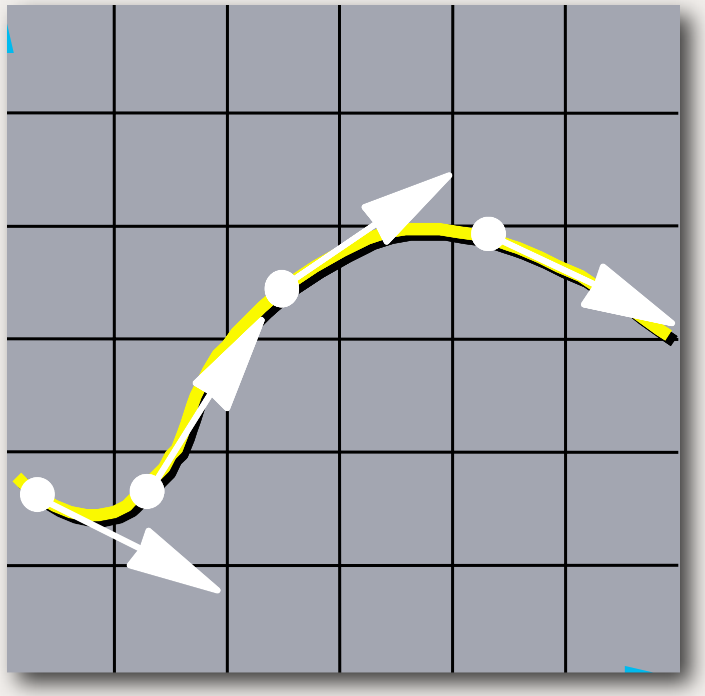
Particle Systems

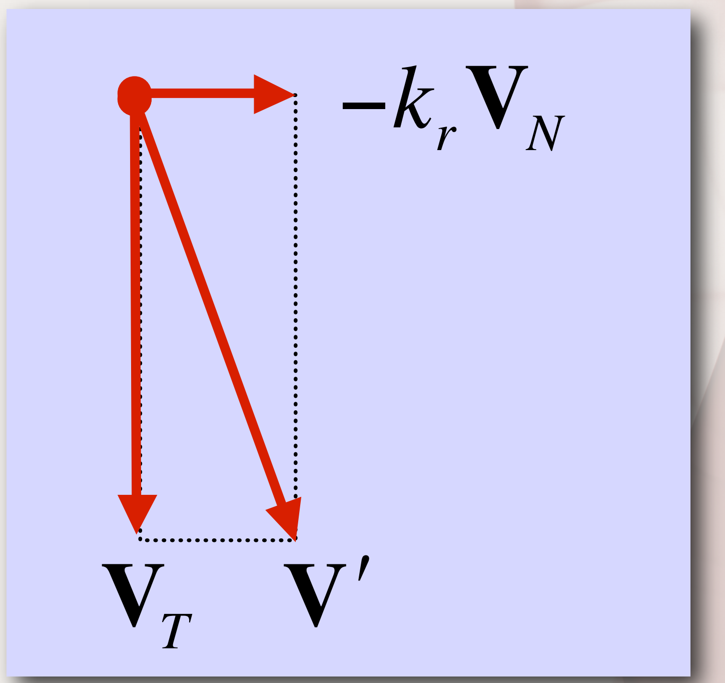
(and fun things we can do with them)

Adrien Treuille

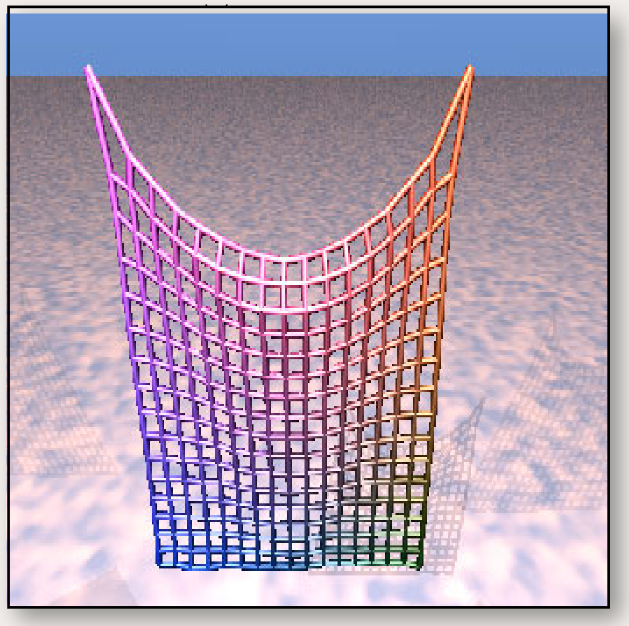
Overview



DiffEQ Review



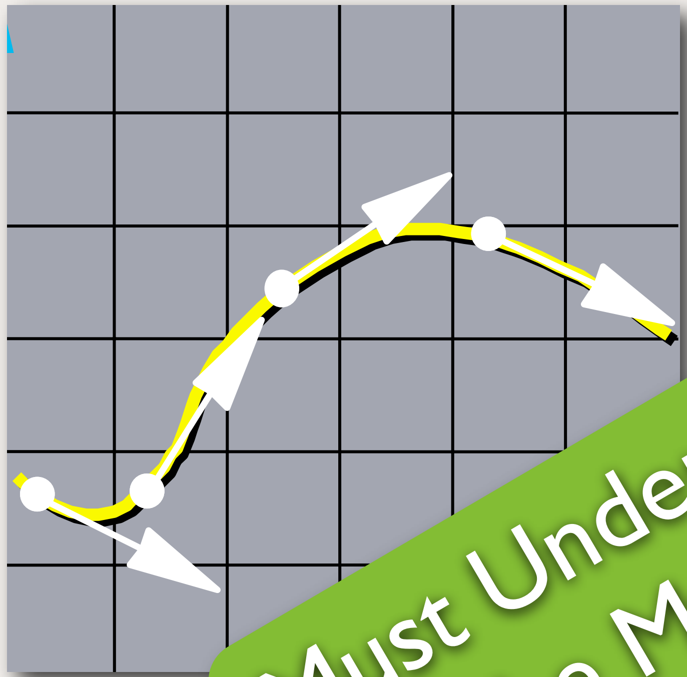
Particle Dynamics



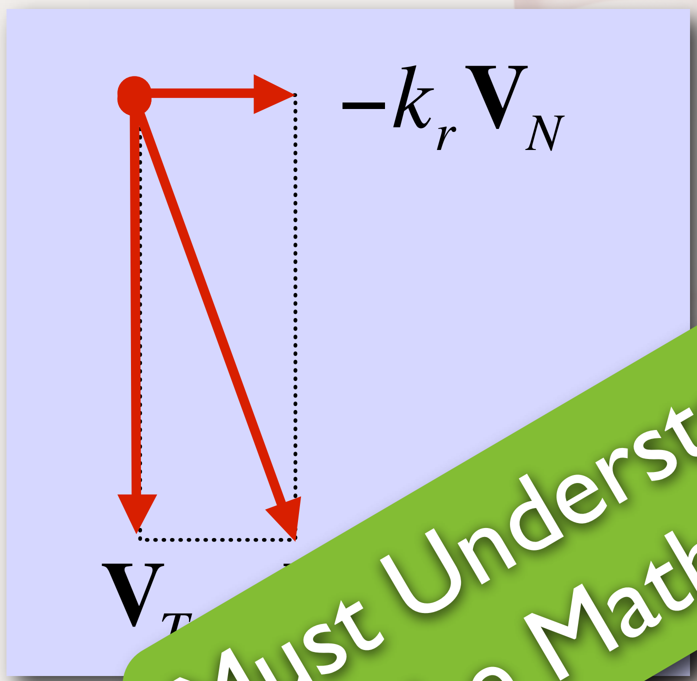
Cloth

Hair

Overview



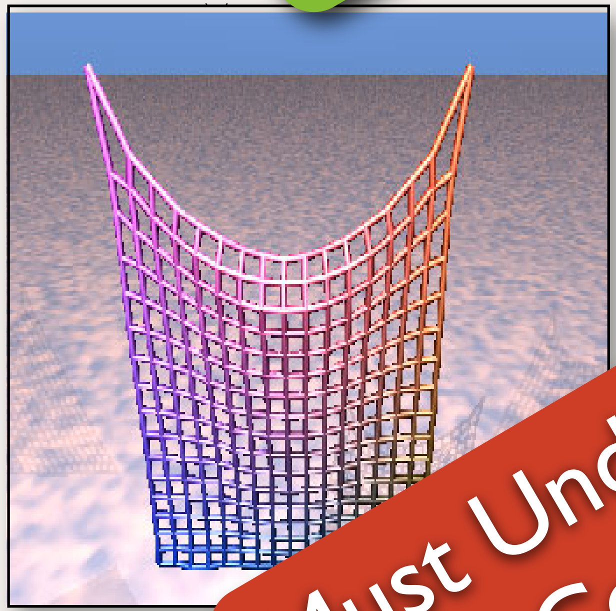
DiffE... ...ew



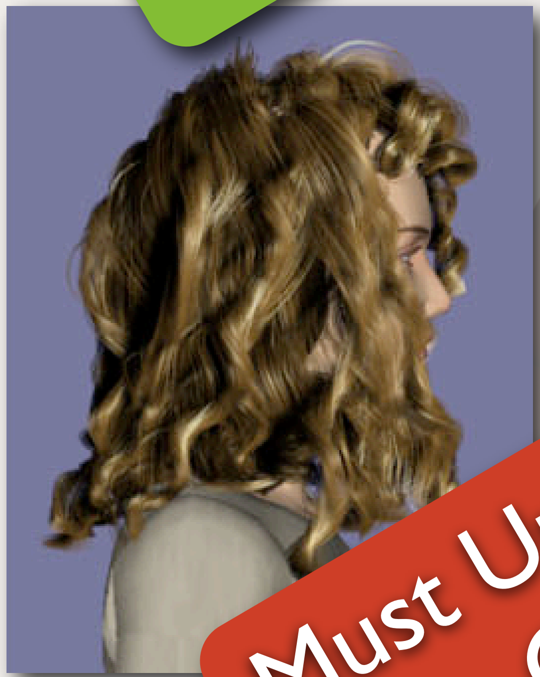
Parti... ...amics

Must Understand the Math!

Must Understand the Math!



Cl...

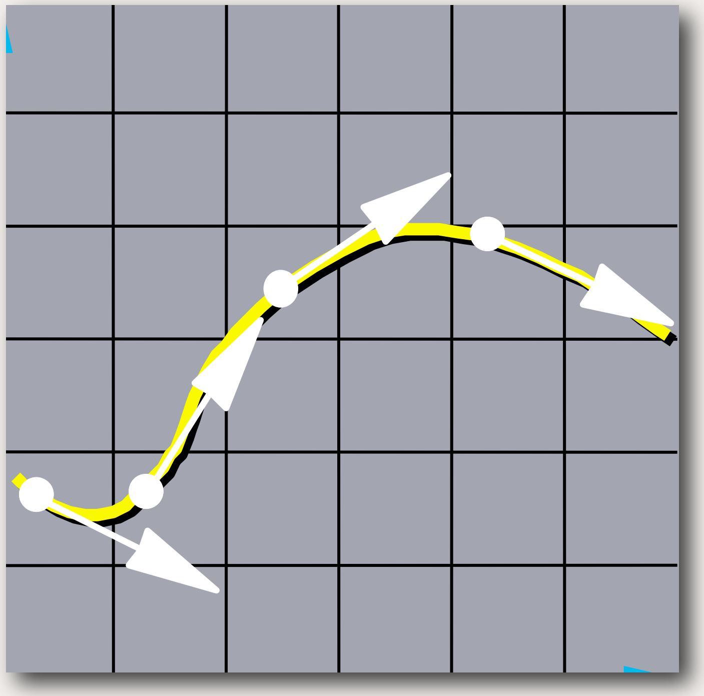


Ha...

