

Numerical solution of ODE (Ordinary Differential Equation)

000

What is an ODE ?

Example:

f is an unknown **function**

$$f'(x) = af(x) + b$$

dérivative of f depends of f

f is 1D function
(otherwise PDE)

Can also be written as: $y' = ay + b$

001

What is an ODE ?

More examples:

Linear, constant coefficients

$$f'(x) = 4f(x) + 2$$

Linear, variable coefficients

$$f'(x) = 4(x - 5)f(x) + 2x^2 - 7$$

Non linear

$$f'(x) = 4x \sin(xf(x)) + 2/f^2(x)$$

General expression

$$f'(x) = \mathcal{F}(x, f)$$

Even more general: Implicit formulation

$$\mathcal{R}(x, f, f') = 0$$

002

What is an ODE?

Order of ODE:

Second order, linear

$$f''(x) = 2f'(x) + 3f(x) - 4$$

Third order, non linear

$$f^{(3)}(x) = 2x^2 \sin(f''(x) + f'(x)) - f^2(x)$$

General definition:

$$f^{(n)}(x) = \mathcal{F}(x, f, f', \dots, f^{(n-1)})$$

Even more general:

$$\mathcal{R}(x, f, f', \dots, f^{n-1}, f^{(n)}) = 0$$

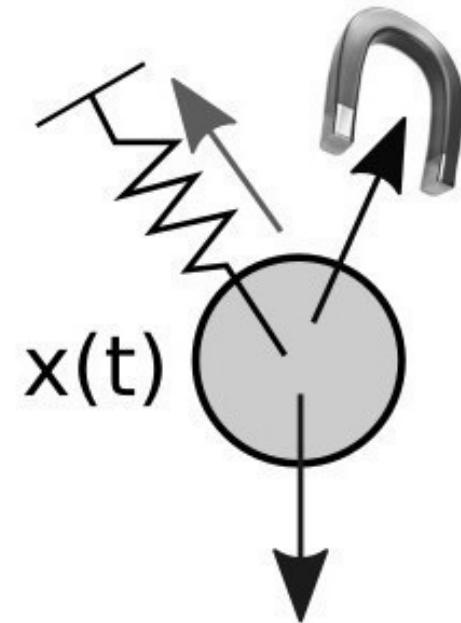
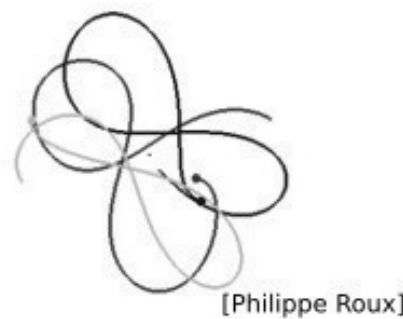
003

Why do we need ODE?

Physics:

$$m a(t) = \sum F(x(t), t)$$

$$x''(t) = \frac{1}{m} F(x(t))$$



004

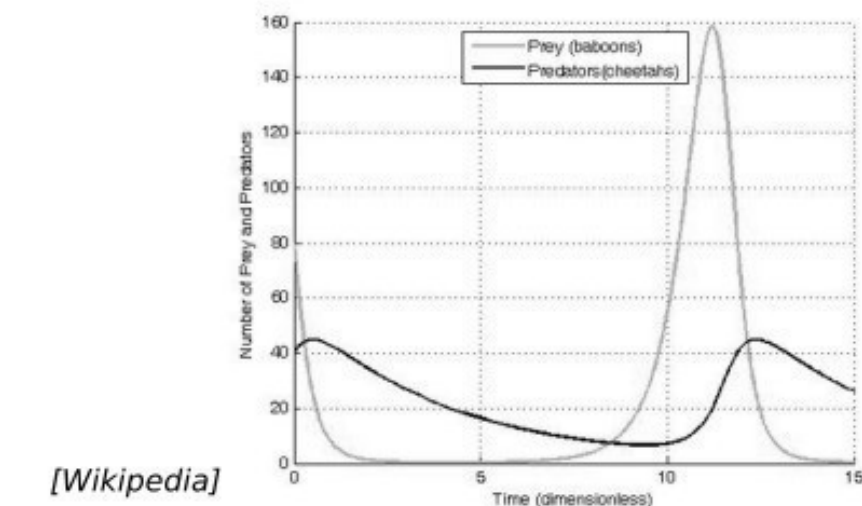
Why do we need ODE?

Biology:

Population grows

$$\begin{aligned} f_1 : \text{prey} \\ f_2 : \text{predator} \end{aligned} \quad \begin{cases} f_1'(t) = f_1(t)(\alpha - \beta f_2(t)) \\ f_2'(t) = -f_2(t)(\gamma - \delta f_1(t)) \end{cases}$$

Lotka-Volterra equation

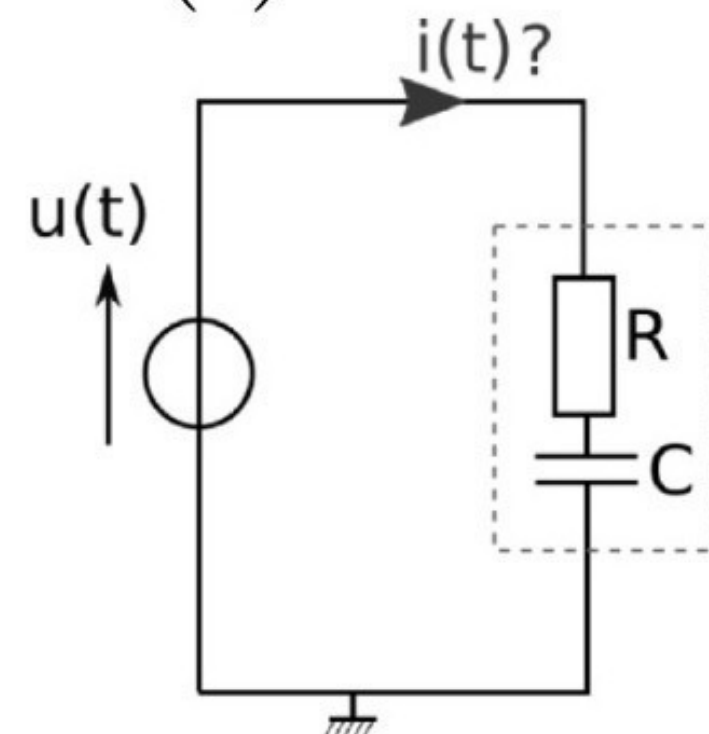
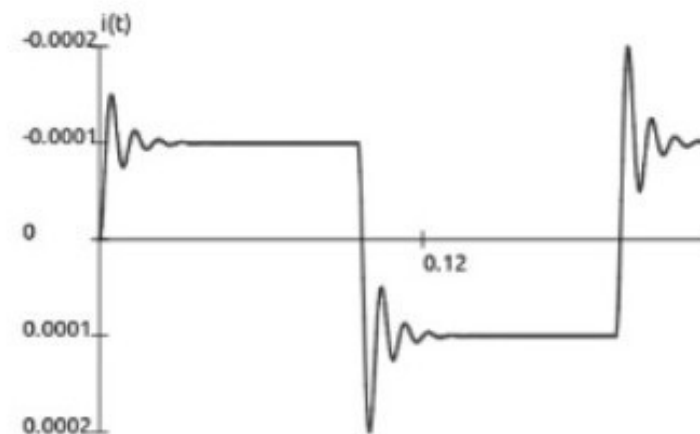


005

Why do we need ODE?

Physics:

$$RCi'(t) + i(t) = Cu'(t)$$



006

What can we solve analytically?

Linear + constant coefficient

$$a_0 f(x) + a_1 f'(x) + \dots + a_n f^{(n)}(x) = r(x)$$

Linear + variable coefficient + low order

$$a_0(x) f(x) + a_1(x) f'(x) + a_2(x) f''(x) = r(x)$$

Non linear: Almost never
Existence and uniqueness ?

