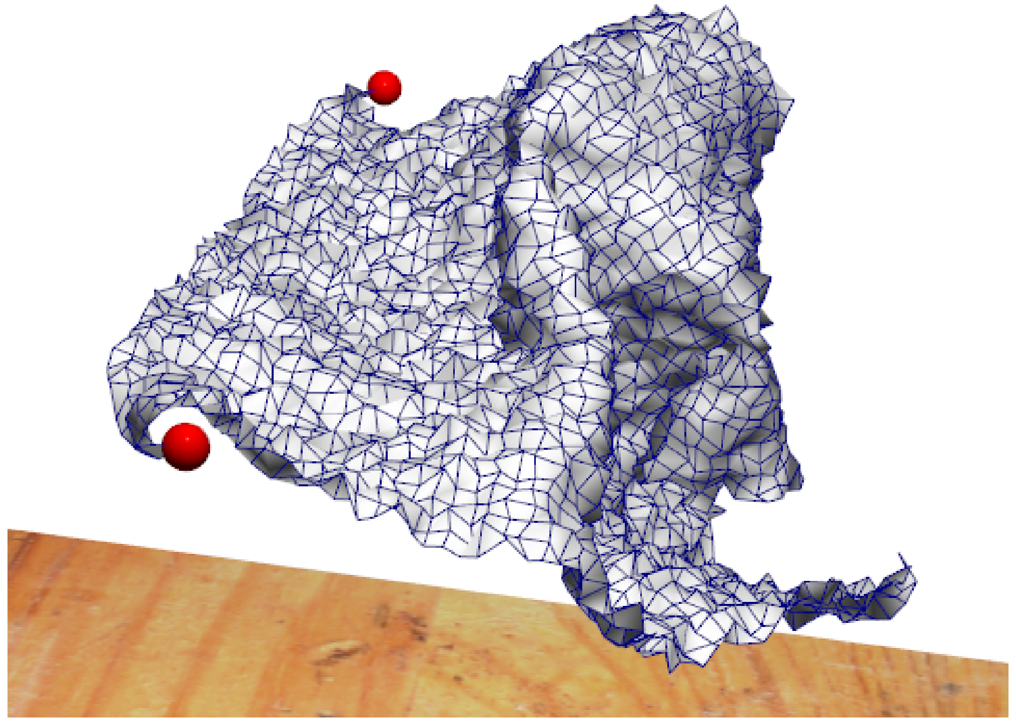
Q. How can we model the effect of the wind ?

Note on Mass-Spring numerical solution

- Non-linear ODE

- Large K_{ij} : good length preservation, but stiff ODE
  $\Rightarrow$ divergence of explicit schemes.
- Avoid explicit Euler (divergence)

- Semi-implicit Euler/Verlet works fine for low K_{ij}
  Semi-implicit Euler + PBD allows simple integration + stable stiff springs
  [Muller et al. PBD , Inextensible clothing in Computer Games]



- RK4 more accurate (but higher complexity than Verlet)