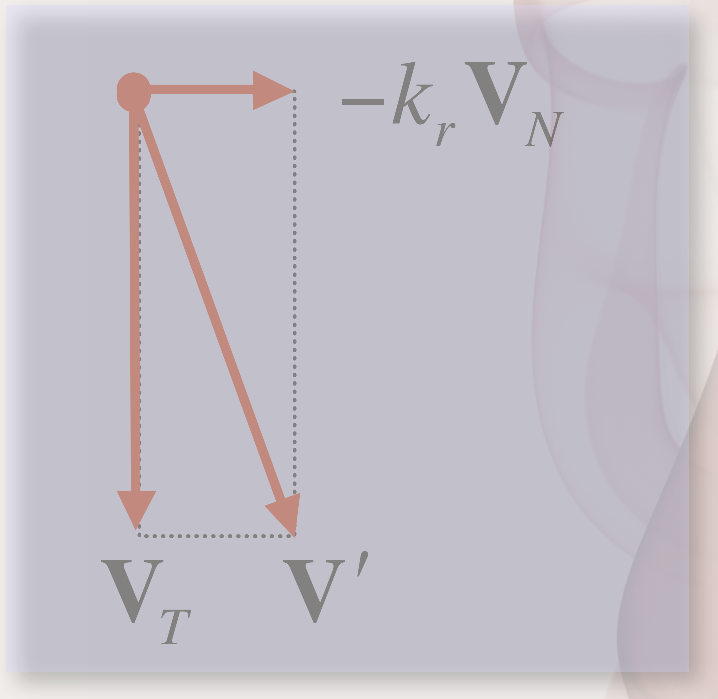
Must Understand the Concepts!

Must Understand the Concepts!