007

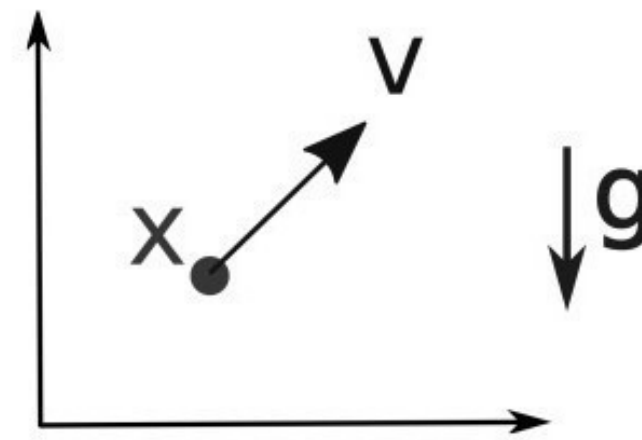
Case study: Free fall under gravity

Motion equations:

Initial conditions:

$$x(t=0)=x_0$$

$$v(t=0)=v_0$$



008

Numerical approach

Motion equations:

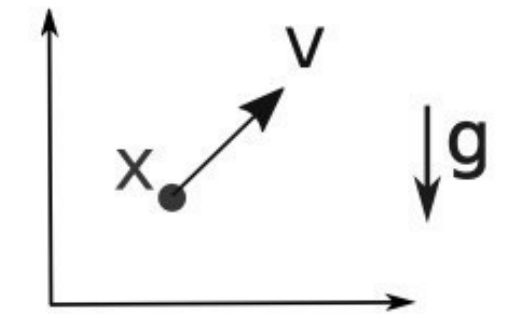
$$x'(t)=v(t)$$

$$v'(t)=g$$

Initial conditions:

$$x(t=0)=x_0$$

$$v(t=0)=v_0$$



Approximation:

$$f'(t) \simeq \frac{f(t + dt) - f(t)}{dt}$$

009

Numerical approach

Motion equations:

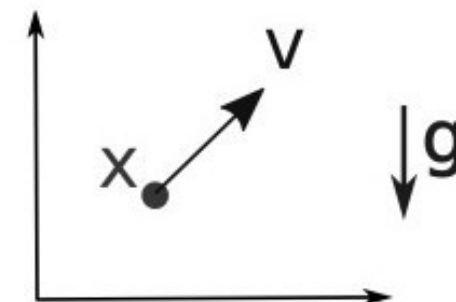
$$x'(t)=v(t)$$

$$v'(t)=g$$

Initial conditions:

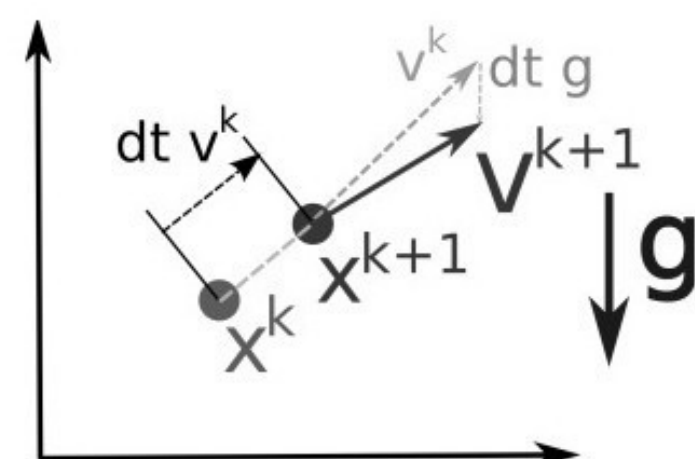
$$x(t=0)=x_0$$

$$v(t=0)=v_0$$



Solution

$$\begin{cases} v^{k+1} = v^k + (\Delta t)g \\ x^{k+1} = x^k + (\Delta t)v^k \end{cases}$$



Code;

```
x=x0;
v=v0
for (k=0; k<N; ++k)
{
    x=x+dt*v;
    v=v+dt*g;
}
```

010

Numerical approach

Motion equations:

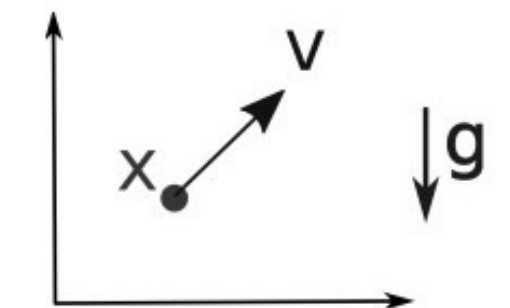
$$x'(t)=v(t)$$

$$v'(t)=g$$

Initial conditions:

$$x(t=0)=x_0$$

$$v(t=0)=v_0$$



Numerical solution:

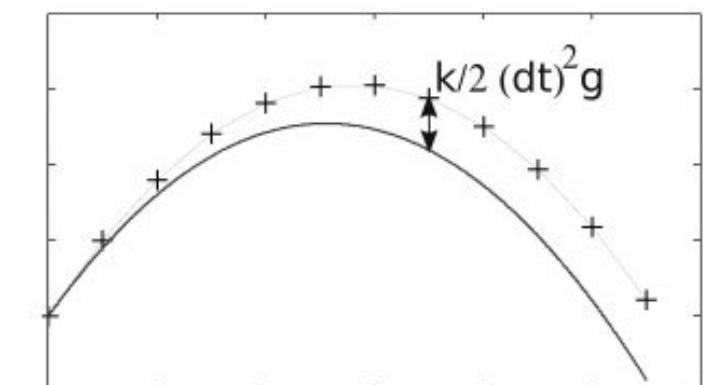
$$\begin{cases} v^{k+1} = v^k + (\Delta t)g \\ x^{k+1} = x^k + (\Delta t)v^k \end{cases}$$

$$\Rightarrow \begin{cases} x^{k+2} = 2x^{k+1} - x^k + (\Delta t)^2 g \\ x^0 = x_0 \\ x^1 = x_0 + \Delta t v_0 \end{cases}$$

$$\Rightarrow x(t = k\Delta t) = x_0 + (k\Delta t)v_0 + \frac{k(k-1)}{2}(\Delta t)^2 g$$

Real solution:

$$\tilde{x}(t = k\Delta t) = x_0 + (k\Delta t)v_0 + \frac{1}{2}(k\Delta t)^2 g$$



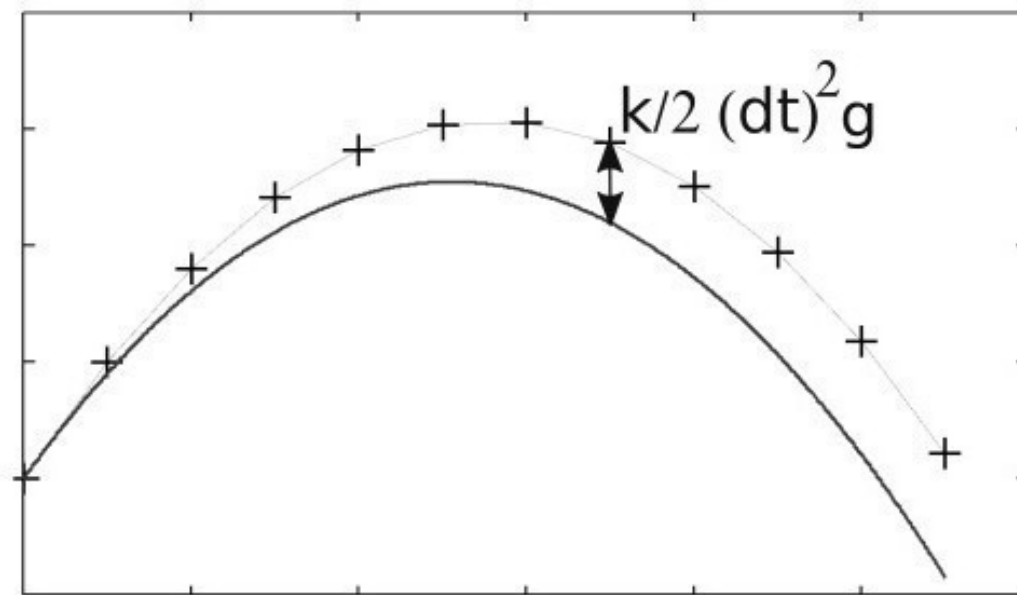
011

Accuracy

Numerical error: $\|x(k\Delta t) - \tilde{x}(k\Delta t)\| = \frac{k}{2}(\Delta t)^2 g$

Definition of accuracy of order h:

$$\|x(k\Delta t) - \tilde{x}(k\Delta t)\| = \mathcal{O}((\Delta t)^{h+1})$$



012

Matrix formulation

Linear ODE of order n
= system of ODE of order 1 $u(t) = \begin{pmatrix} x(t) \\ x'(t) \end{pmatrix}$

$$u'(t) = Au(t) + b(t)$$

with $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ $b(t) = \begin{pmatrix} 0 \\ g \end{pmatrix}$ $u(t) = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix}$

013

Matrix formulation

$$u'(t) = Au(t) + b(t) \quad u(t) = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} \quad A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ b(t) = \begin{pmatrix} 0 \\ g \end{pmatrix}$$

Same approach:

$$u^{k+1} = (I + \Delta t A) u^k + \Delta t b$$

Same solution ...

