Nucleus

Towards a Unified Dynamics Solver for Computer Graphics

Jos Stam

Senior Research Scientist

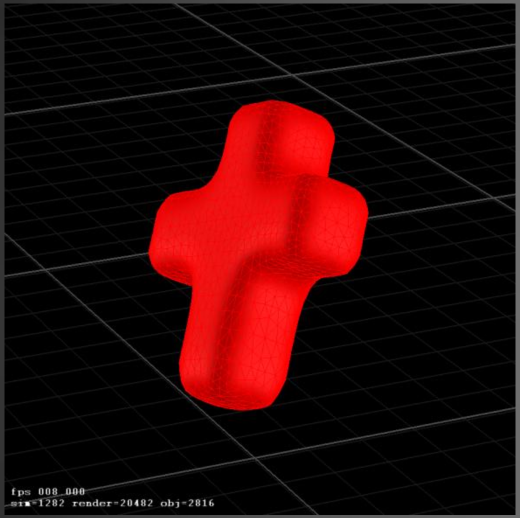
Autodesk, Inc.

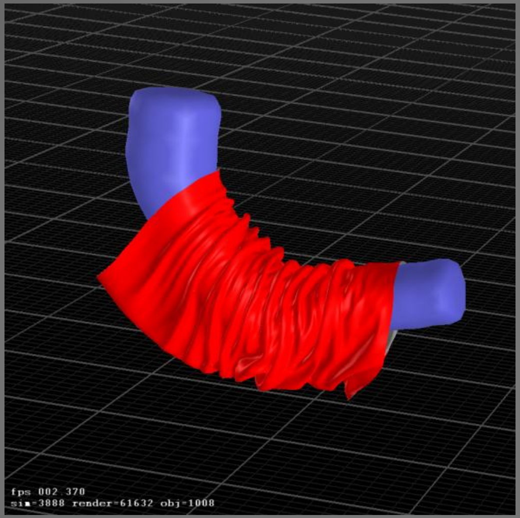
Toronto, Canada

Motivation

Ruysdael (1628-1682)

Too many Solvers





rigid bodies

cloth

fluids

Solvers have to interact

Unified Solver

Two-way interaction

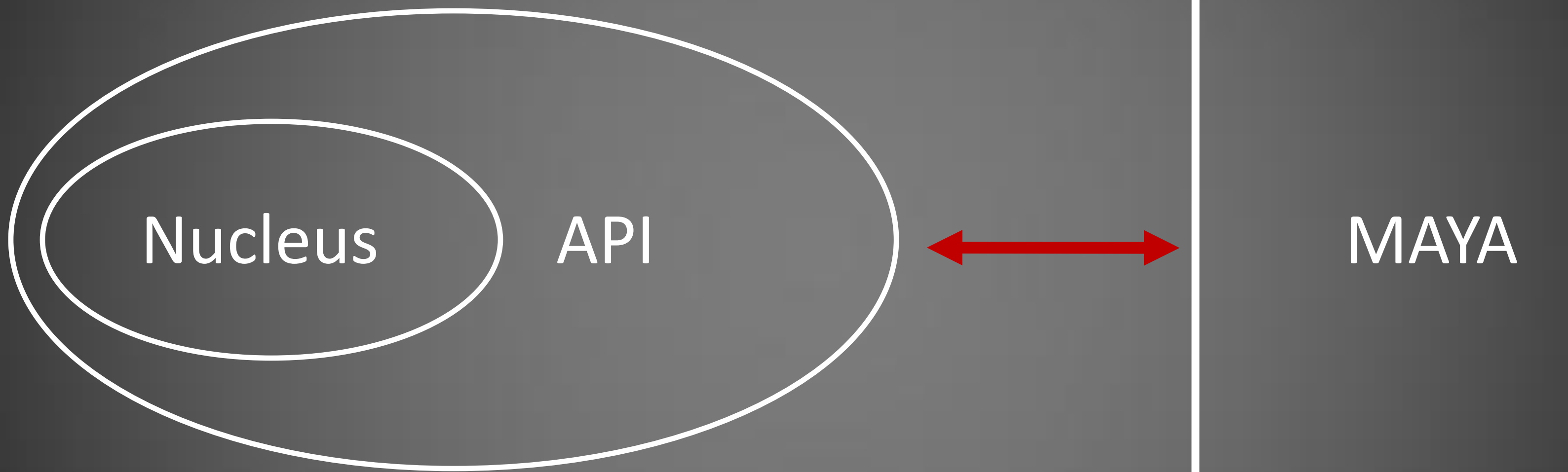
Unified Solver

King Dome, Seattle, March 26, 2000

History

Simple idea	: 2000
Fooling around (palm demo)	: 2001
First prototypes	: 2003-2005
First **real** implementation	: 2005
First public demo	: 2006
Released in MAYA 8.5 (nCloth)	: 2007
nParticles	: 2008

Nucleus



Nucleus

Nucleus

MAYA

~ 100 files

~ 40,000 files

Approach

General Shape Model

Stable Dynamics

Shape Model

“Simplicial Approximation Theorem”

1 Theorem (Brouwer 1910)

Every continuous mapping can be approximated by a piece-wise linear simplicial map.

Fundamental theorem of CG modeling

Shape Model

"Simplicial Approximation Theorem"

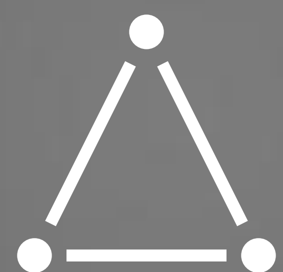
1 Theorem

Let K and L be complexes; let K be finite. Given a continuous map $h{:}|K| \to |L|$ there is an N such that h has a simplicial approximation $f : \mathrm{sd}^N K \to L$.

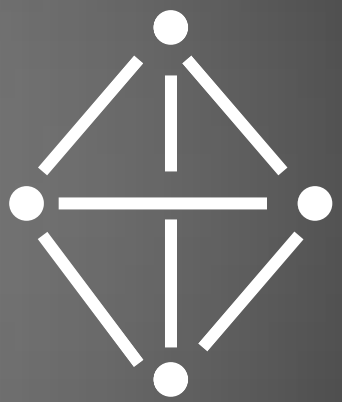
Simplicial?

0-simplex

1-simplex



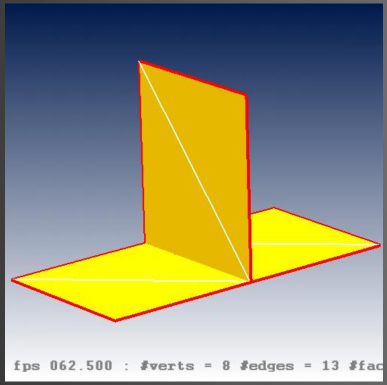
2-simplex

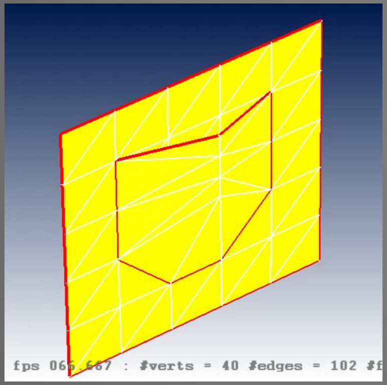


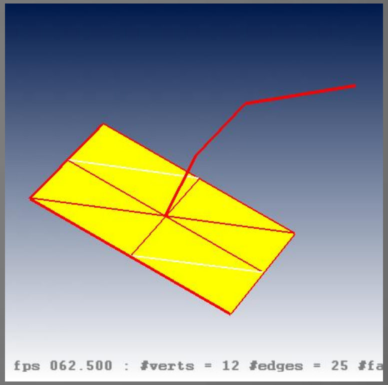
3-simplex

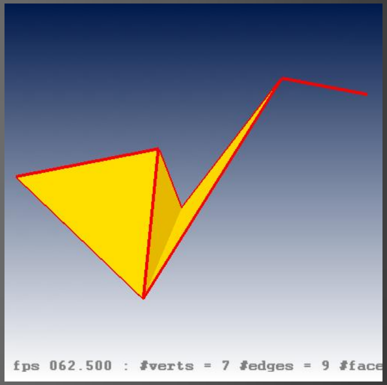
k-simplex

Simplicial Complex

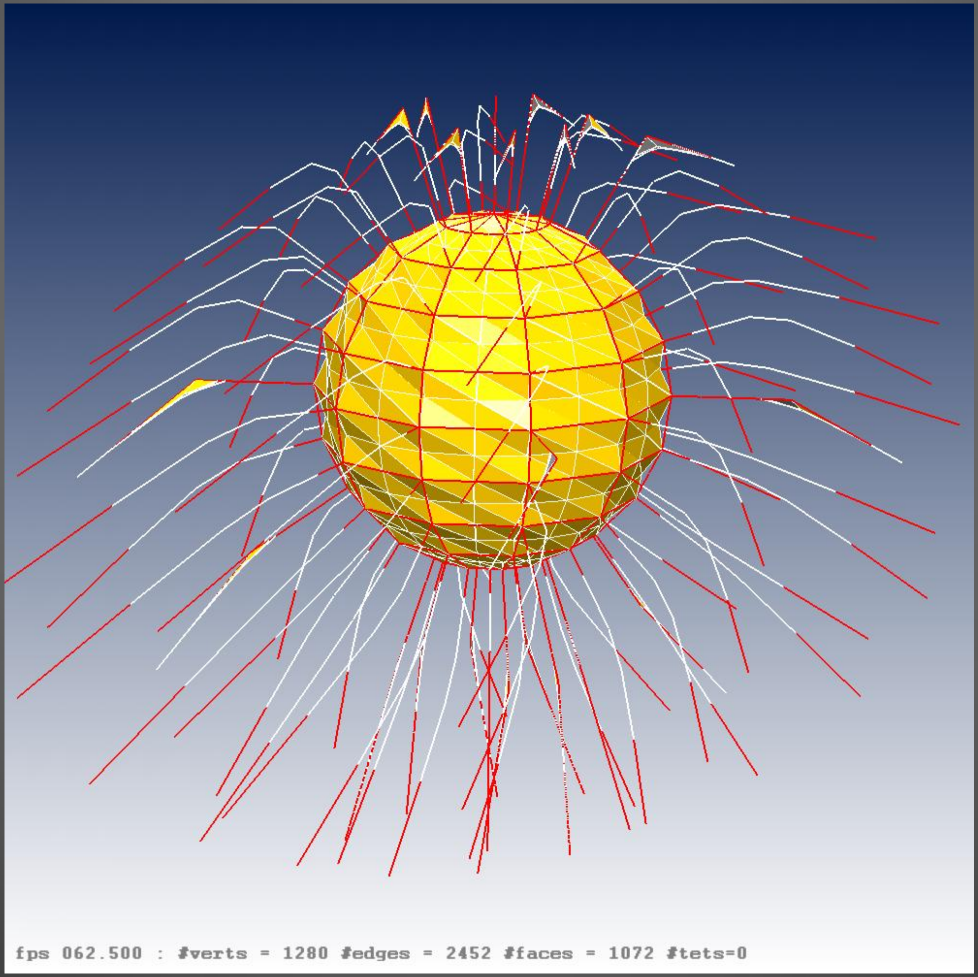




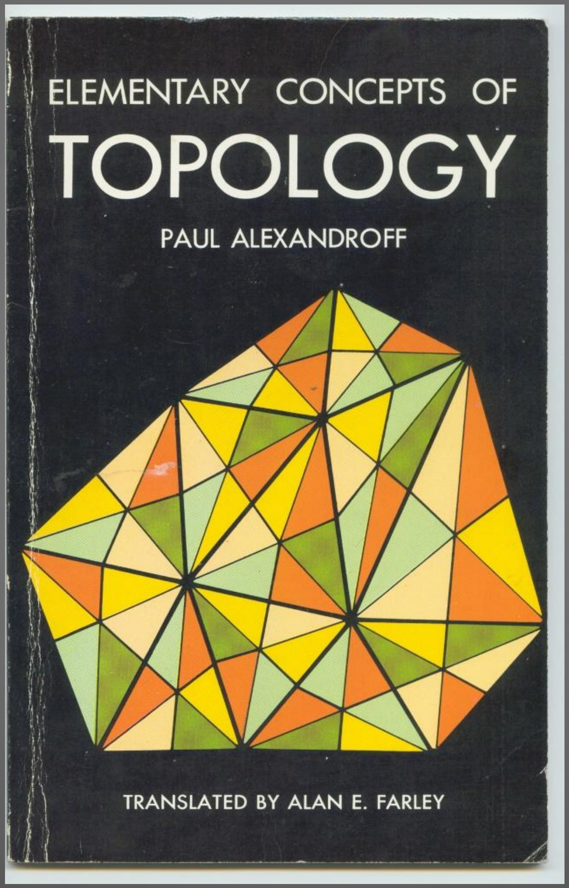




Simplicial Complex

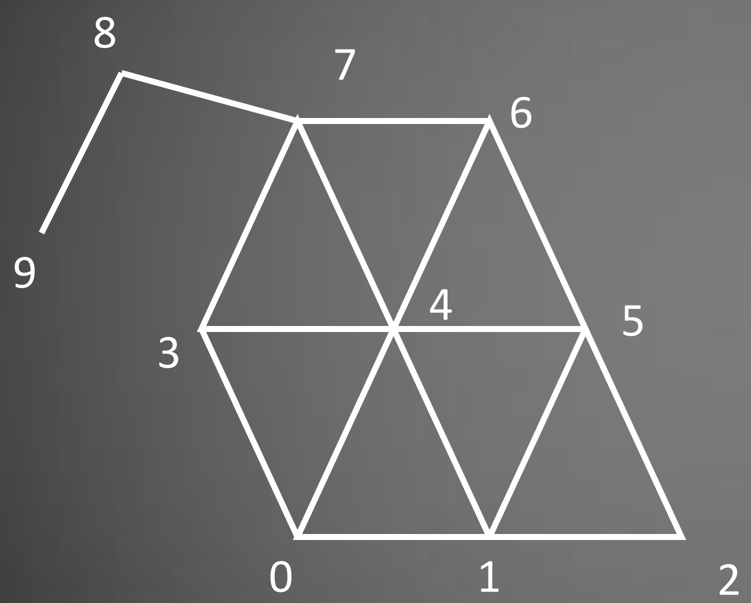


Simplicial Complex



Dover: $2.75

Shape Model



2 edges:
-(8,9) +(7,8)

7 triangles:
+(0,1,4), +(1,2,5)
-(0,3,4), -(1,4,5)
+(3,4,7), +(4,5,6),
+(4,6,7)

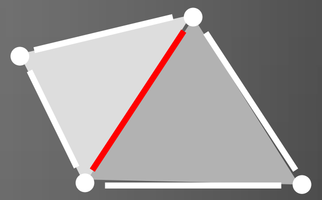
(i,j,k) = -(i,k,j)

Definition purely topological

Simplicial Complex

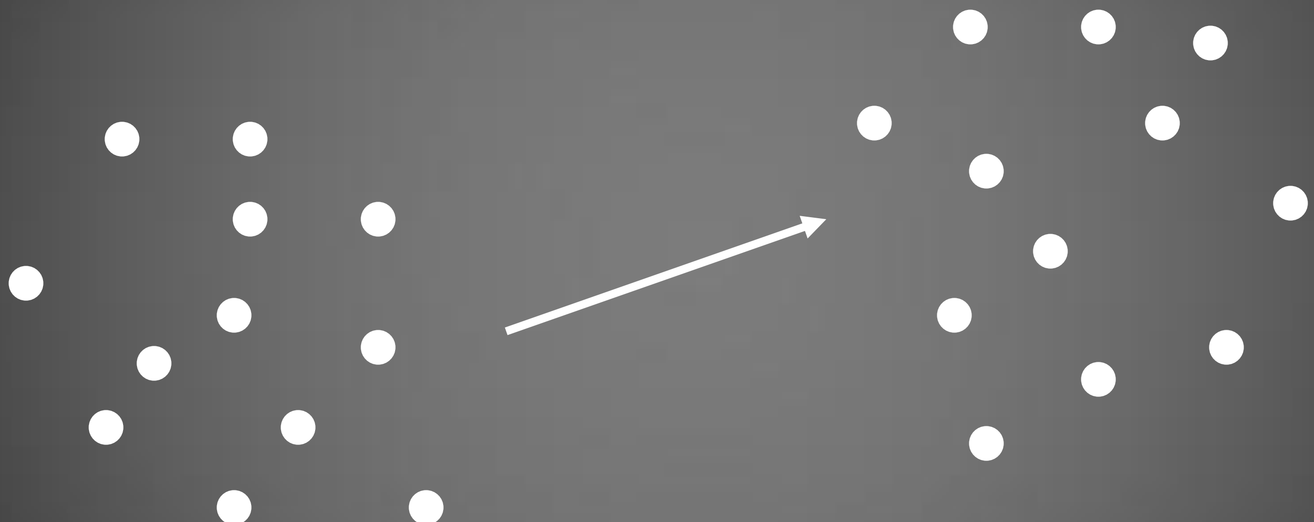
Single class for all primitives:

```
class simplex {
    int k;
    int sign;
    int vertex[k+1];
    int child[k+1];
    int n_parents;
    int parent[n_parents];
};
```



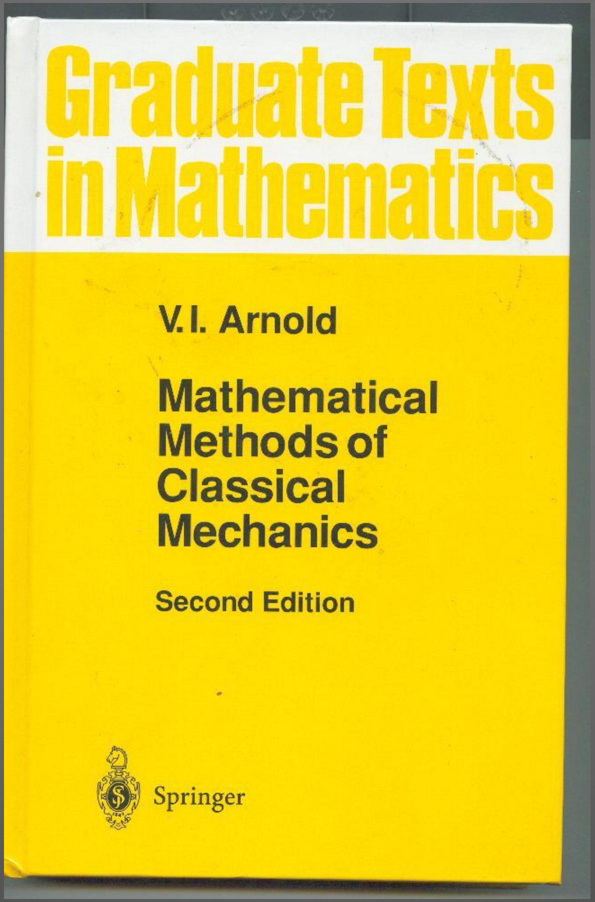
Code not specific for a primitive: simple

Dynamics



$$\mathbf{x}(t) = (x_1(t), \cdots, x_N(t)) \in \mathbf{R}^{3N}$$

Dynamics


Graduate Texts in Mathematics
V.I. Arnold
Mathematical Methods of Classical Mechanics
Second Edition
Springer

Newton's Law

$$\boxed{\ddot{\mathbf{x}} = -\nabla \mathbf{f}(\mathbf{x}) + \mathbf{f}_e}$$

$$\mathbf{x}(0) = \mathbf{x}_0$$

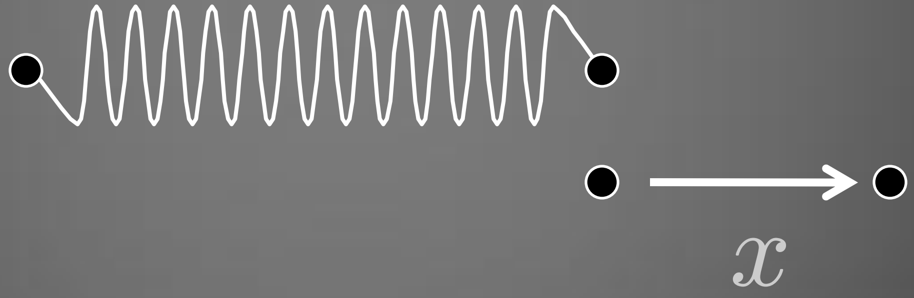
$$\dot{\mathbf{x}}(0) = \mathbf{v}_0$$

$$E = \frac{1}{2}\dot{\mathbf{x}}^2 + \mathbf{f}(\mathbf{x}) - \mathbf{f}_e \cdot \mathbf{x}$$

Isaac Newton

Isaac Newton's *Principia* 1687

Simple Example



Spring

$$\boxed{\ddot{x} = -x}$$

$$x(0) = x_0$$

$$\dot{x}(0) = v_0$$

$$E = \frac{1}{2}\dot{x}^2 + \frac{1}{2}x^2$$

Spring

$$\boxed{\ddot{x} = -x}$$

$$z = x + i\,\dot{x} \qquad \in \mathbf{C}$$

$$\boxed{\dot{z} = -i\,z}$$

$$z(0) = z_0$$

Spring

$$\boxed{\dot{z} = -i\,z}$$

$$z(0) = z_0$$

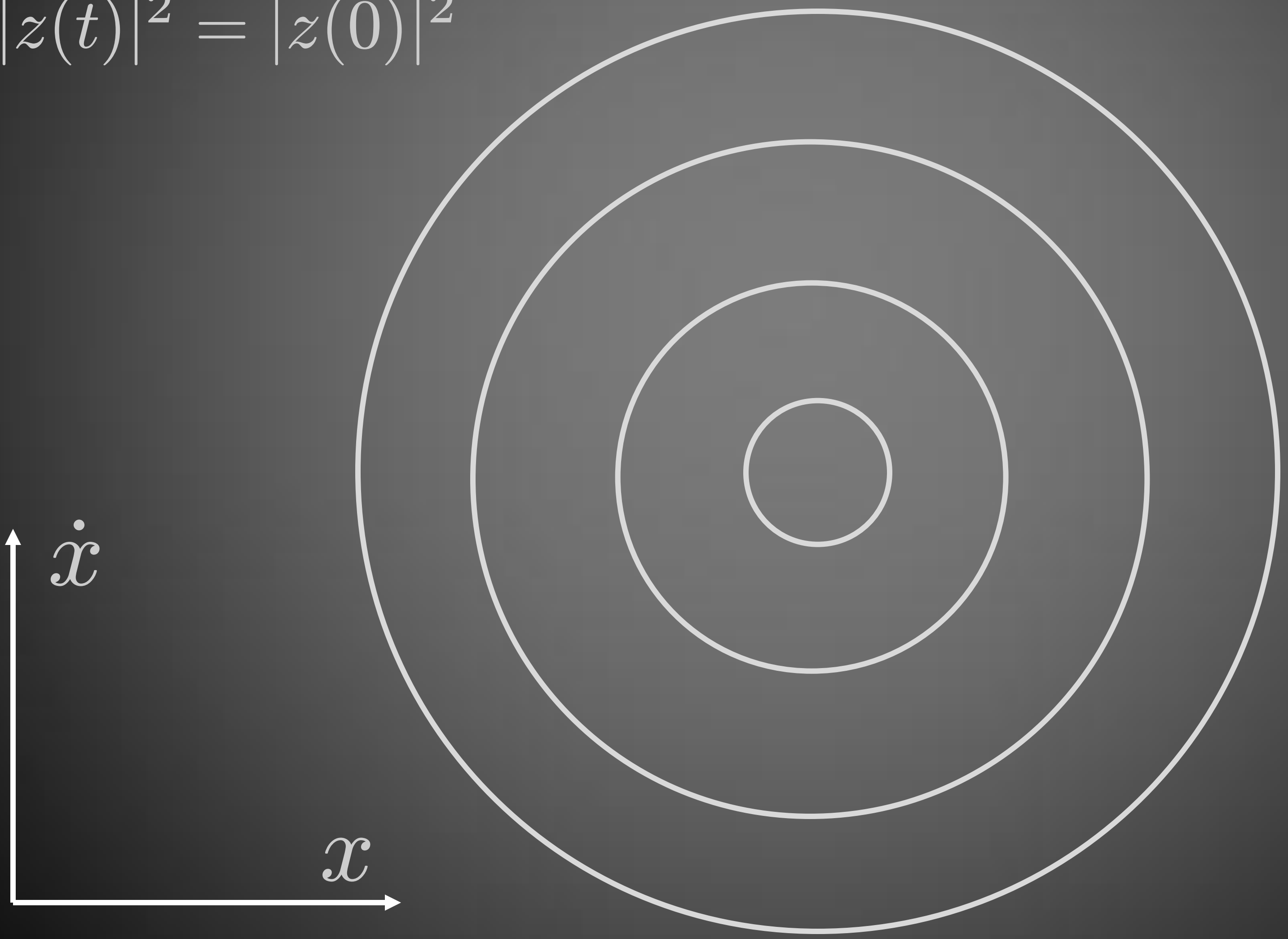
$$z(t) = e^{-it}\,z(0)$$

A Thought…

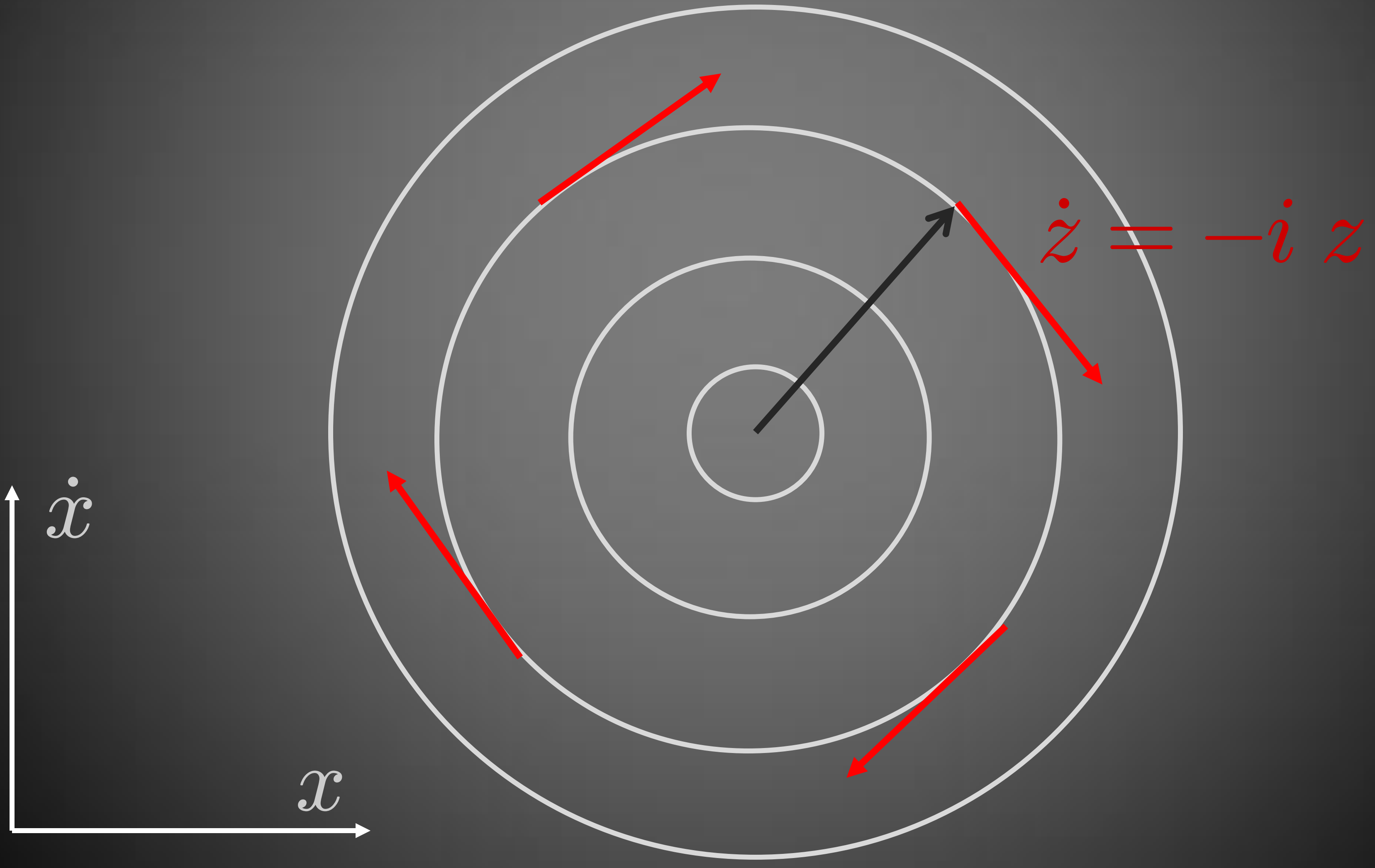
$$\mathbf{R}^{6n} \longrightarrow \mathbf{C}^{3n}$$