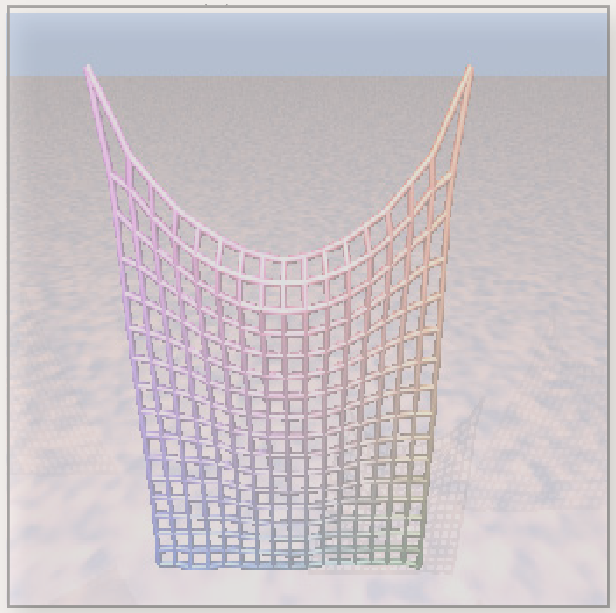
Overview



DiffEQ Review



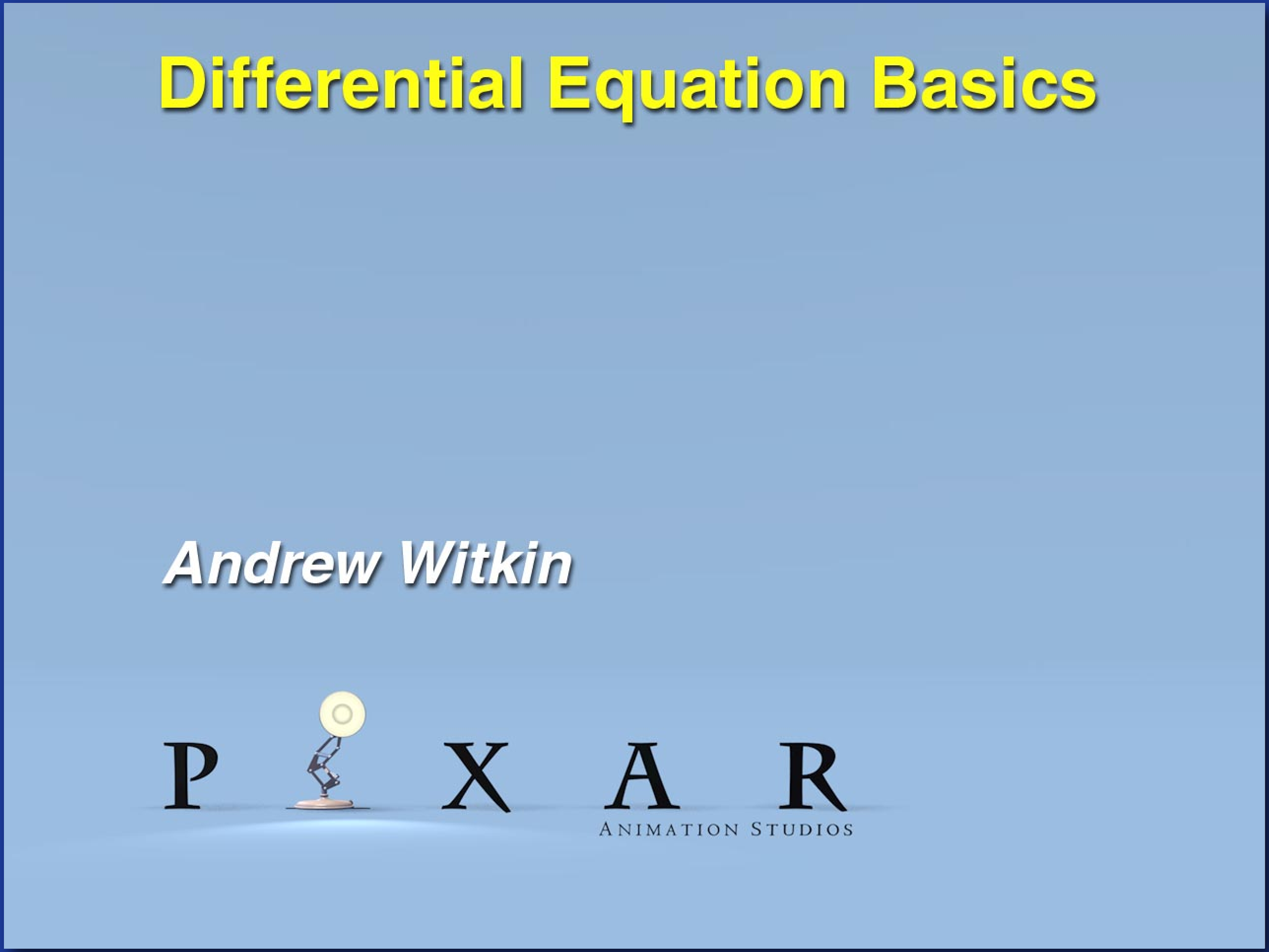
Particle Dynamics



Cloth

Hair

DiffEQ Integration

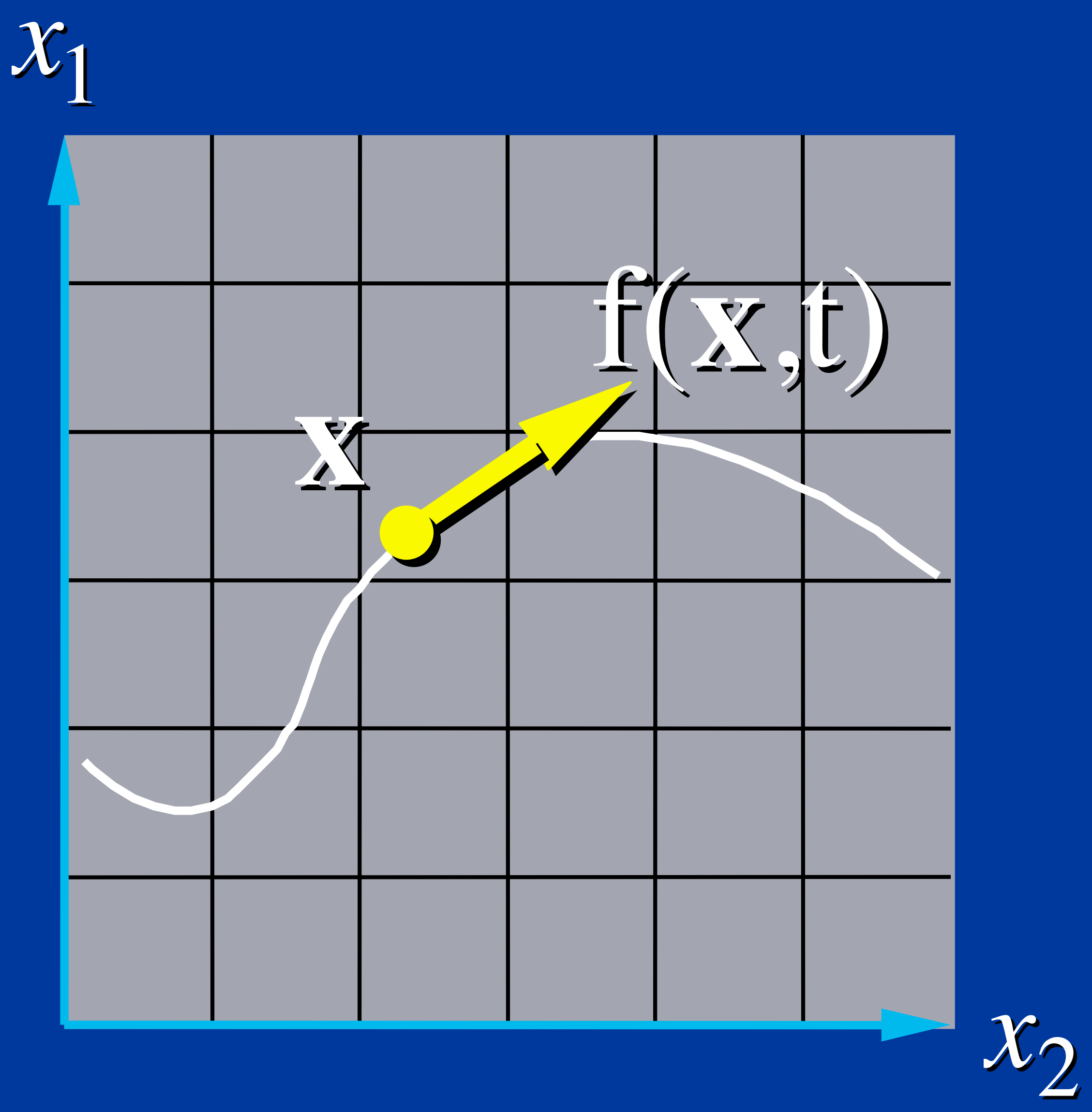


Differential Equation Basics

Andrew Witkin

PIXAR
ANIMATION STUDIOS

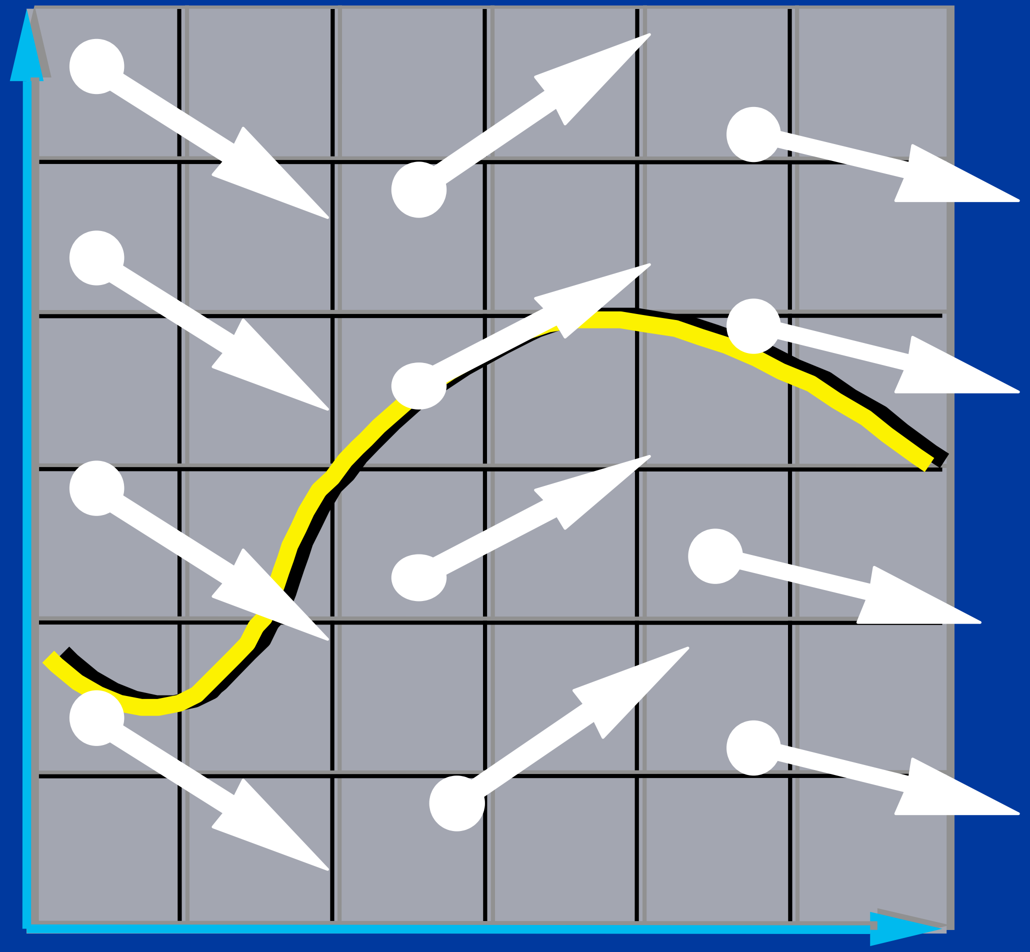
A Canonical Differential Equation



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

- **x(*t*): a moving point.**
- **f(x,*t*): x's velocity.**

Vector Field



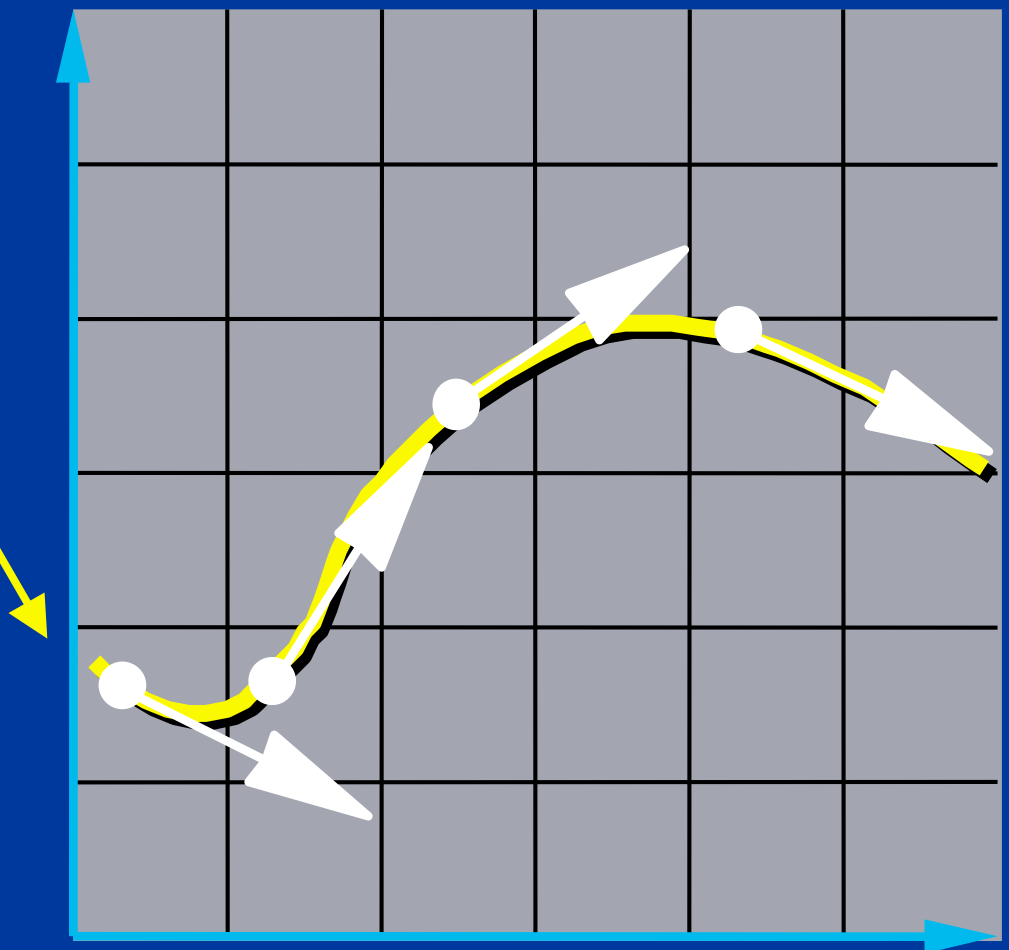
The differential equation

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

defines a vector field over x.

Integral Curves

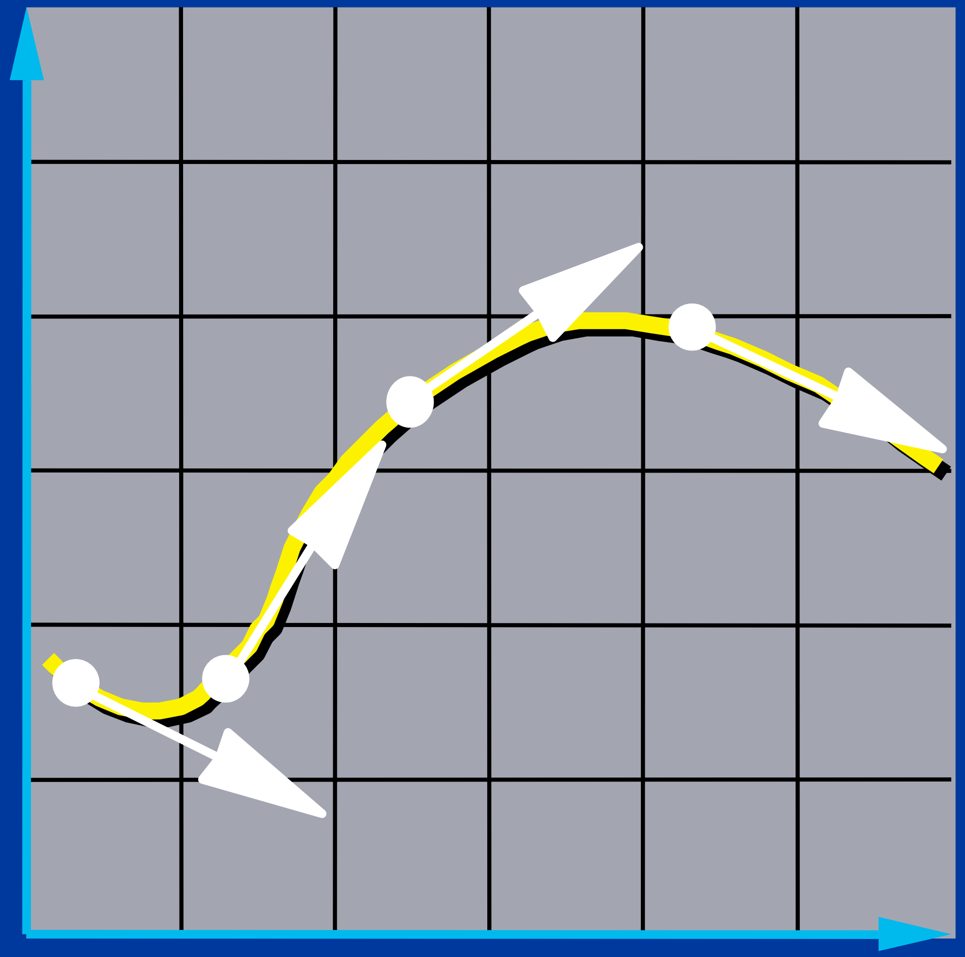
Start Here



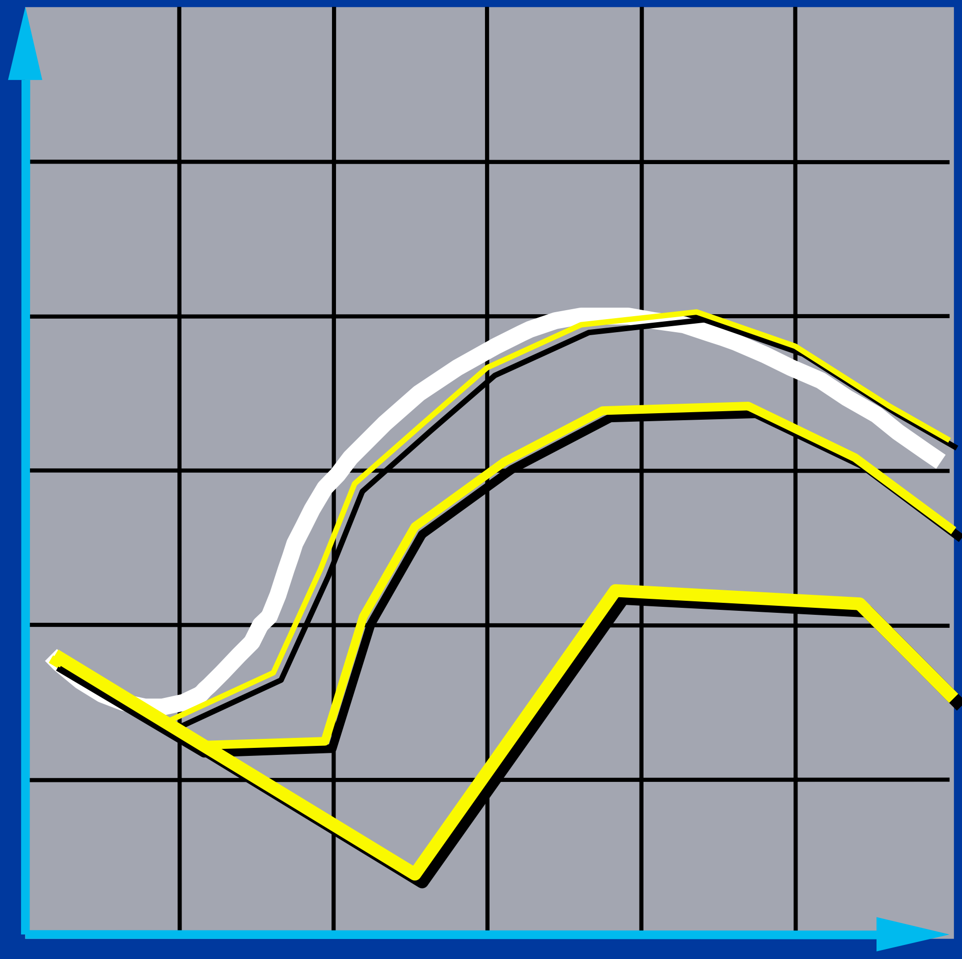
Pick any starting point, and follow the vectors.

Initial Value Problems

Given the starting point, follow the integral curve.