014

Matrix formulation

$$u'(t) = Au(t) + b(t) \quad u(t) = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} \quad A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ b(t) = \begin{pmatrix} 0 \\ g \end{pmatrix}$$

Explicit Euler: $\frac{u^{k+1} - u^k}{\Delta t} = A u^k + b$

015

Matrix formulation

$$u'(t) = Au(t) + b(t) \quad u(t) = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} \quad A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad b(t) = \begin{pmatrix} 0 \\ g \end{pmatrix}$$

Explicit Euler: $\frac{u^{k+1} - u^k}{\Delta t} = A u^k + b$

Implicit Euler: $\frac{u^{k+1} - u^k}{\Delta t} = A u^{k+1} + b$

now unknown

016

Implicit Euler

$$u^{k+1} = (I - \Delta t A)^{-1} (u^k + \Delta t b)$$

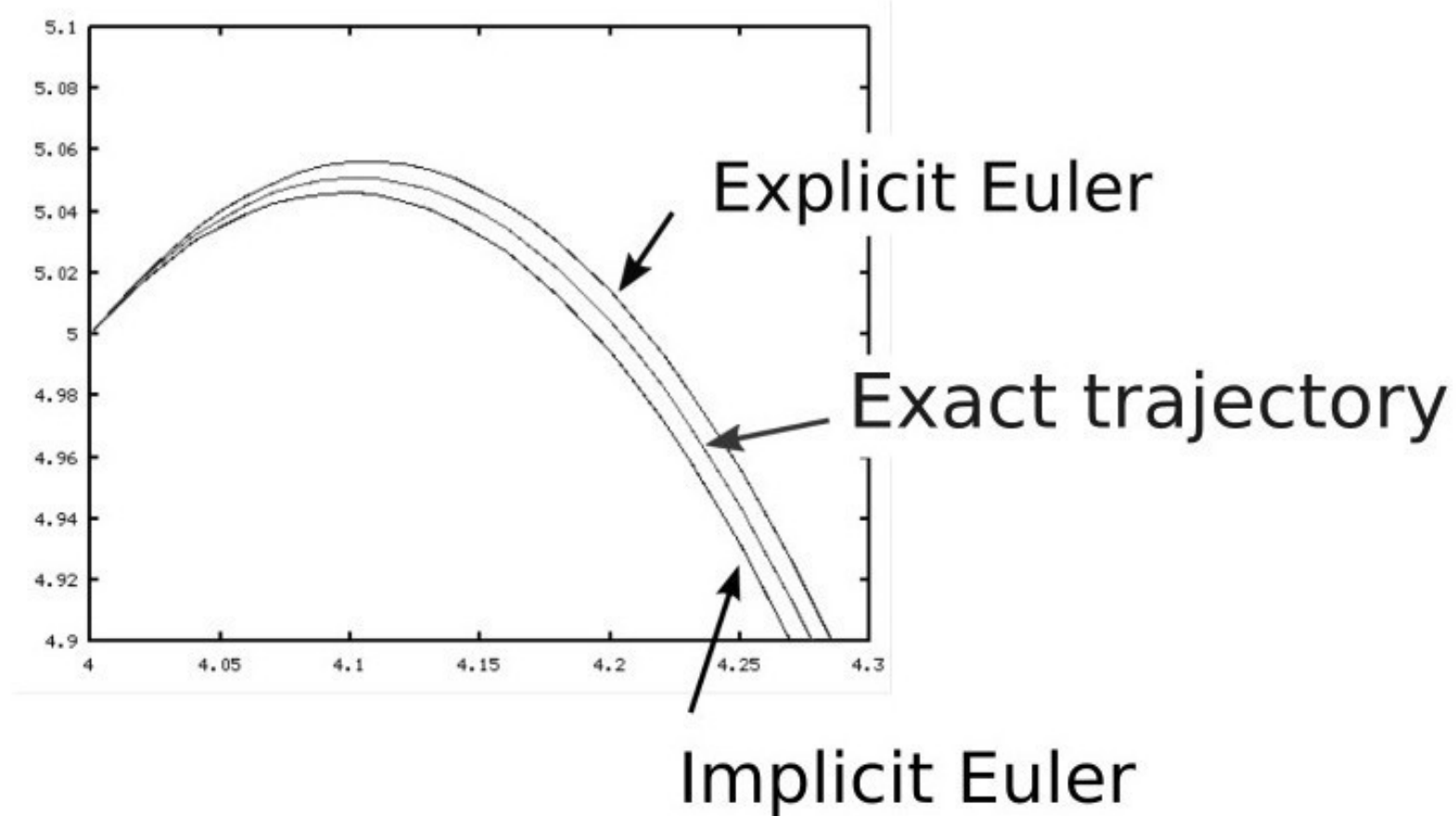
$$\begin{cases} x^{k+2} = 2x^{k+1} - x^k + (\Delta t)^2 g \\ x^0 = x_0 \\ x^1 = x_0 + \Delta t v_0 + (\Delta t)^2 g \end{cases}$$