Spring

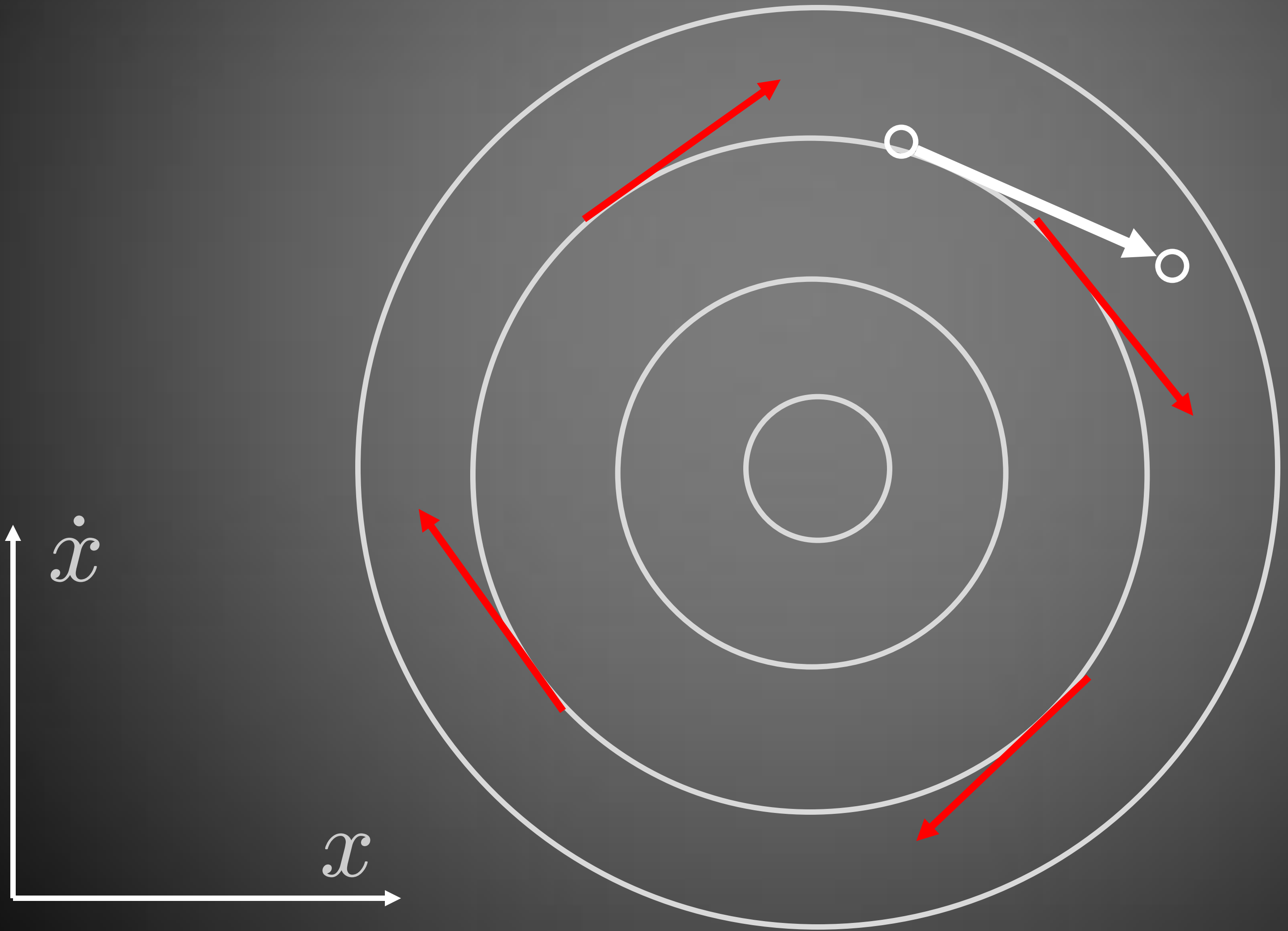
$$|z(t)|^2 = |z(0)|^2$$



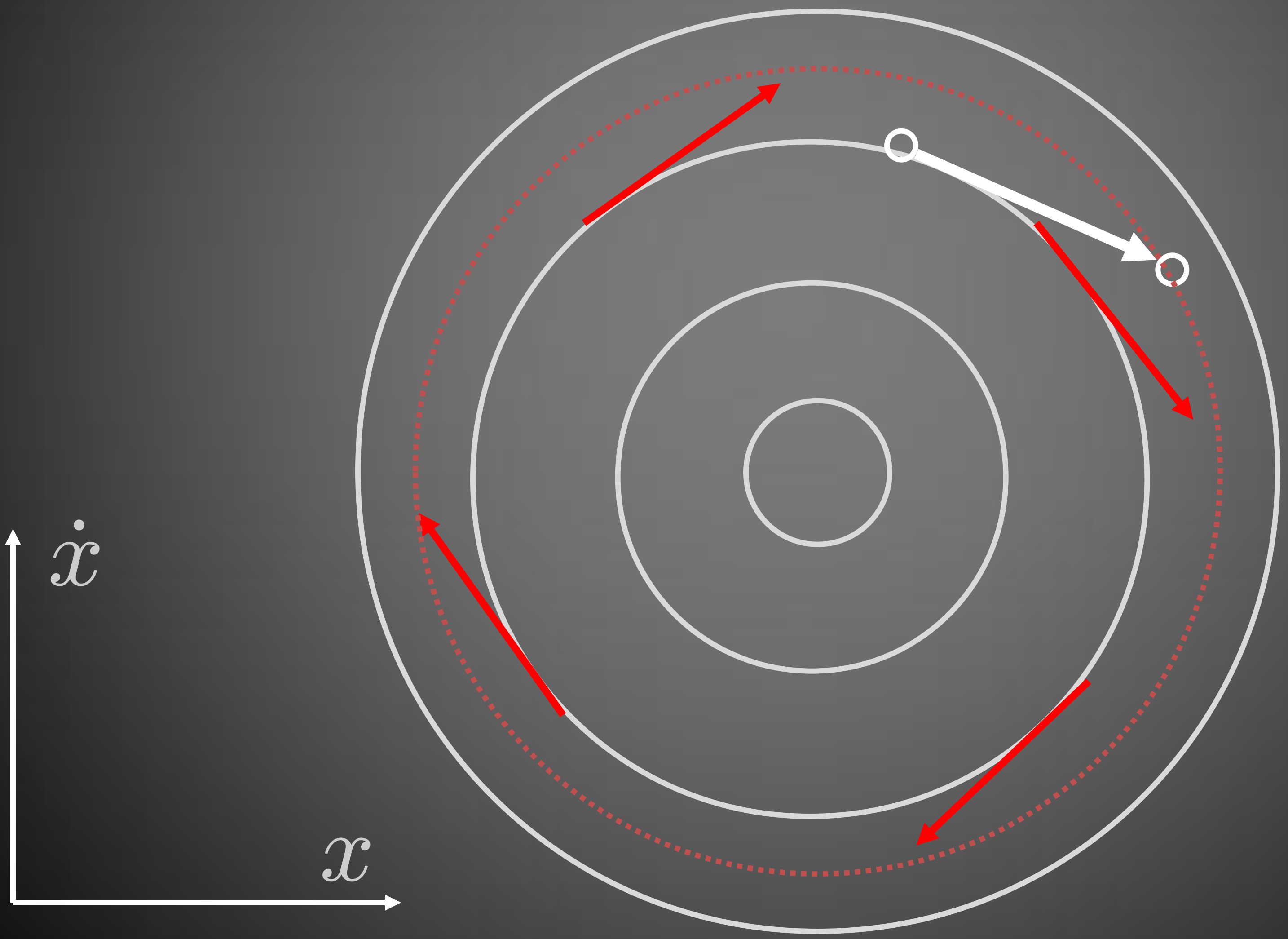
Spring



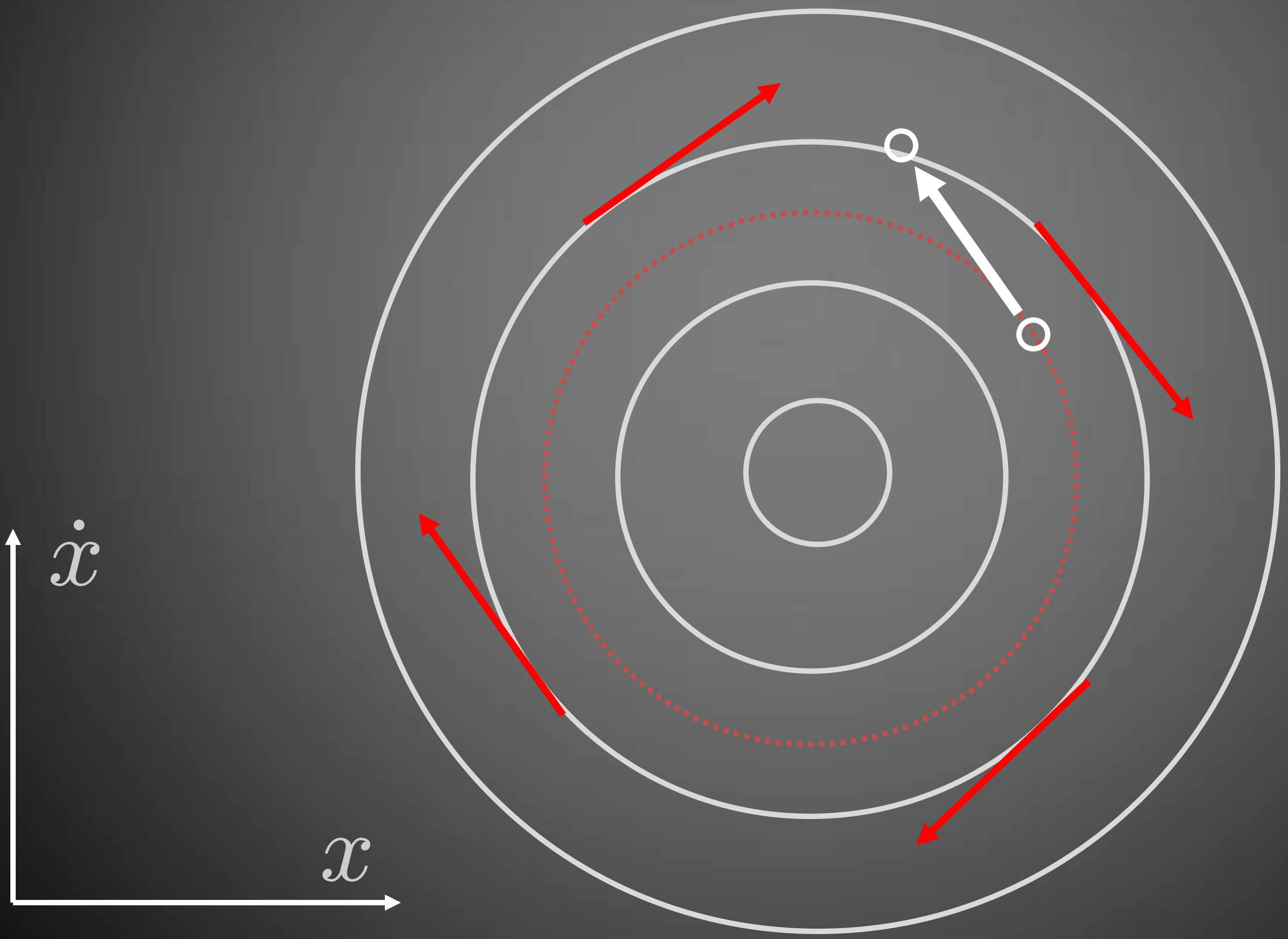
Explicit Solver



Explicit Solver



Implicit Solver



Explicit Solver

$$\frac{1}{h}\left(z^n - z^{n-1}\right) = -i\, z^{n-1}$$

$$z^n = z^{n-1} - ih\, z^{n-1}$$

$$z^n = \left(1 + h^2\right) e^{-ih}\ z^{n-1}$$

$$z^n = \left(1 + h^2\right)^n e^{-inh}\ z^0$$

Implicit Solver

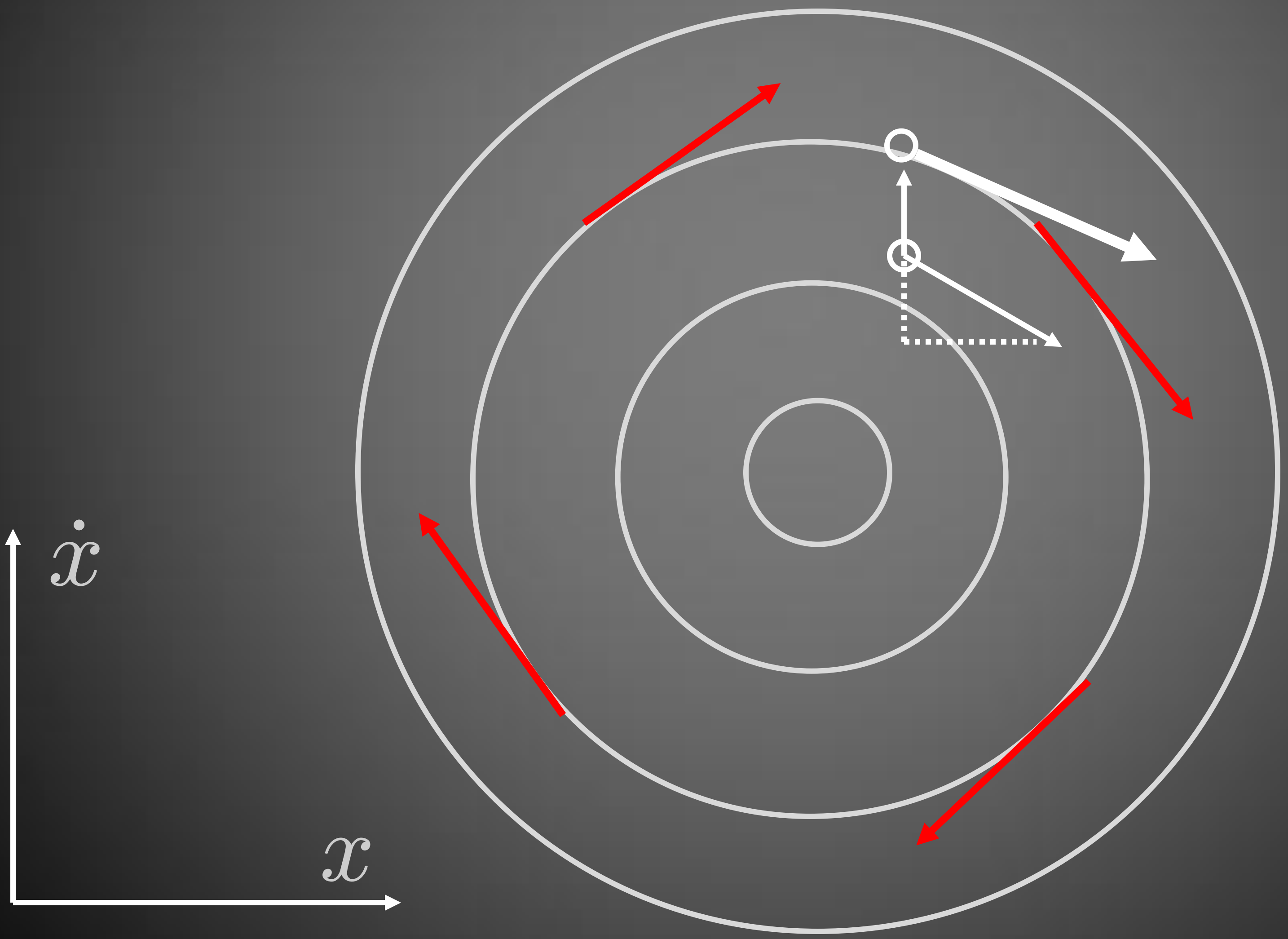
$$\frac{1}{h}\left(z^n - z^{n-1}\right) = -i\, z^n$$

$$(1 + ih)\, z^n = z^{n-1}$$

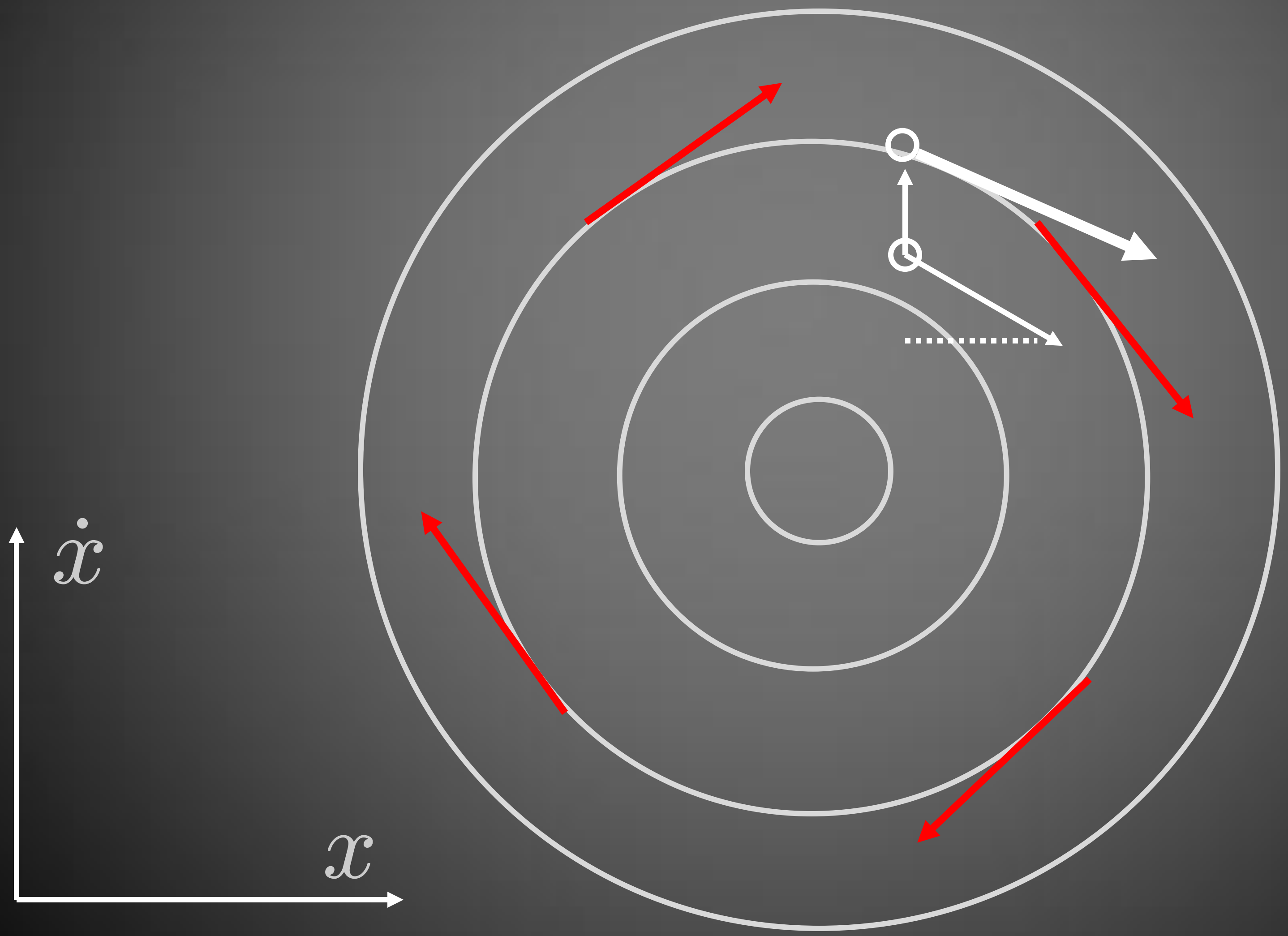
$$z^n = \frac{1}{1+h^2} e^{-ih}\ z^{n-1}$$

$$z^n = \frac{1}{(1+h^2)^n} e^{-inh}\ z^0$$

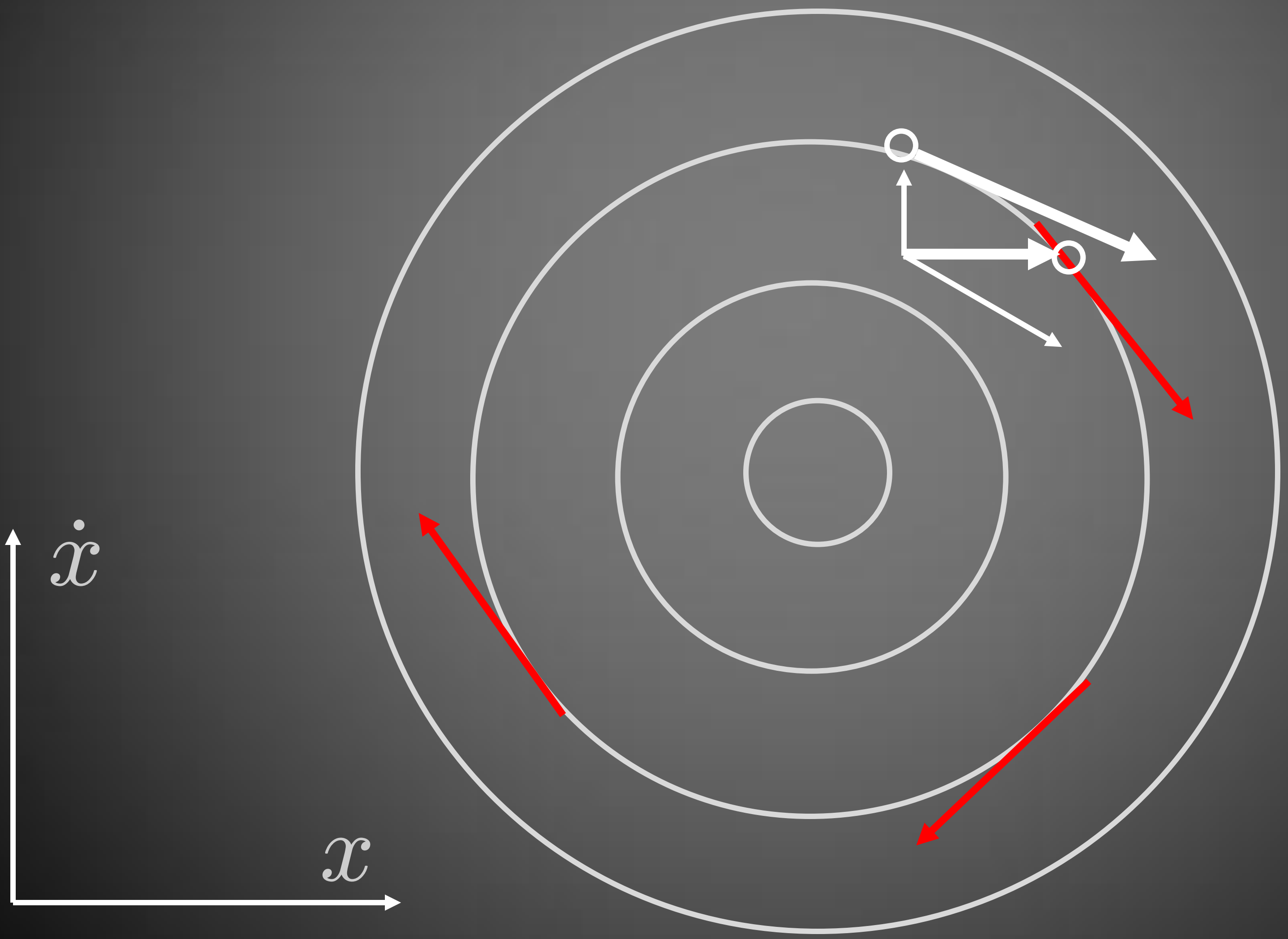
Symplectic Solver



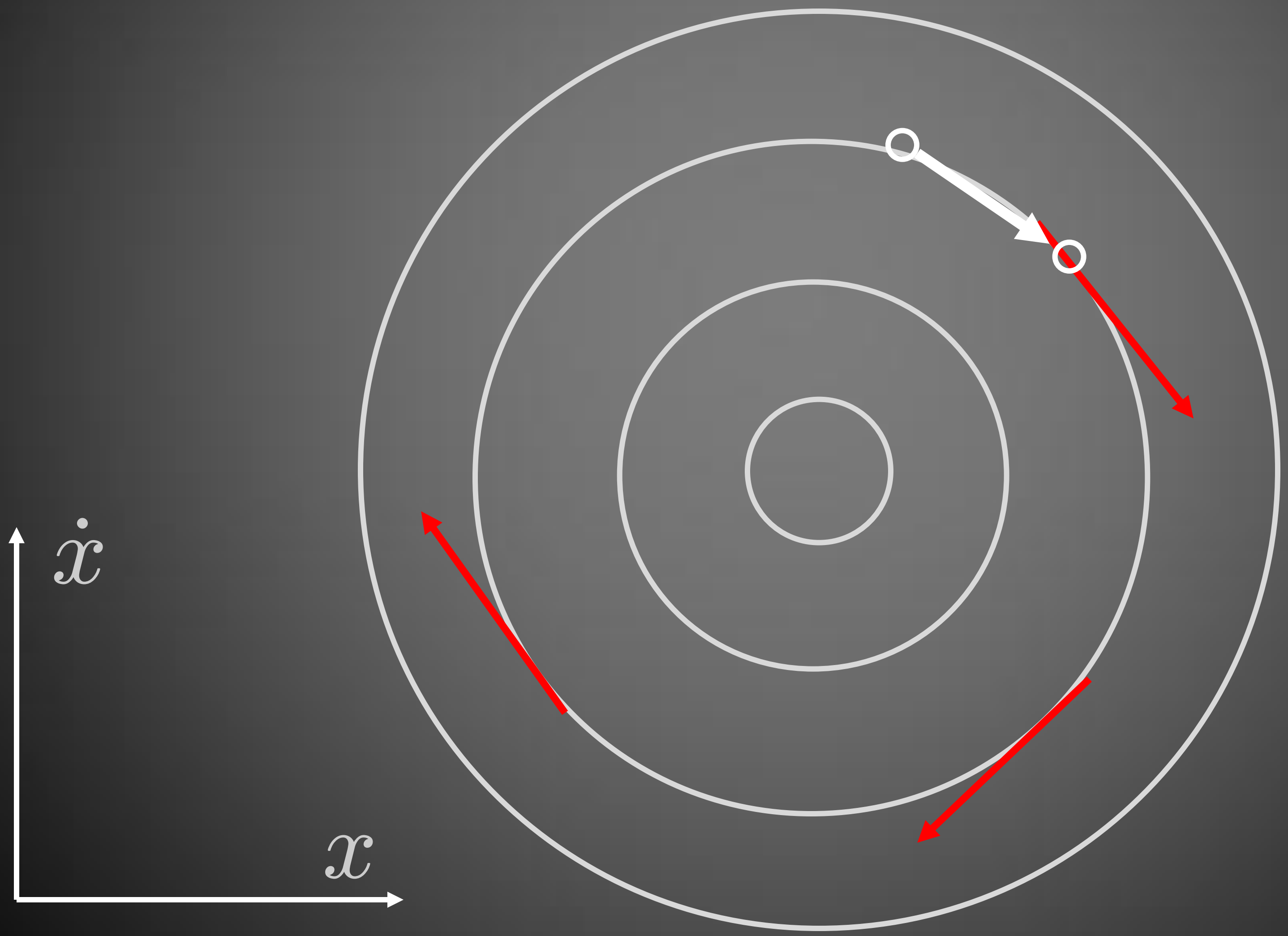
Symplectic Solver



Symplectic Solver



Symplectic Solver



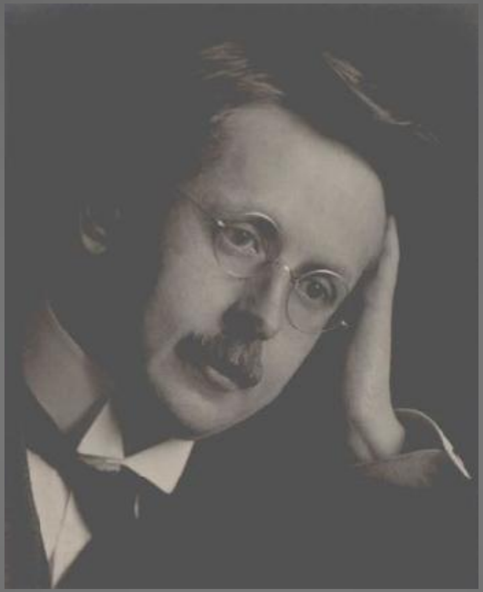
Symplectic?

“Official Definition”:

“Plaiting or joining together; -- said of a bone next above the quadrate in the mandibular suspensorium of many fishes, which unites together the other bones of the suspensorium”.

Fish bones?

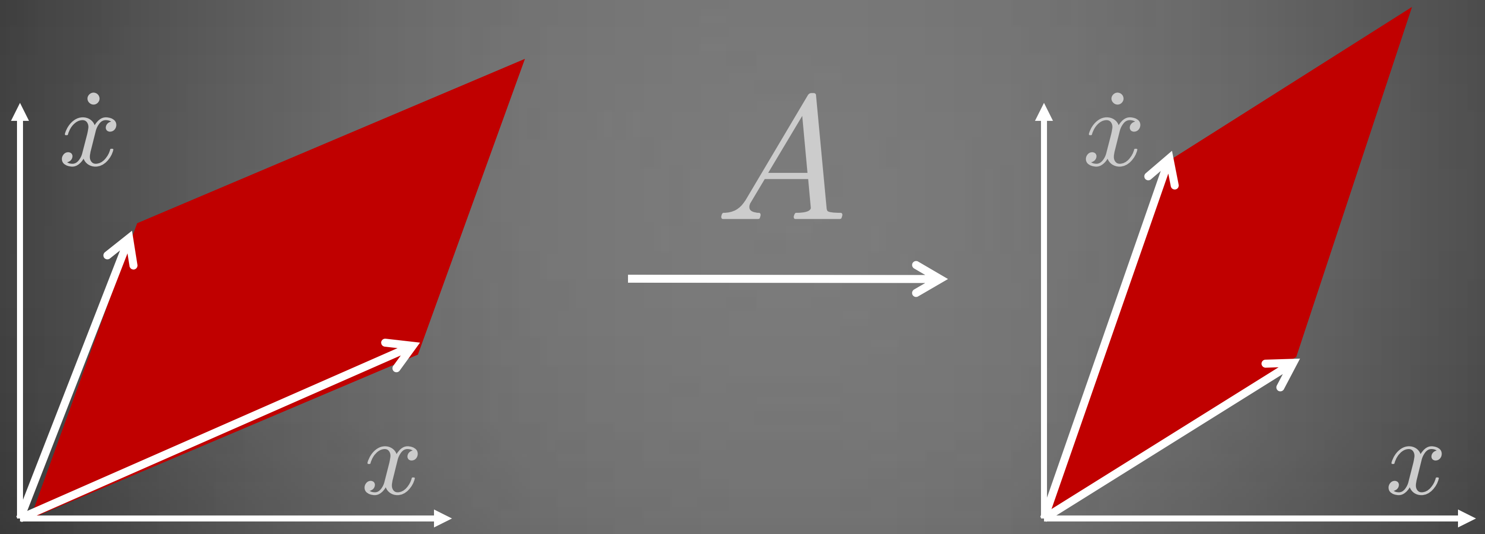
Symplectic?



In math: Hermann Weyl (1930's) : com-plex → sym-plectic

Com : Latin root
Sym : Greek root

Symplectic?



Symplectic?

$$\begin{pmatrix} \dot{x}^1 \\ x^1 \end{pmatrix} = \boxed{\begin{pmatrix} 1 & -h \\ h & 1-h^2 \end{pmatrix}} \begin{pmatrix} \dot{x}^0 \\ x^0 \end{pmatrix}$$

$$A$$

31 Springer Series in Computational Mathematics

Geometric Numerical Integration

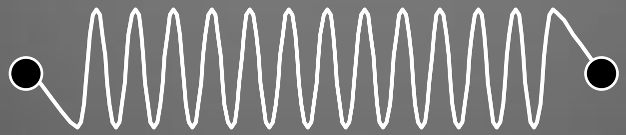
Structure-Preserving Algorithms for Ordinary Differential Equations

E. Hairer
C. Lubich
G. Wanner

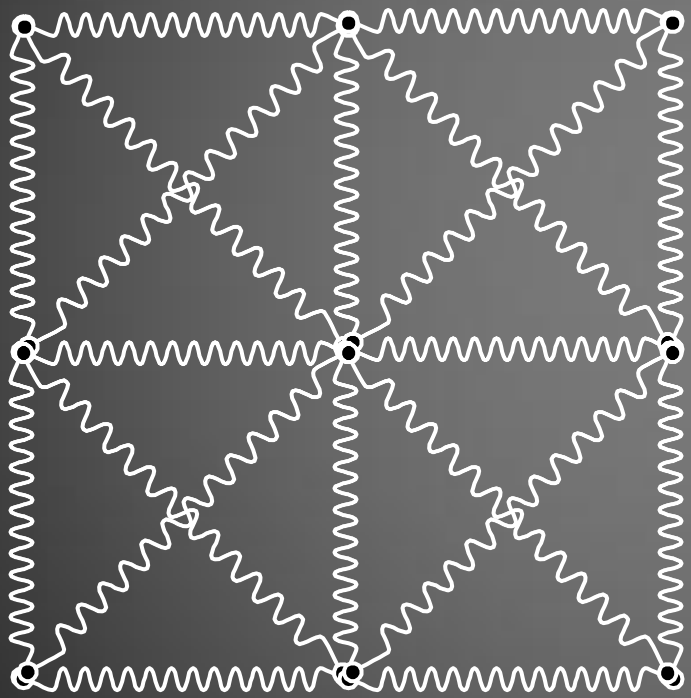
Springer

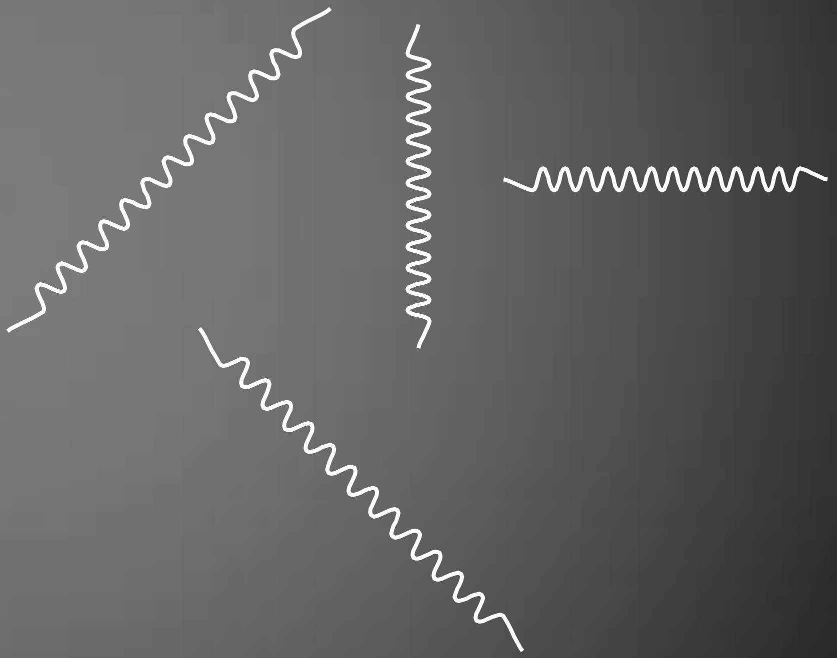
Demo

Spring Demo

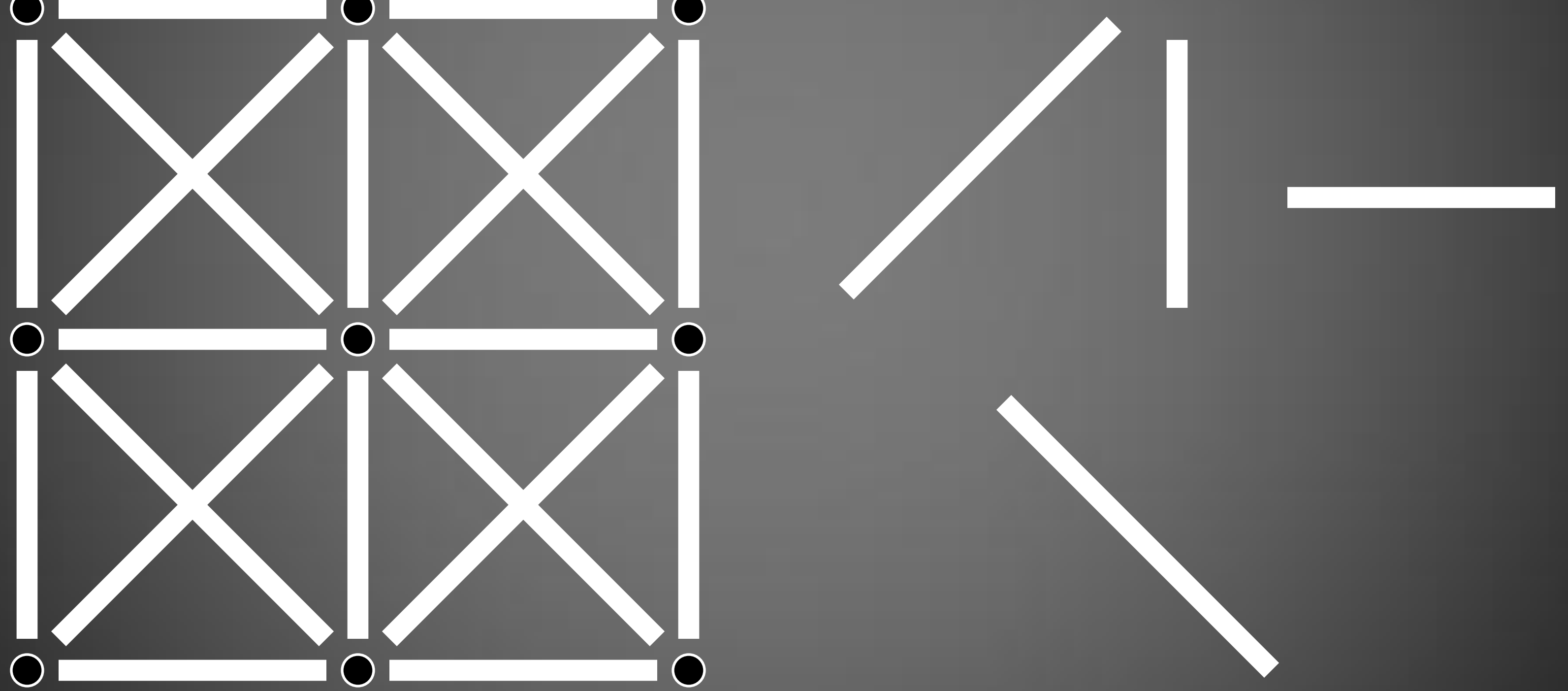


Simple Idea





Simple Idea



Nucleus

Strategy:

- Use constraints
- Implicit on velocity
- Explicit on position

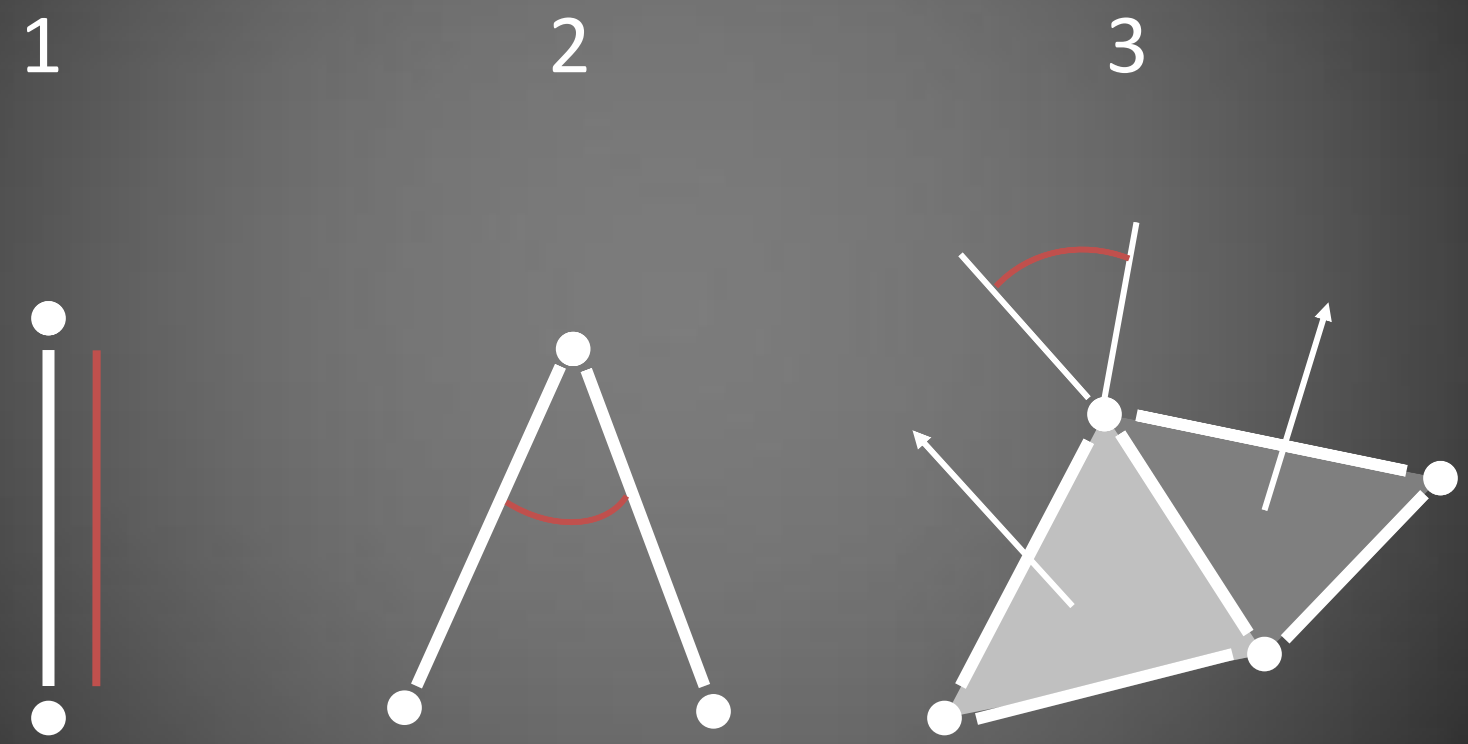
Nucleus

$$C\left(\mathbf{x}+\mathbf{v}+\delta\mathbf{v}\right)=0$$

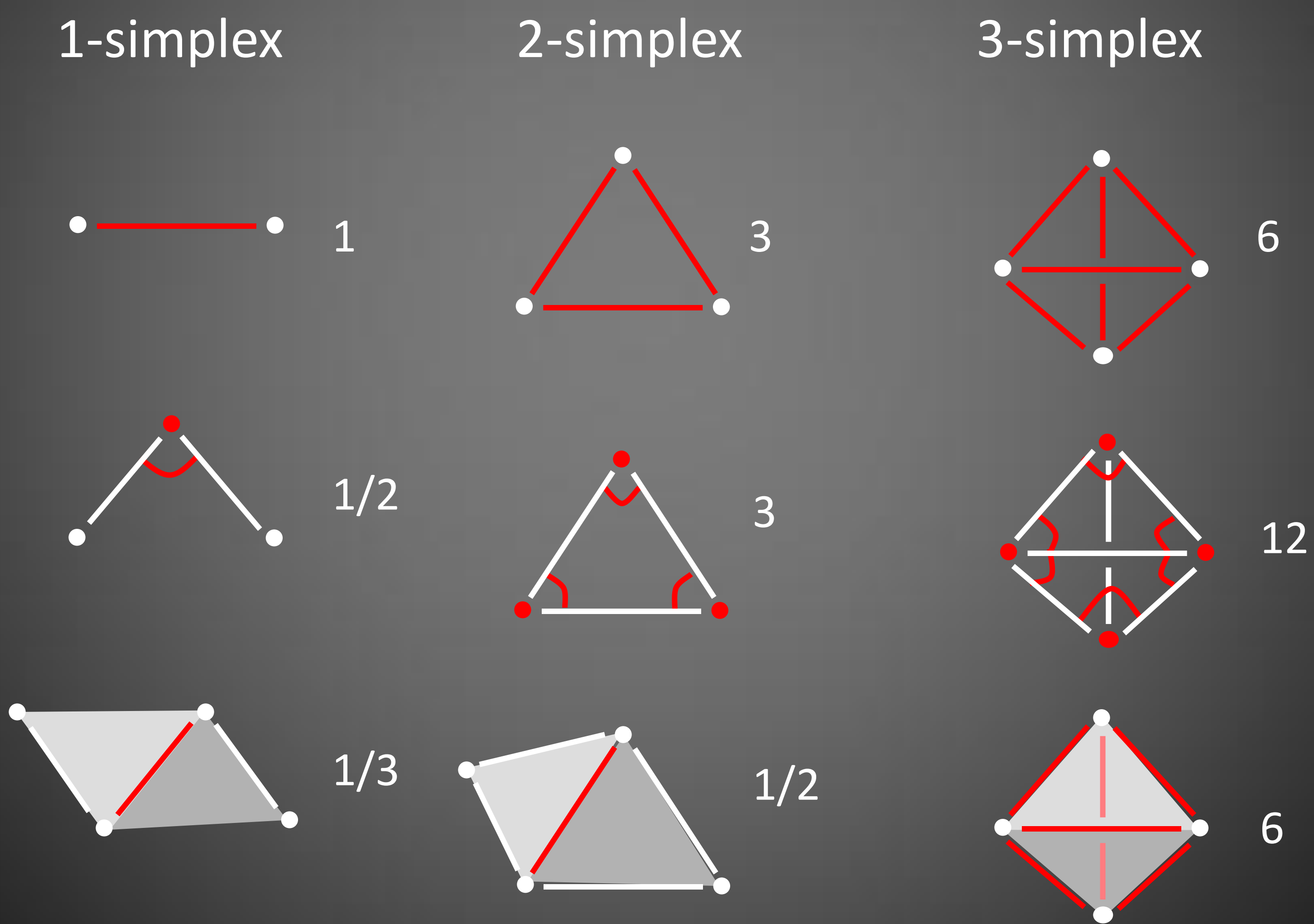
Solve for: $\delta\mathbf{v}$

$$\mathbf{v}=\mathbf{v}+\delta\mathbf{v}$$
$$\mathbf{x}=\mathbf{x}+\mathbf{v}$$

Deformations



Deformations



How To Solve?

$$C\left(\mathbf{x}+\mathbf{v}+\delta\mathbf{v}\right)=0$$

Non-linear…

One constraint at a time

Linearize

BFGS

One Constraint

$$f_k(t) = C_k\left(\mathbf{x} + \mathbf{v} + t\,\mathbf{d}_k\right) = 0$$

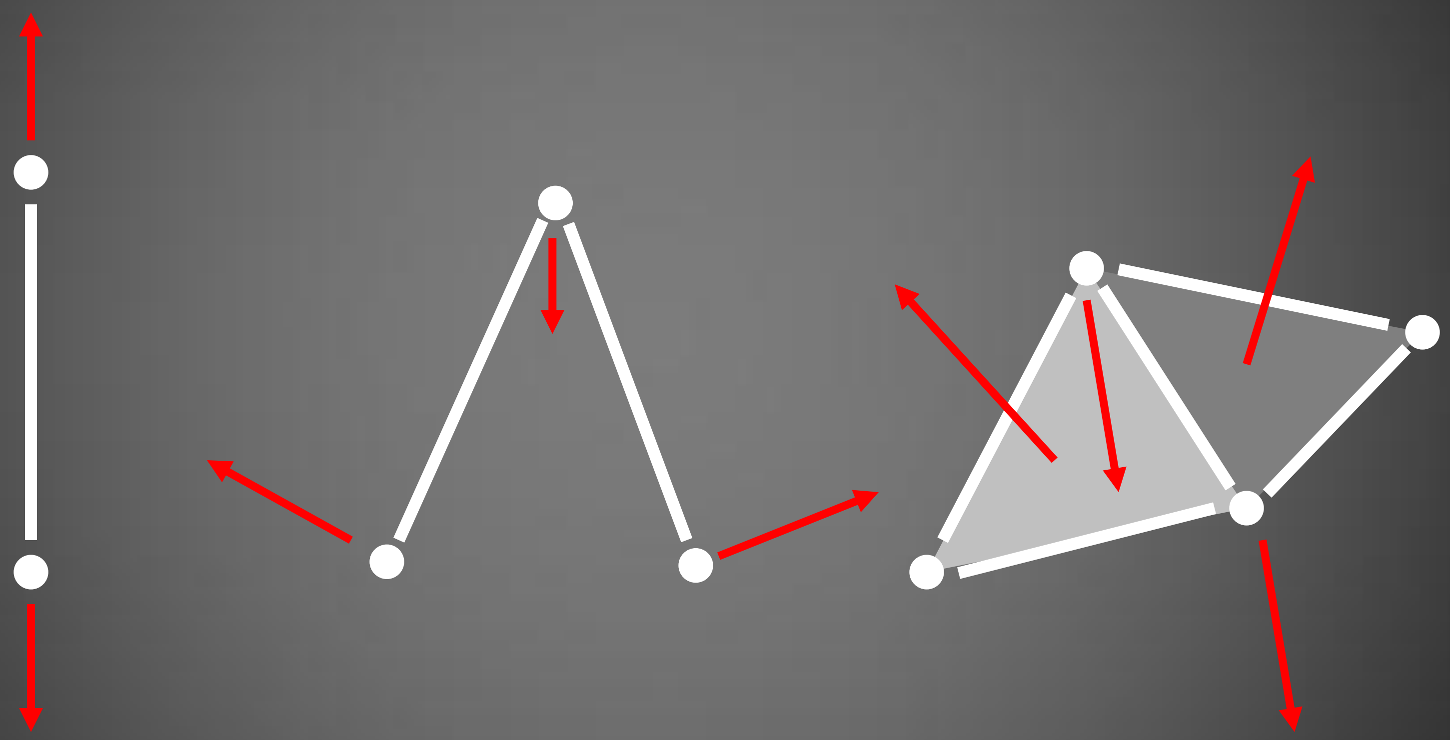
$$f_k(t) \approx f_k(0) + t\;f_k^{'}(0) = 0$$

$$t = -f_k(0)/f_k^{'}(0)$$

Search Direction

What about $\mathbf{d}_k$?

Search Direction



Bridson, Marino and Fedkiw SCA 2003

Linearize

$$\nabla C\left(\mathbf{x}+\mathbf{v}\right)\delta\mathbf{v} = -C\left(\mathbf{x}+\mathbf{v}\right)$$

$$Au = b \qquad 3n \times m$$

$$AA^T v = b \quad u = A^T v$$

LSQR, CGLS

BFGS

$$C\left(\mathbf{x}+\mathbf{v}+\delta\mathbf{v}\right) \rightarrow$$

$$\nabla C\left(\mathbf{x}+\mathbf{v}+\delta\mathbf{v}\right) \rightarrow$$

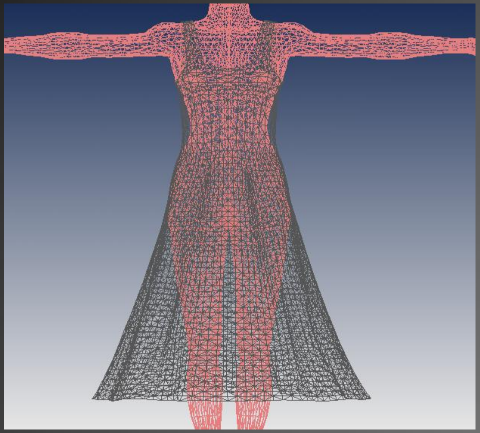
$$\rightarrow \delta\mathbf{v}$$

One Constraint

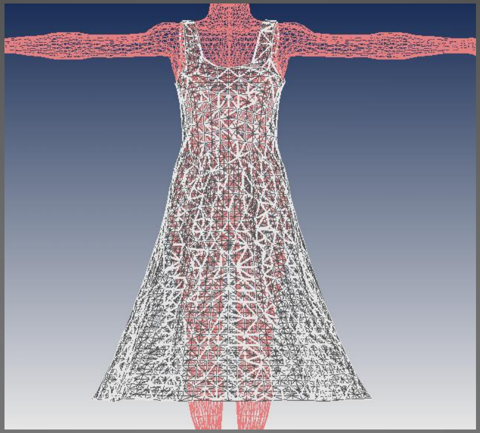
Update velocity

Sequential	vs	Parallel
Gauss-Seidel		Jacobi
Better convergence		Slower convergence
Bias		No bias
Hard to parallelize		Easy to parallelize

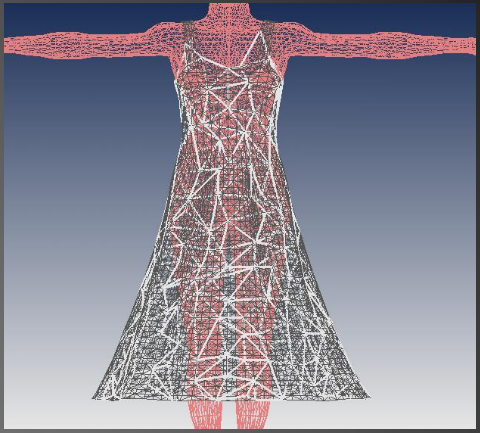
Multigrid ?