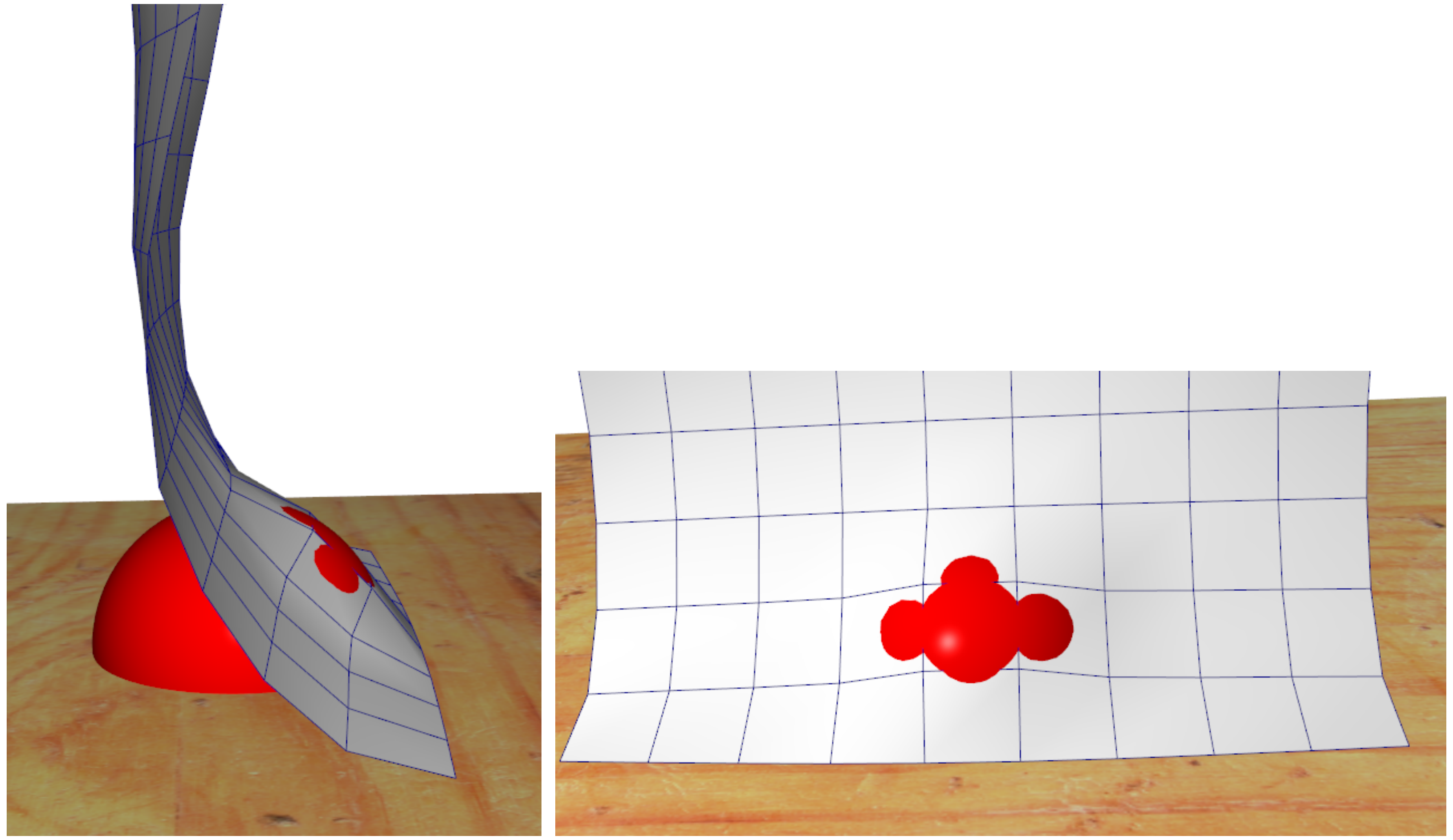
- Implicit Euler : requires linearization, but very stable

Collisions

- Simple approach : Handled as collision between particles and shapes
  - (+) Simple and efficient
  - (-) Collision may still appears within a triangle

$\Rightarrow$ Simplest solution: take an ϵ-margin around each collider.

Or Exaustive approach: edges + faces (more coslty).