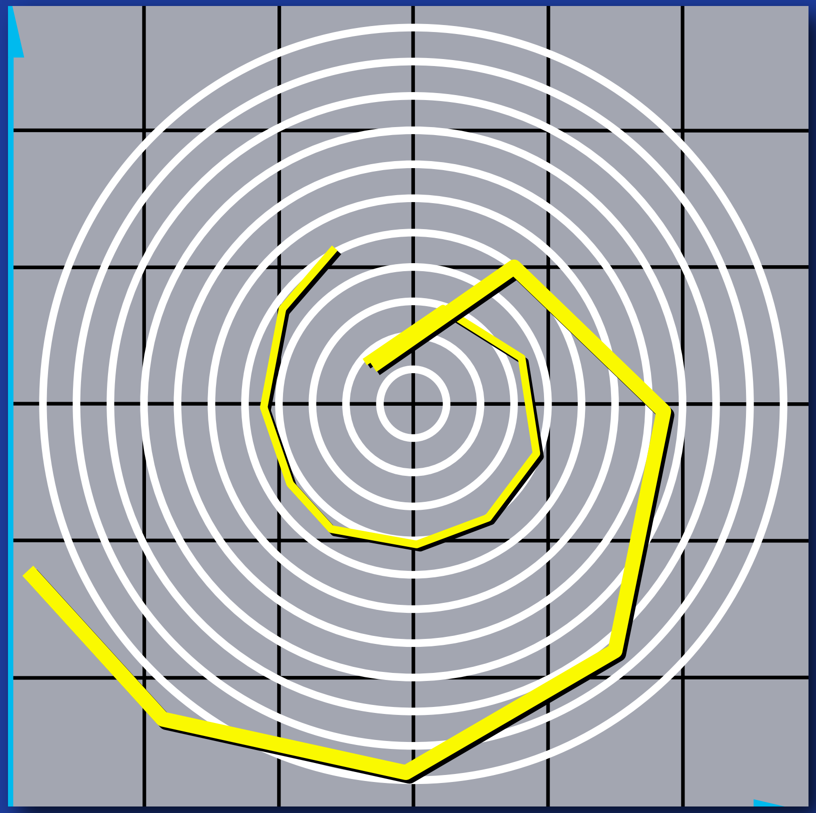
Euler's Method

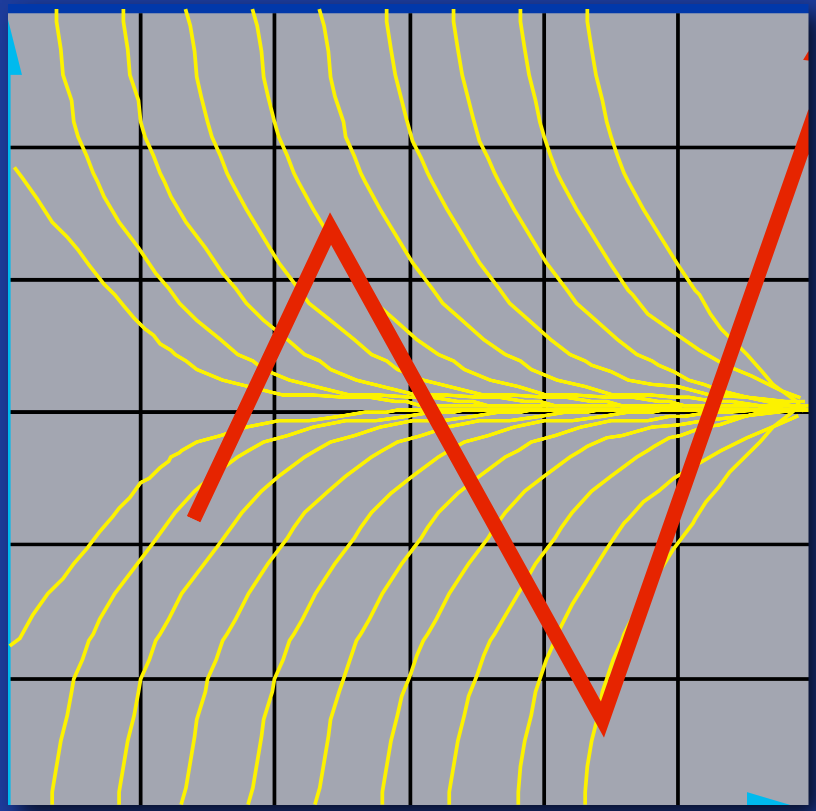


- Simplest numerical solution method
- Discrete time steps
- Bigger steps, bigger errors.

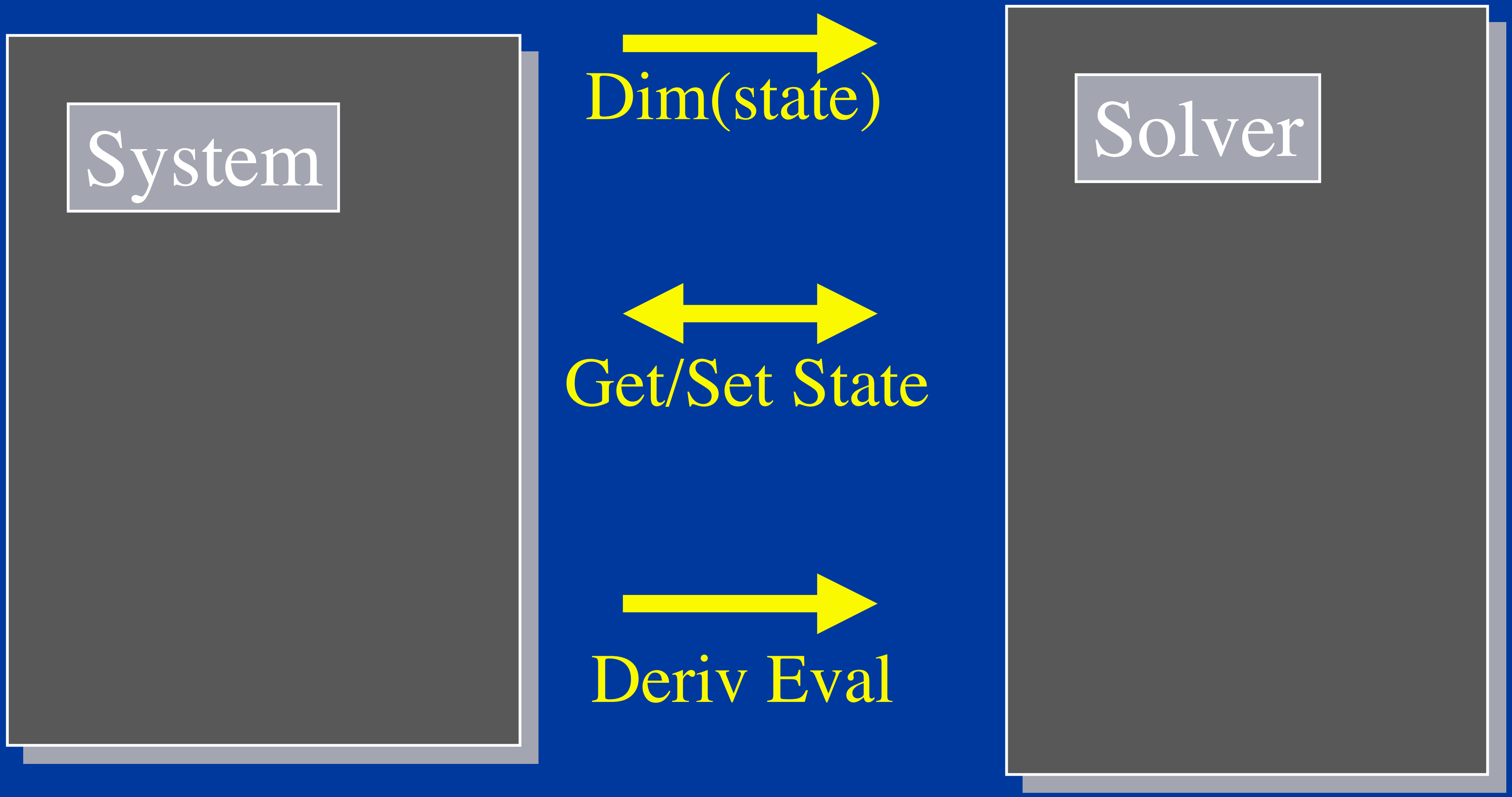
$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t\,\mathbf{f}(\mathbf{x},t)$$

Two Problems


Accuracy


Instability

Solver Interface

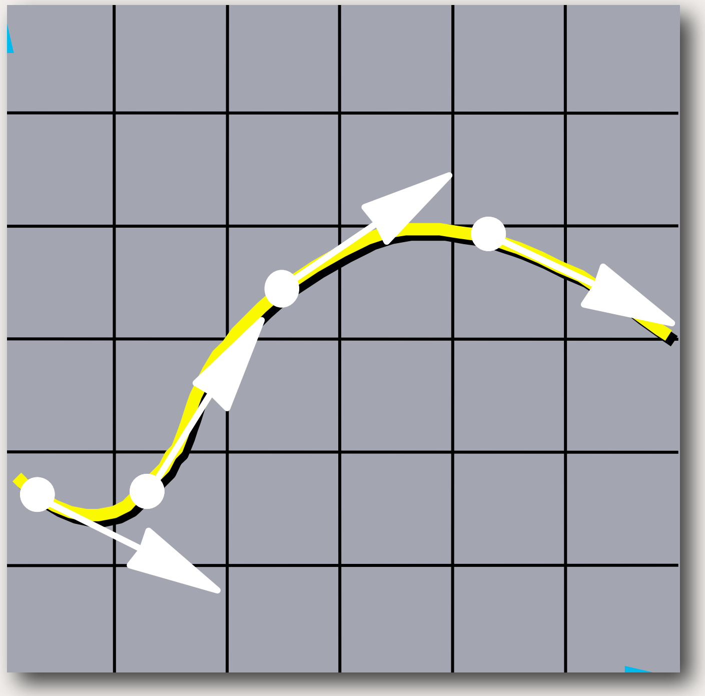


A Code Fragment

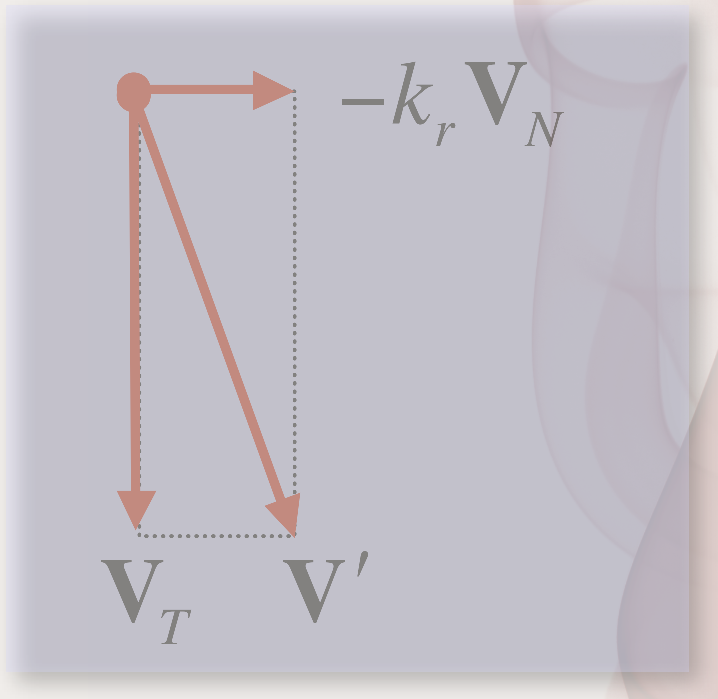
```
void eulerStep(Sys sys, float h) {
   float t = getTime(sys);
   vector<float> x0, deltaX;

   t = getTime(sys);
   x0 = getState(sys);
   deltaX = derivEval(sys,x0, t);
   setState(sys, x0 + h*deltaX, t+h);
}
```