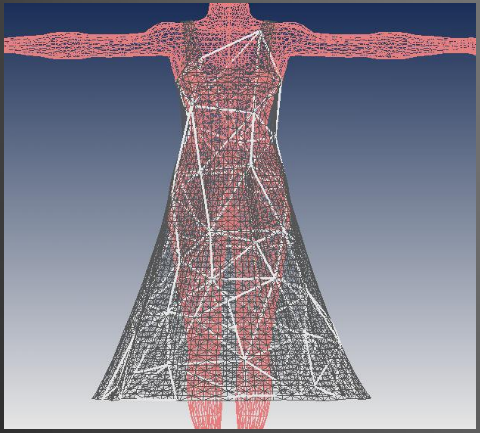
level 0



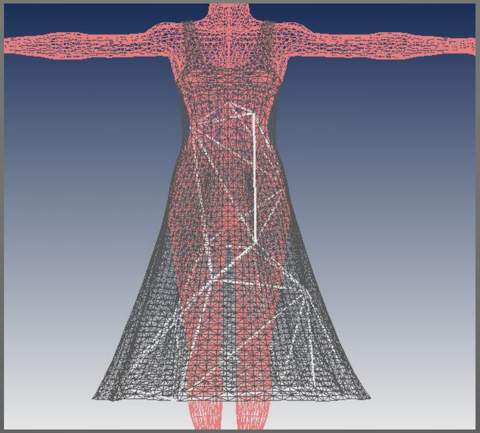
level 1



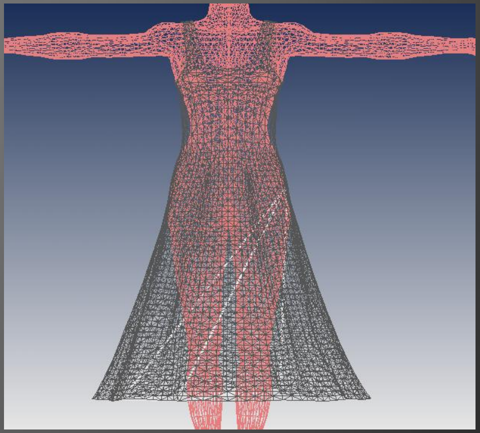
level 2



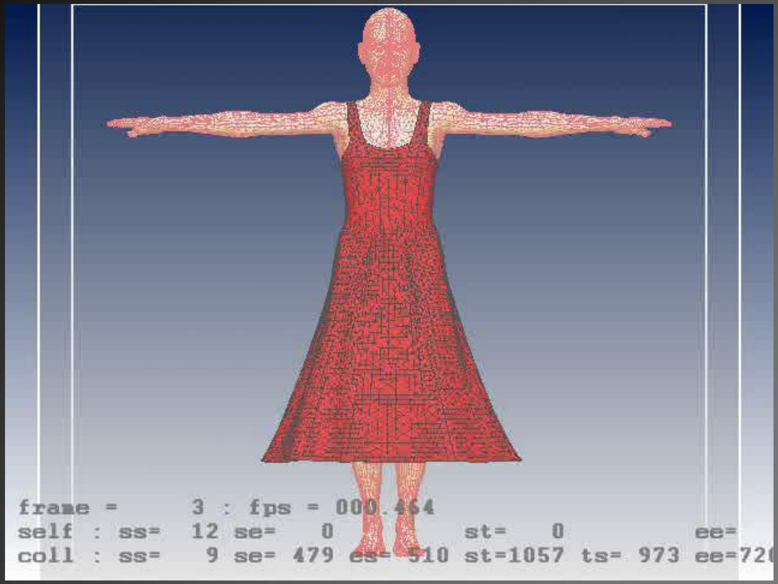
level 3



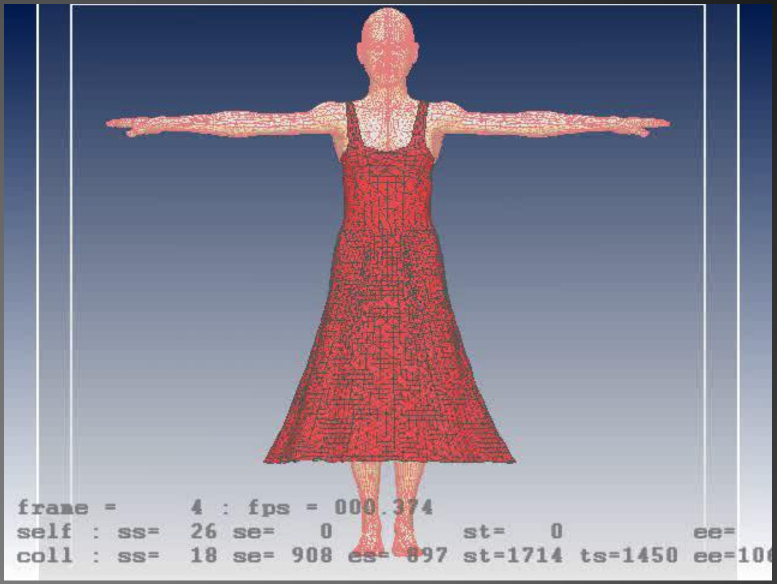
level 4



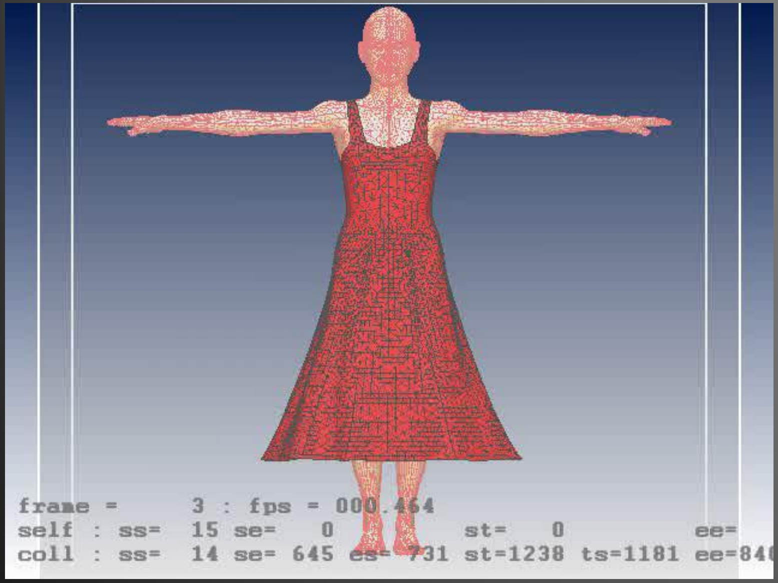
level 5



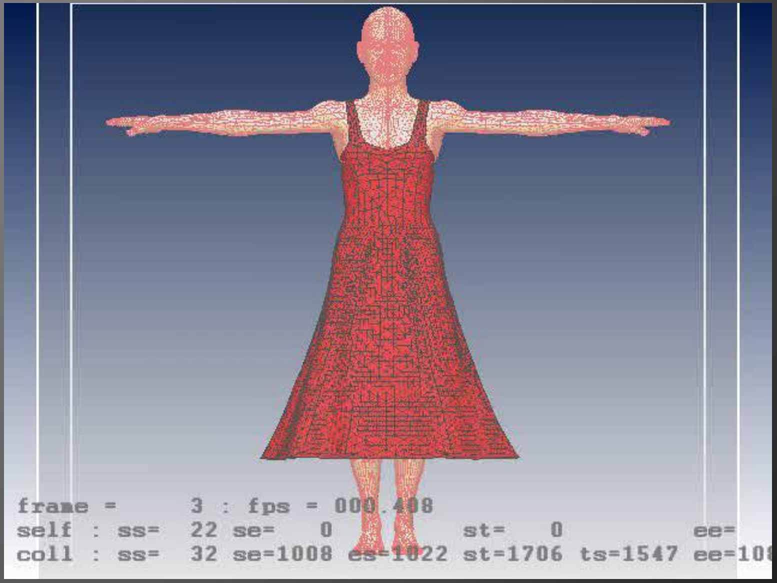
parallel / no multigrid



parallel / with multigrid

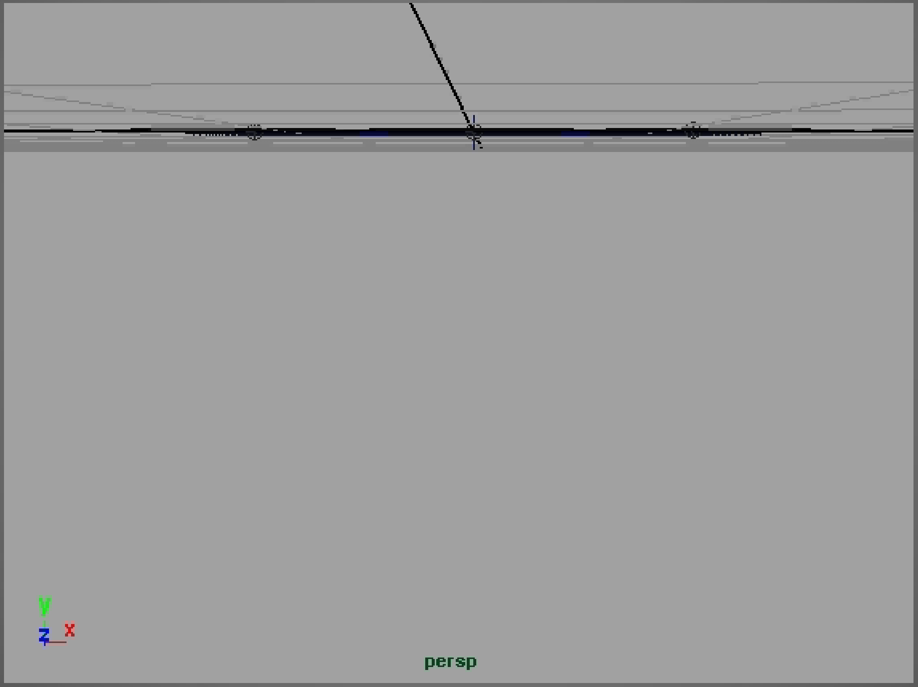


sequential / no multigrid

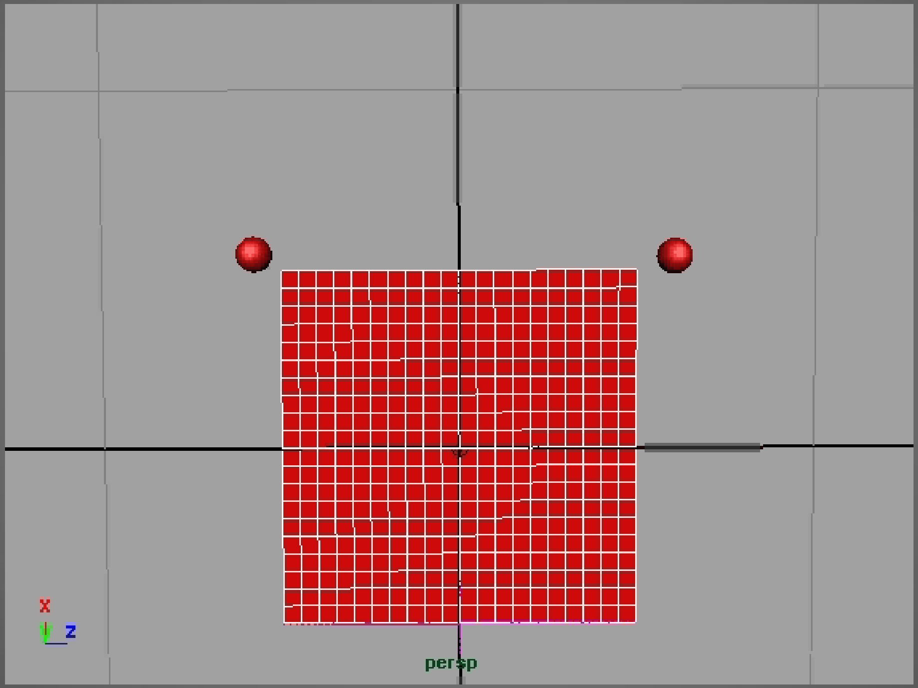


sequential / multigrid

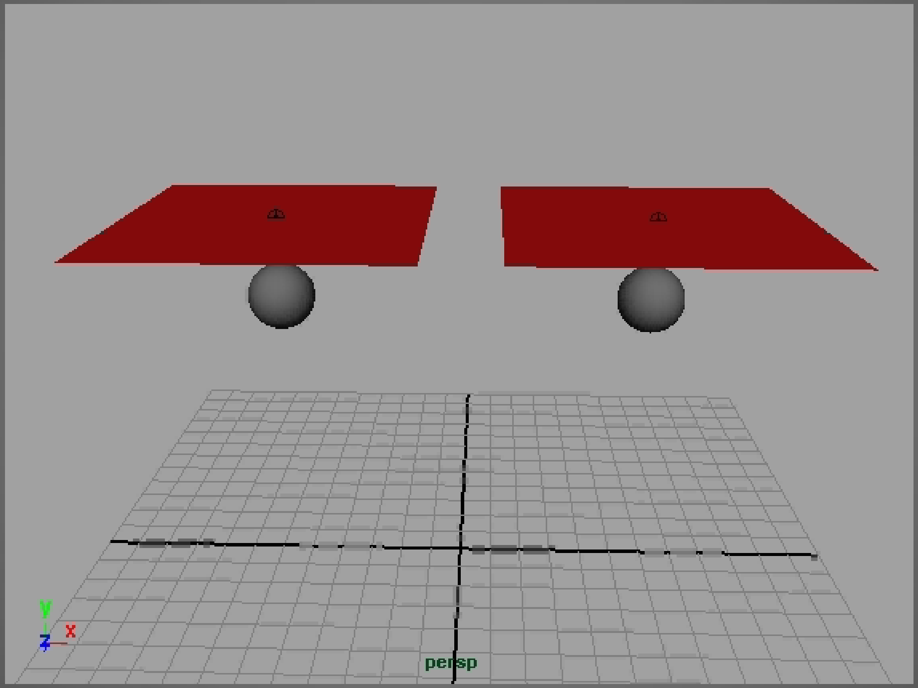
Stretch/Compression



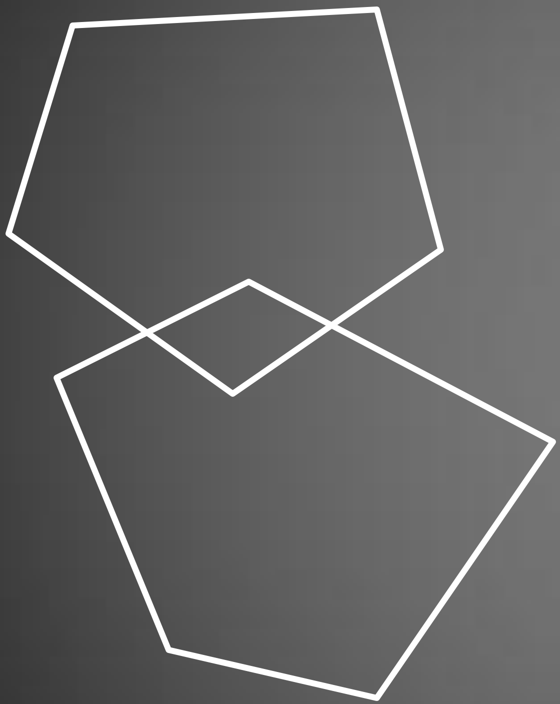
Shear

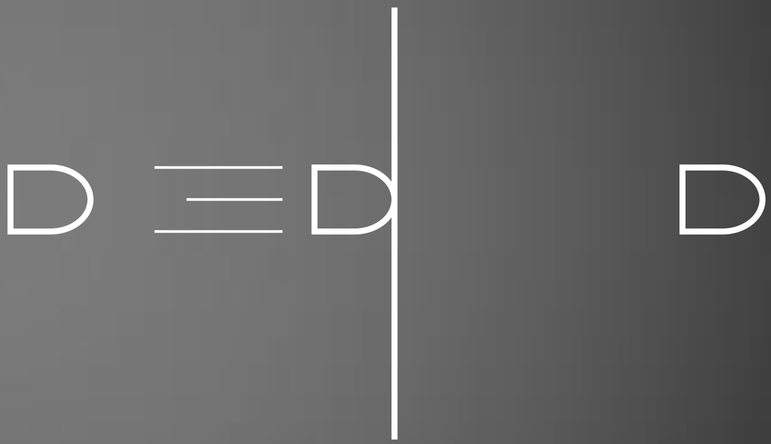


Bending



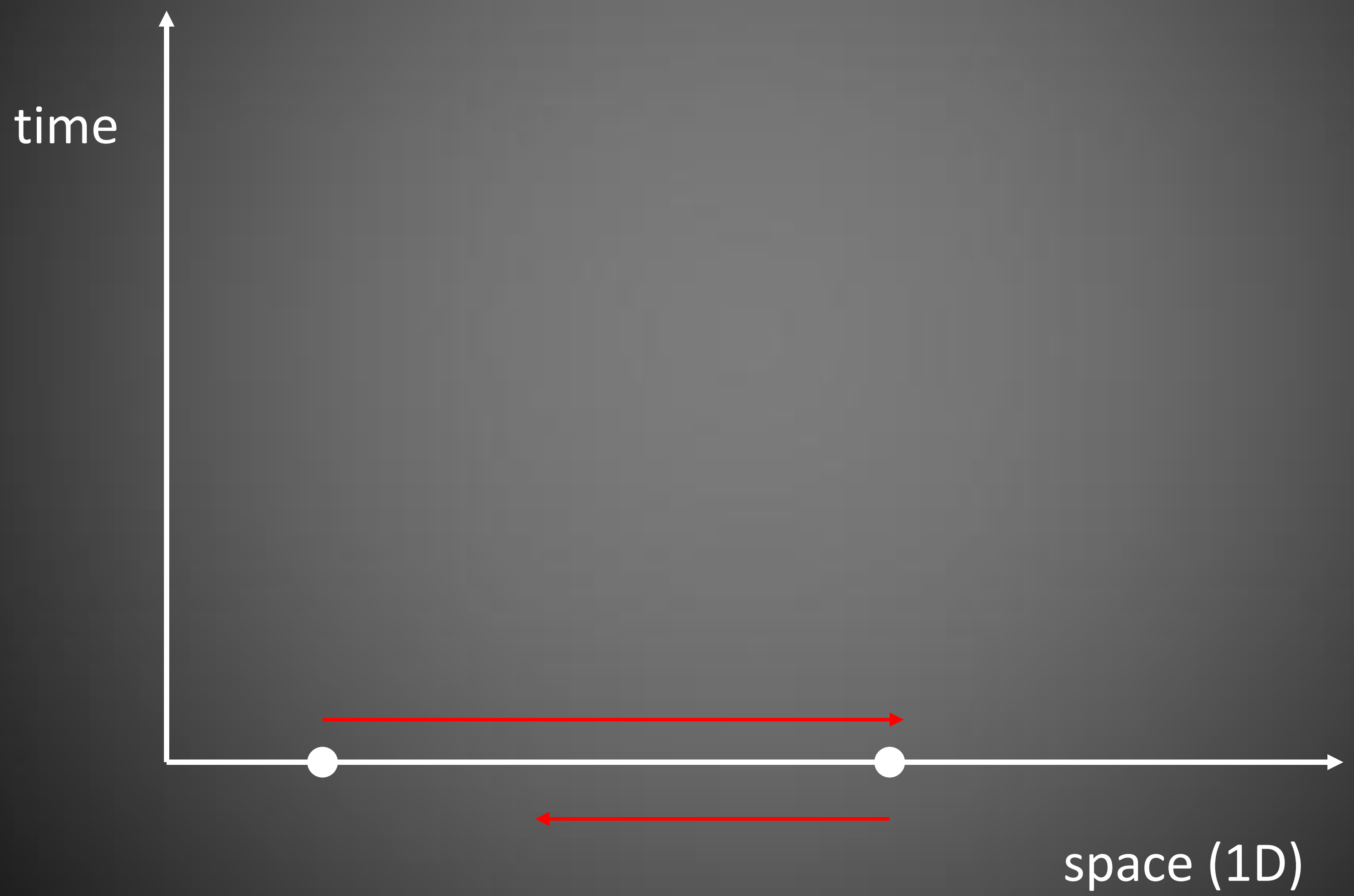
Collisions



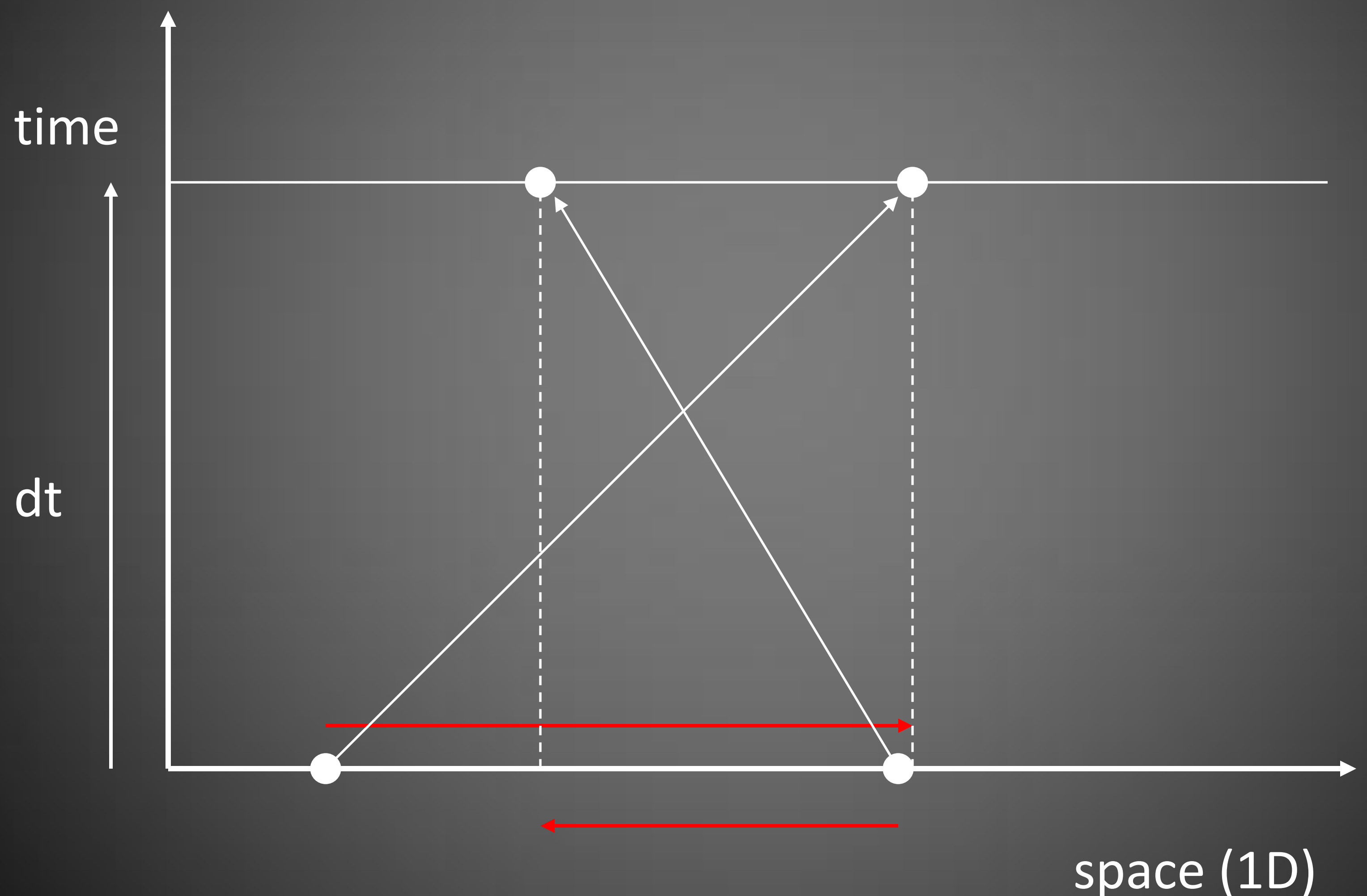


penalty vs space-time

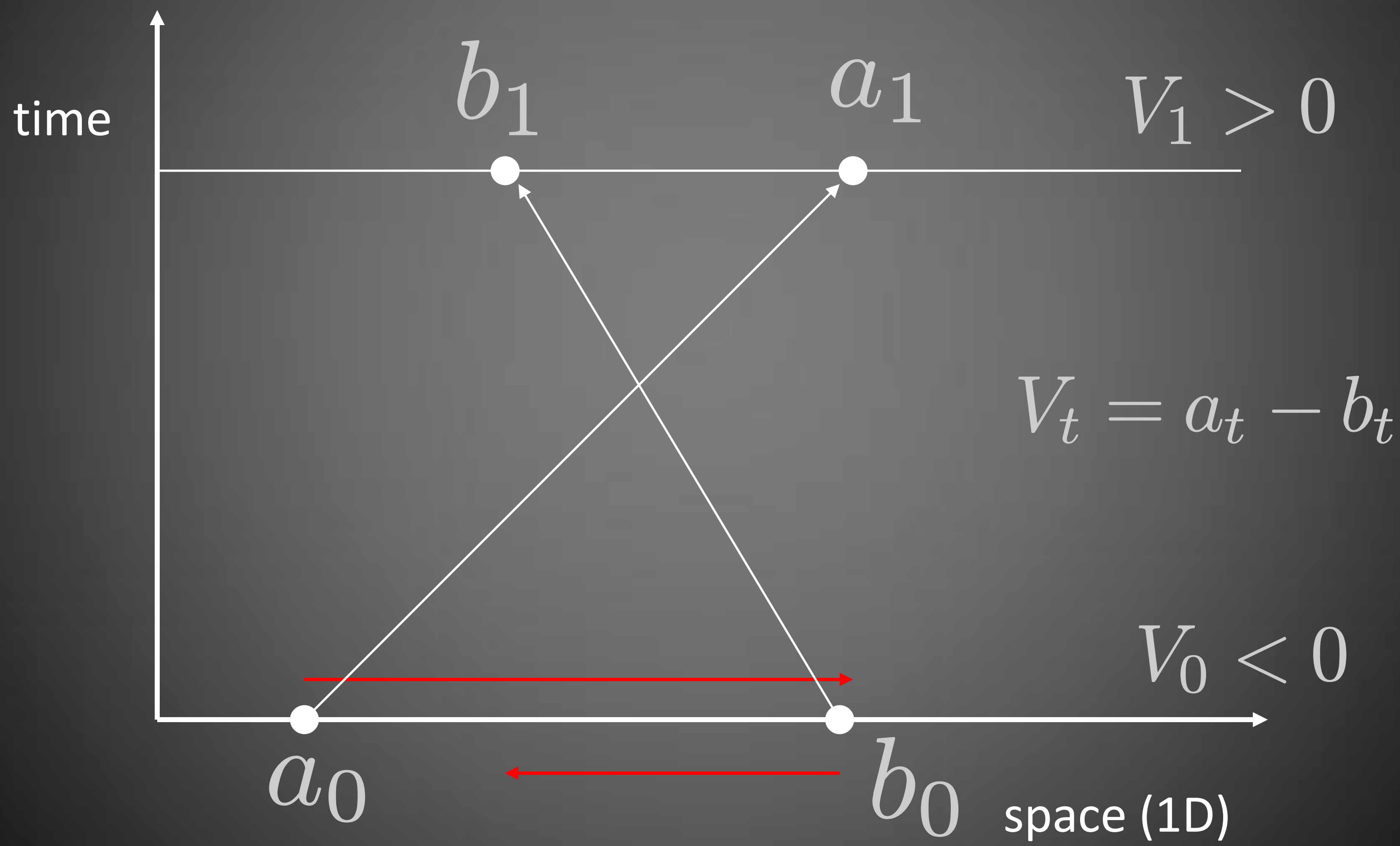
Space-Time



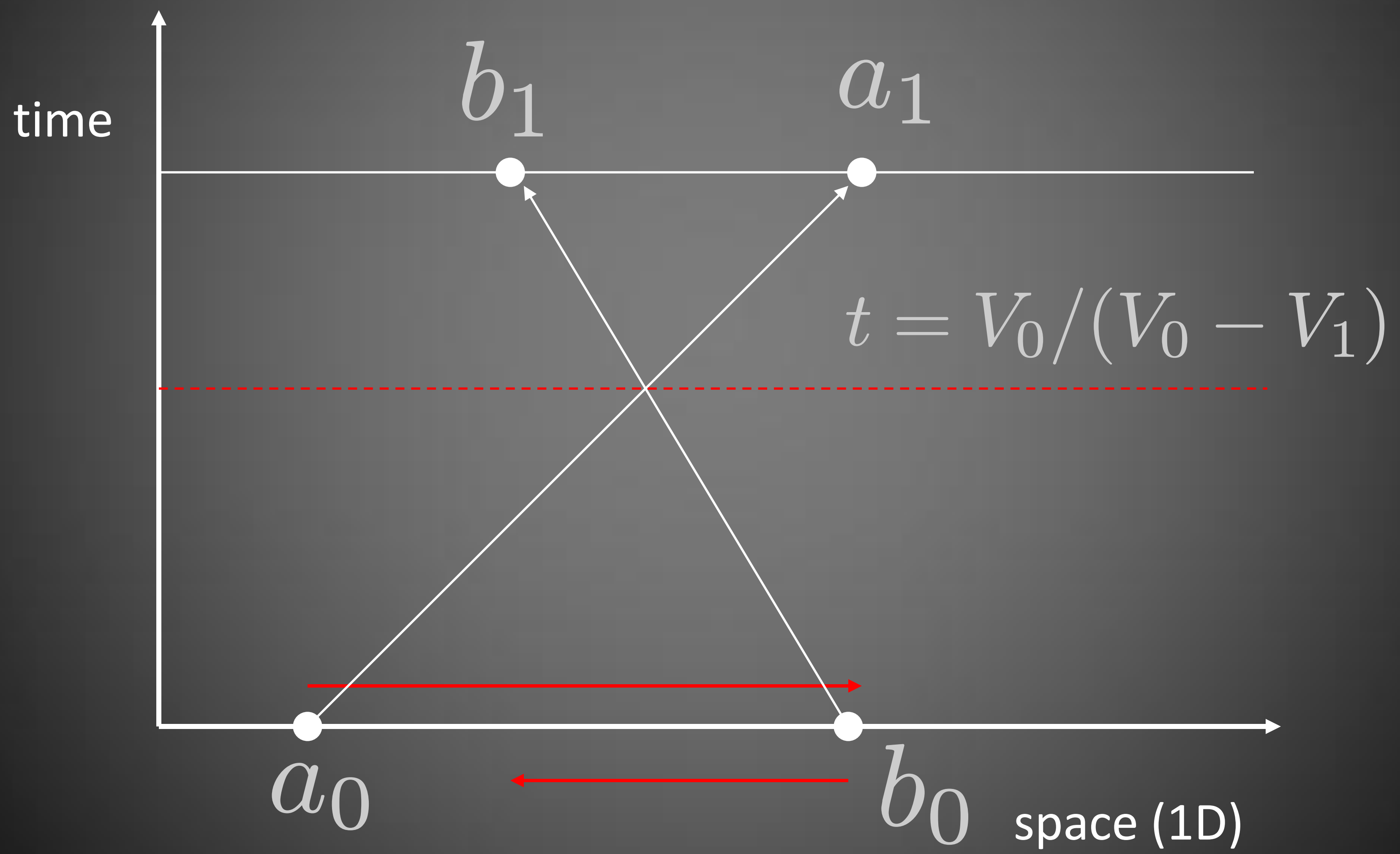
Space-Time



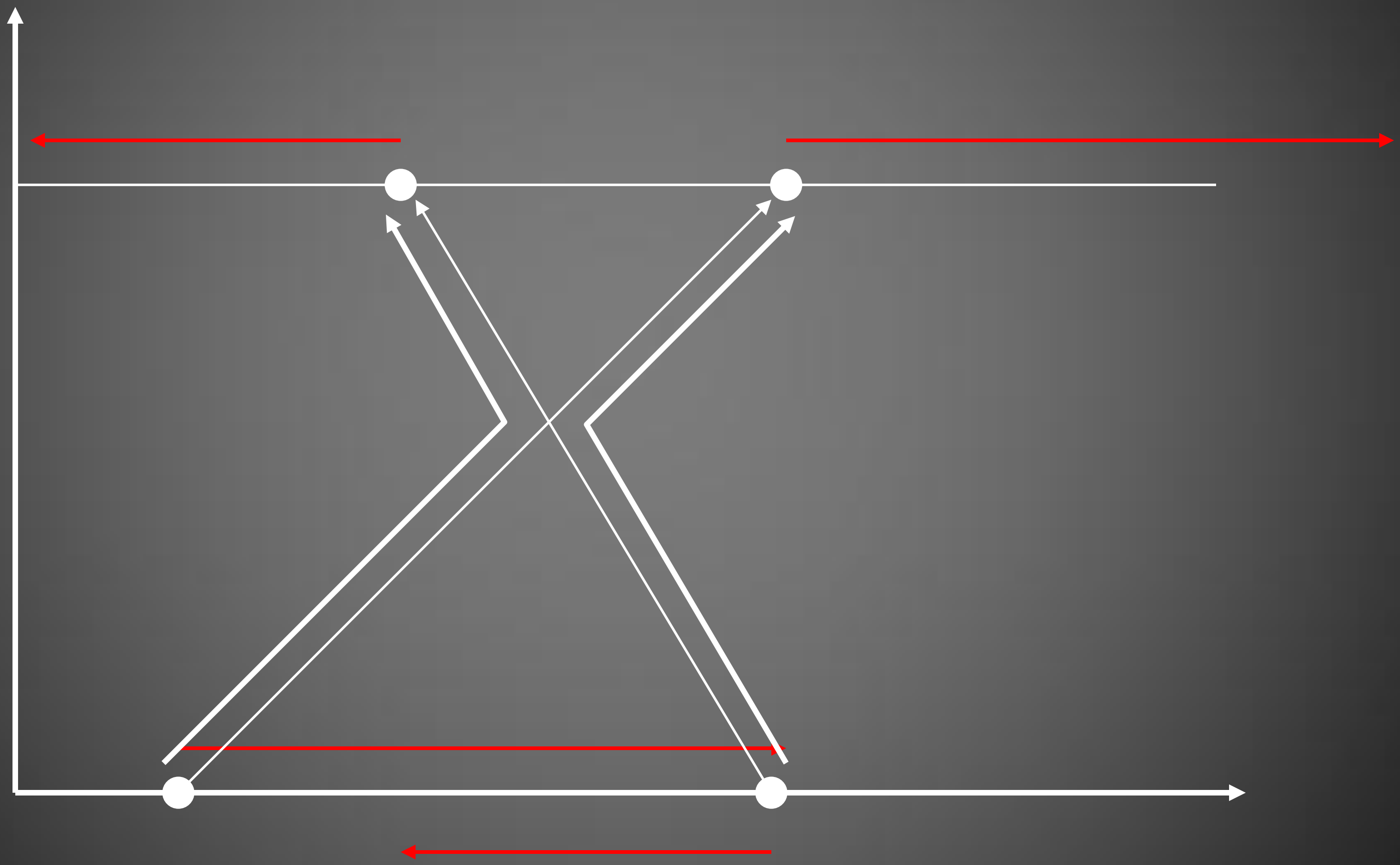
Space-Time



Space-Time

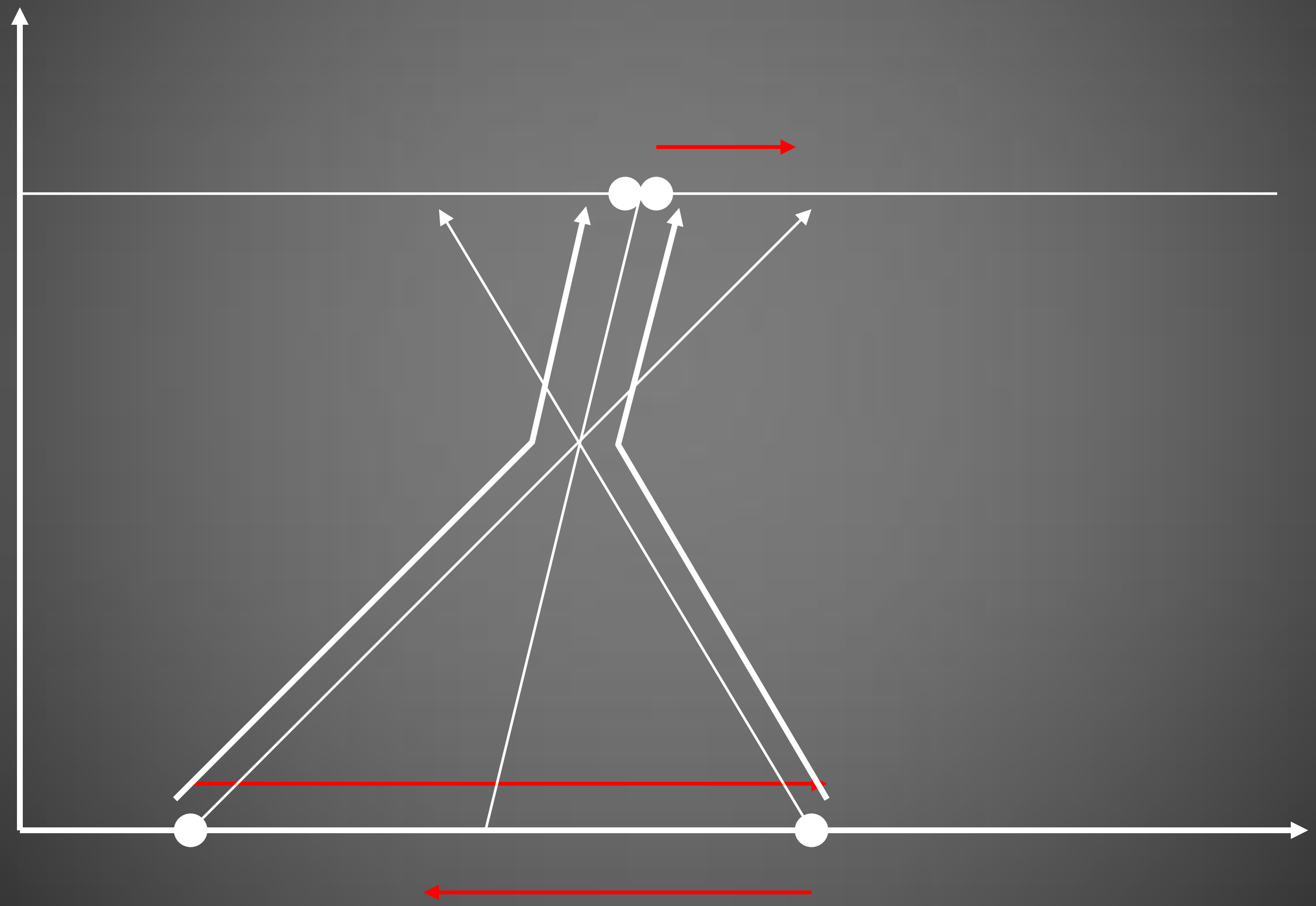


Space-Time



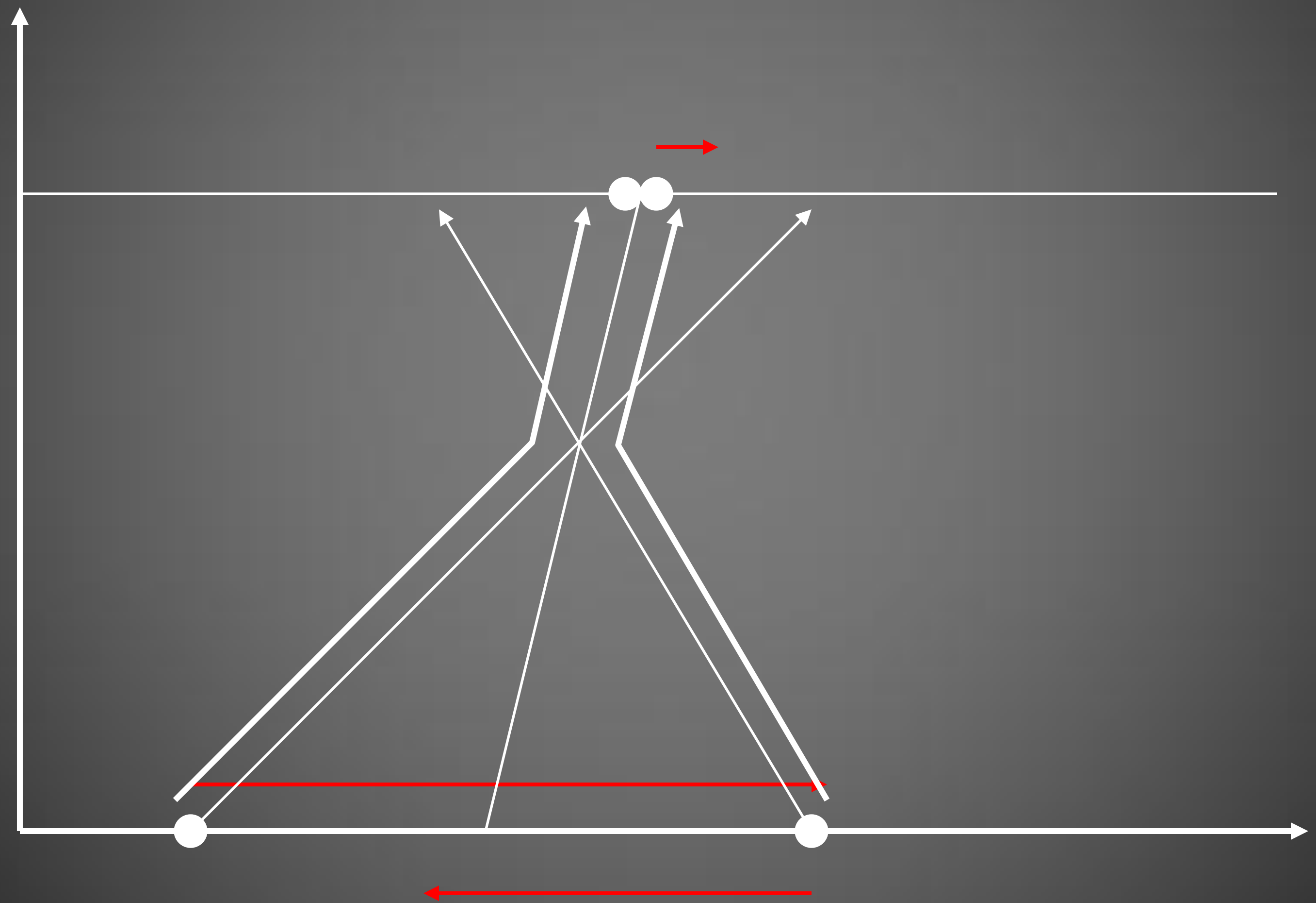
elastic

Space-Time



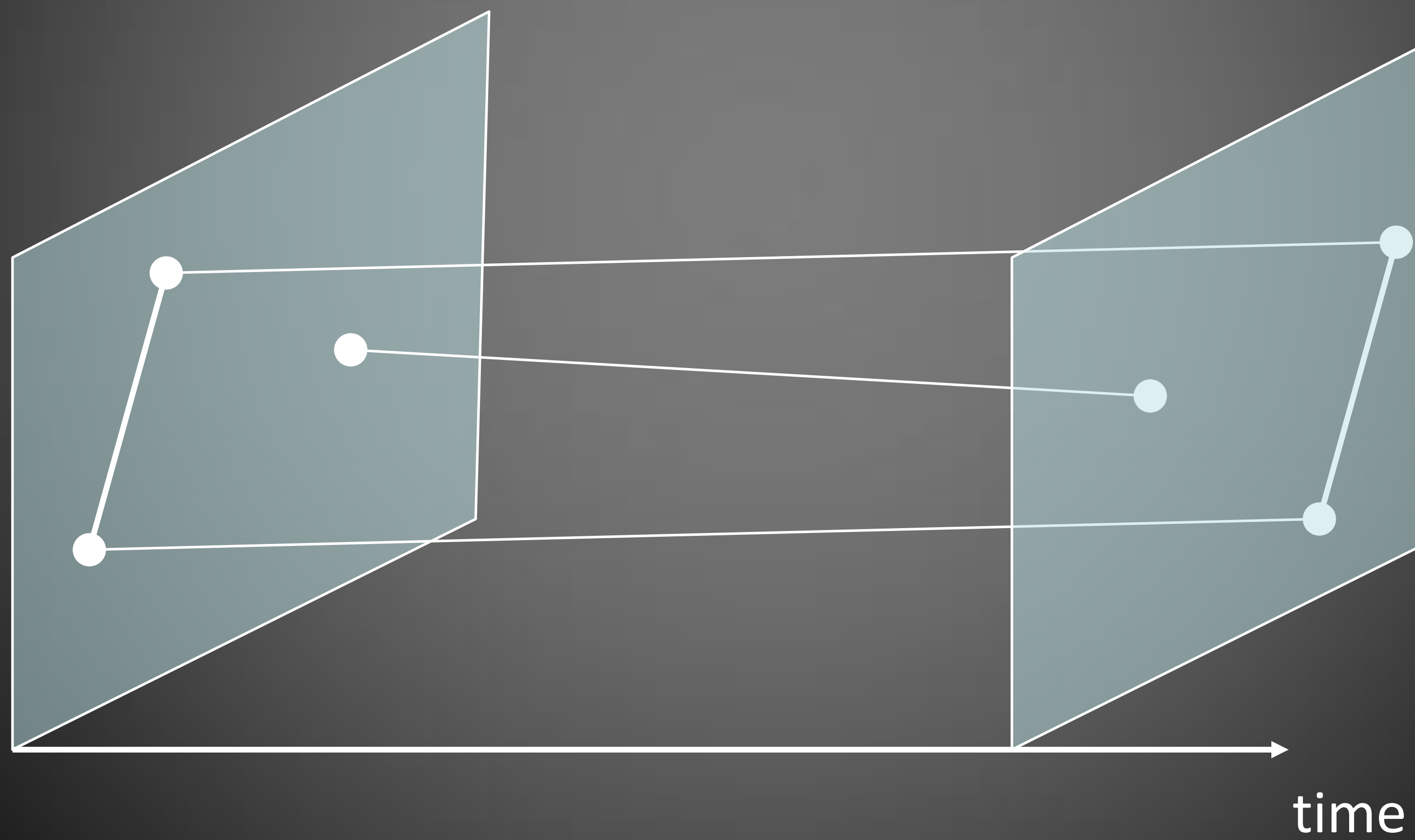
inelastic

Space-Time

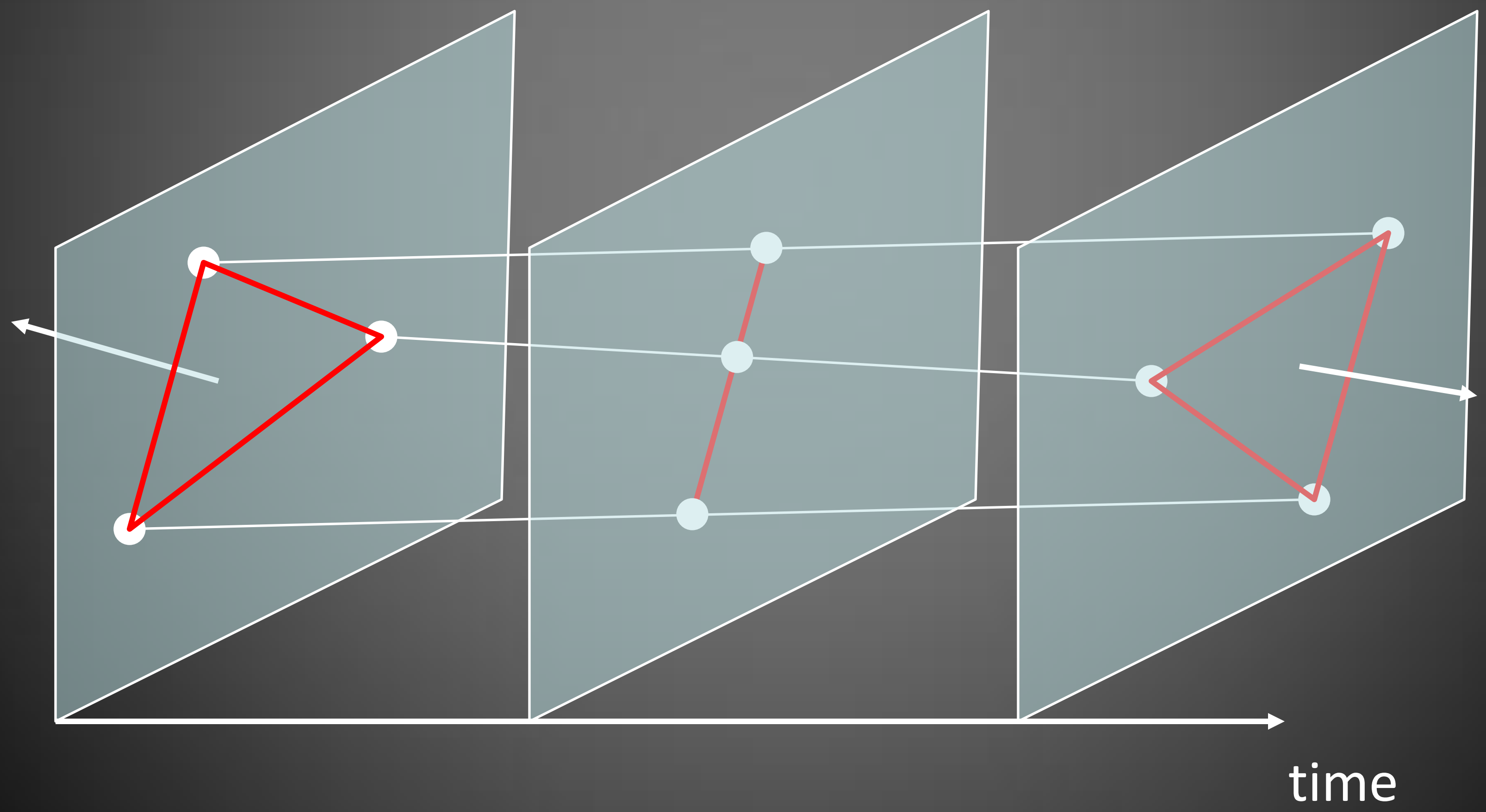


friction

Space-Time (2D)

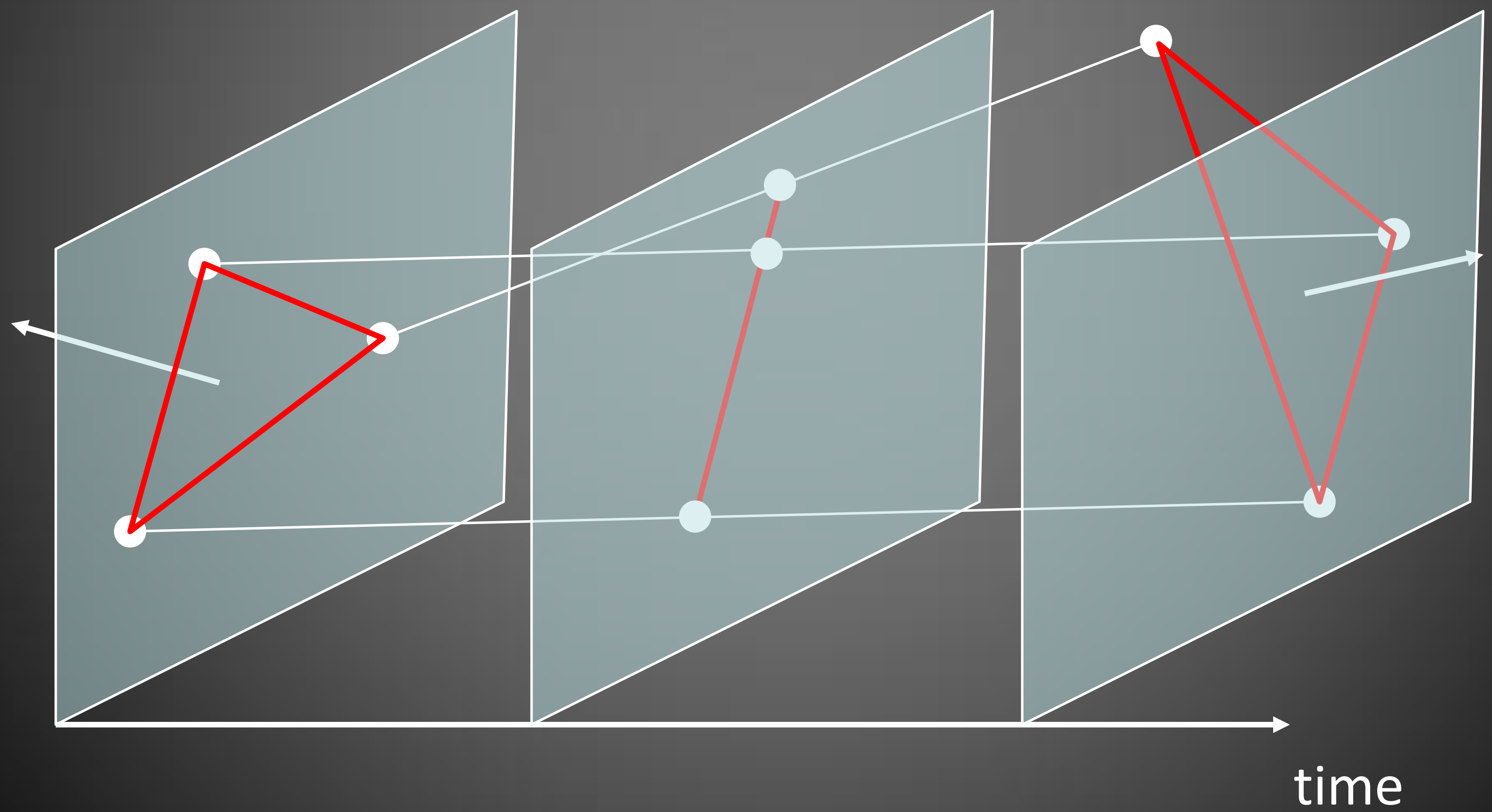


Space-Time (2D)