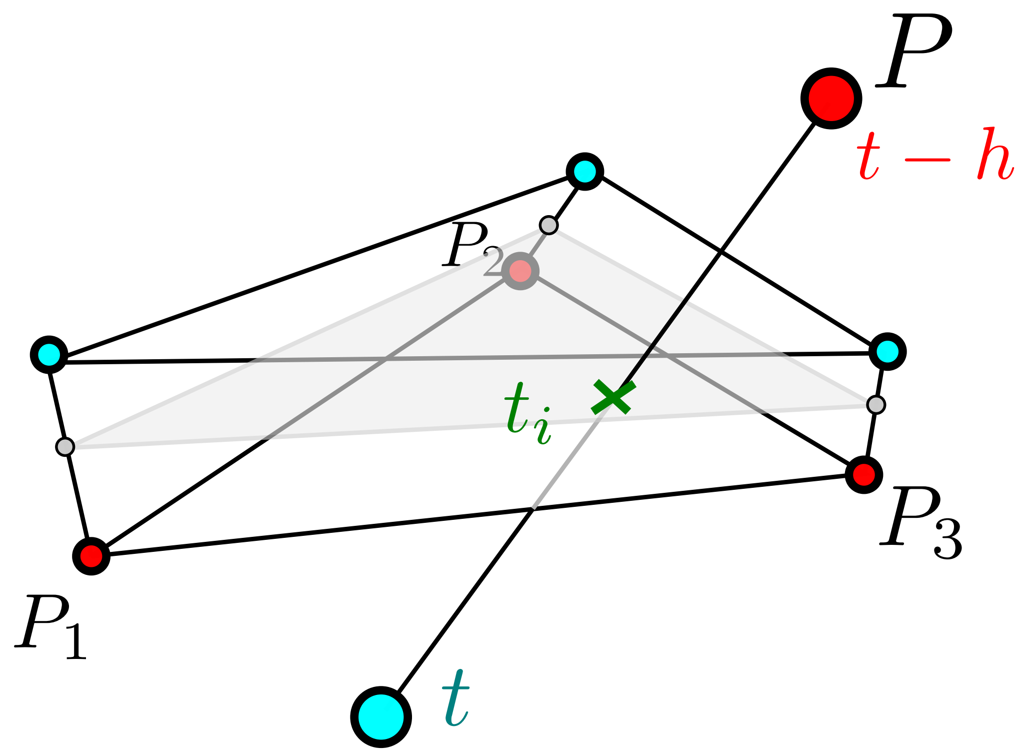


Detecting self collision

Handled as moving point in collision with moving triangle

Inputs

- Triangle $P_1(t)P_2(t)P_3(t)$, a point $P(t)$
- Each position $P_k(t) = P_k(0) + t\, v_{P_k}$



Computing intersection

Necessary condition

- Find $t_i \in [0, h]$ such that $P(t_i)$ is in triangle plane

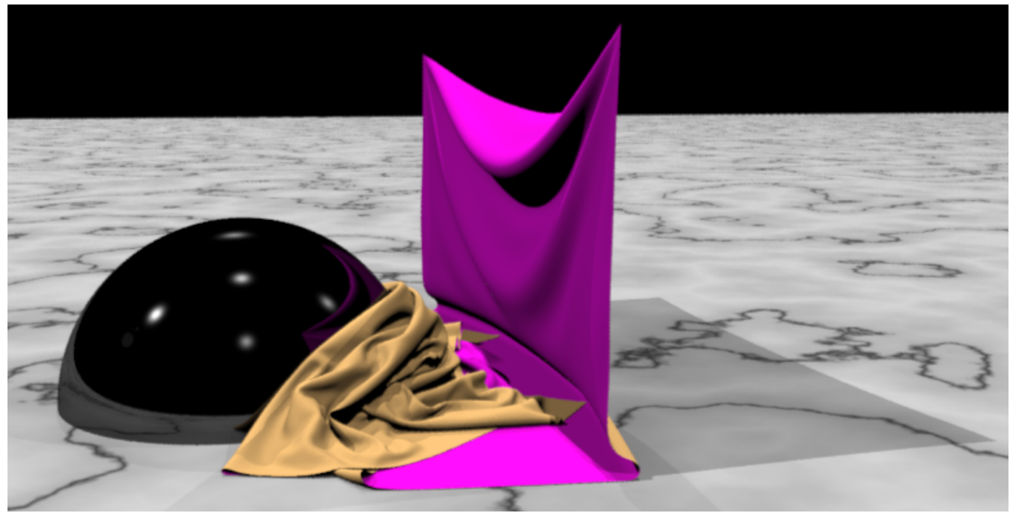
$$(P(t_i) - P_1(t_i)) \times n(t_i) = 0$$

$n(t_i)$: normal of the triangle at time t_i

Sufficient condition

- Check $P(t_i)$ is inside the triangle

$$P(t_i) = \alpha P_1(t_i) + \beta P_2(t_i) + \gamma P_3(t_i)$$
$$(\alpha, \beta, \gamma) \in [0, 1]^3, \alpha + \beta + \gamma = 1$$



[X. Provot. Collision and self-collision handling in cloth model dedicated to design garments. Graphics Interface 1997.]