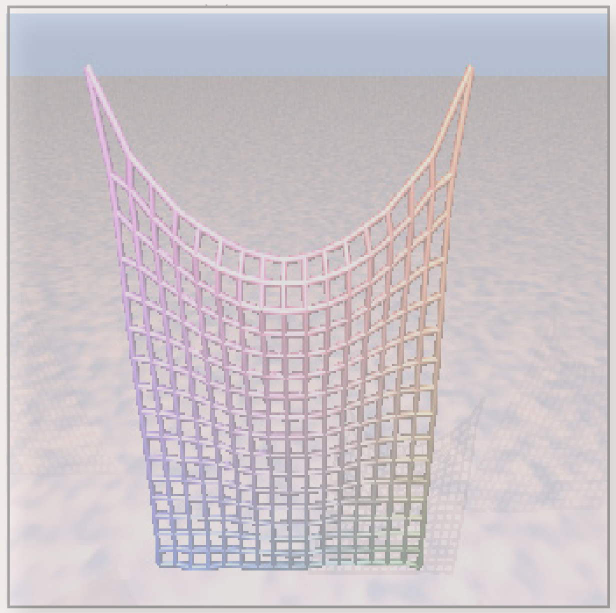
Overview



DiffEQ Review



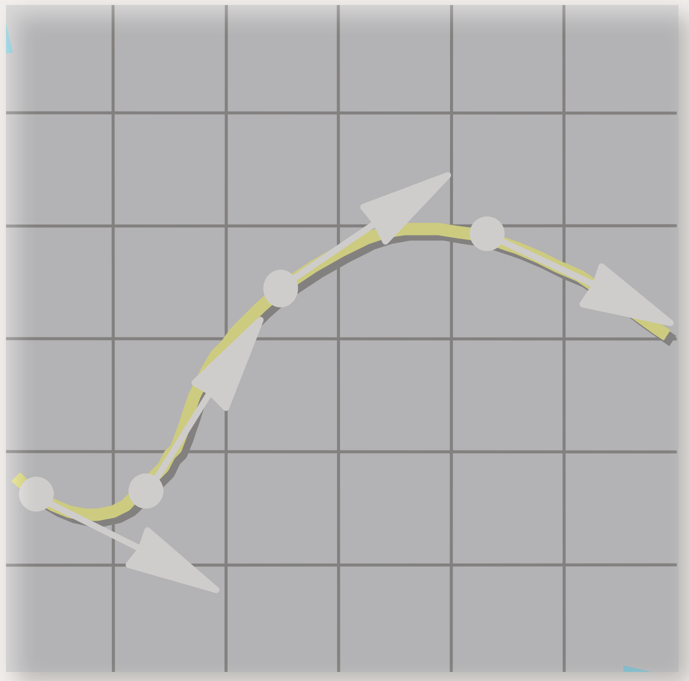
Particle Dynamics



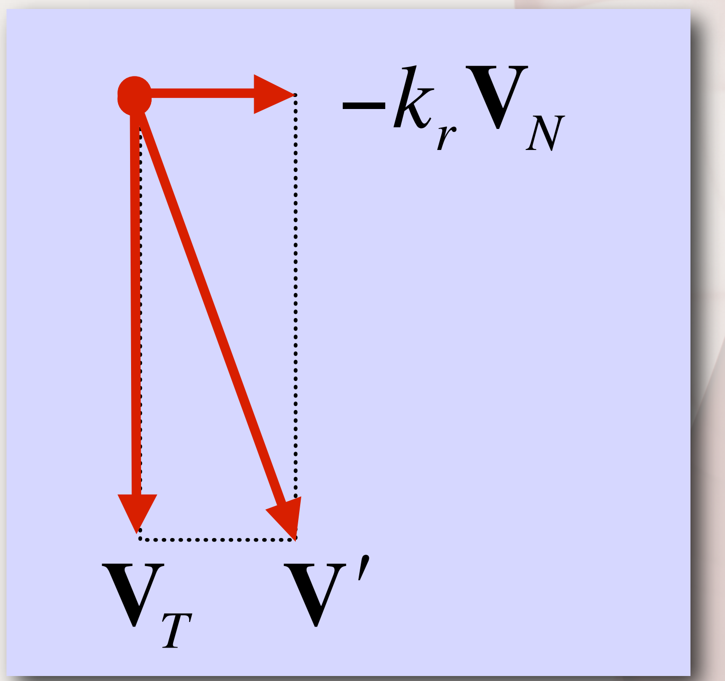
Cloth

Hair

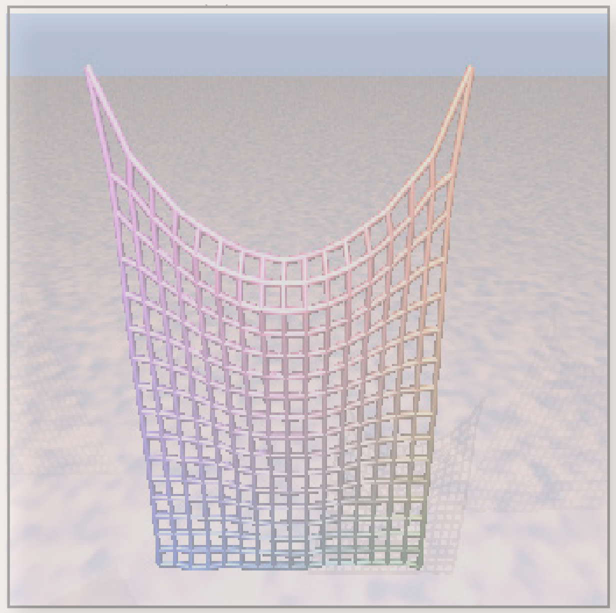
Overview



DiffEQ Review

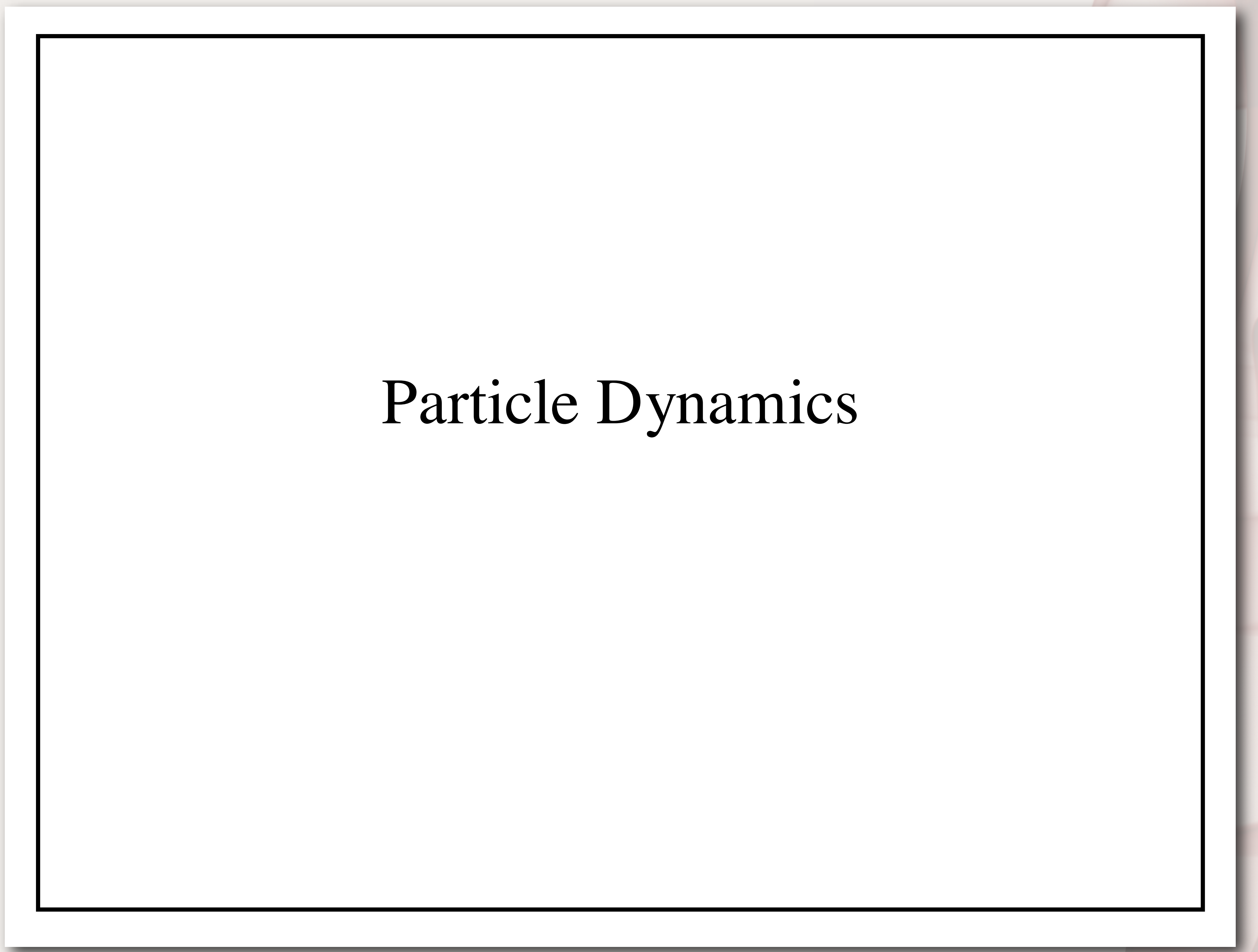


Particle Dynamics



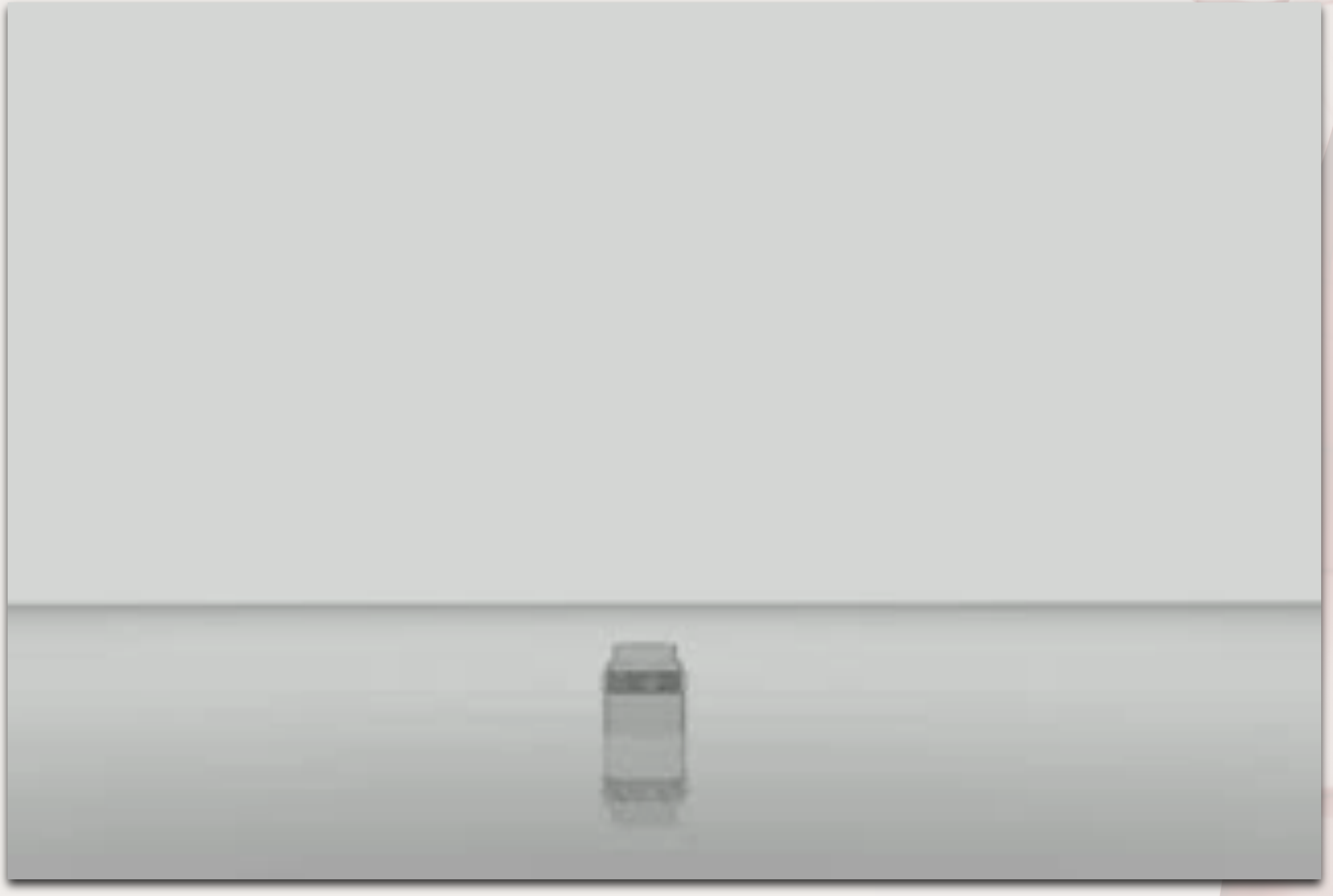
Cloth

Hair



Particle Dynamics

from Zoran Popović


http://www.youtube.com/watch?v=N8xNlp00kss

Overview

- One lousy particle
- Particle systems
- Forces: gravity, springs
- Implementation