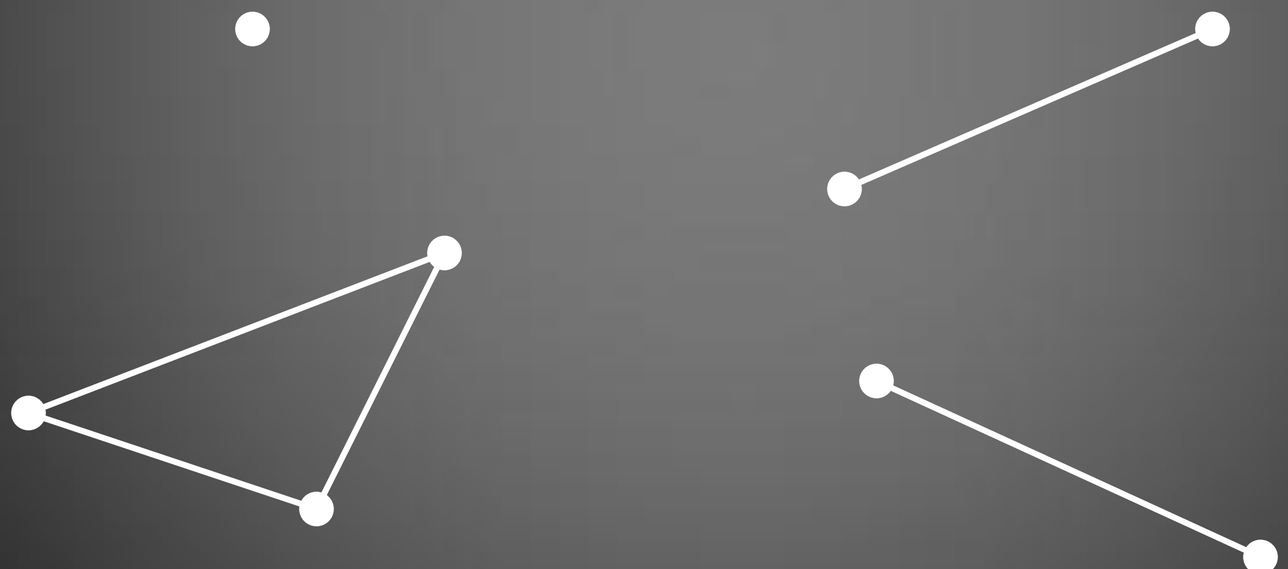


Space-Time (2D)

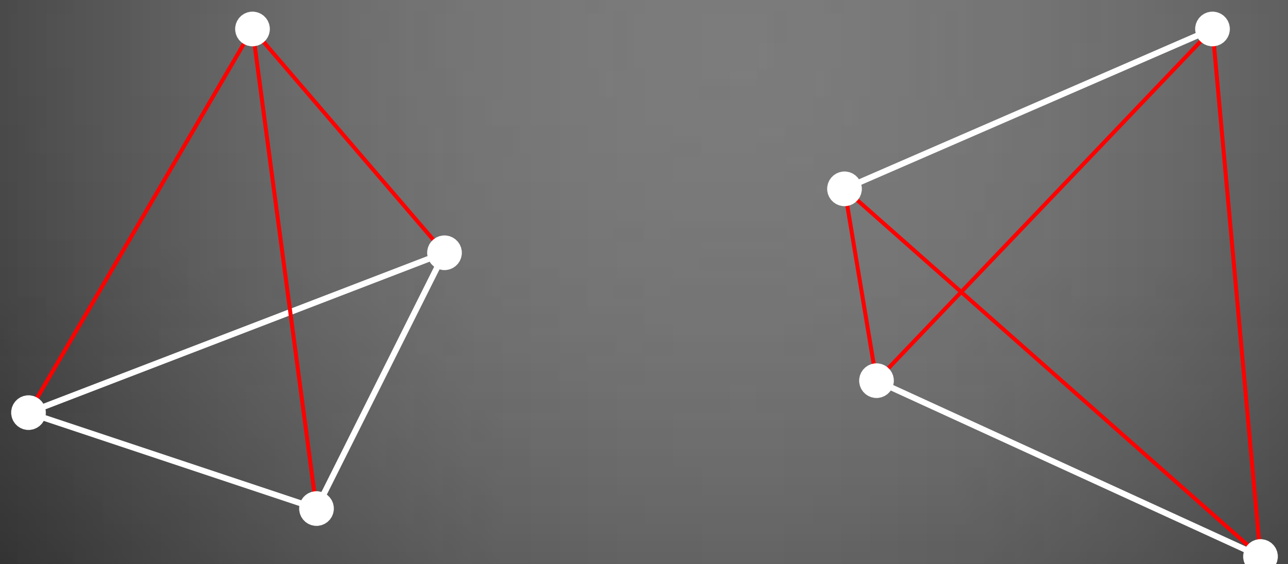
Necessary but **not** sufficient



Space-Time (3D)



Space-Time (3D)



Summary

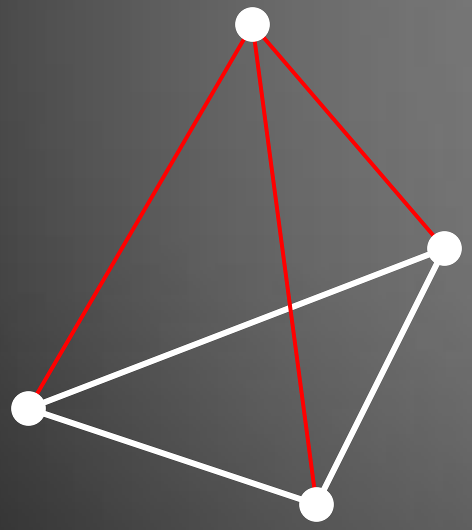
If $V_0 \cdot V_1 > 0$ stop

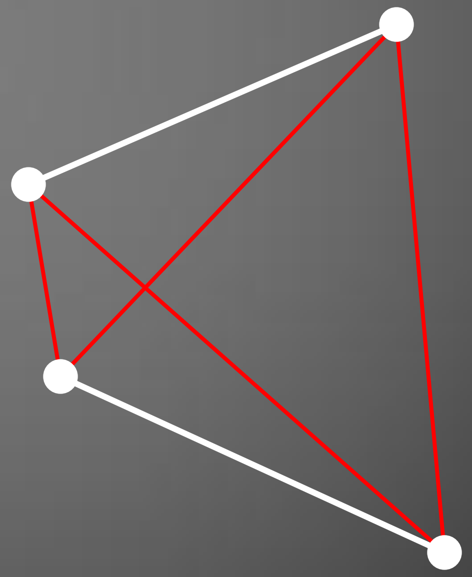
Find t such that $V_t = 0$

Check if primitives overlap at t

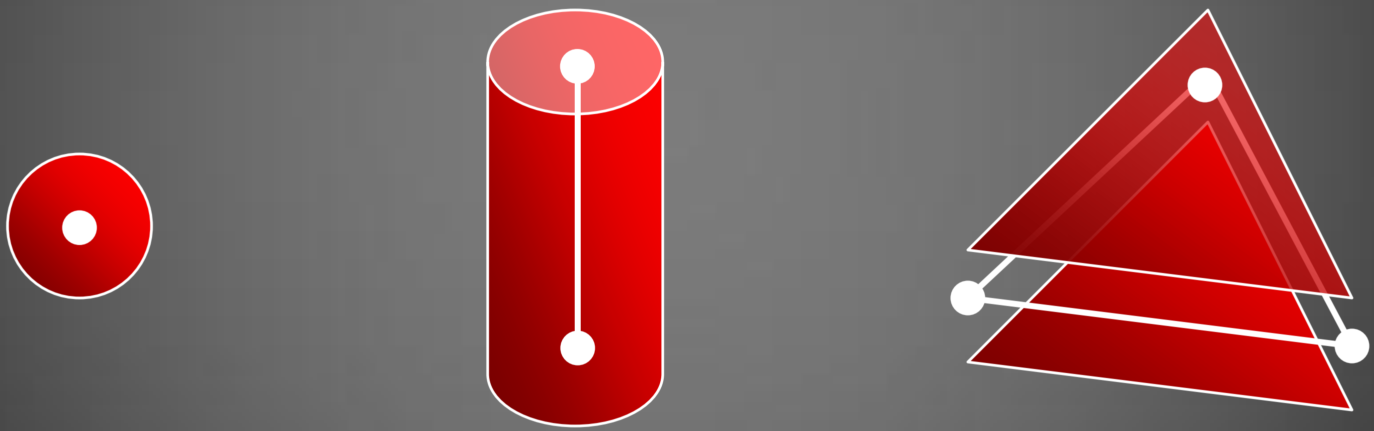
If yes handle collision

Higher Dimensions?

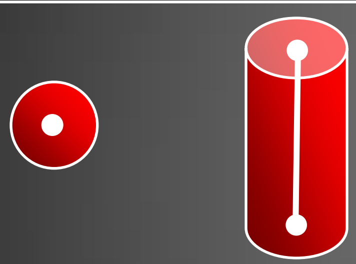
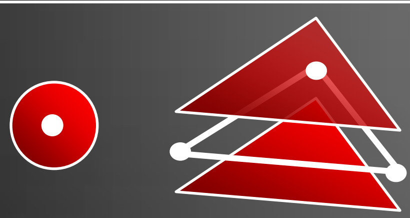
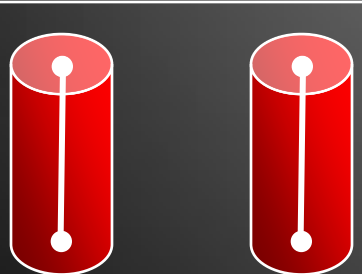




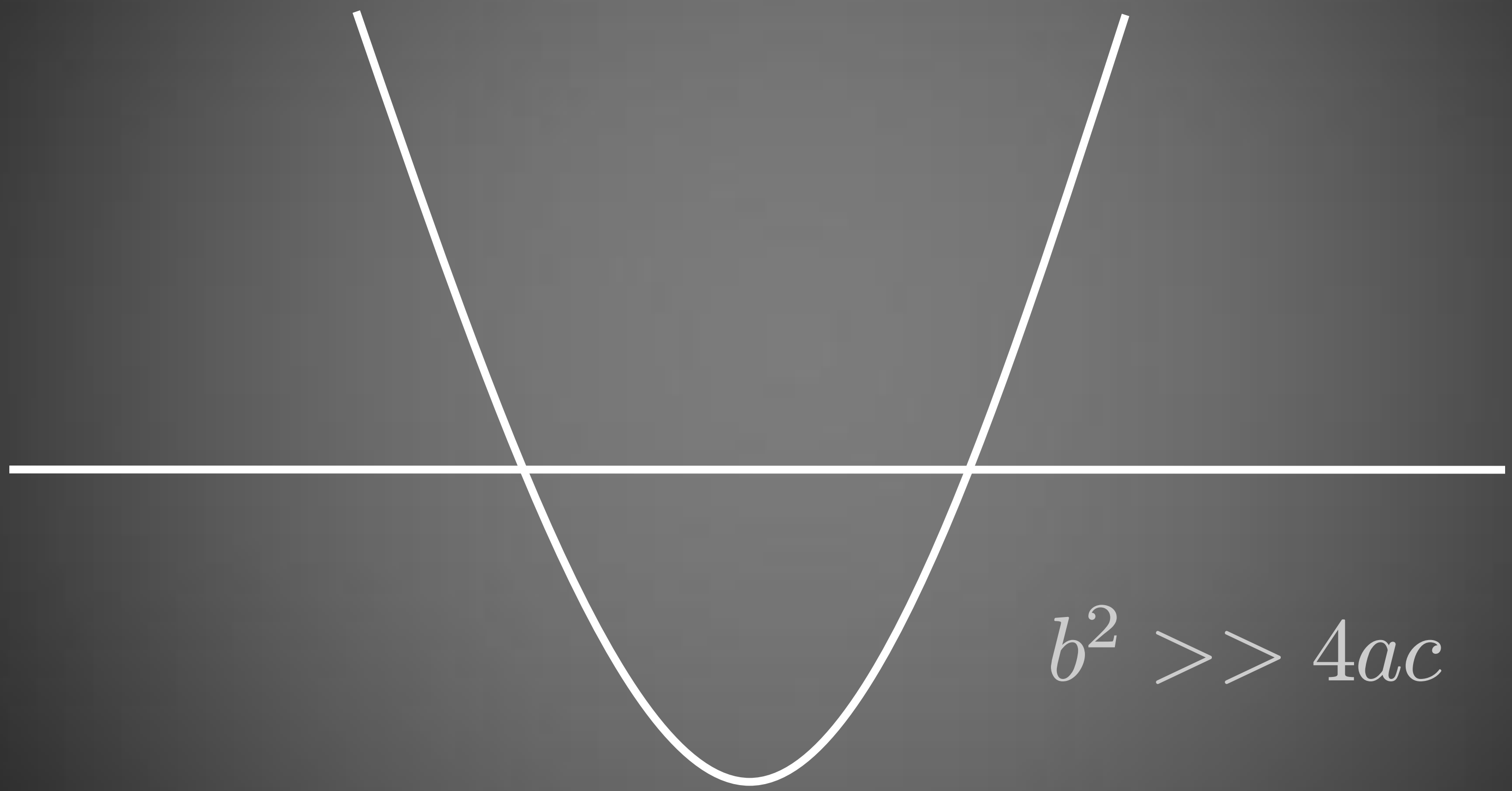
Thickness



Thickness

	Quadratic (2)
	Quartic (4)
	Sextic (6)
	Sextic (6)

Roots of a Polynomial



$$b^2 >> 4ac$$

$$t = \frac{1}{2a}\left(b \pm \sqrt{b^2 - 4ac}\right)$$

Roots of a Polynomial

$$q = -b - \operatorname{sgn}(b)\sqrt{b^2 - 4ac}$$

$$t_1 = \frac{2c}{q}$$

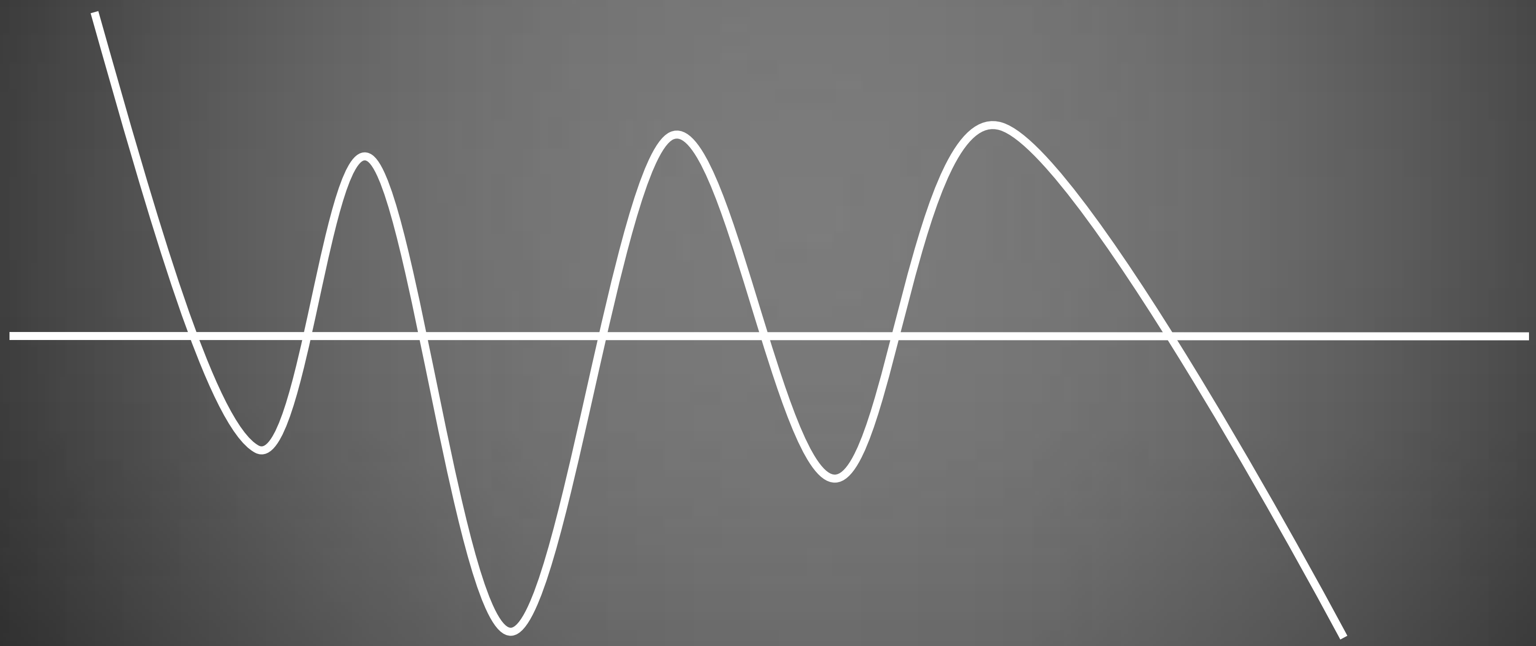
$$t_2 = \frac{q}{2a}$$

Roots of a Polynomial

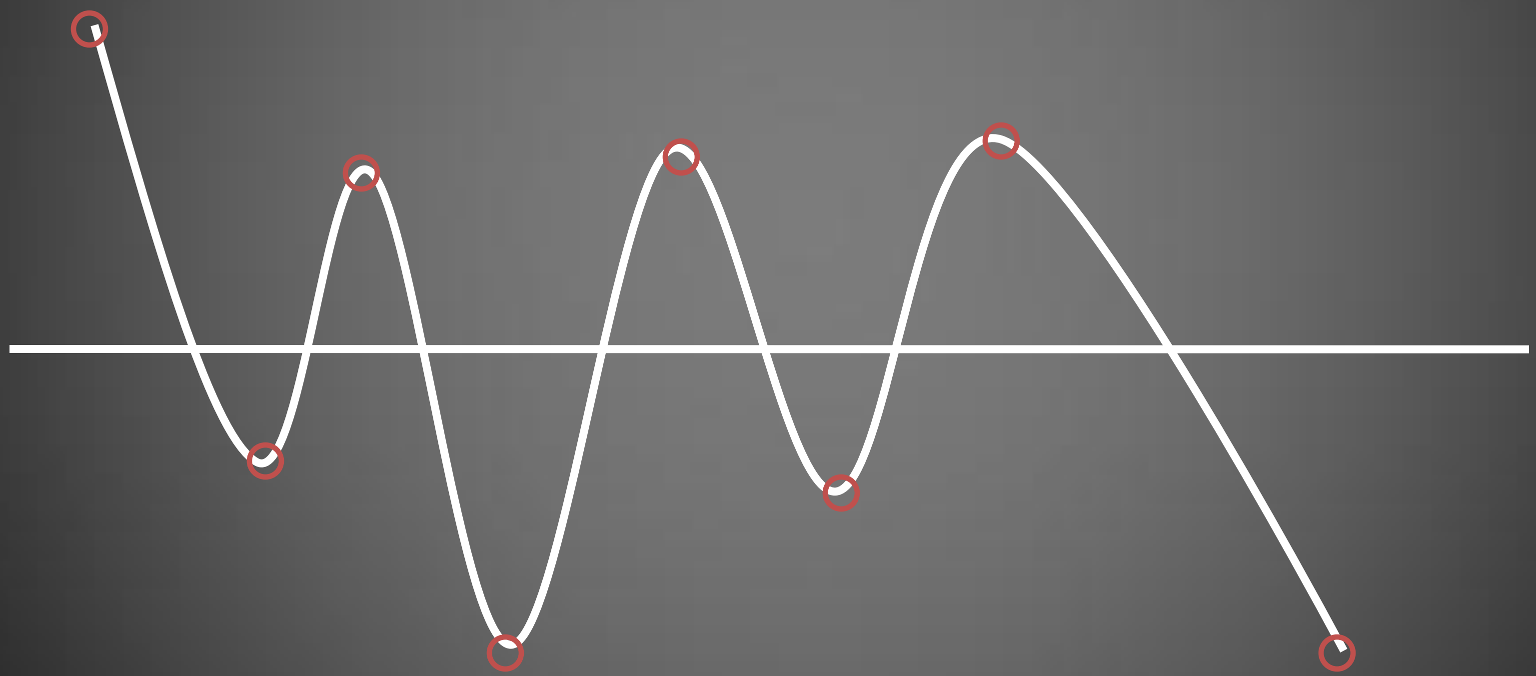
No stable formulas for degrees 3 and 4.

No formulas for degree > 4 (Abel + Galois)

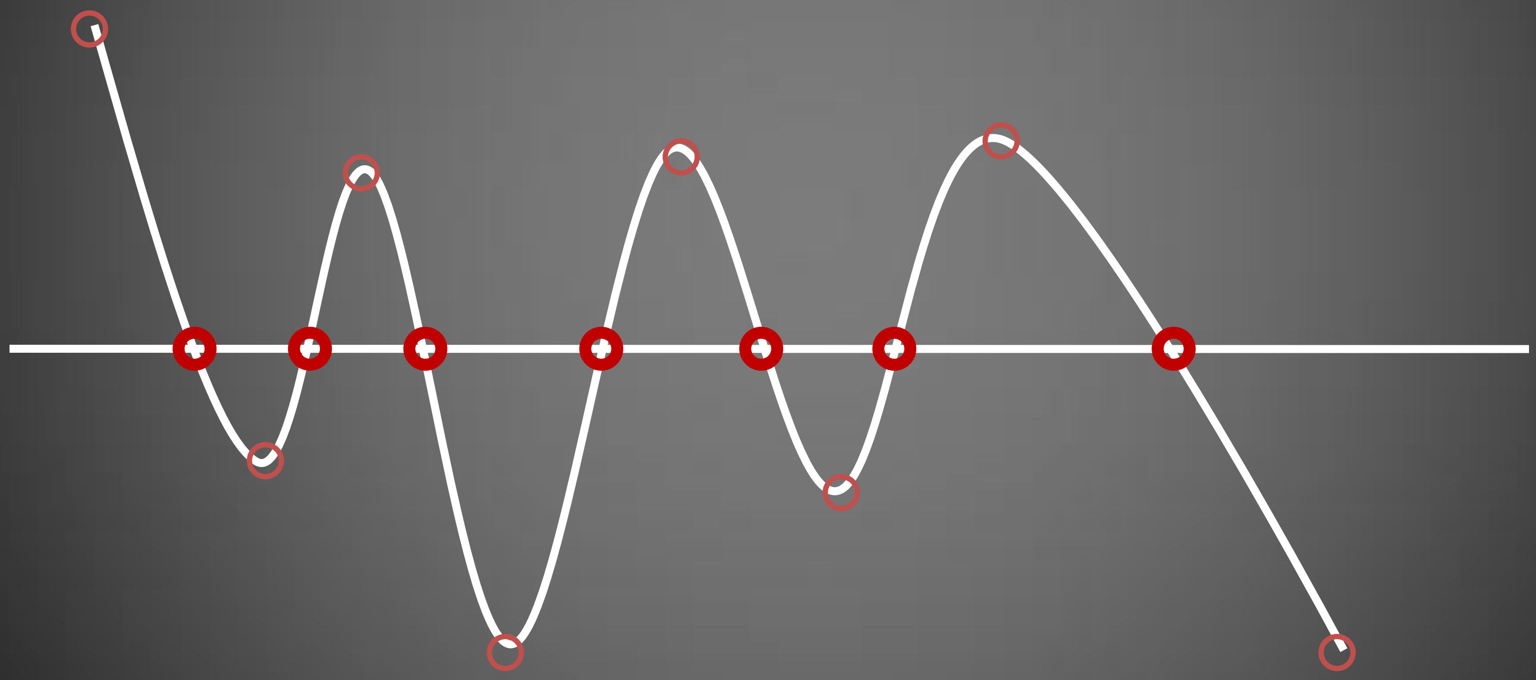
Roots of a Polynomial



Roots of a Polynomial

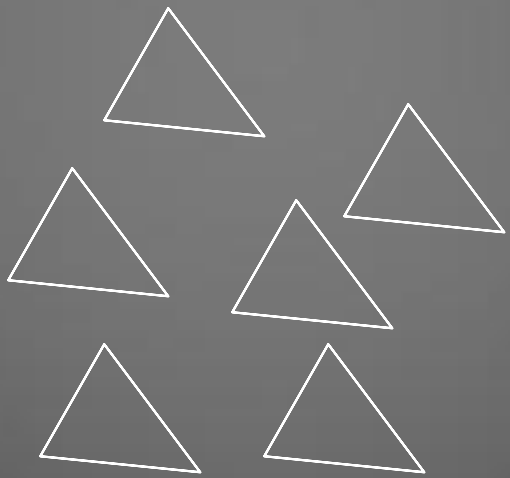


Roots of a Polynomial



Many Primitives

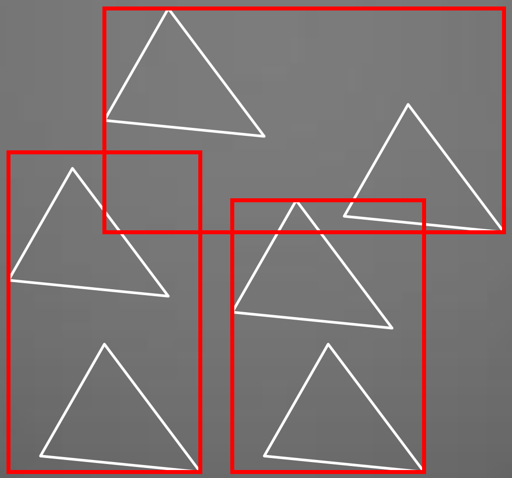
Use hierarchical data structures for speed



AABB tree (simple), actually kDOP

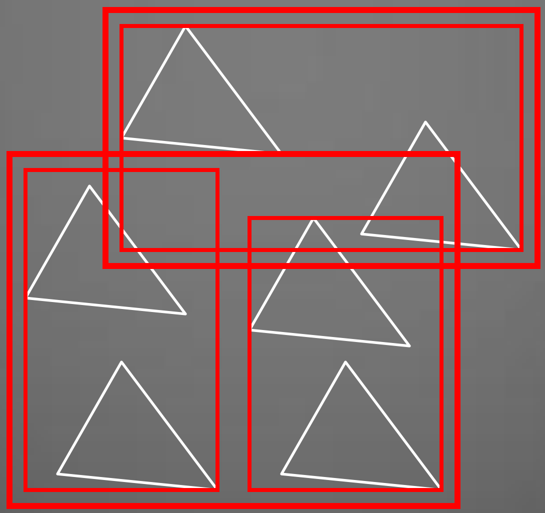
Many Primitives

Use hierarchical data structures for speed



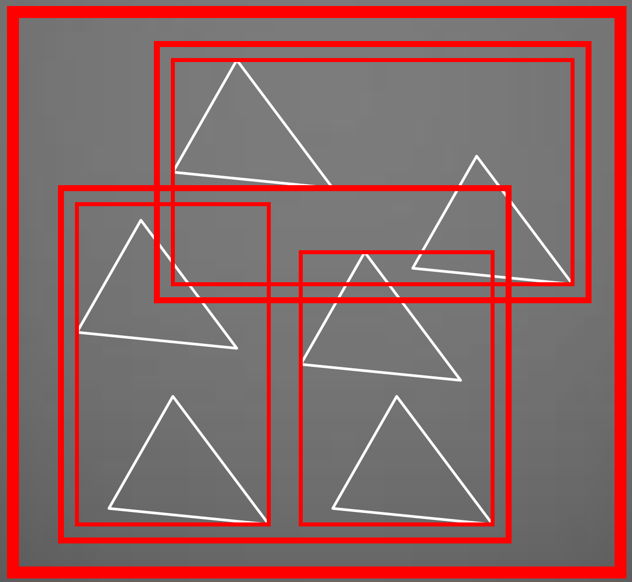
Many Primitives

Use hierarchical data structures for speed

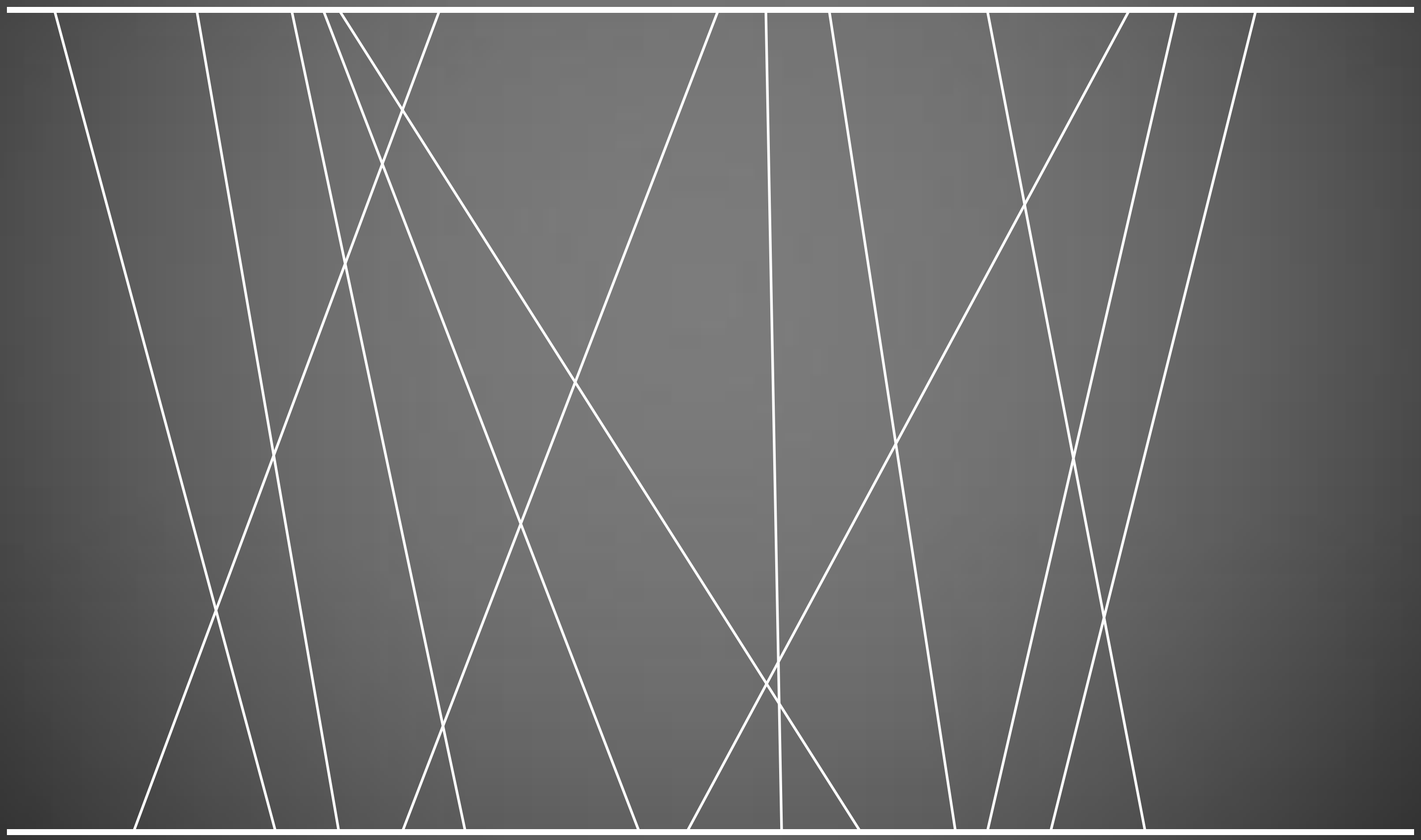


Many Primitives

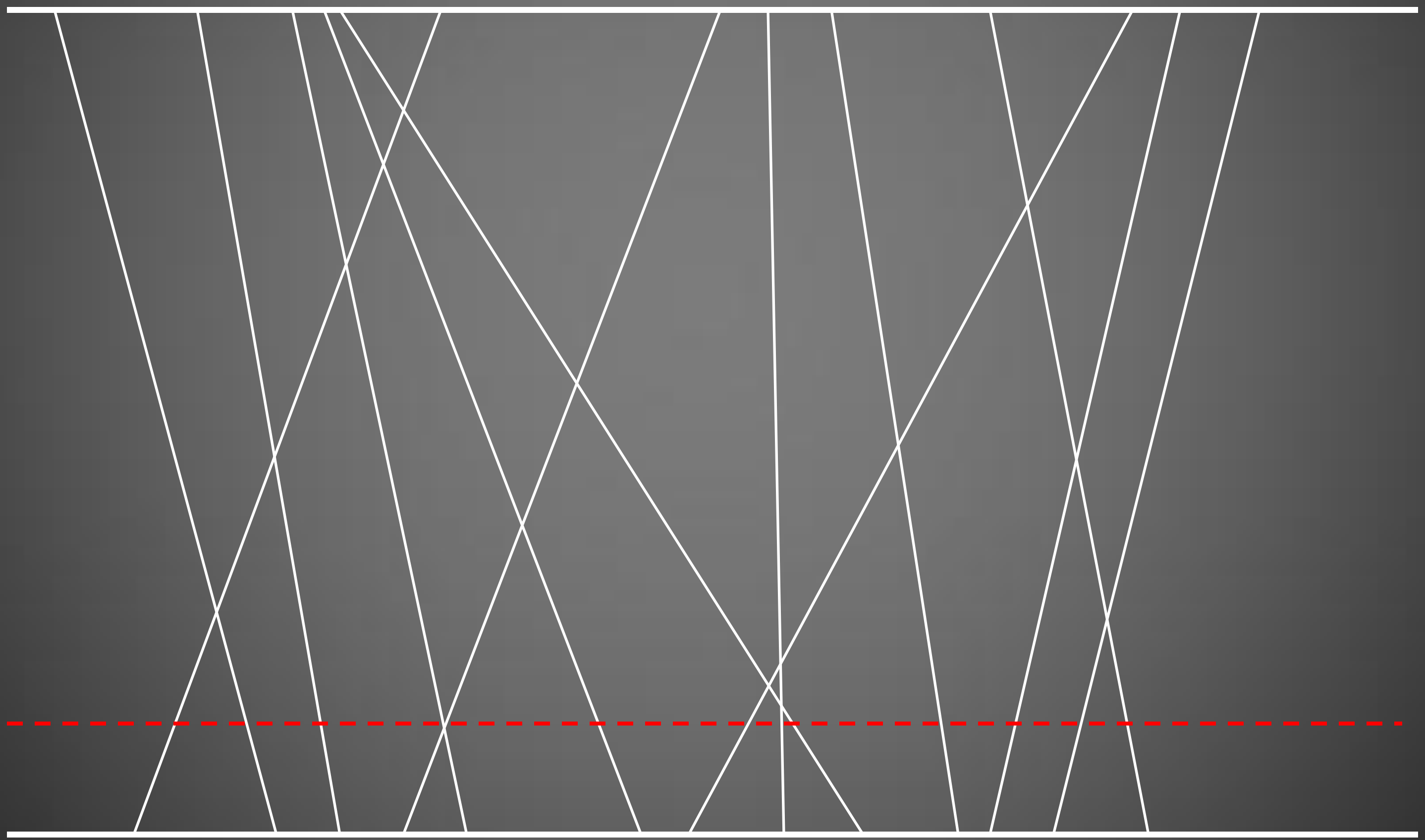
Use hierarchical data structures for speed