$$x(k\Delta t) = x_0 + (k\Delta t) v_0 + \frac{k(k+1)}{2} (\Delta t)^2 g$$

Still not the true solution

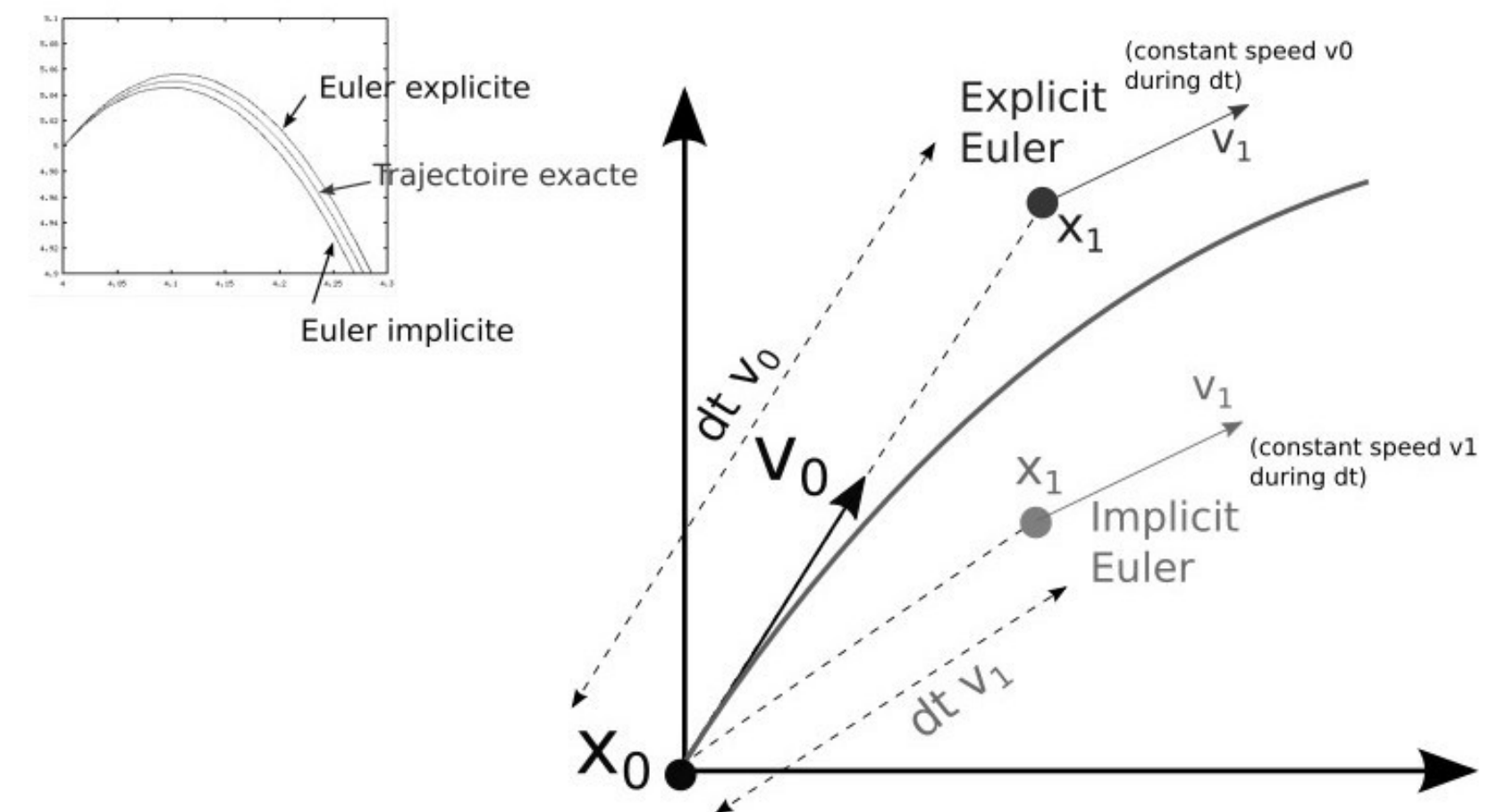
017

Implicit Euler



018

Implicit Euler



019

Summary Explicit/Implicit Euler

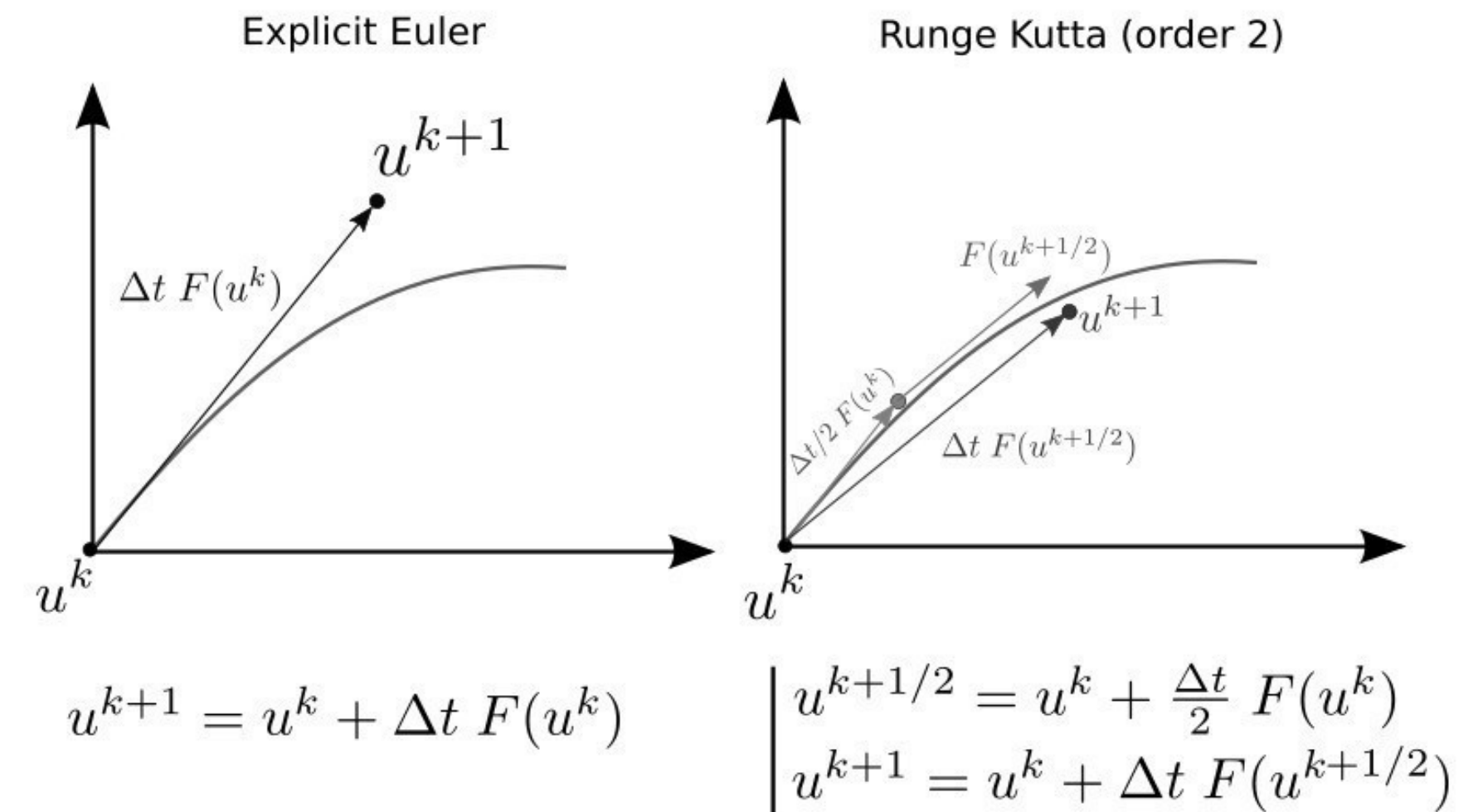
$$u'(t) = F(t, u)$$

Explicit Euler: $\frac{u^{k+1} - u^k}{\Delta t} = F(u^k)$

Implicit Euler: $\frac{u^{k+1} - u^k}{\Delta t} = F(u^{k+1})$
(F must be inverted)

020

Runge Kutta



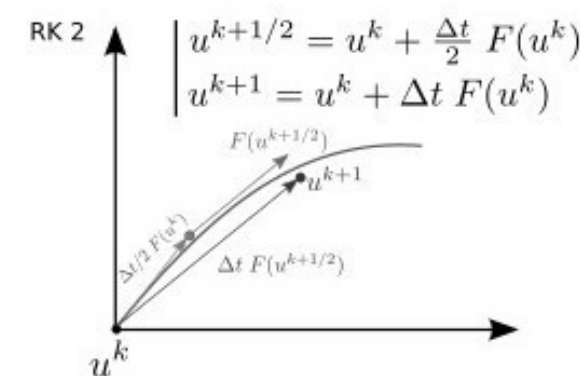
021

Runge Kutta: Free fall

$$F : u^k = (x^k, v^k) \mapsto \begin{pmatrix} v^k \\ g \end{pmatrix}$$

$$u^{k+1/2} = \begin{pmatrix} x^k + \Delta t/2 v^k \\ v^k + \Delta t/2 g \end{pmatrix}$$

$$F(u^{k+1/2}) = \begin{pmatrix} v^k + \frac{\Delta t}{2} g \\ g \end{pmatrix}$$

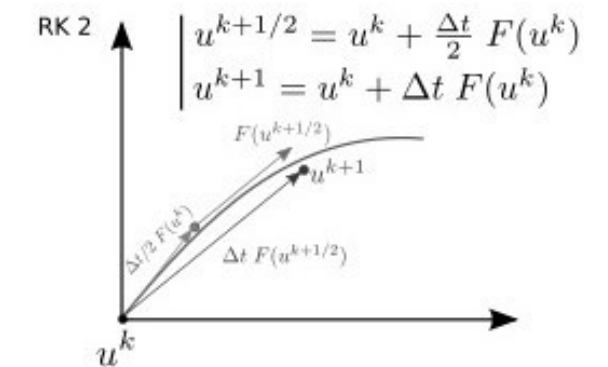


022

Runge Kutta: Free fall

$$\begin{aligned} x^{k+1} &= x^k + \Delta t v^k + \frac{(\Delta t)^2}{2} g \\ v^{k+1} &= v^k + \Delta t g \end{aligned}$$

$$\begin{cases} x^{k+2} = 2x^{k+1} - x^k + (\Delta t)^2 g \\ x^0 = x_0 \\ x^1 = x_0 + \Delta t v_0 + \frac{(\Delta t)^2}{2} g \end{cases} \Rightarrow \boxed{x^k = x_0 + (k \Delta t) v_0 + \frac{(k \Delta t)^2}{2} g}$$



023

Runge Kutta: ode45

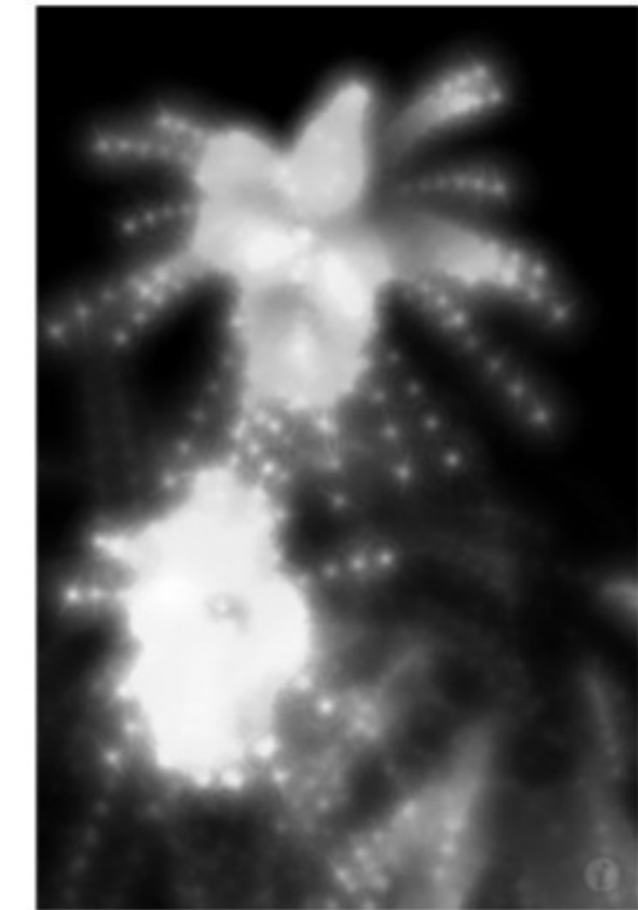
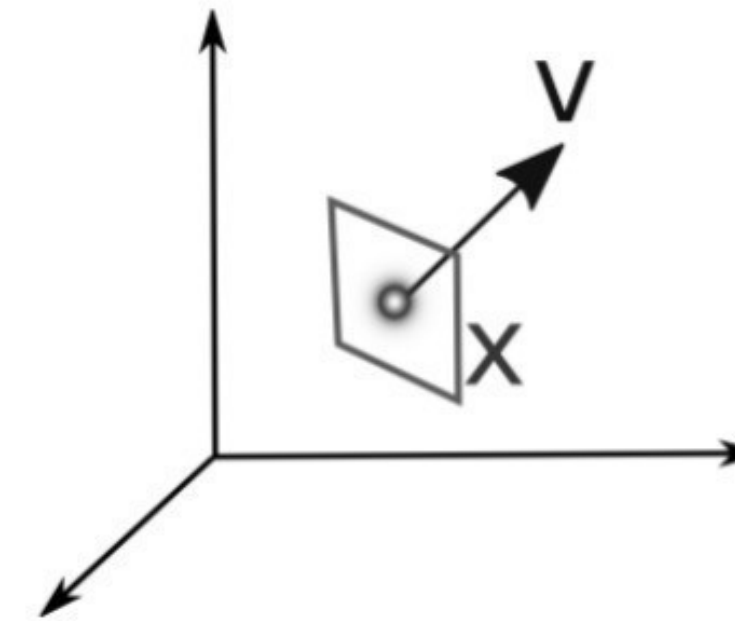
$$\begin{aligned}
 k_1 &= F(t_n, u_n) \\
 k_2 &= F\left(t_n + \frac{1}{5}\Delta t, u_n + \frac{1}{5}\Delta t k_1\right) \\
 k_3 &= F\left(t_n + \frac{3}{10}\Delta t, u_n + \left(\frac{3}{40}k_1 + \frac{9}{40}k_2\right)\Delta t\right) \\
 k_4 &= F\left(t_n + \frac{3}{5}\Delta t, u_n + \left(\frac{3}{10}k_1 - \frac{9}{10}k_2 + \frac{6}{5}k_3\right)\Delta t\right) \\
 k_5 &= F\left(t_n + \Delta t, u_n + \left(-\frac{11}{54}k_1 + \frac{5}{2}k_2 - \frac{70}{27}k_3 + \frac{35}{27}k_4\right)\Delta t\right) \\
 k_6 &= F\left(t_n + \frac{7}{8}\Delta t, u_n + \left(\frac{1631}{55296}k_1 + \frac{175}{512}k_2 - \frac{575}{13824}k_3 + \frac{44275}{110592}k_4 + \frac{253}{4096}k_5\right)\Delta t\right) \\
 u_{n+1}^4 &= u_n + \Delta t \left(\frac{2825}{27648}k_1 + \frac{18575}{48384}k_3 + \frac{13525}{55296}k_4 + \frac{277}{14336}k_5 + \frac{1}{4}k_6\right) \\
 u_{n+1}^5 &= u_n + \Delta t \left(\frac{37}{378}k_1 + \frac{250}{621}k_3 + \frac{125}{594}k_4 + \frac{512}{1771}k_6\right)
 \end{aligned}$$