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

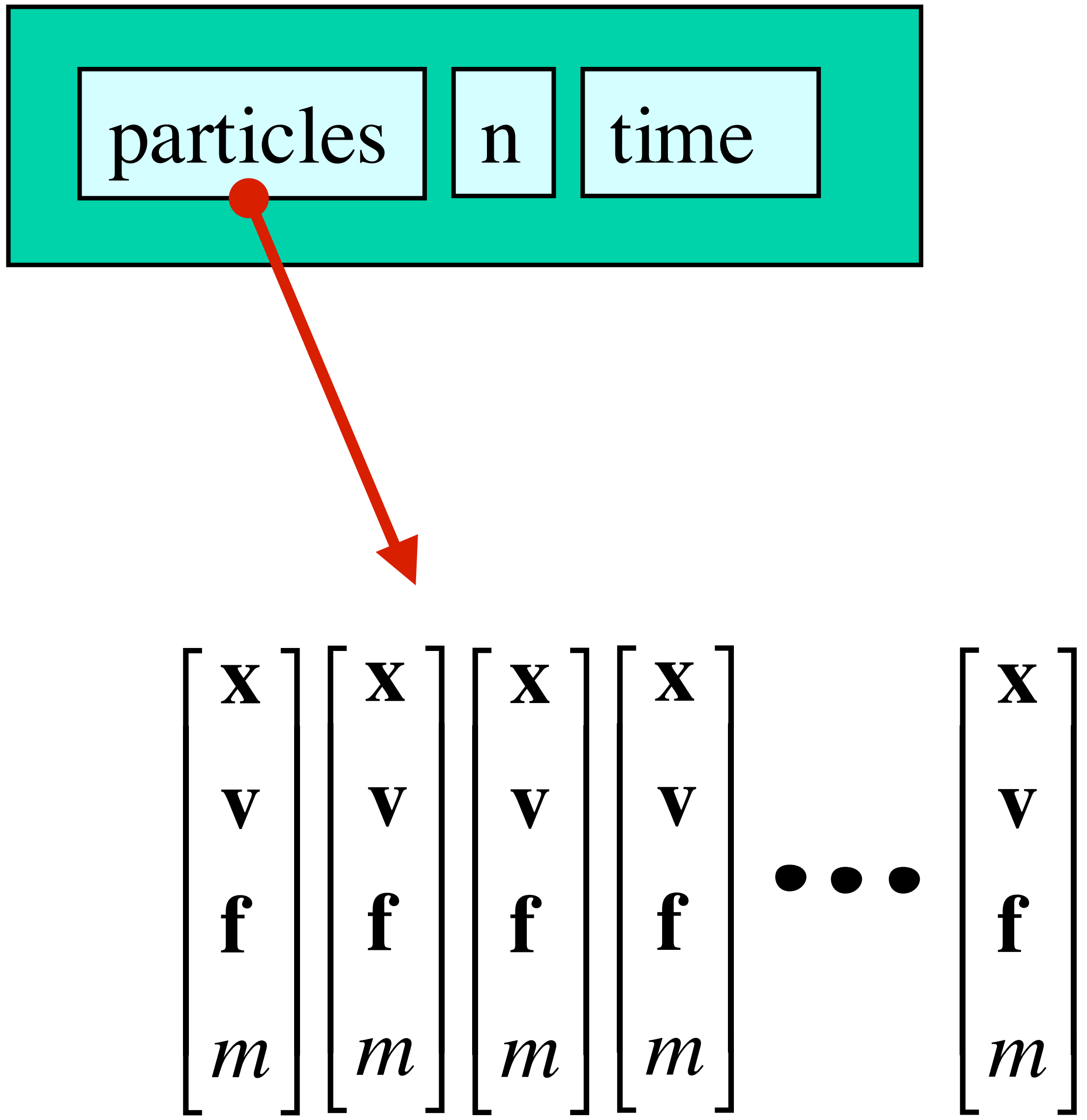
Position in phase space

$$\begin{tikzcd} \left[\mathbf{x}\right. & \text{position} \arrow[l] \\ \mathbf{v} & \text{velocity} \arrow[l] \\ \mathbf{f} & \text{force accumulator} \arrow[l] \\ \left. m\right] & \text{mass} \arrow[l] \end{tikzcd}$$

Solver interface

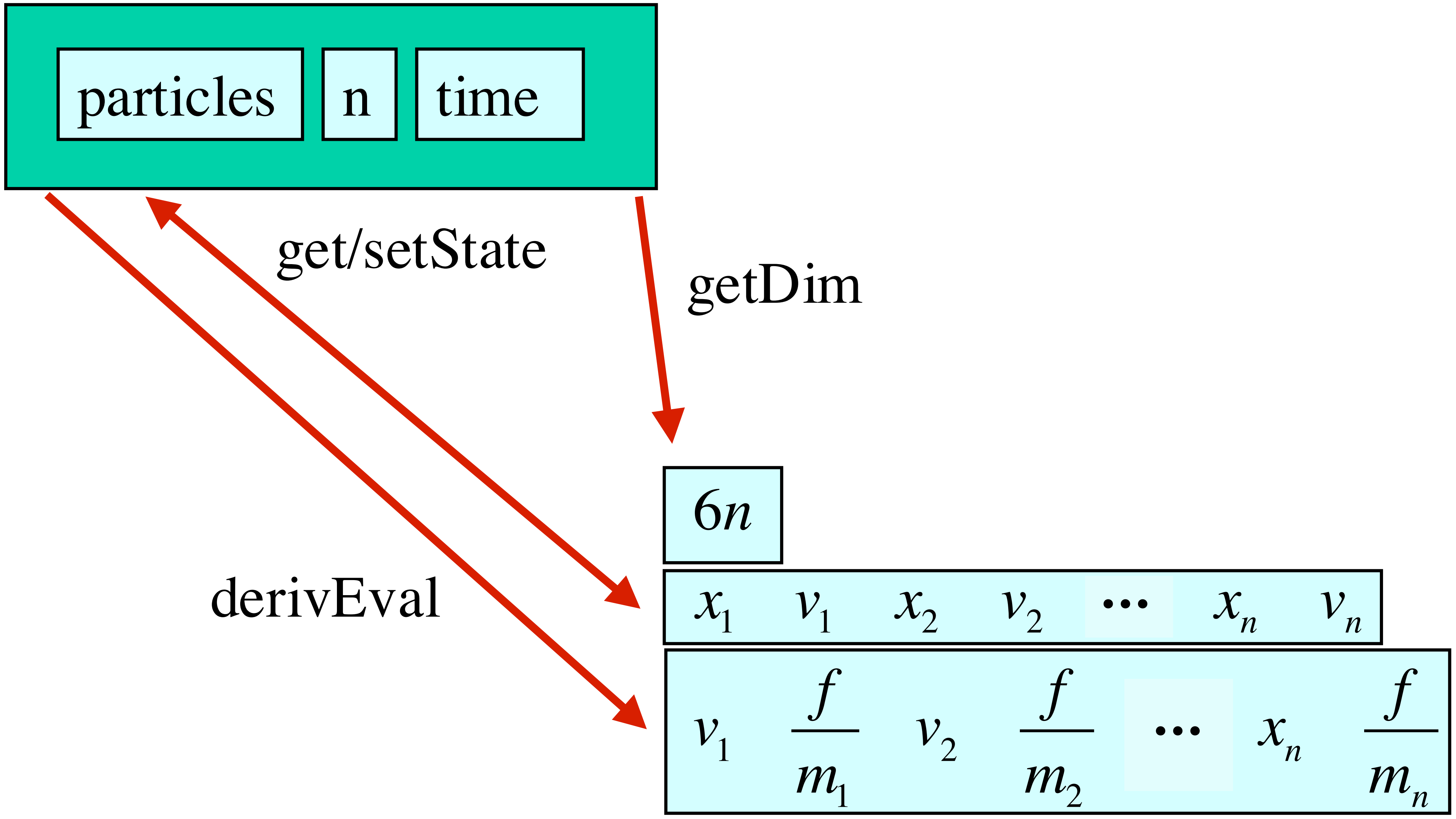
$$\begin{tikzcd}
 & \begin{bmatrix}6\end{bmatrix} \\
\begin{bmatrix}\mathbf{x}\\ \mathbf{v}\\ \mathbf{f}\\ m\end{bmatrix} \arrow[ur, "\text{getDim}"] \arrow[r, shift left, "\text{getState}"] \arrow[dr, "\text{derivEval}"'] & \begin{bmatrix}\mathbf{x}\\ \mathbf{v}\end{bmatrix} \arrow[l, shift left, "\text{setState}"] \\
 & \begin{bmatrix}\mathbf{v}\\ \mathbf{f}/m\end{bmatrix}
\end{tikzcd}$$

Particle systems



$$\begin{bmatrix}\mathbf{x}\\ \mathbf{v}\\ \mathbf{f}\\ m\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{v}\\ \mathbf{f}\\ m\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{v}\\ \mathbf{f}\\ m\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{v}\\ \mathbf{f}\\ m\end{bmatrix}\cdots\begin{bmatrix}\mathbf{x}\\ \mathbf{v}\\ \mathbf{f}\\ m\end{bmatrix}$$

Solver interface


particles
n
time
get/setState
getDim
derivEval
$6n$
$x_1 \quad v_1 \quad x_2 \quad v_2 \quad \cdots \quad x_n \quad v_n$
$v_1 \quad \frac{f}{m_1} \quad v_2 \quad \frac{f}{m_2} \quad \cdots \quad x_n \quad \frac{f}{m_n}$

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)