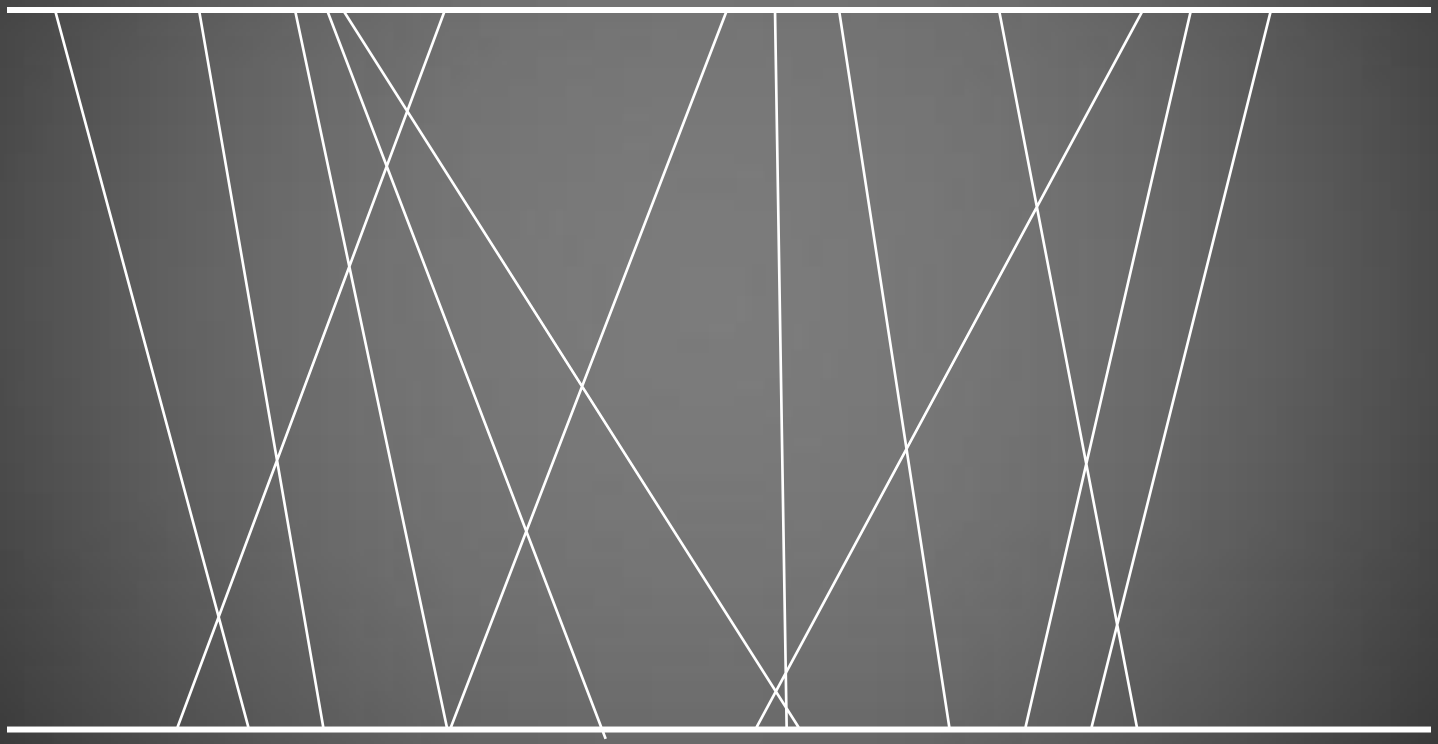
Many Primitives



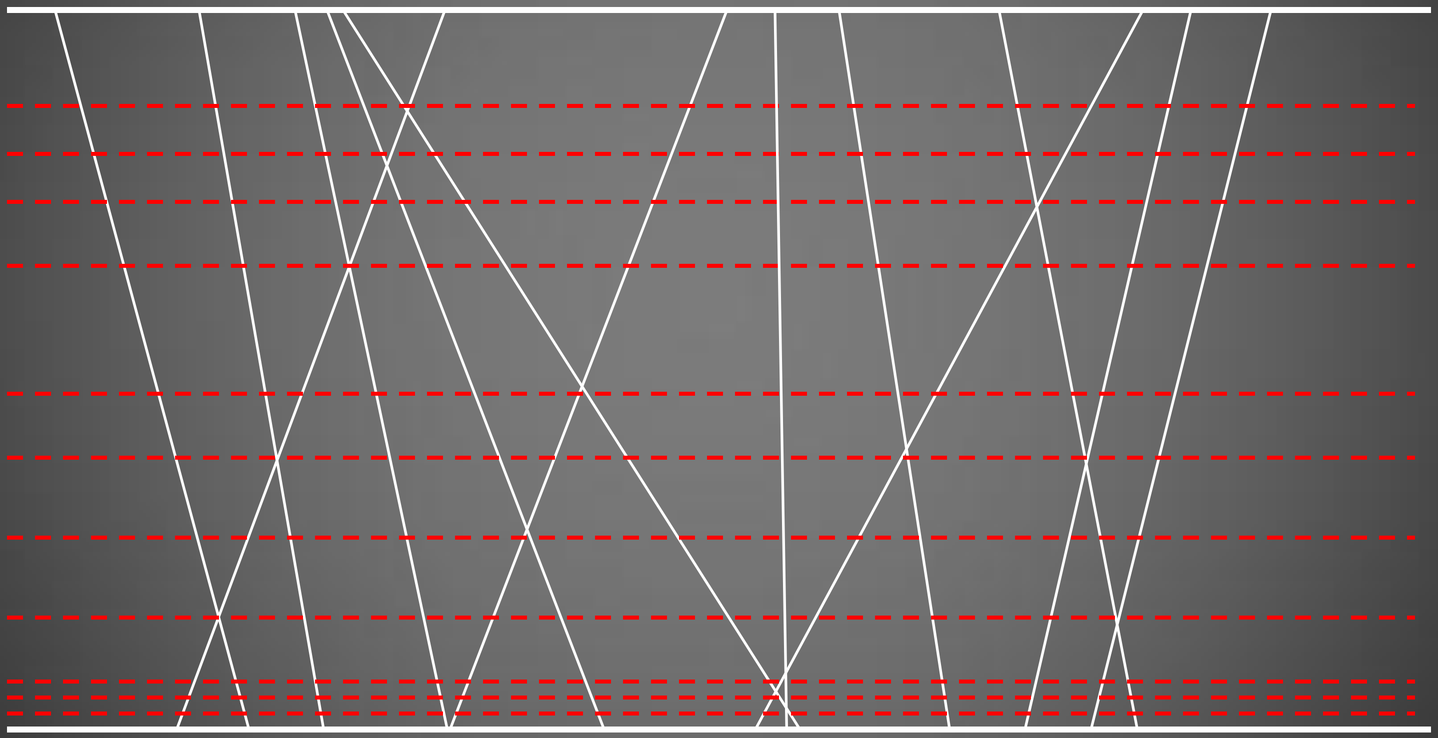
Many Primitives



Many Primitives



Many Primitives

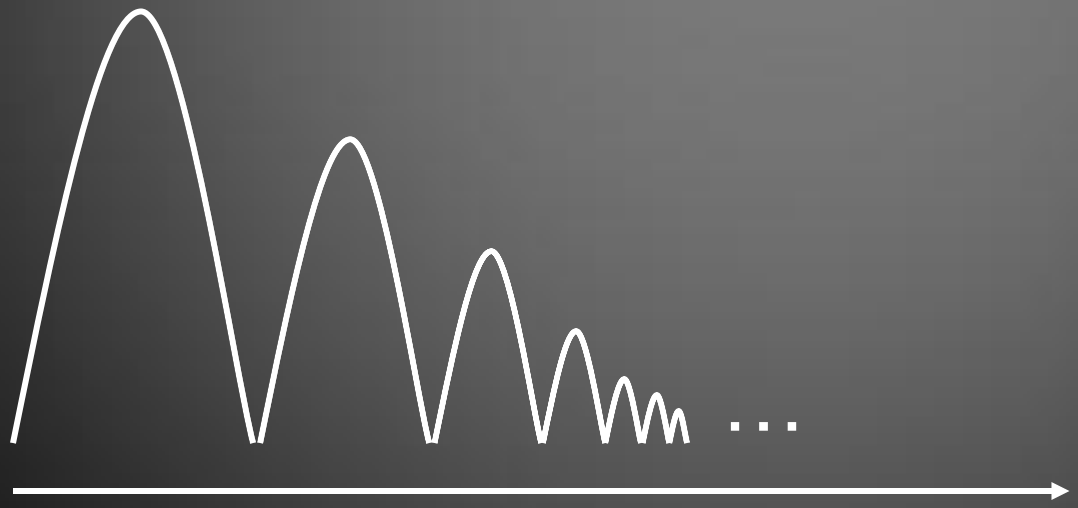


Etc. Expensive in General

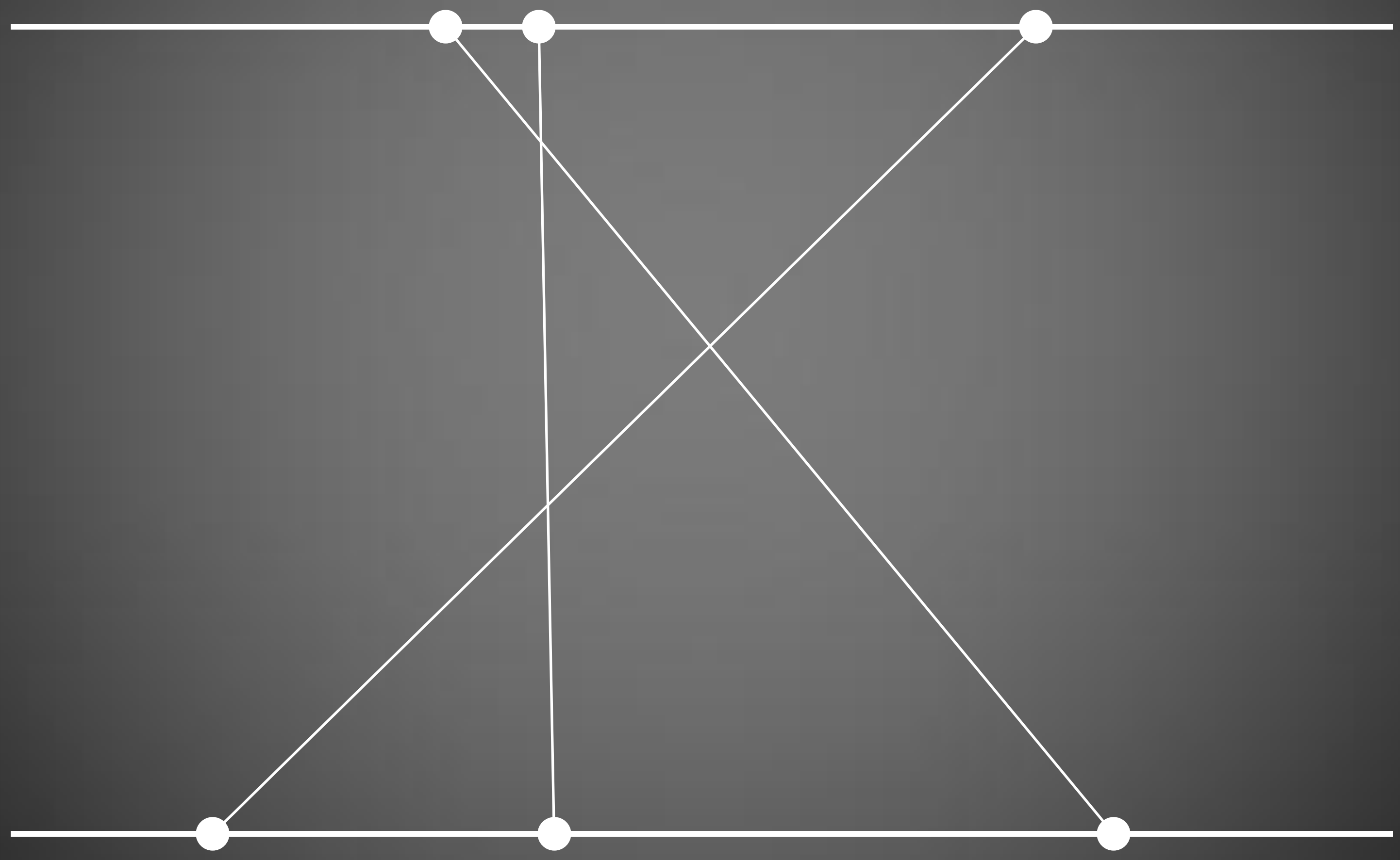
Many Primitives

Our approach:
Iterate over entire time step
Until all collisions resolved.