024

Application to particles systems

Sprites:

Particles falling under gravity
Limited life time
Fill with animated transparent texture



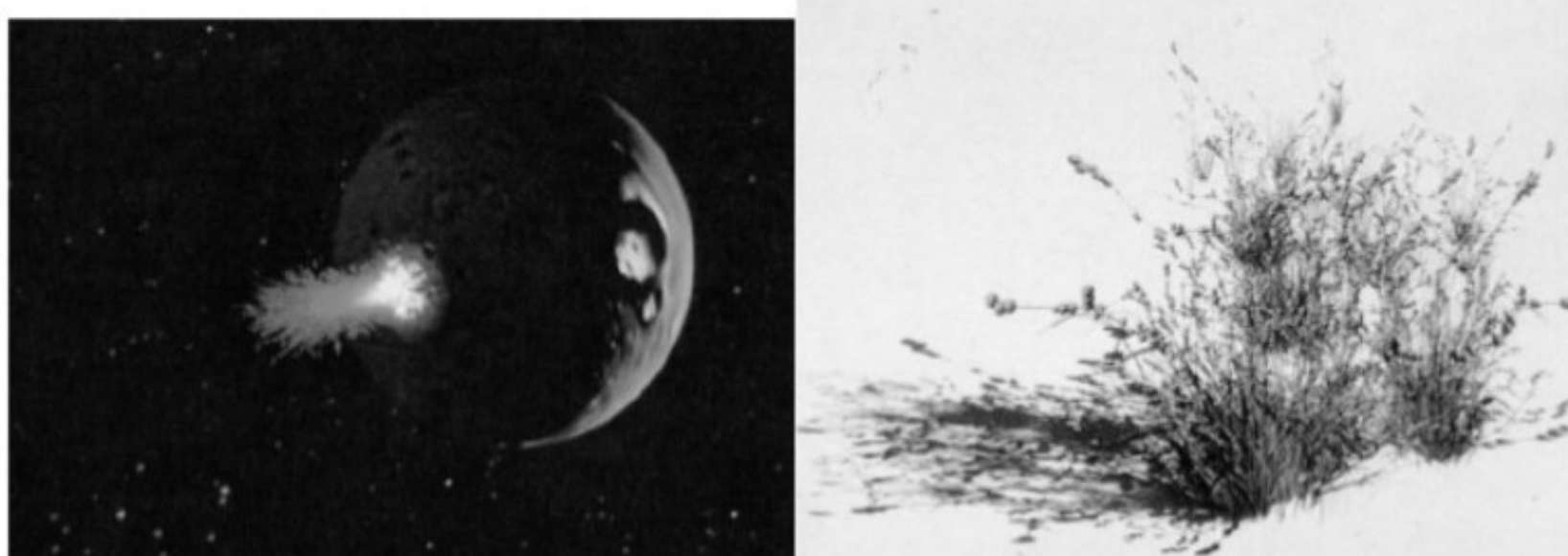
© Apple

025

Application to particles systems

Drawing trajectories

[William T. Reeves. **Particle Systems. A Technique for Modeling a Class of Fuzzy Objects.** *ACM Transaction on Graphics*, 17(3). 1983]



© Lucasfilm, Star Trek II

[Reeves, TOG 83]

026

Spring Mass System

Spring force

$$F(t) = K(L_0 - x(t))$$

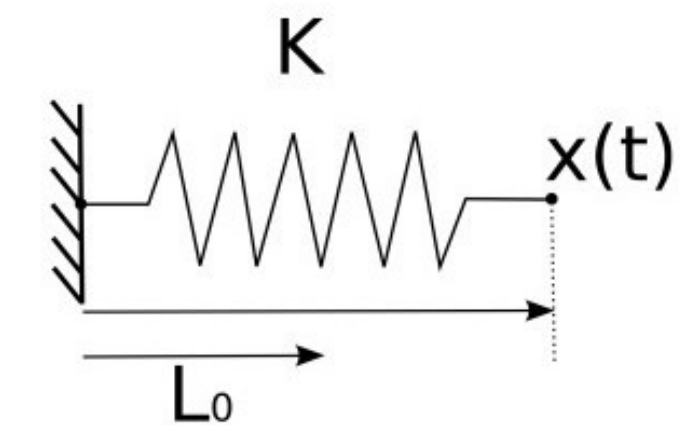
Equation of motion

$$x''(t) = K/m(L_0 - x(t))$$

ODE formulation

$$u'(t) = Au(t) + B$$

$$A = \begin{pmatrix} 0 & 1 \\ -K/m & 0 \end{pmatrix} \quad b = \begin{pmatrix} 0 \\ L_0 K/m \end{pmatrix}$$

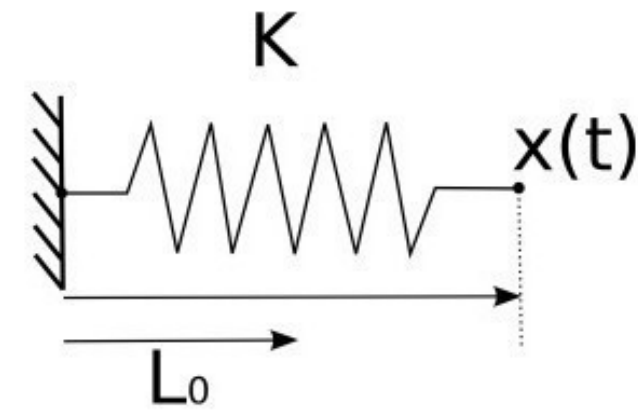


027

Spring Mass System

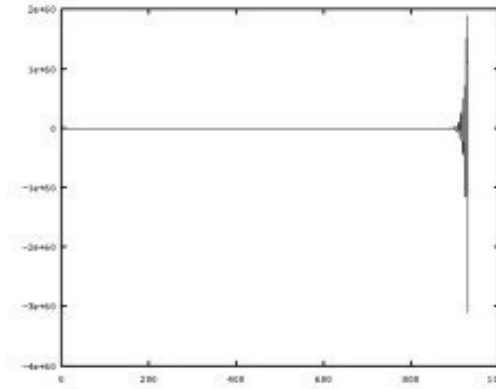
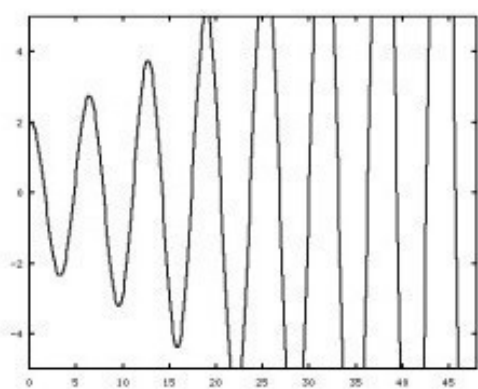
Spring force

$$F(t) = K(L_0 - x(t))$$



Explicit Euler:

$$x^{k+2} = 2x^{k+1} - \left(1 + (\Delta t)^2 \frac{K}{m}\right) x^k + (\Delta t)^2 \frac{K}{m} L_0$$