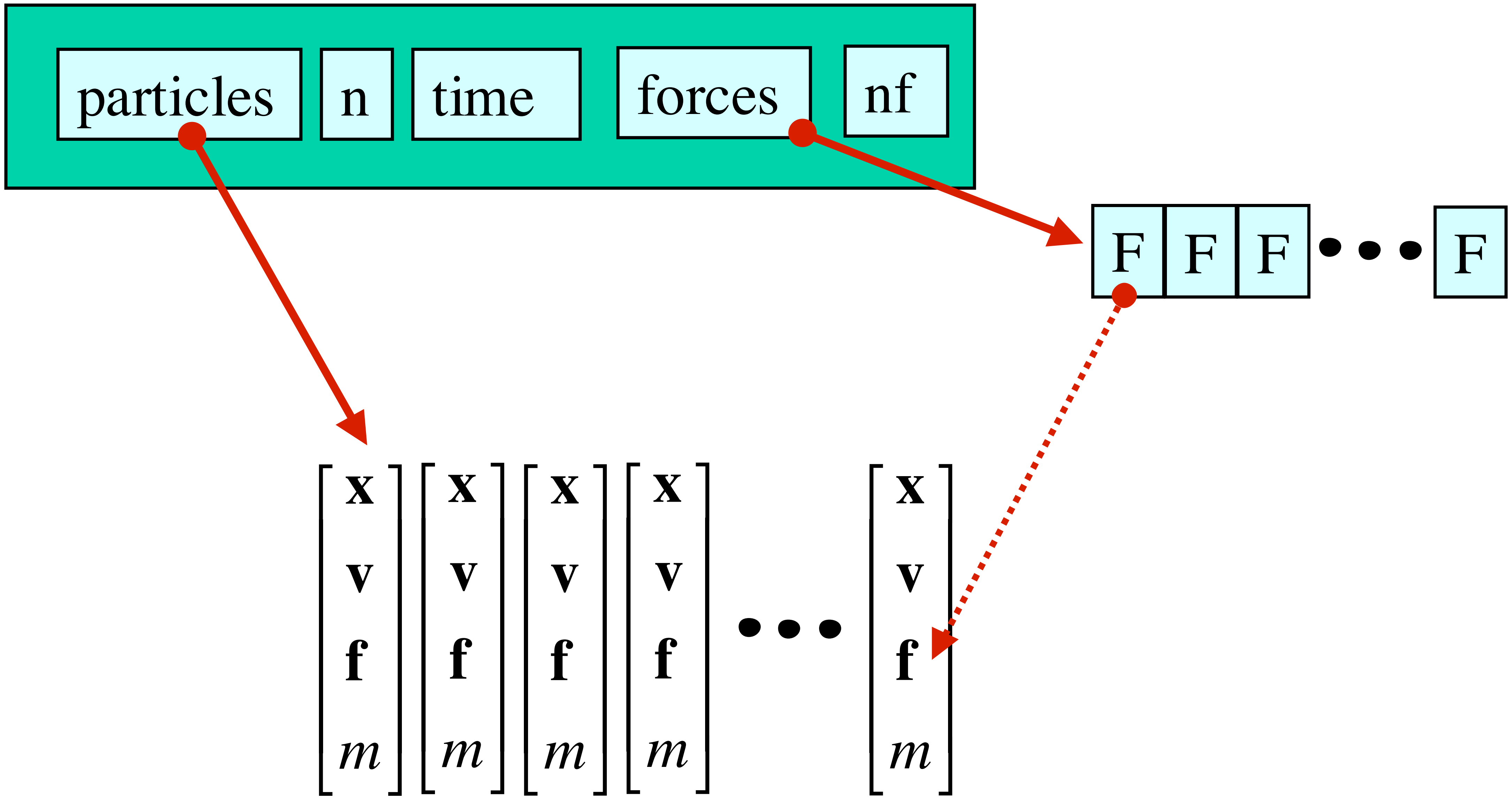
Force structures

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
```

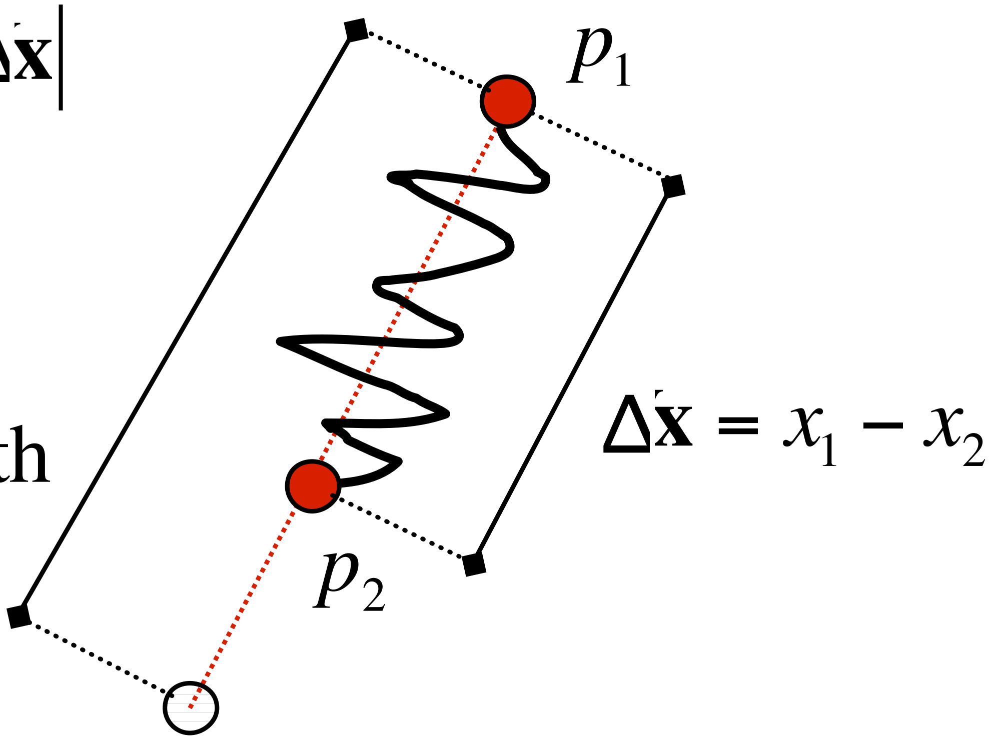
Damped spring

Force law:

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

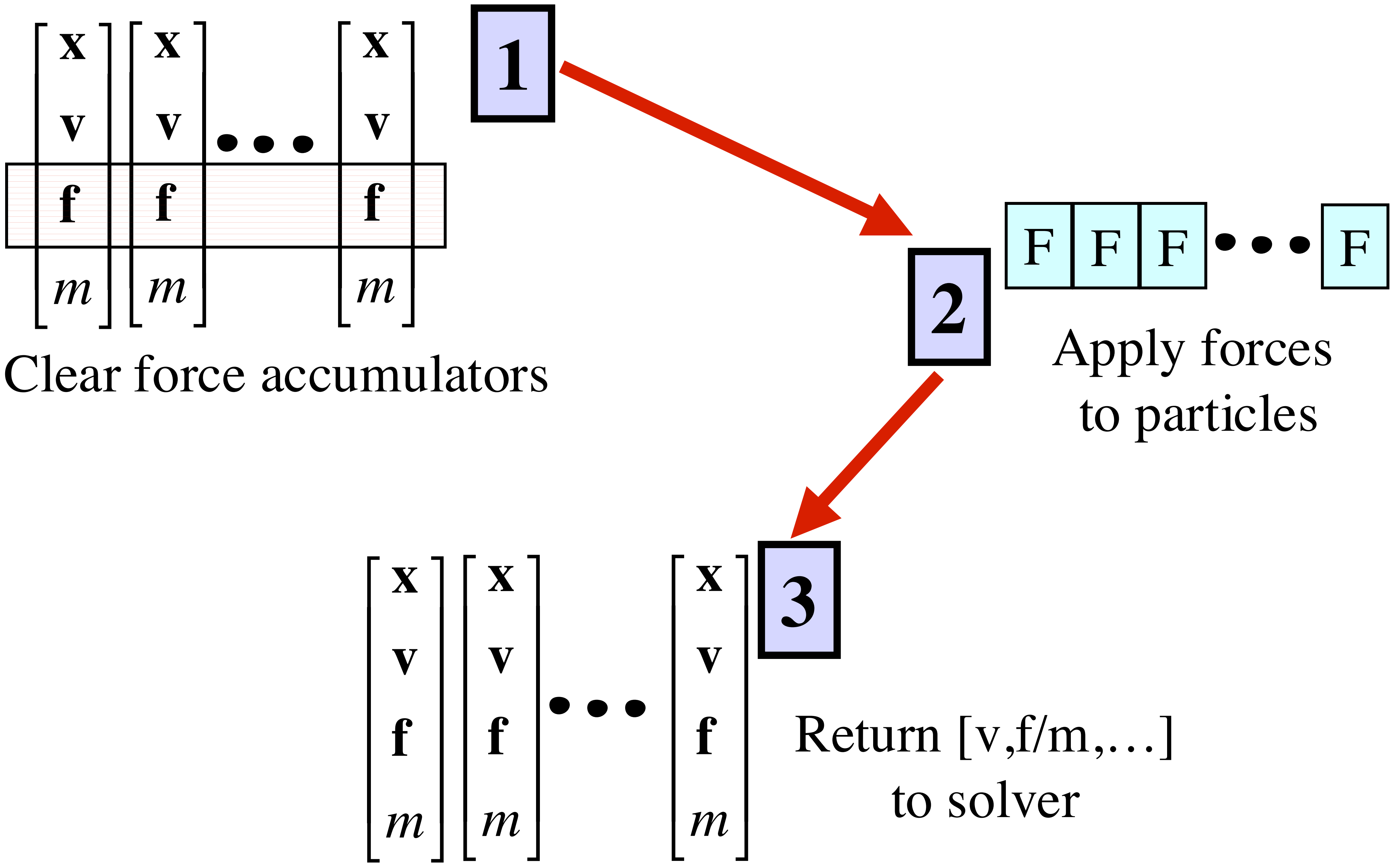
$$\mathbf{f}_2 = -\mathbf{f}_1$$

$\mathbf{r}$ = rest length



http://video.google.com/videoplay?docid=-2182452945242275492&ei=ZSjSSbvGKo2grwKm-LHPAQ&q=particle+system+spring&hl=en&client=safari

derivEval Loop

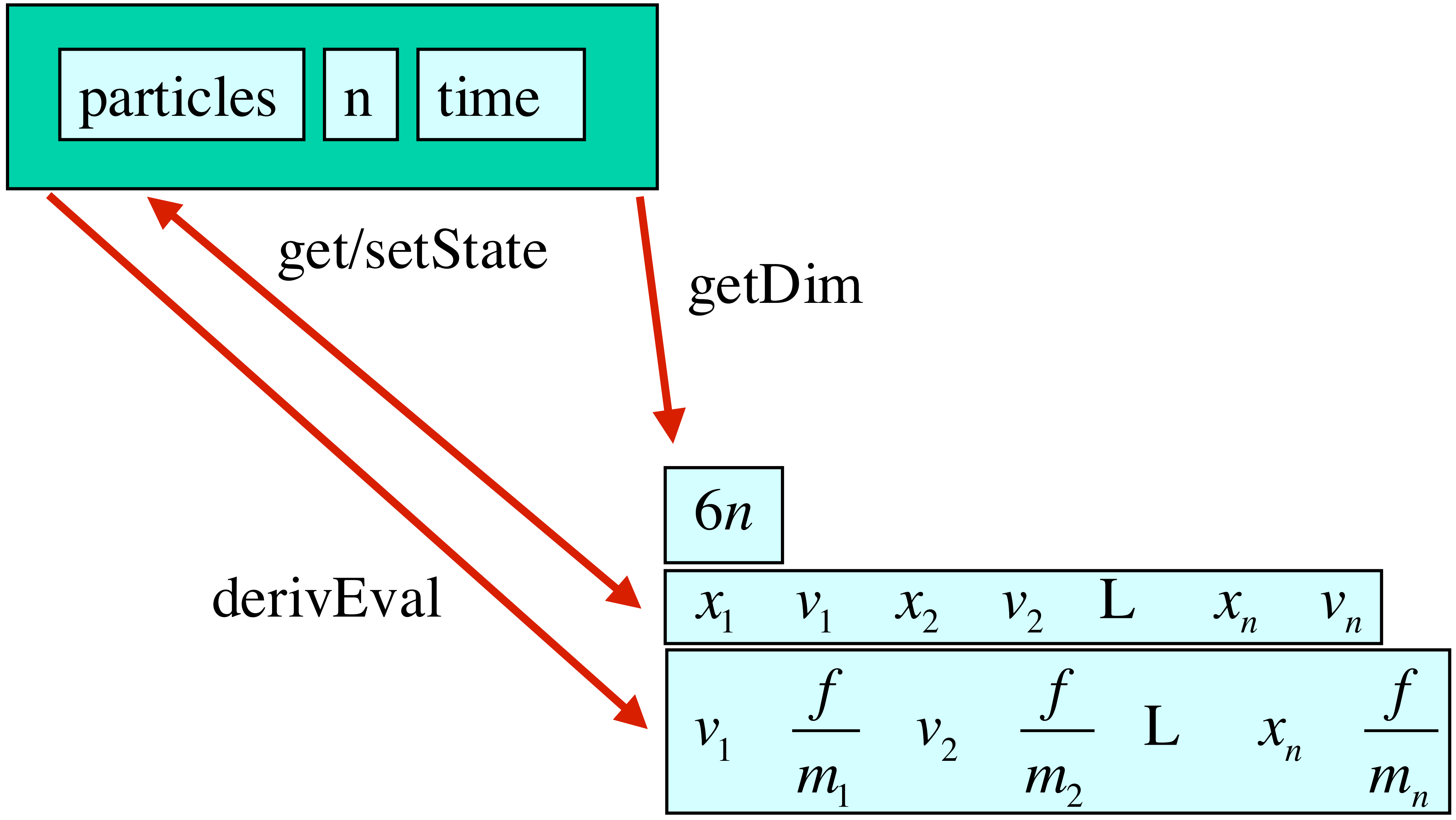


Clear force accumulators

Apply forces to particles

Return [v,f/m,…] to solver

Solver interface

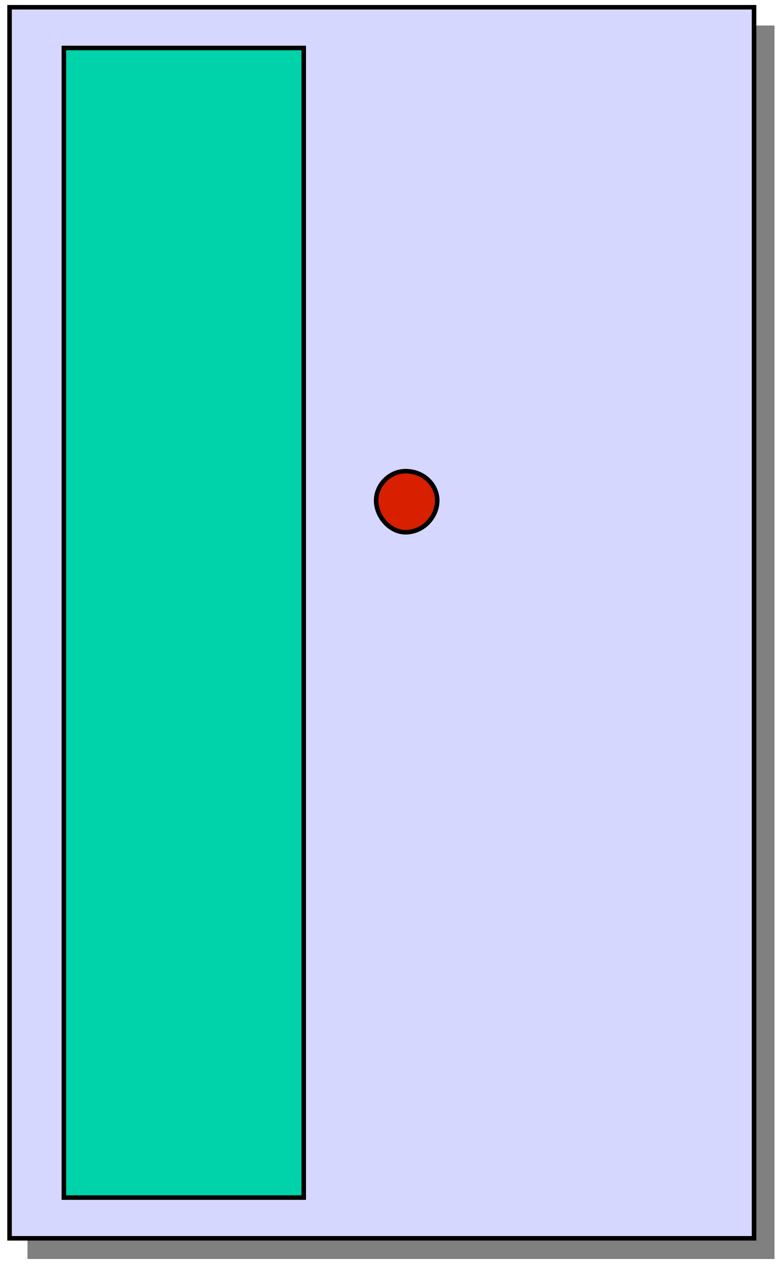

particles
n
time
get/setState
getDim
derivEval
$6n$
$x_1 \quad v_1 \quad x_2 \quad v_2 \quad \cdots \quad x_n \quad v_n$