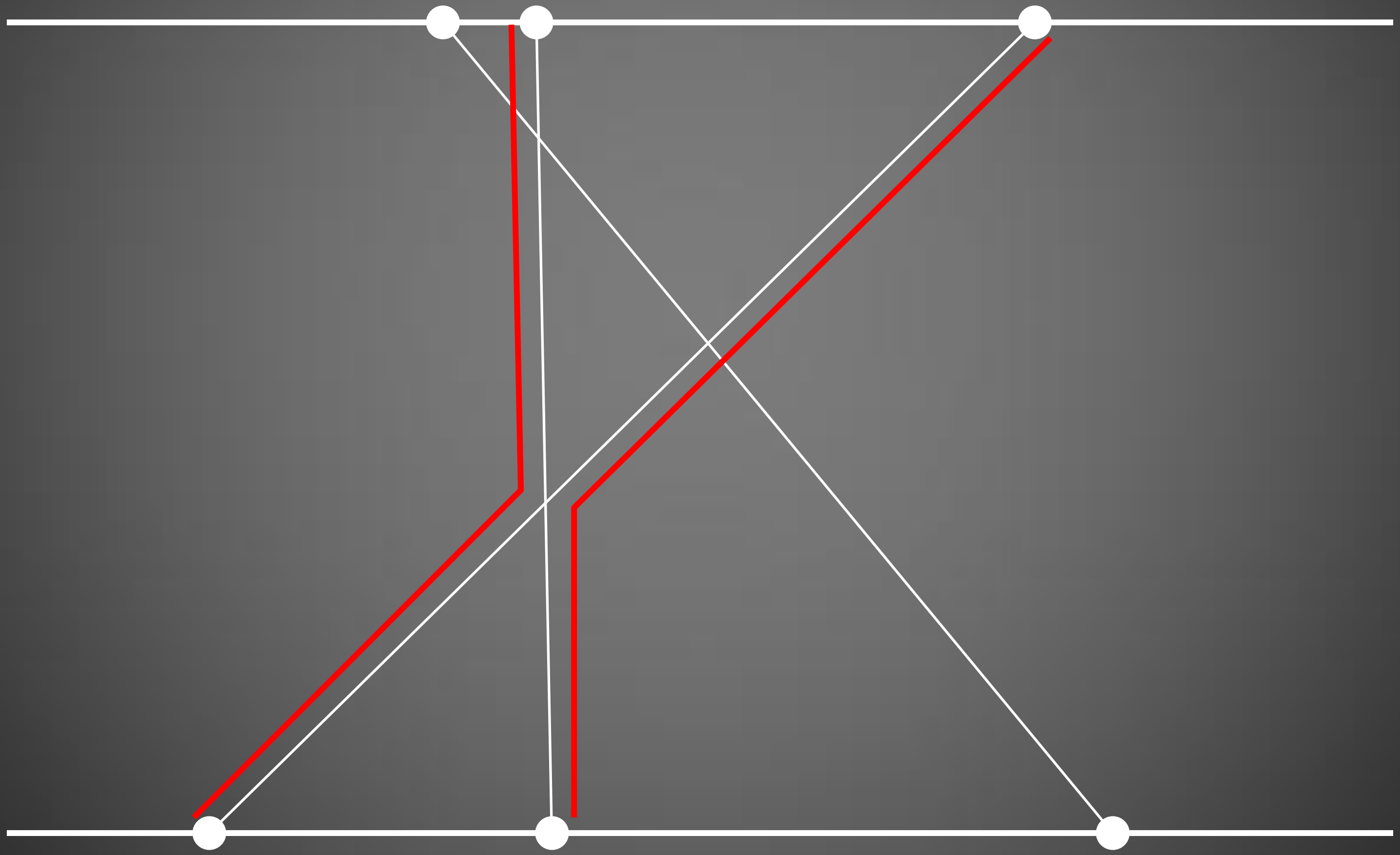
Avoids lockups and Zeno's paradox



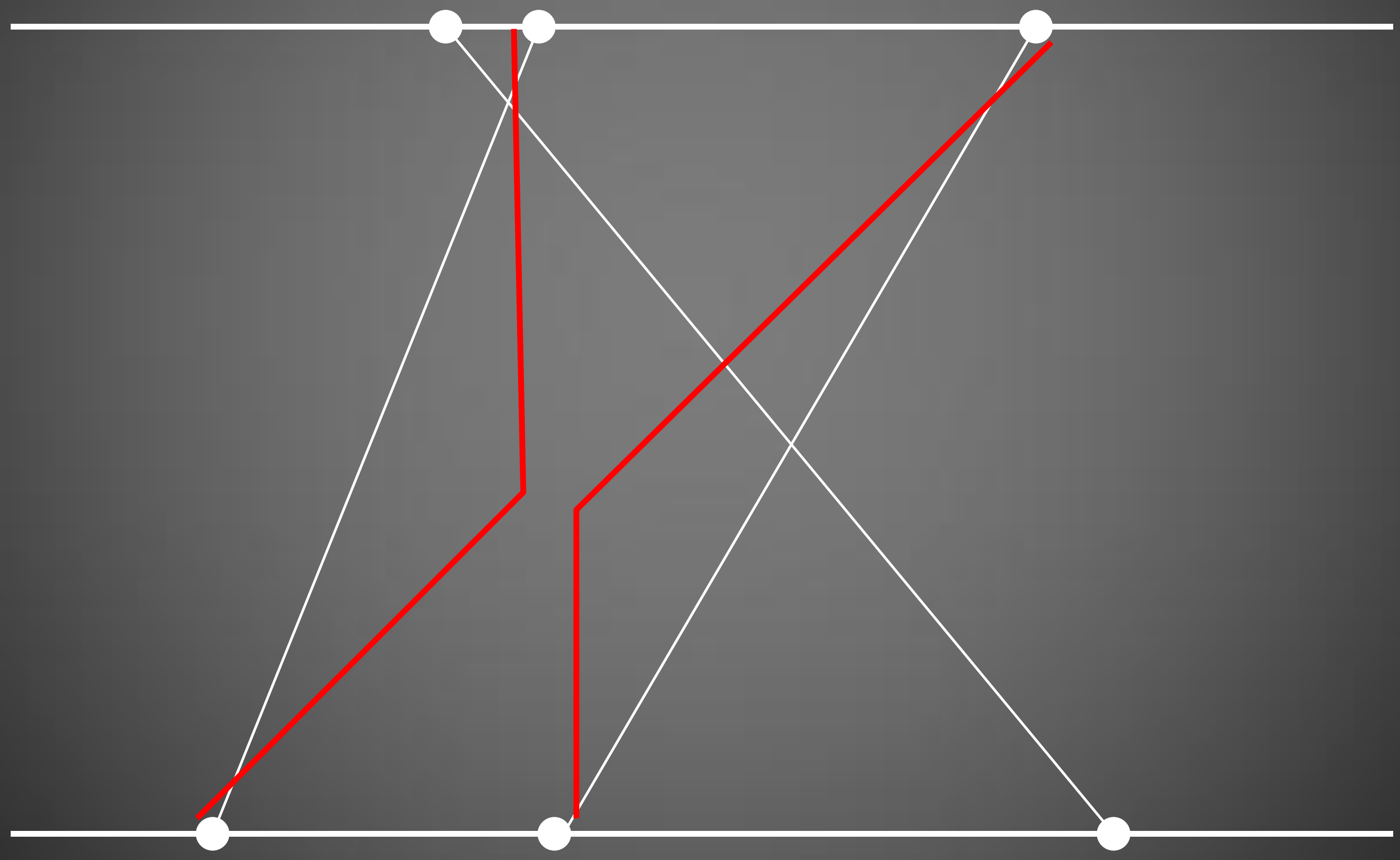
Many Primitives



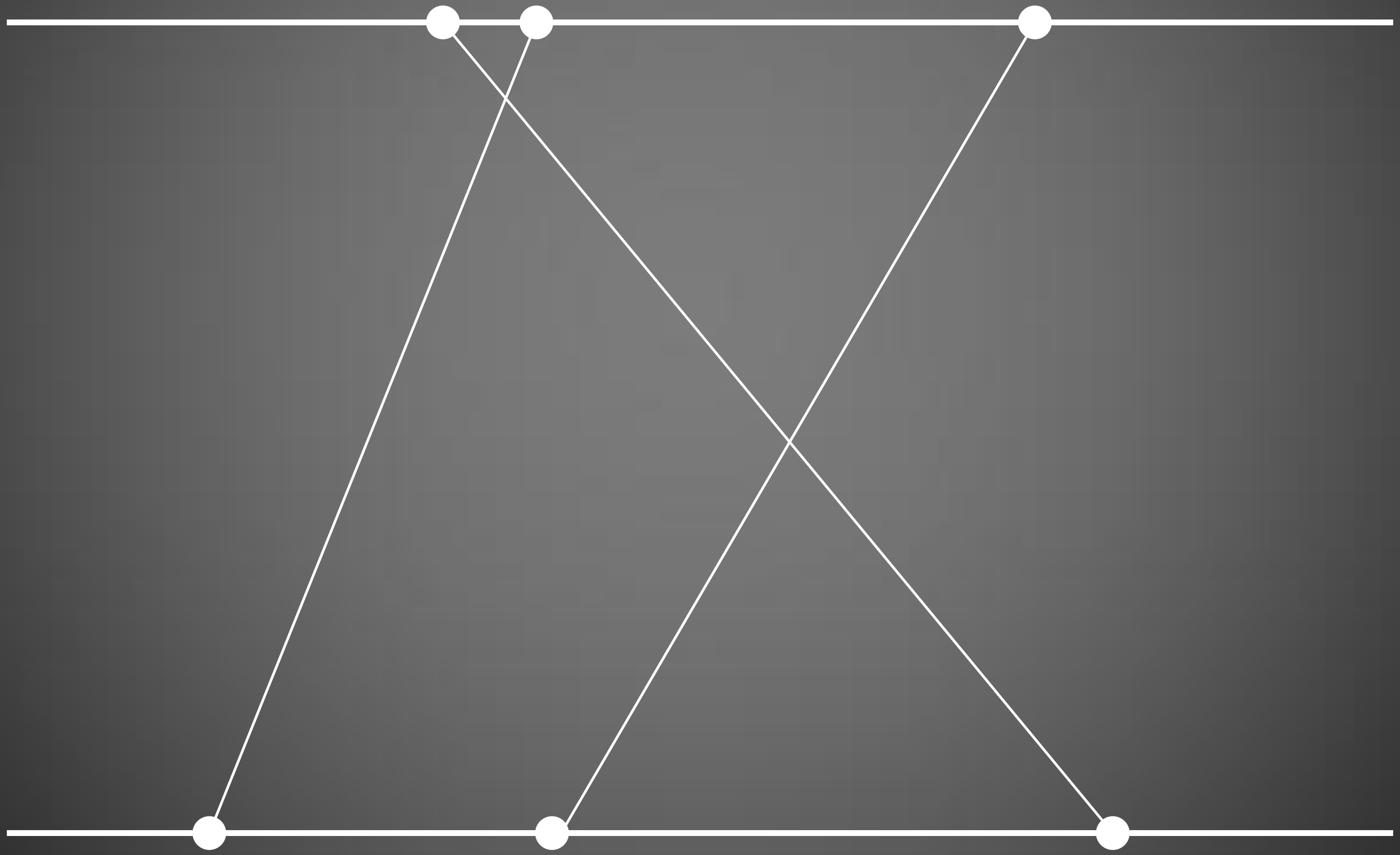
Many Primitives



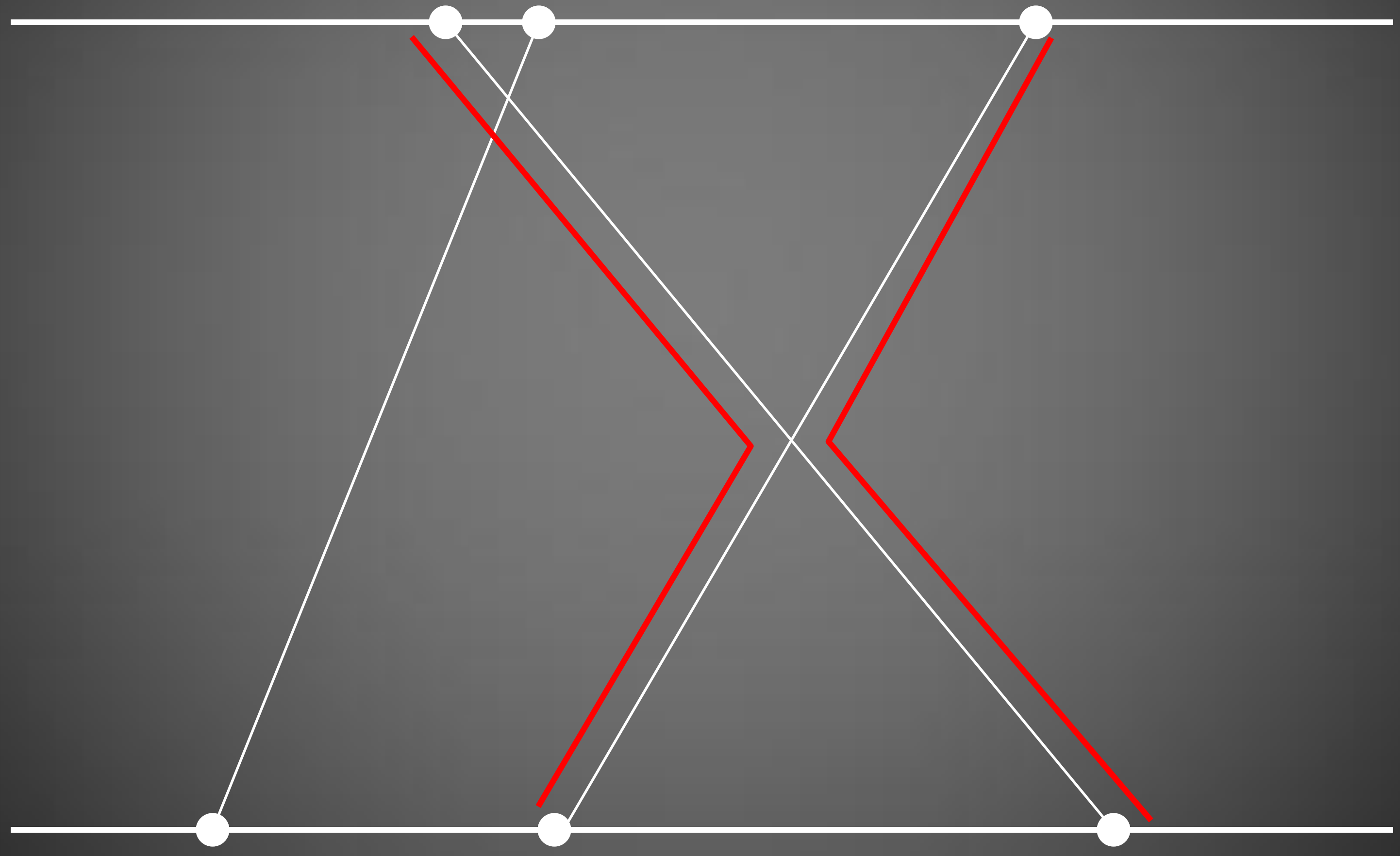
Many Primitives



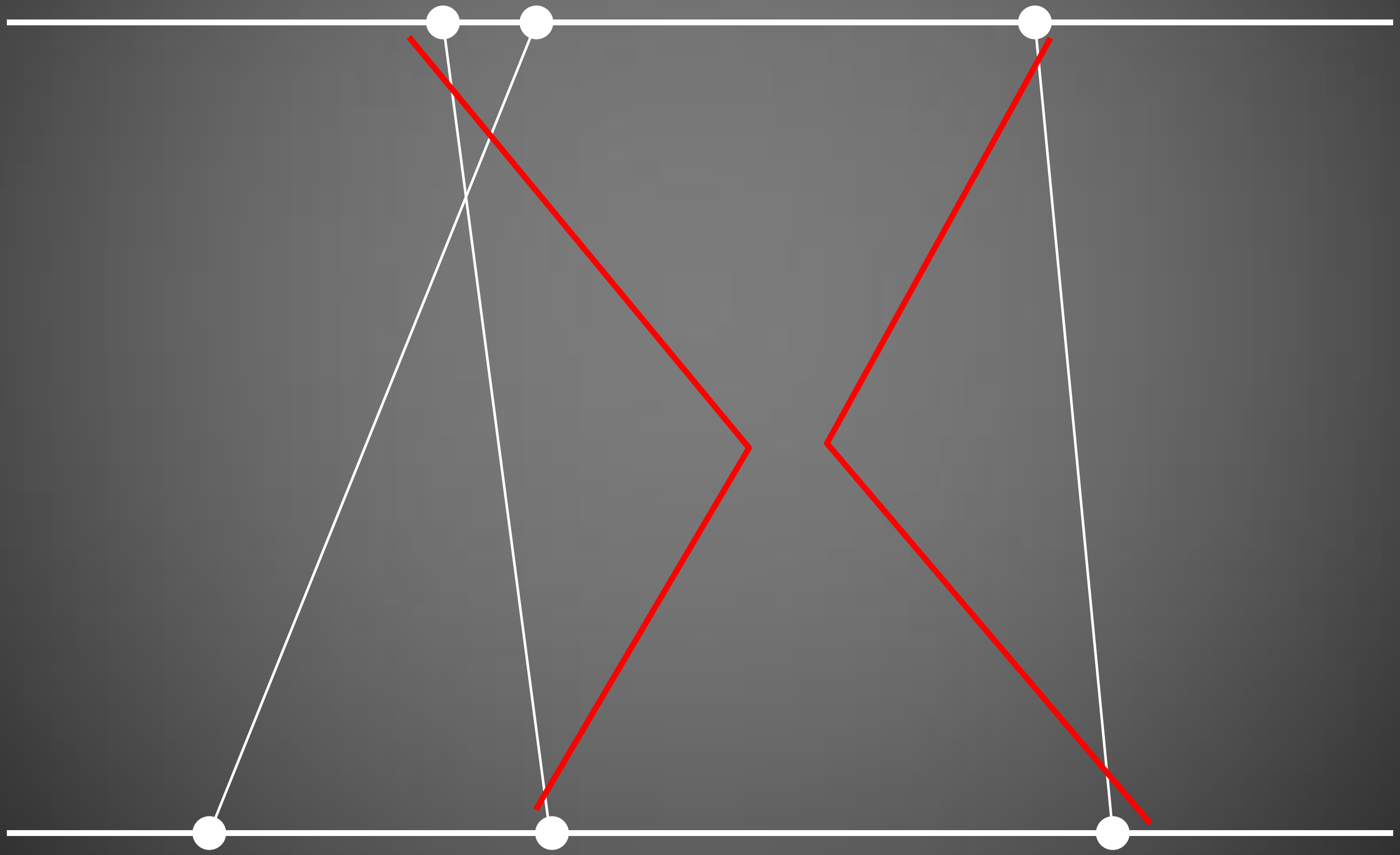
Many Primitives



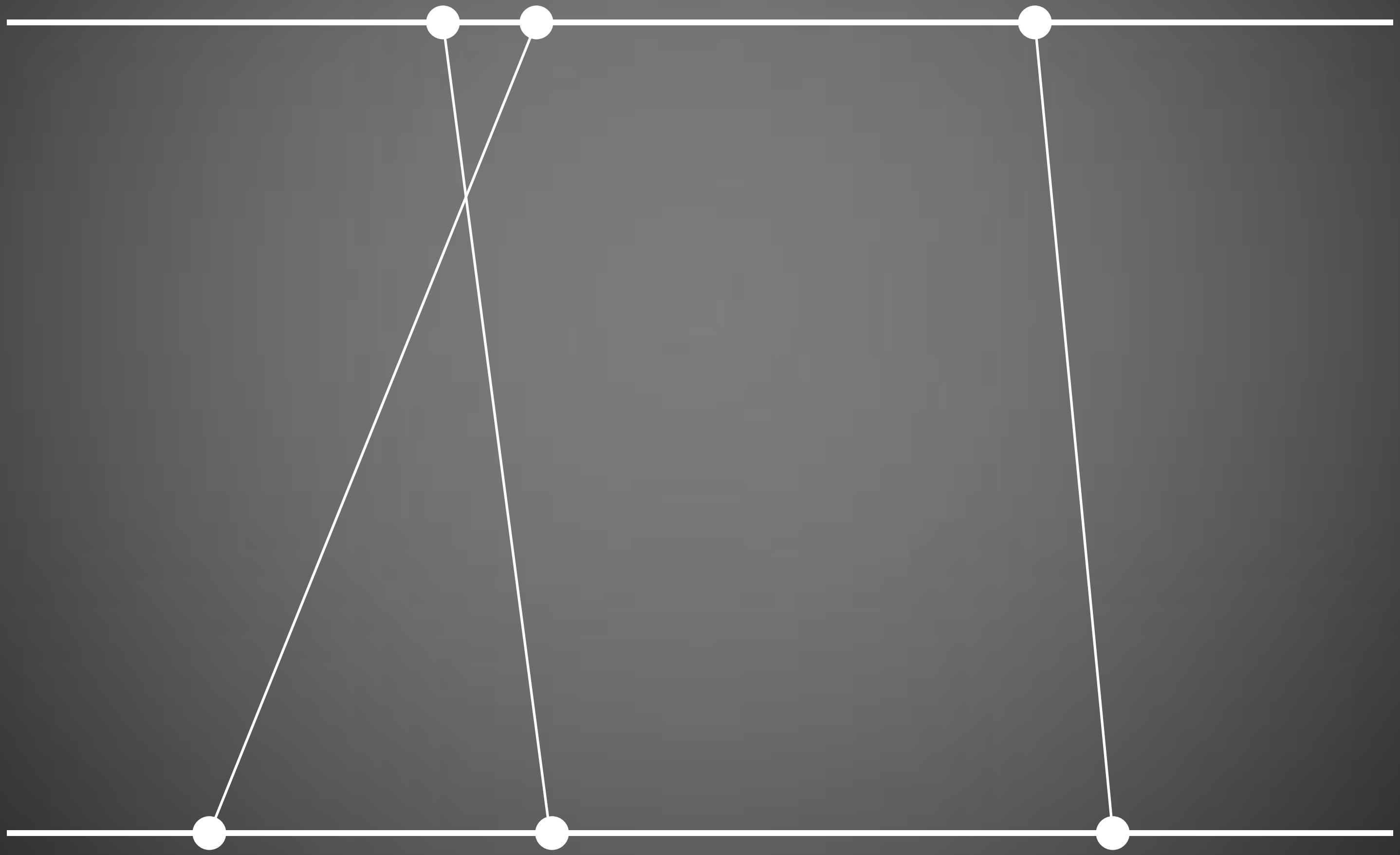
Many Primitives



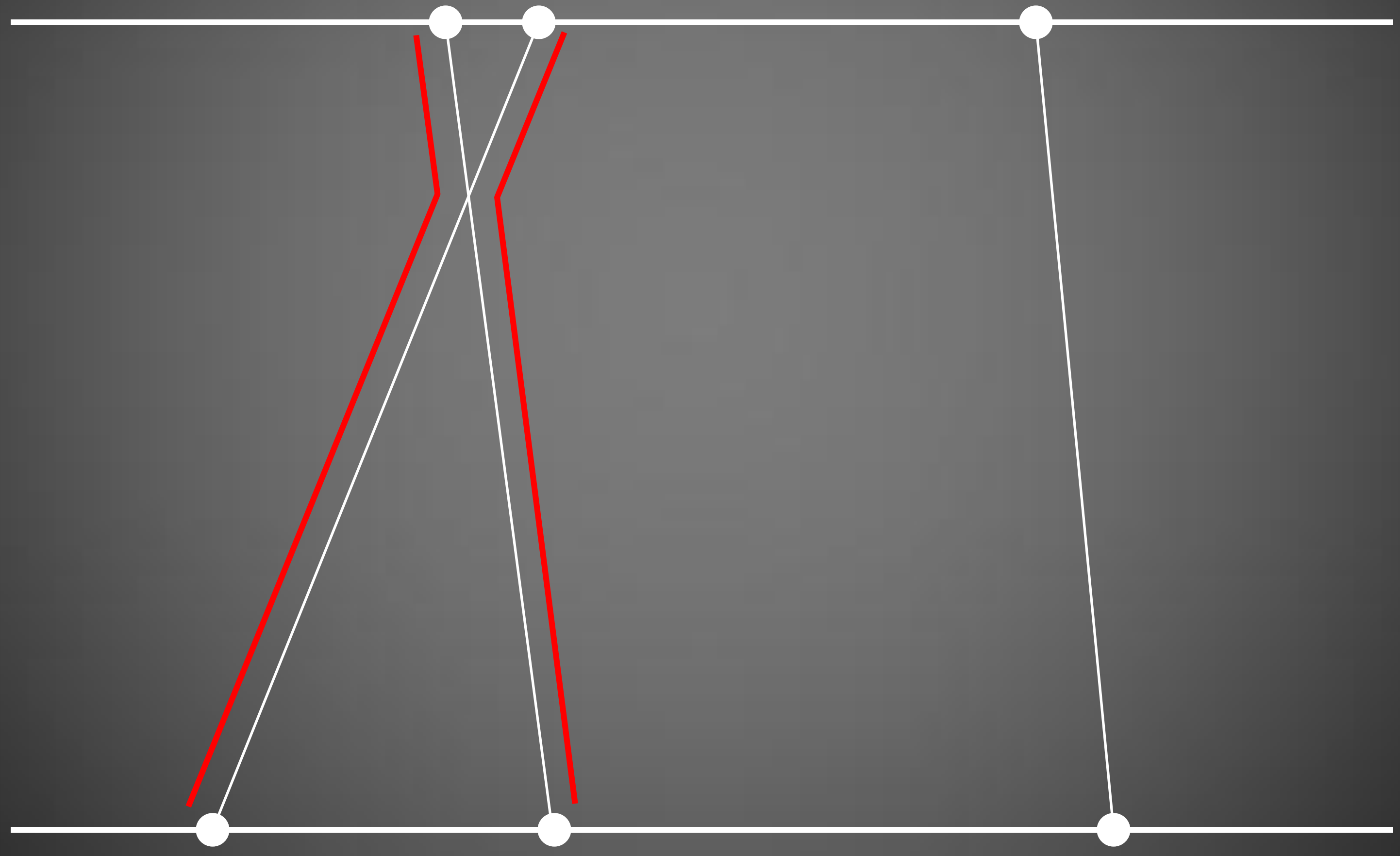
Many Primitives



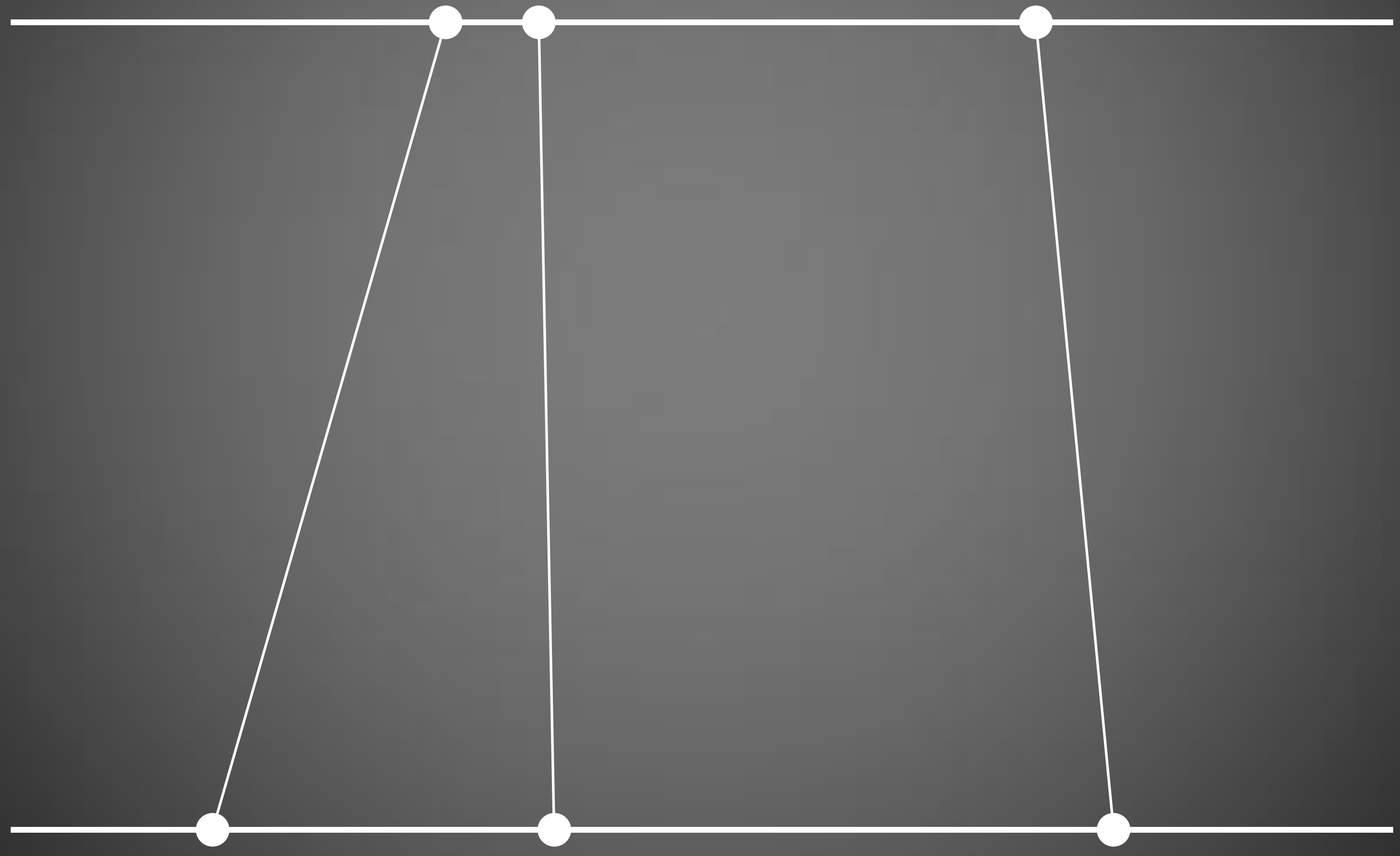
Many Primitives



Many Primitives



Many Primitives



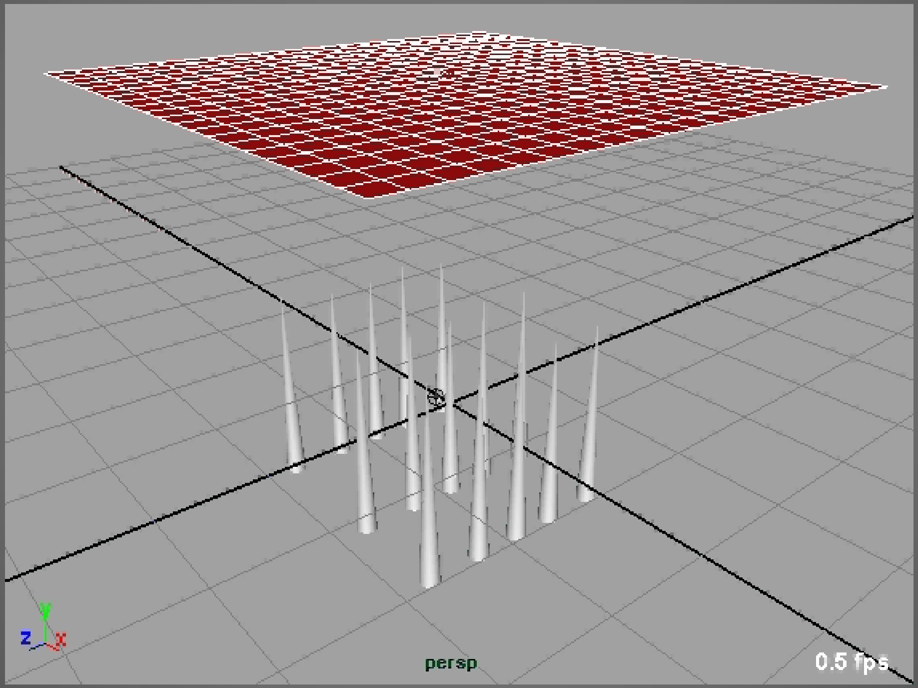
Demos

1D demo

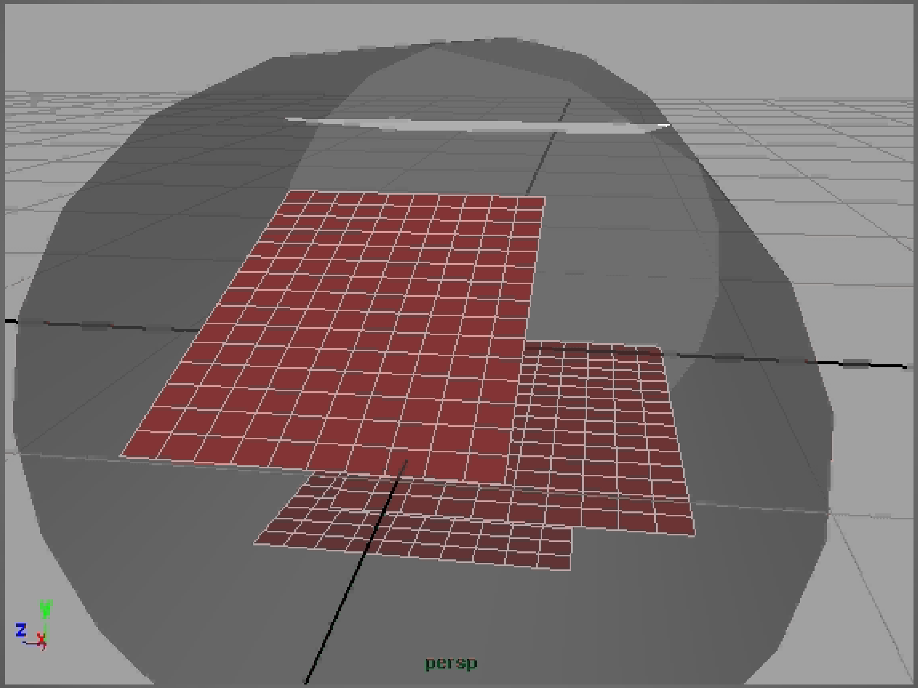
(t key for space-time)

3D demo

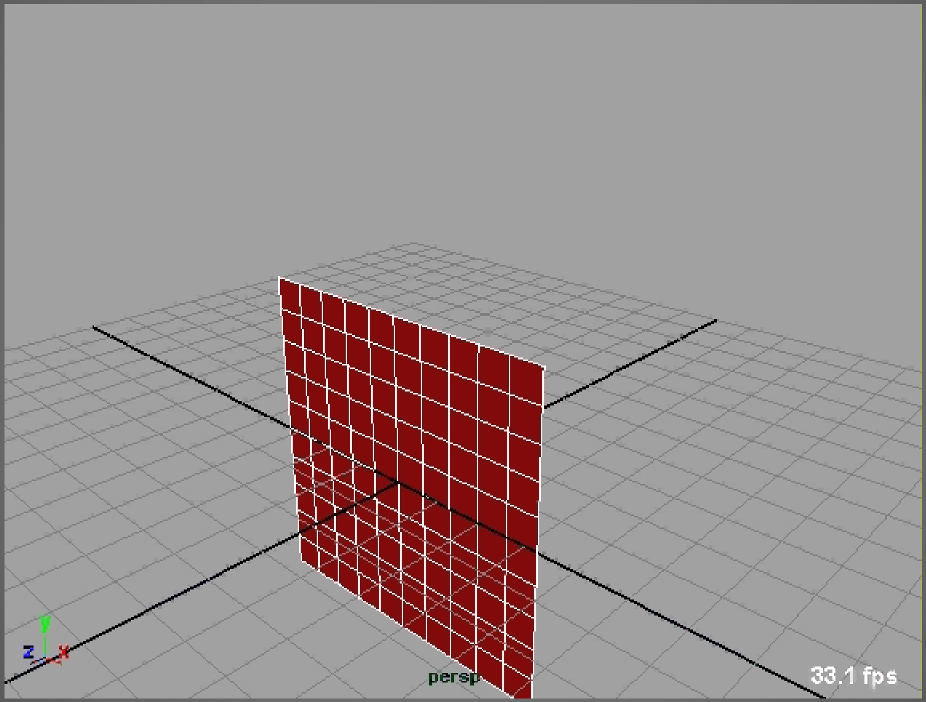
Torture Tests



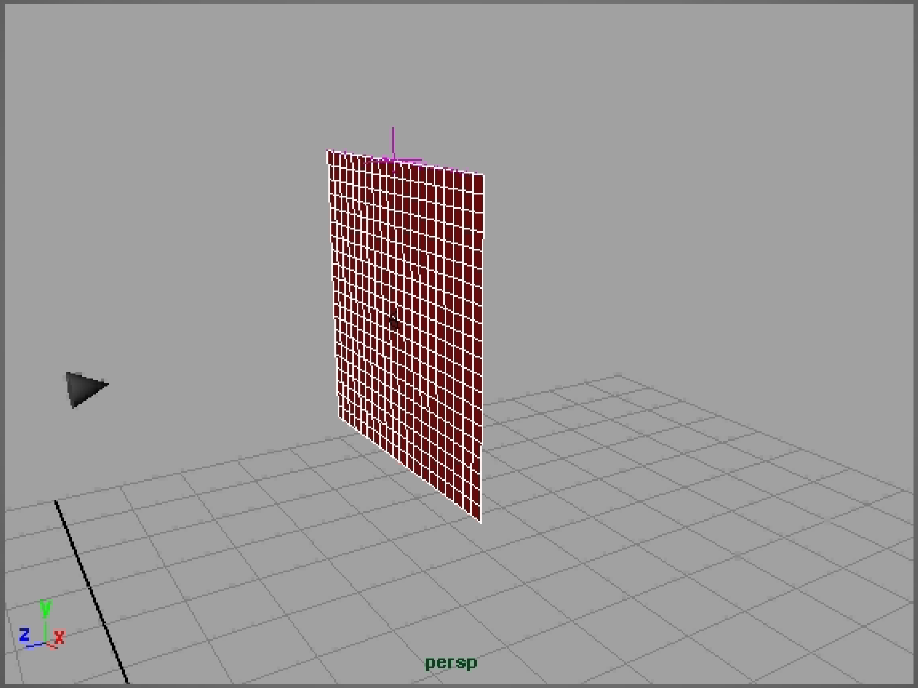
Torture Tests



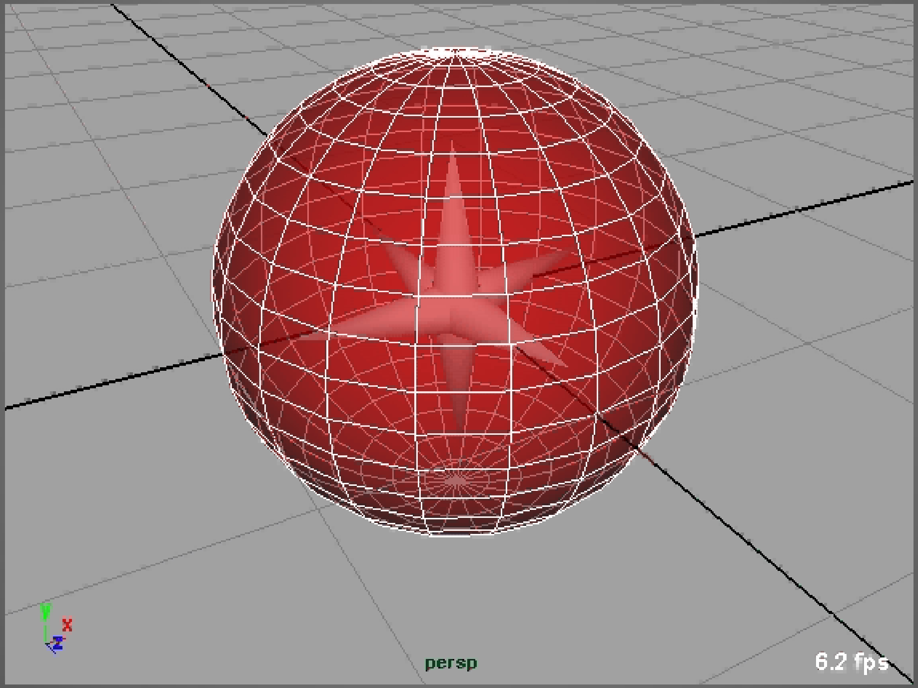
Torture Tests



Torture Tests



Torture Tests



Torture Tests

Full render 1

Full render 2

Pressure Model

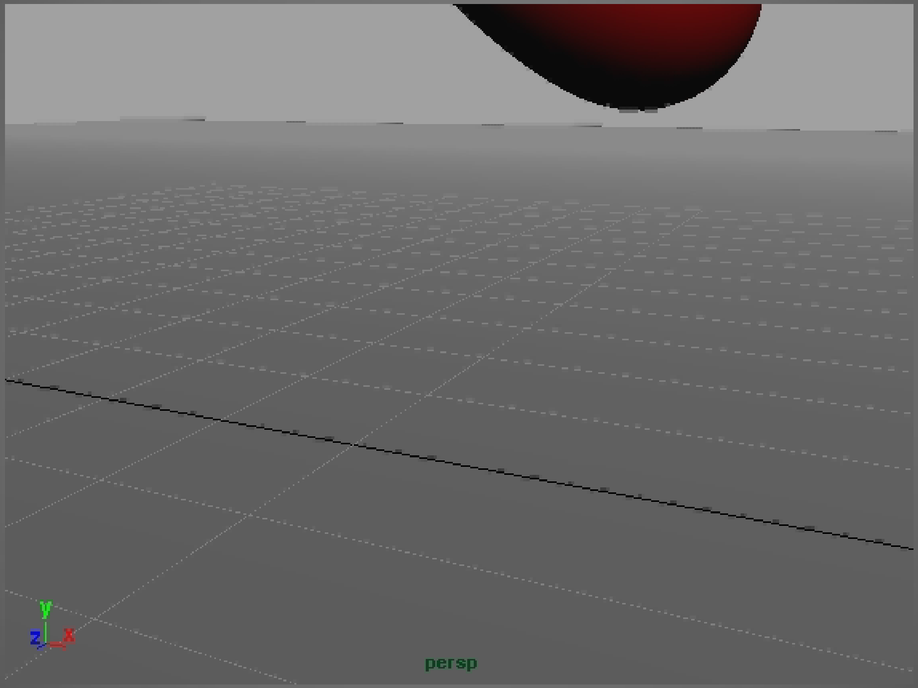
$$PV\left(\mathbf{x}+\mathbf{v}+\delta\mathbf{v}\right)=cM$$