[R. Bridson et al. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. ACM SIGGRAPH 2002]

Limitation of mass spring model and continuous model

- Does mass-spring system converge toward a unique solution when sampling increase ?

  $\Rightarrow$ No :(

  Depends on the connectivity $\rightarrow$ bad for physical accuracy

Corollary

- Mass-springs work well for grid-mesh structure (draping)
- Less for arbitrary triangular meshes

1st improvment: Change toward energy formulation for bending springs (limits locking effect)

$$F = \frac{\partial E}{\partial p}$$

$$E = \frac{1}{2} K\, L\, \kappa^2,\ \kappa\text{: curvature}$$

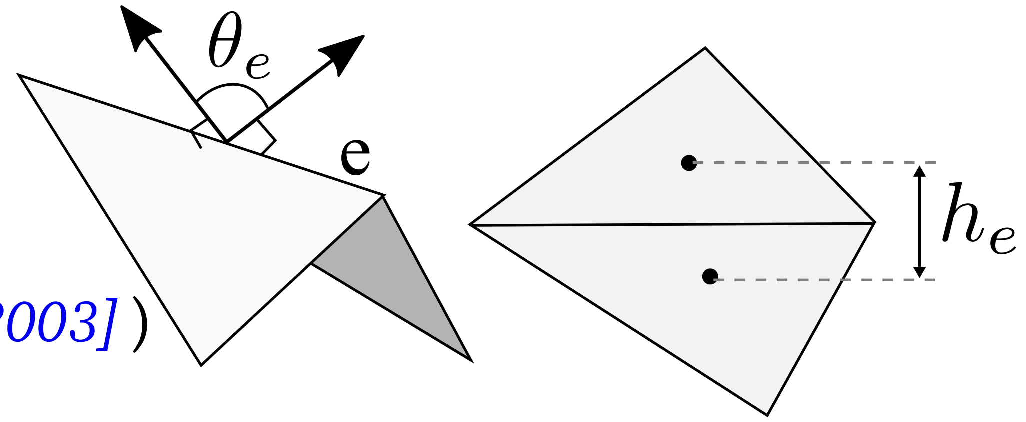
[Cho et al, Stable but Responsive Cloth, ACM SIGGRAPH 2002]

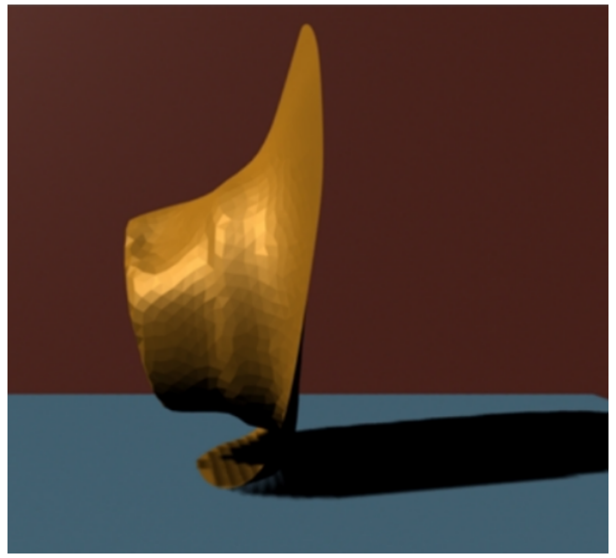
Triangle as continuous elements

- Defining Bending Energy between triangles
  - $W_B(x) = \sum_{edges\ e} (\theta_e - \theta_e^0) \frac{\|e^0\|}{h_e^0}$

    [E. Grinspun et al., Discrete Shells, SCA 2003]

    (or expressed using forces in [R. Bridson et al., SCA 2003])





- Going toward full FEM numerical resolution
  - B. Thomaszewski et al. [SCA 2006], [VRIPHYS 2008], [EG 2009].