Expect:

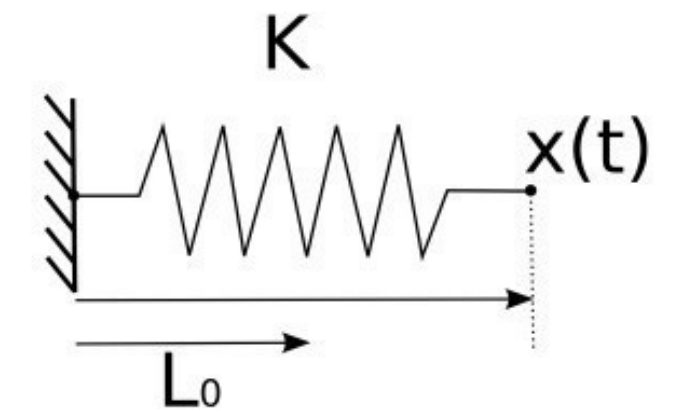
$$x(t) = A \sin(\omega t + \varphi)$$

028

Spring Mass System

Spring force

$$F(t) = K(L_0 - x(t))$$



Explicit Euler:

$$x^{k+2} = 2x^{k+1} - \left(1 + (\Delta t)^2 \frac{K}{m}\right) x^k + (\Delta t)^2 \frac{K}{m} L_0$$

Do not diverge if $1 + \sqrt{(\Delta t)^2 \frac{K}{m}} < 1$

=> Always diverge to infinity !

029

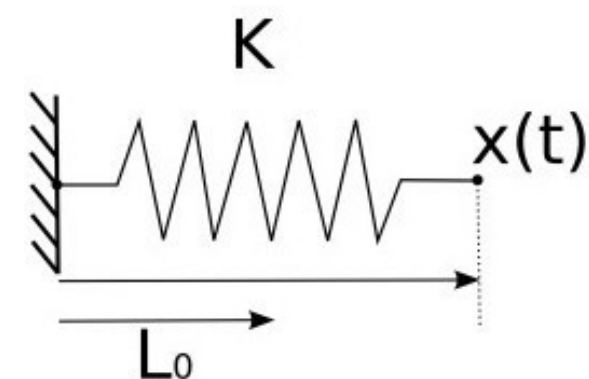
Spring Mass System

Spring force

$$F(t) = K(L_0 - x(t))$$

Add fluid damping

$$F_d(t) = -\mu v(t)$$



New equation for explicit Euler:

$$x^{k+2} - \left(2 - \frac{\mu}{m} \Delta t\right) x^{k+1} + \left(1 + (\Delta t)^2 \frac{K}{m} - \frac{\mu}{m} \Delta t\right) x^k = (\Delta t)^2 \frac{K}{m} L_0$$

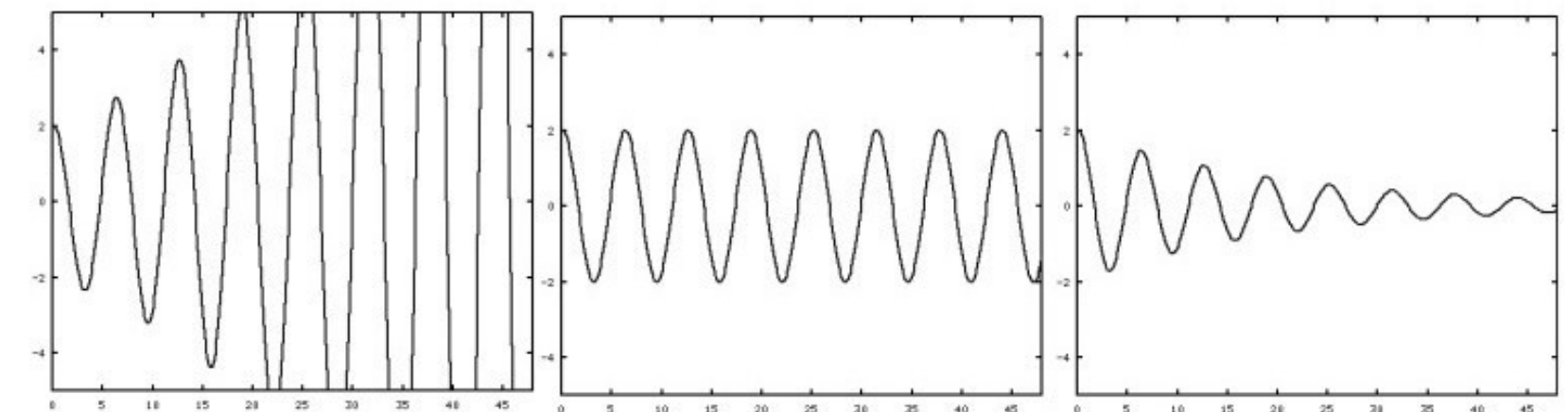
Conditionnaly stable

Large K => Stiff springs

Stiff ODE

030

Accuracy != Stability



031

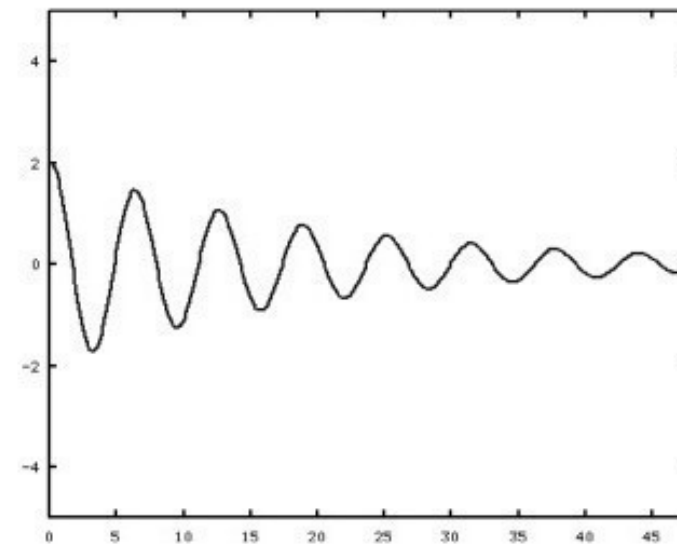
Implicit Euler

$$\begin{pmatrix} 1 & -\Delta t \\ -\Delta t \frac{K}{m} & 1 \end{pmatrix} u^{k+1} = u^k + \Delta t \begin{pmatrix} 0 \\ K \frac{L_0}{m} \end{pmatrix}$$