P : pressure

M : mass

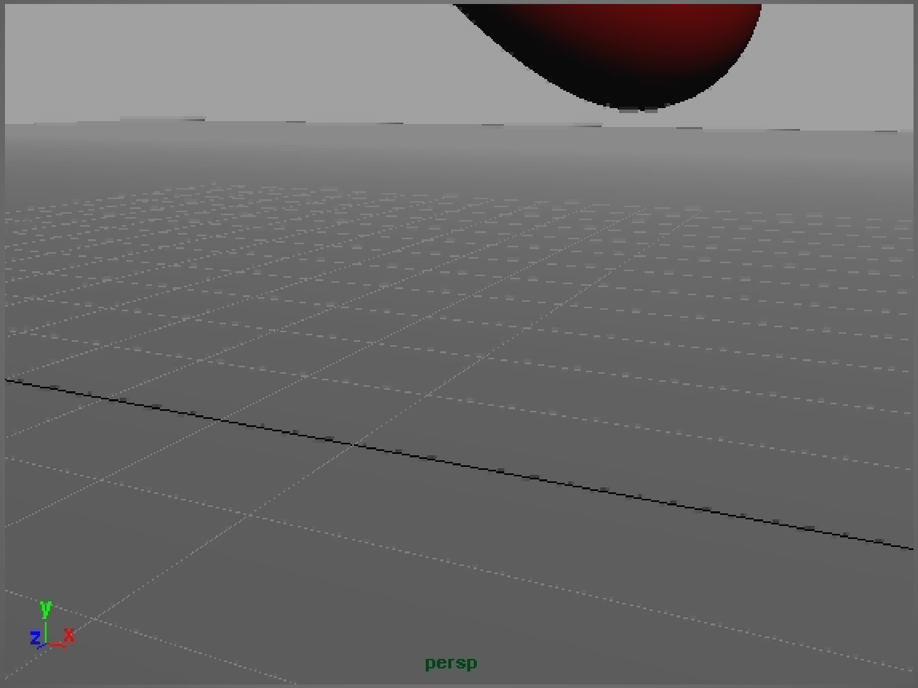
V : volume

Pressure Model



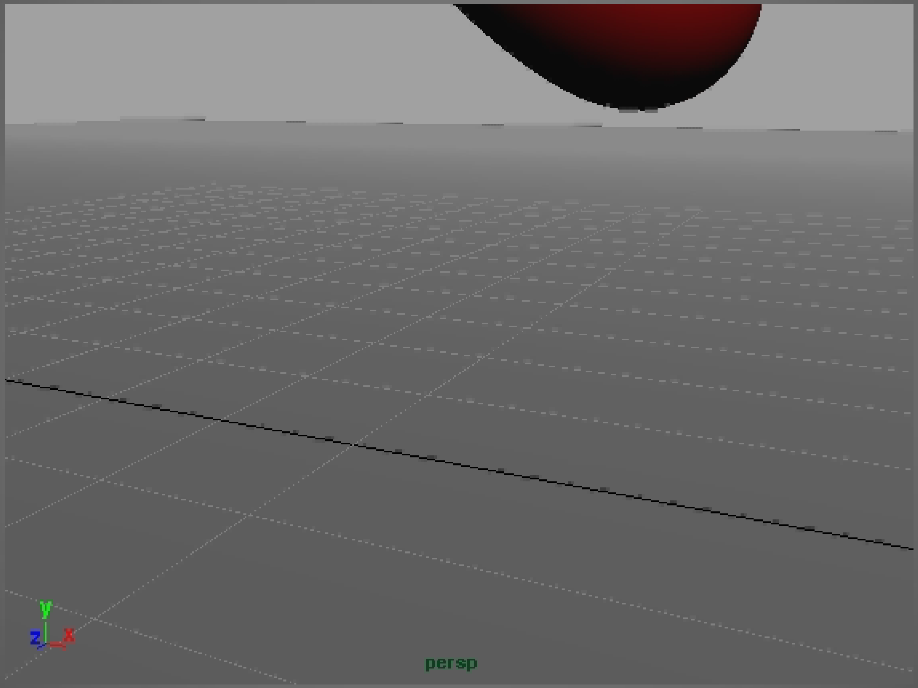
No pressure

Pressure Model



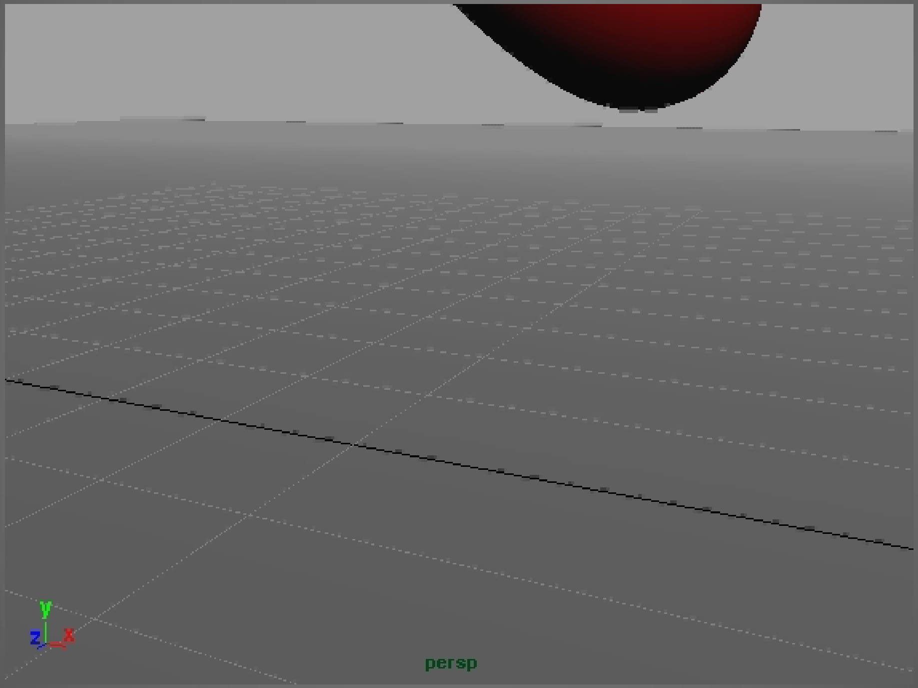
Volume conservation

Pressure Model



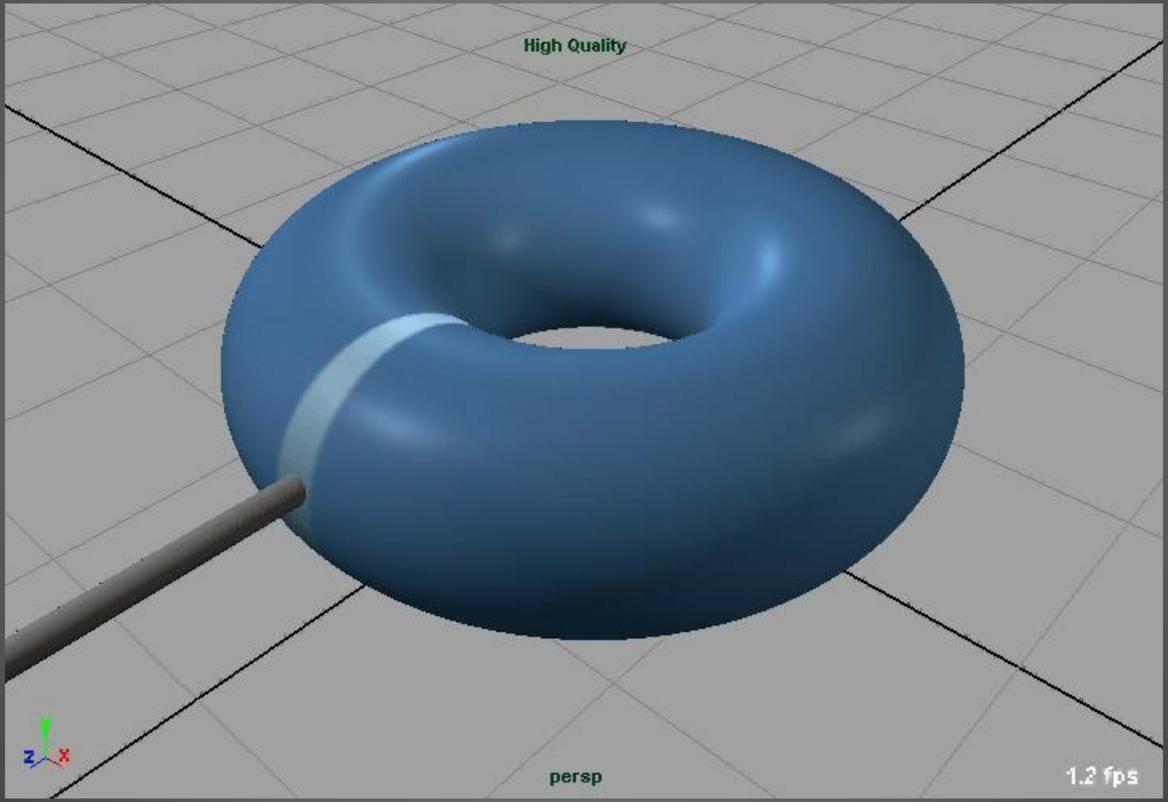
Under pressure

Pressure Model



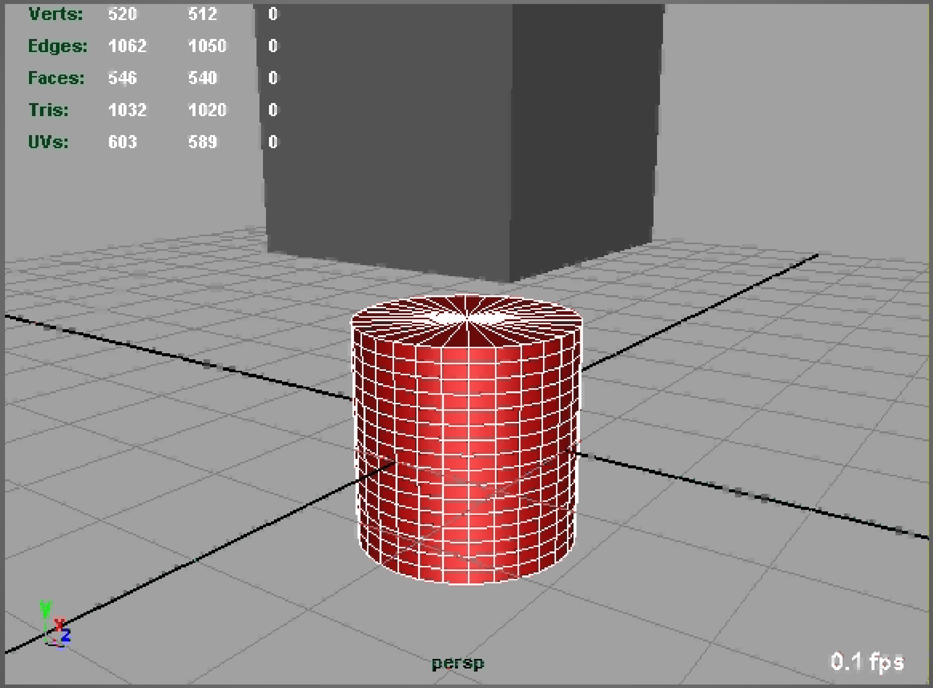
Under a lot of pressure

Pressure Model



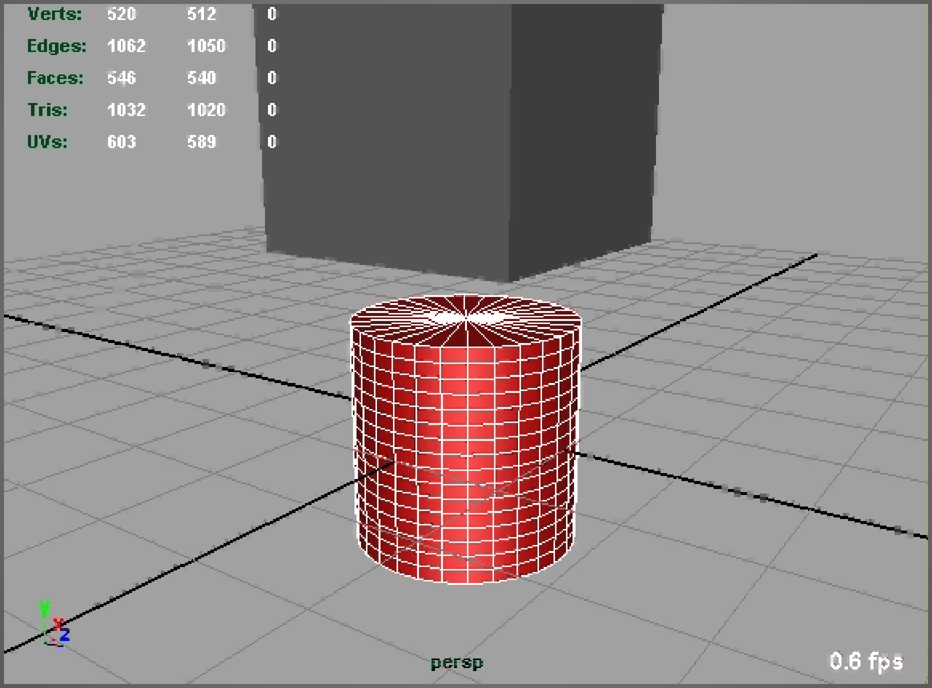
Air tightness + Pump rate

Bend



Fully rendered

Bend

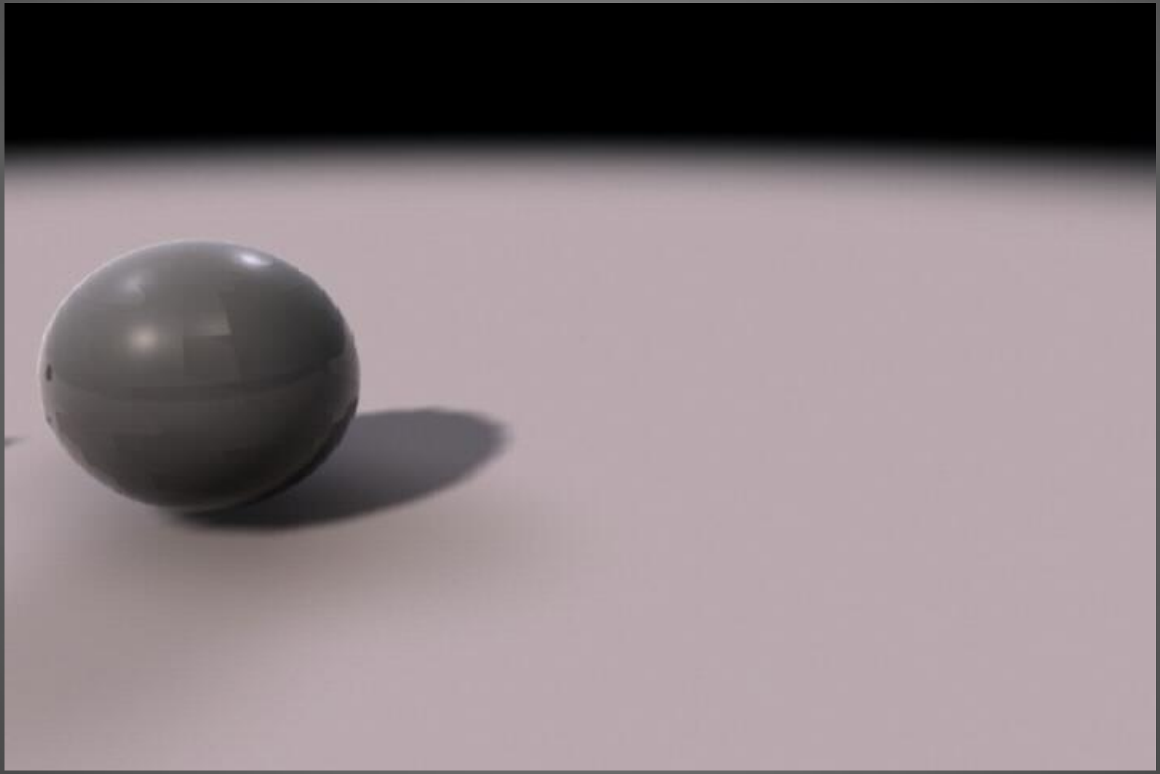


Fully rendered

Bend

Car crash

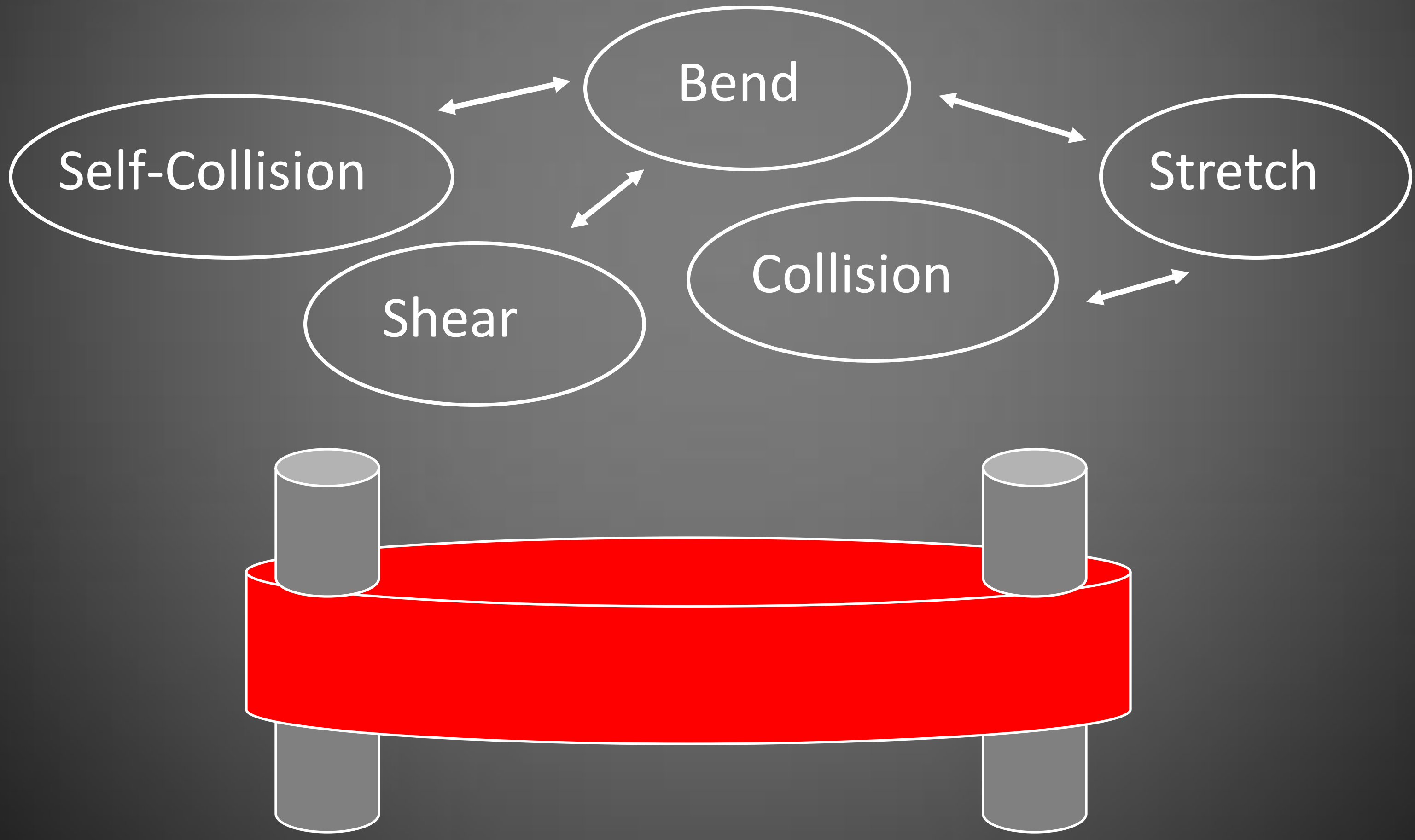
Fracture



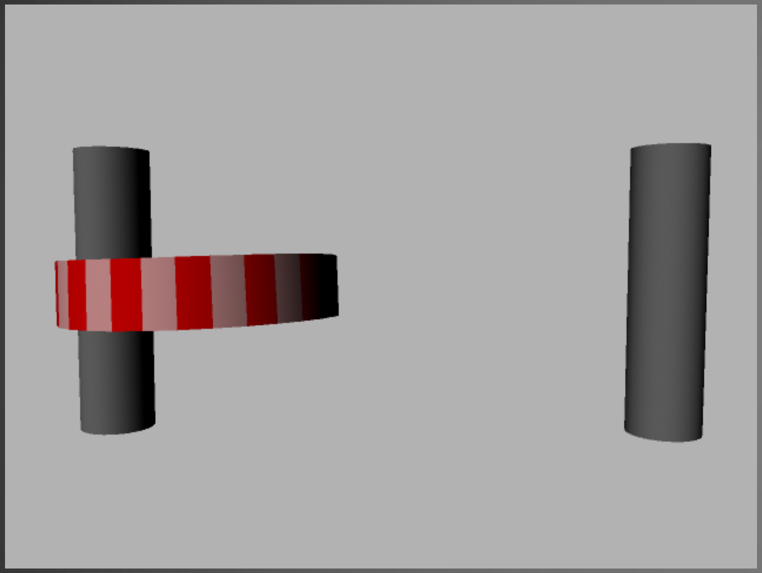
Animation

General Solver

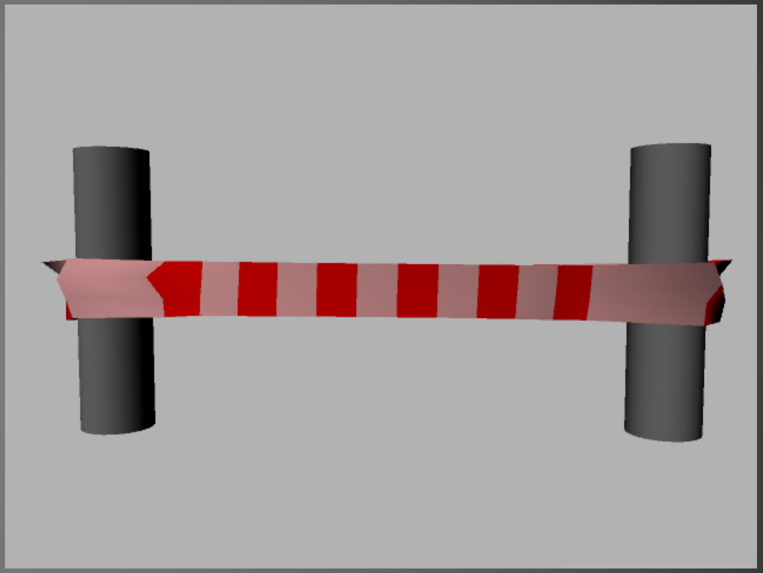
Battle of the constraints



General Solver



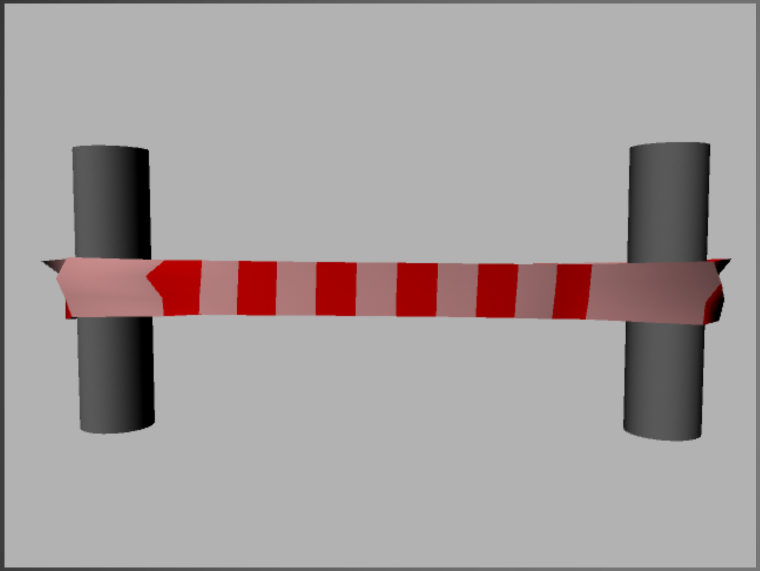
Collisions/stretch



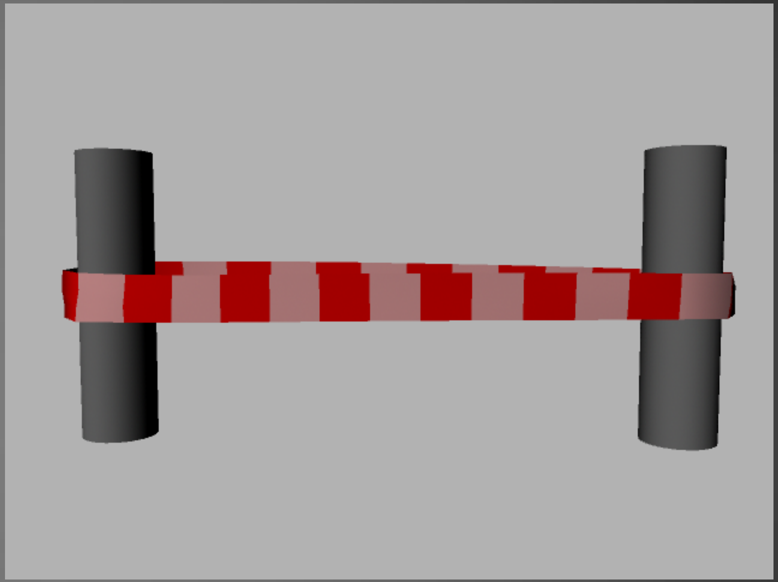
Stretch/collisions

General Solver v1.0

General Solver

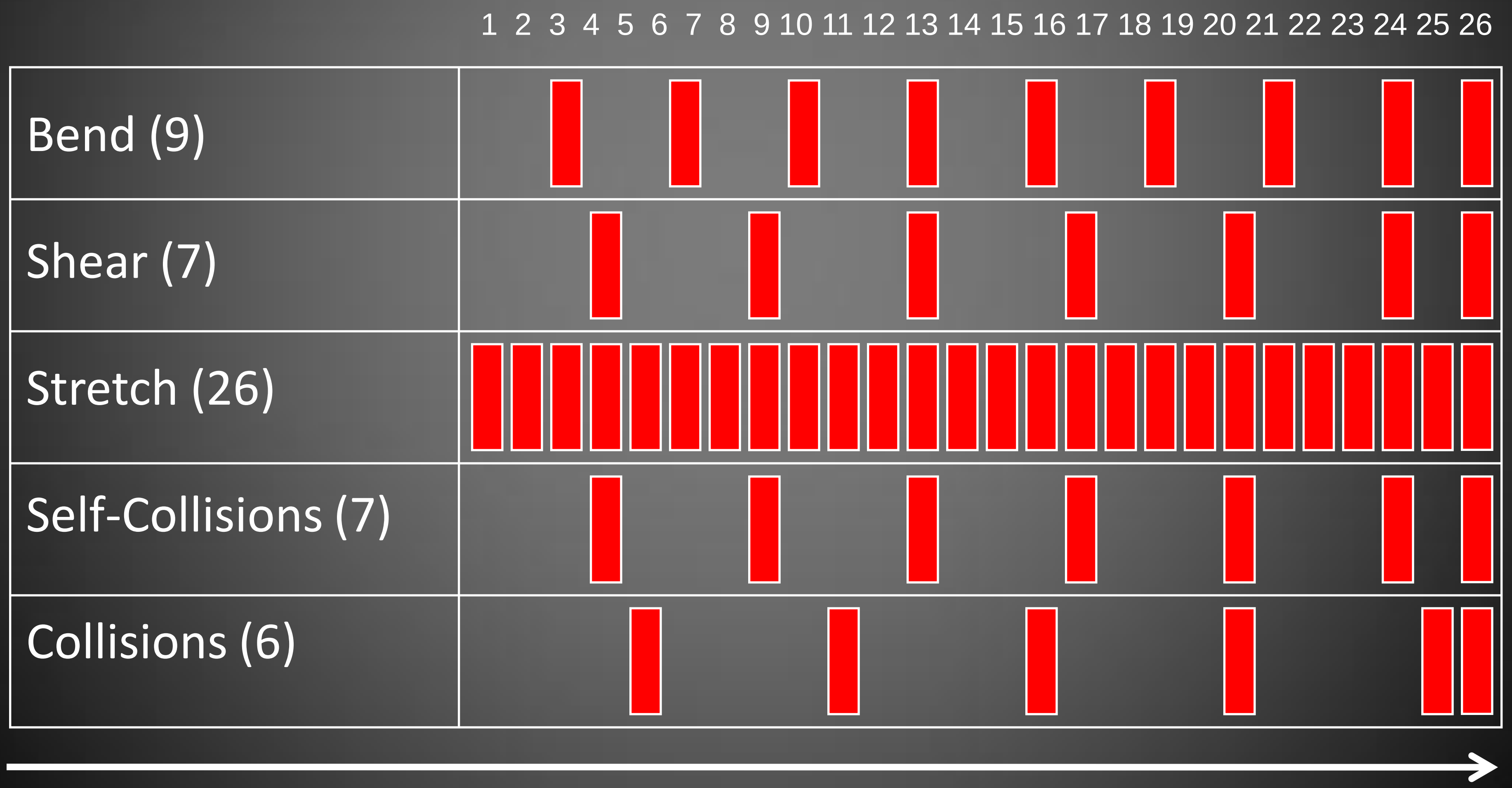


not interleaved

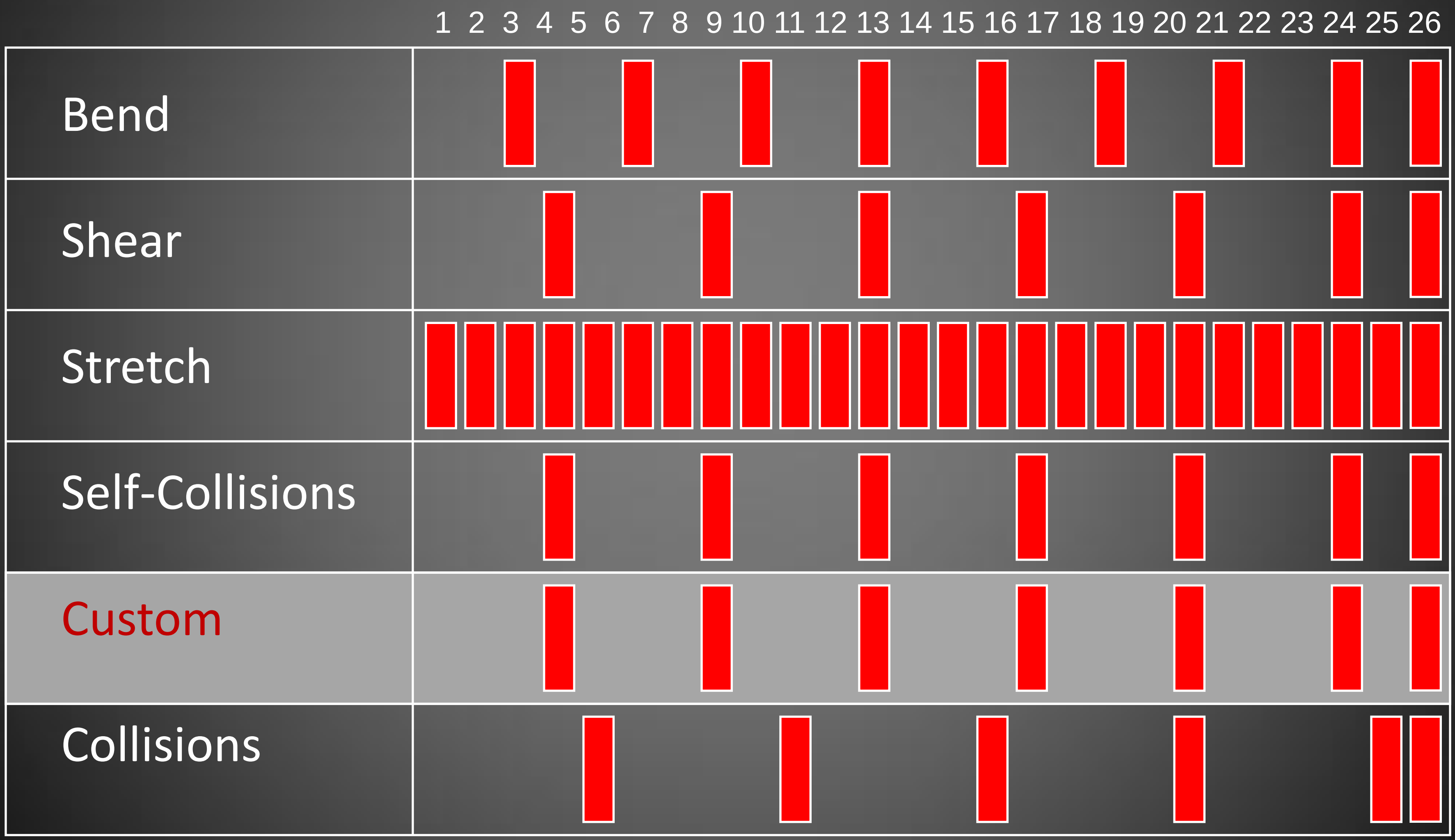


interleaved

General Solver v2.0



General Solver



Demos

Curve demo

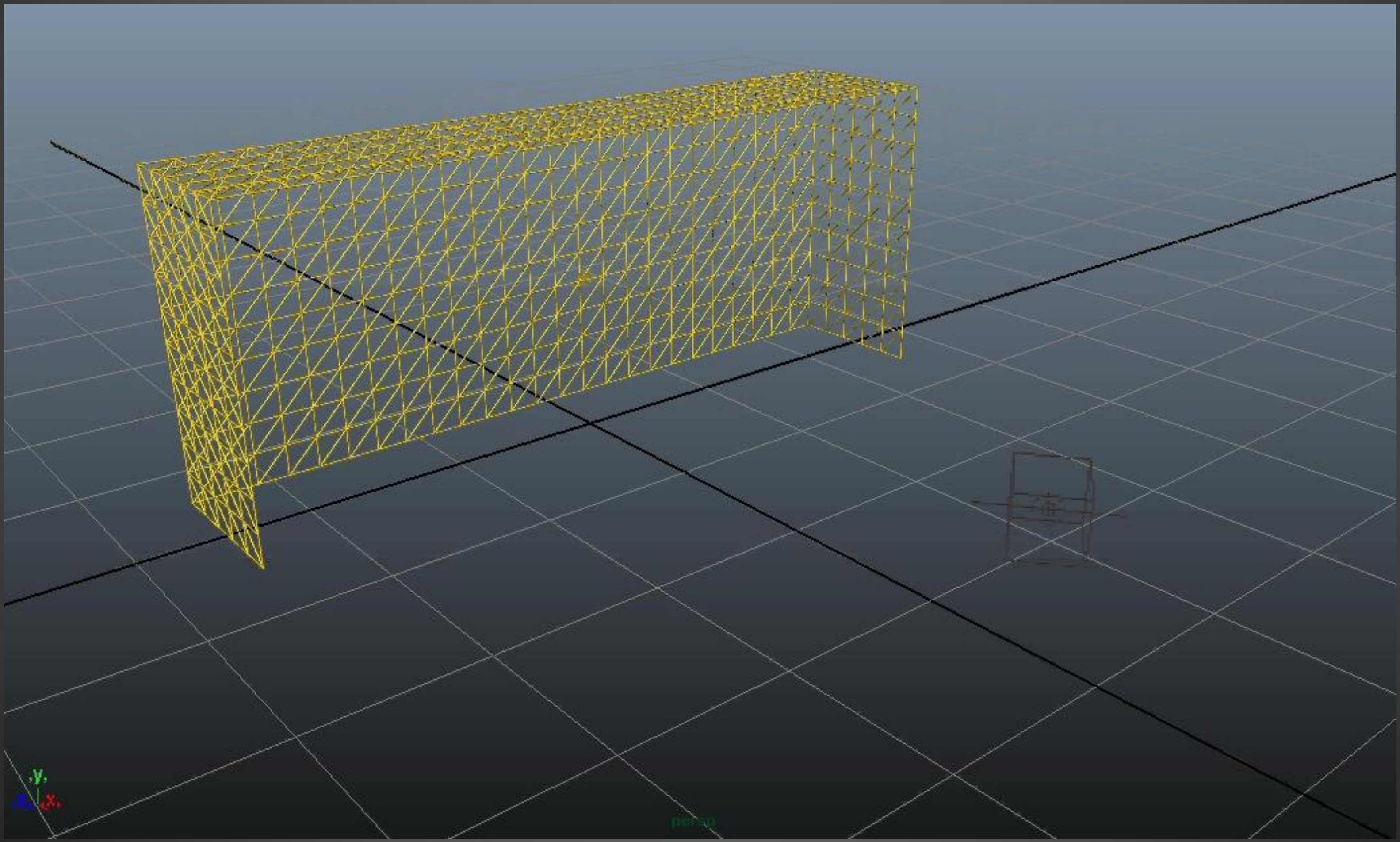
Cloth demo

nParticles

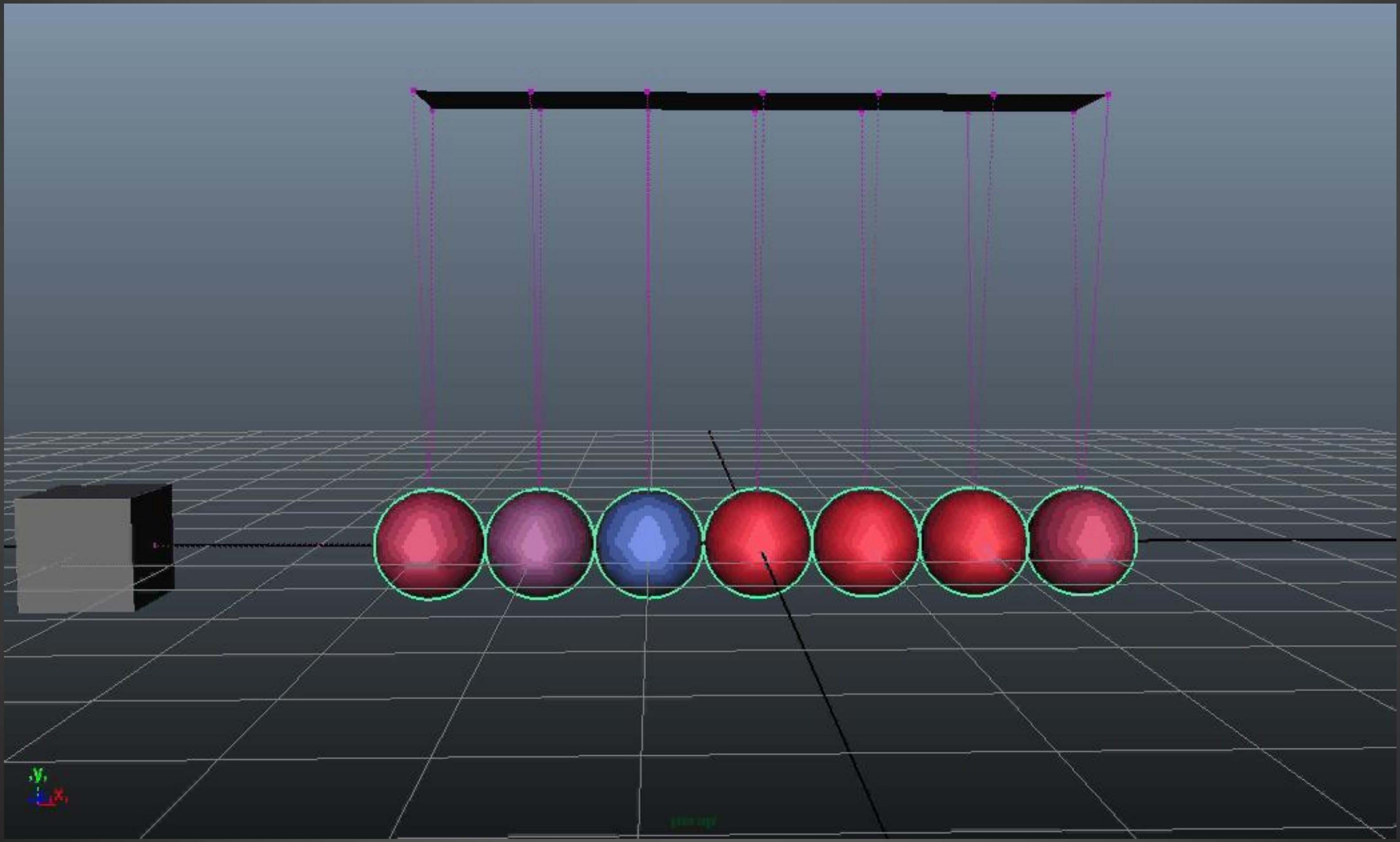
New in MAYA 2009

Extends exisiting particles

nParticles



nParticles



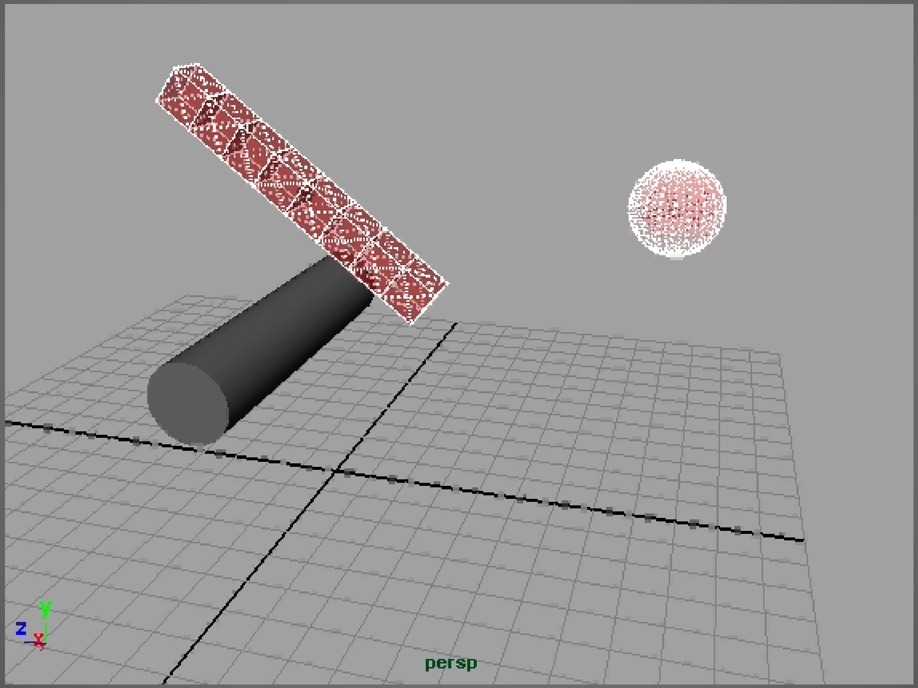
nParticles

$$\rho\left(\mathbf{x}+\mathbf{v}+\delta\mathbf{v}\right)-\rho_0=0$$

$$\rho\left(\mathbf{x}\right)=\sum_{i=1}^{N} m_i W\left(\mathbf{x}-\mathbf{x}_i\right)$$

Stable SPH

Solids



More Cloth Examples

Inflatable Girl

Inflatable Guy

Rigid bodies

ballerina

ballerina 2

Cloth drop

Duncan on the AREA

Brain

http://area.autodesk.com/index.php/blogs_duncan/tag_list/welcome/

Future Work

Other nThings

Improve Collisions

Improve Constraint Solver

The End

Thank You