M

Eigenvalues of M^{-1} $1 - \Delta t \sqrt{\frac{K}{m}} < 1$

Unconditionally stable

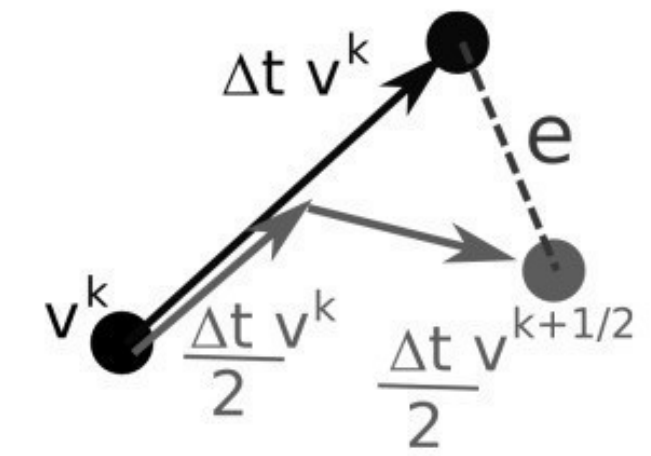


032

Automatic step-size

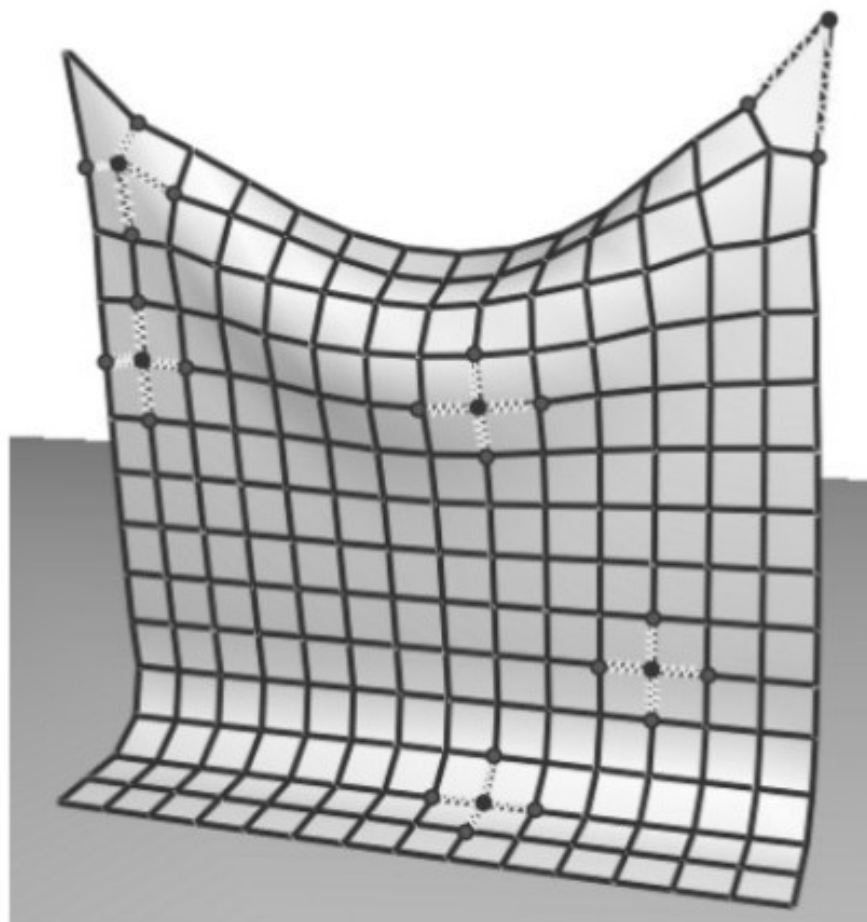
Compute:

- $u_1^{k+1} = u^k + \Delta t F(u^k)$
- $\begin{cases} u_2^{k+1/2} = u^k + \Delta t/2 F(u^k) \\ u_2^{k+1} = u_2^{k+1/2} + \Delta t/2 F(u_2^{k+1/2}) \end{cases}$
- $e = \|u_1^{k+1} - u_2^{k+1}\|$
 $\begin{cases} e < K_{\max} \Rightarrow (\Delta t)_{\text{new}} = \Delta t/2 \\ e < K_{\min} \Rightarrow (\Delta t)_{\text{new}} = 2\Delta t \end{cases}$



033

Cloth Simulation

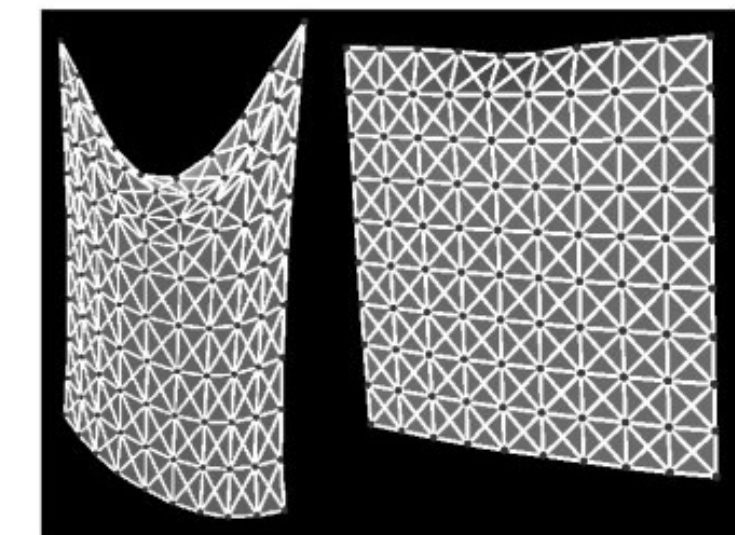
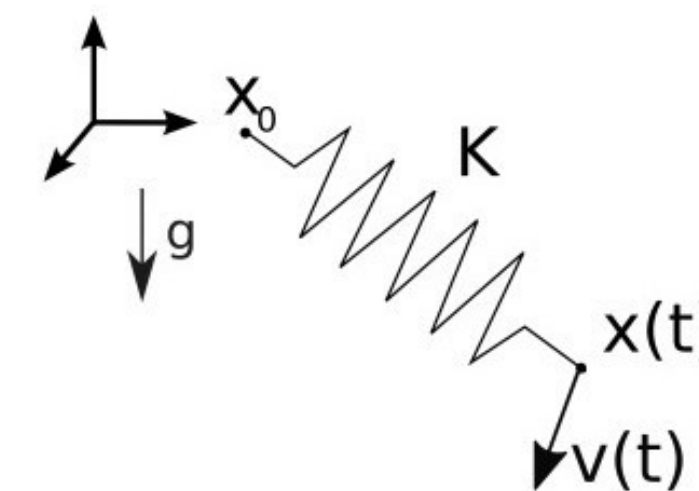


034

Cloth Simulation

In 3D:

$$F(t) = K (L_0 - \|\mathbf{p} - \mathbf{p}_0\|) \frac{\mathbf{p} - \mathbf{p}_0}{\|\mathbf{p} - \mathbf{p}_0\|}$$



035

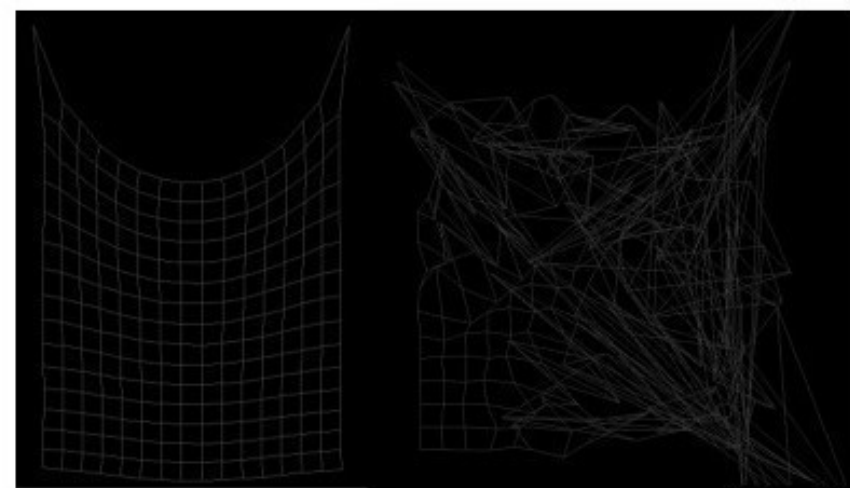
Cloth Simulation: mass springs

Model cloth using coupled springs
Force on a vertex i with neighbors $\mathcal{V}_i$

$$F(x_i, t) = \sum_{j \in \mathcal{V}_i} K^{ij} \left(L_0^{ij} - \|x_i - x_j\| \right) \frac{x_i - x_j}{\|x_i - x_j\|} + g$$

$$\forall i, \begin{cases} x'_i(t) = v_i(t) \\ v'_i(t) = \frac{1}{m_i} \sum_j K^{ij} \left(L_0^{ij} - \|x_i - x_j\| \right) \frac{x_i - x_j}{\|x_i - x_j\|} + g \end{cases}$$

Use you best interpolation scheme



[P. Jacobs]

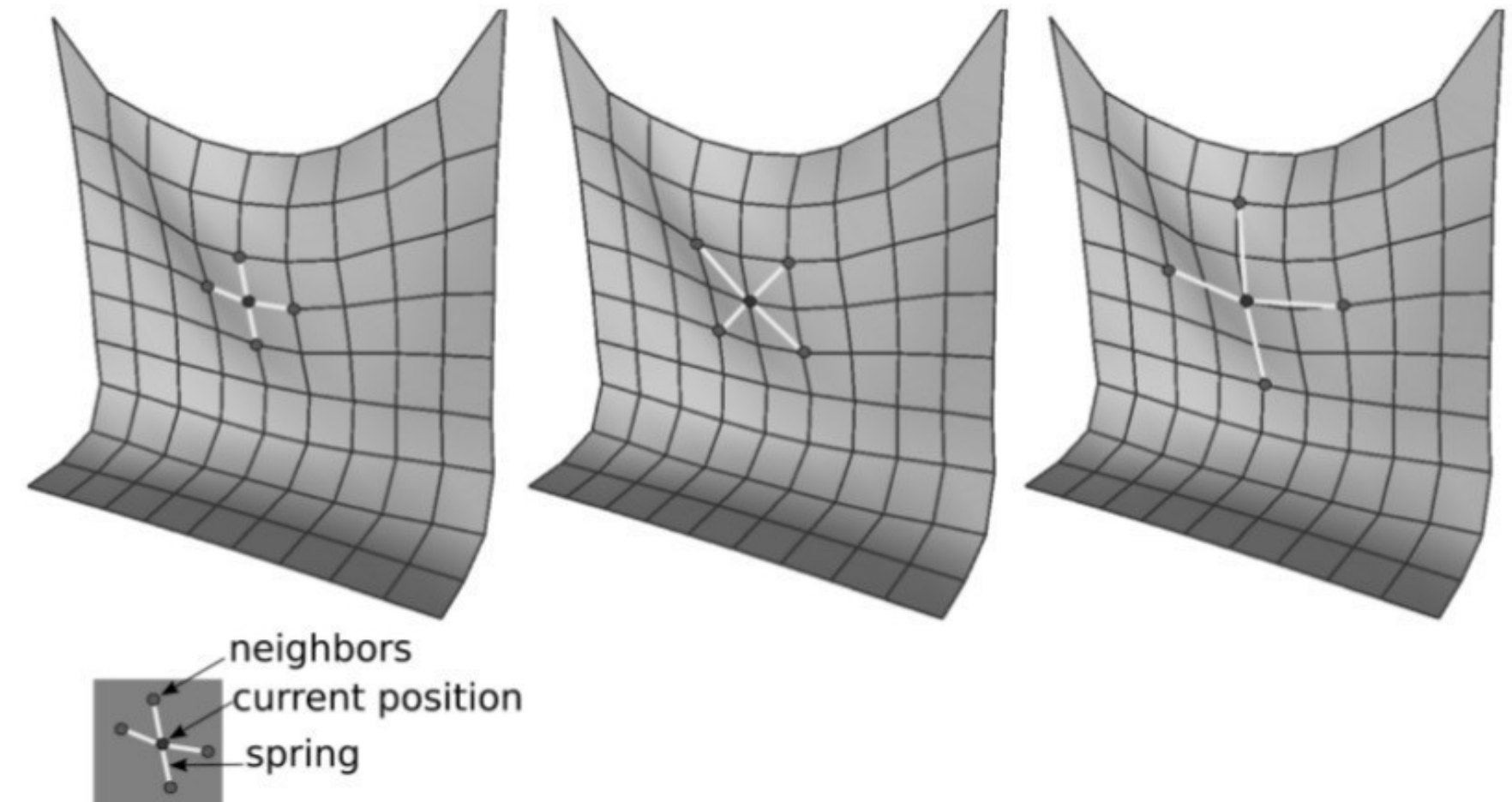
036

Cloth Simulation: spring types

Structural
springs

Shearing
springs

Bending
springs



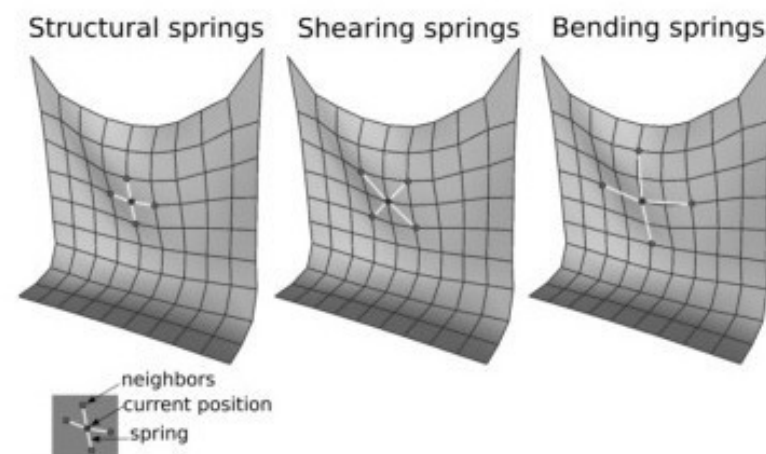
037

Cloth Simulation: Resolution

Explicit Euler

```
//Compute forces
For all i
  For j:  4 direct neighbors (structural): K=K1
         4 diagonal neighbors (shear)   : K=K2
         8 neighbors (bend)            : K=K3
  u=p[i]-p[j]
  F[i] += K (L0-norm(u)) *u/norm(u)
```

```
For all i
  v[i] += dt * F[i]
  p[i] += dt * v[i]
```



038

Complex cloth simulation

Full simulation = Cloth + complex collisions



[Grinspun, SIGGRAPH 09]

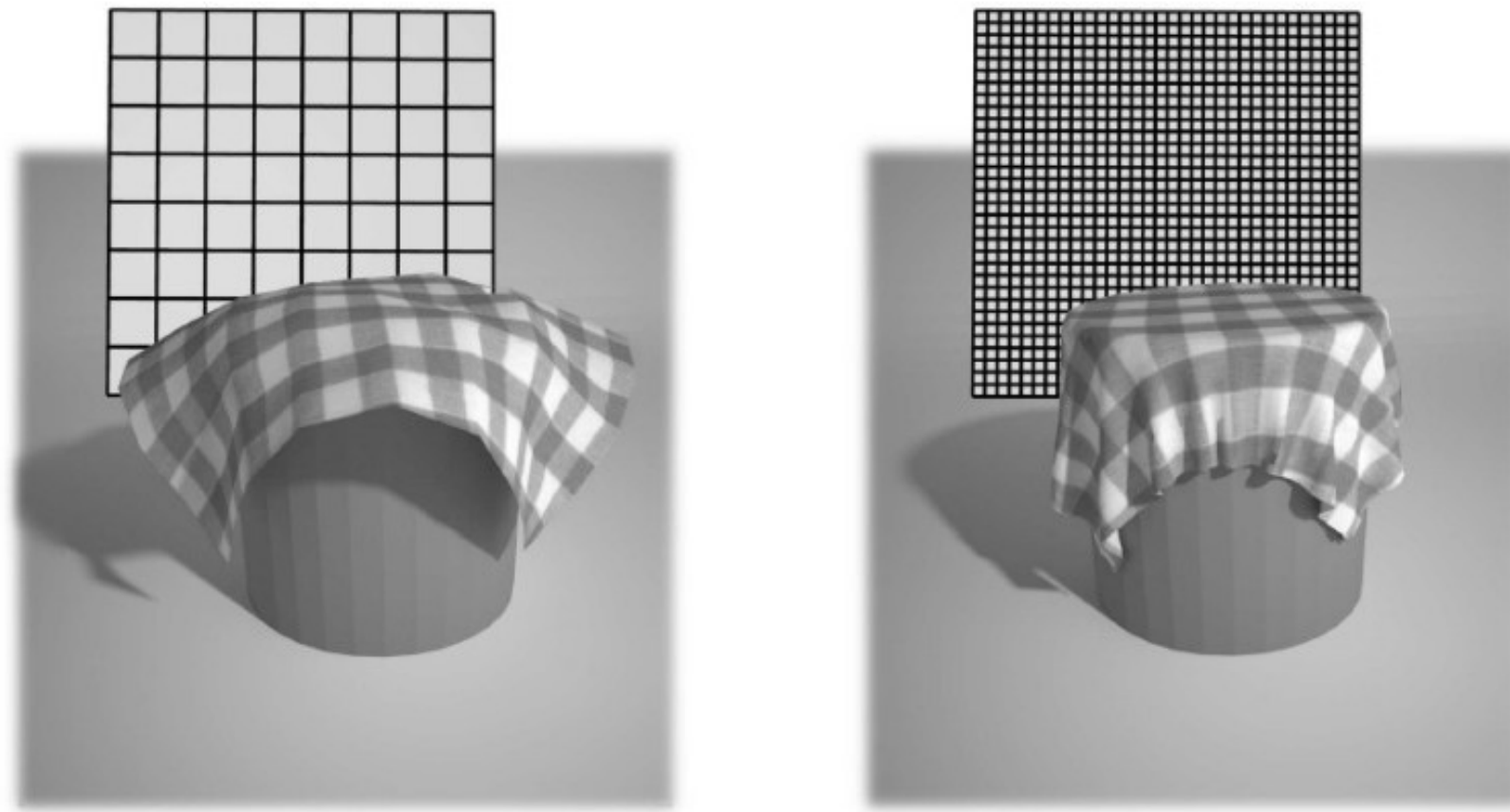


[DAZ3D, Dynamic Clothing]

039

Cloth simulation: Limitations

Mesh influence



040

Implicit scheme

Write the system in a vectorial form

$$u(t) = (p_0(t), p_1(t), \dots, p_{N-1}(t), v_0(t), \dots, v_{N-1}(t))$$

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

Non linear system with N unknown: impossible to invert

Linearize the system in $p(t) - p_0 = \Delta L v(t) + \mathcal{O}((\Delta t)^2)$

Get a linear system.

Loss of the unconditional stability:

In practice, very stable

041

Implicit scheme

Linearization:

$$\mathbf{f}(\mathbf{p}_n + \Delta p, \mathbf{v}_n + \Delta v) = \mathbf{f}_n + \frac{\partial \mathbf{f}}{\partial \mathbf{p}} \Delta p + \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \Delta v$$

Solve a linear system at each time step

$$\mathbf{A} \Delta v = \mathbf{b}$$

$$\mathbf{A} = \mathbf{I} - \Delta t \mathbf{M}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{v}} - \Delta t^2 \mathbf{M}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{p}}$$

$$\mathbf{b} = \Delta t \mathbf{M}^{-1} \left(\mathbf{f}_n + \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{p}} \mathbf{v}_n \right)$$

042