$v_1 \quad \frac{f}{m_1} \quad v_2 \quad \frac{f}{m_2} \quad \cdots \quad x_n \quad \frac{f}{m_n}$

Differential equation solver

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

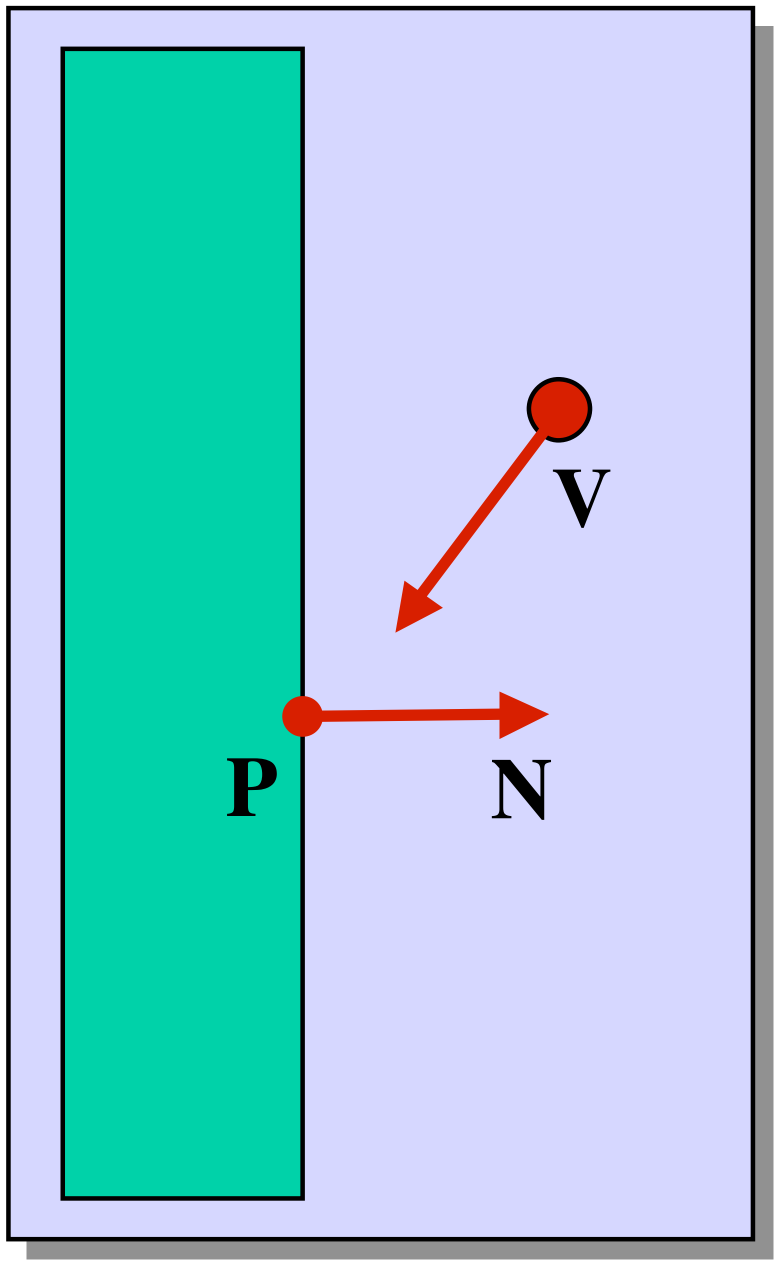
Euler method:
$$\begin{bmatrix} x_1^{i+1} \\ v_1^{i+1} \\ \dots \\ x_n^{i+1} \\ v_n^{i+1} \end{bmatrix} = \begin{bmatrix} x_1^{i} \\ v_1^{i} \\ \dots \\ x_n^{i} \\ v_n^{i} \end{bmatrix} + \nabla t \begin{bmatrix} v_1^{i} \\ f_1^{i}/m_1 \\ \dots \\ v_n^{i} \\ f_n^{i}/m_n \end{bmatrix}$$

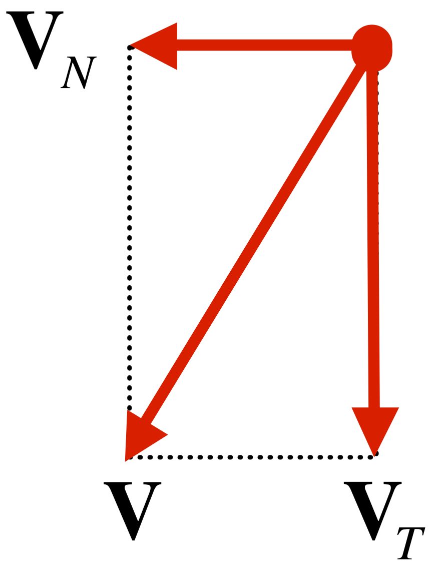
Bouncing off the walls



- Add-on for a particle simulator
- For now, just simple point-plane collisions

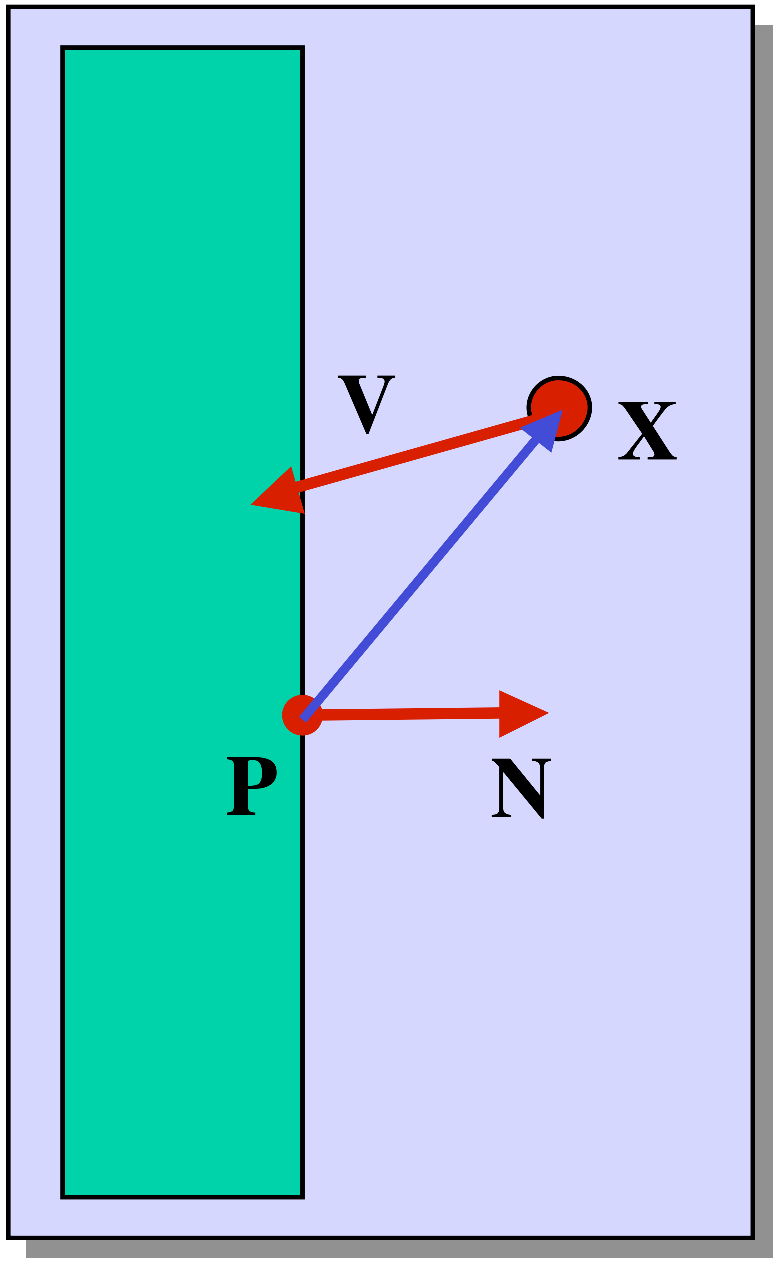
Normal and tangential components





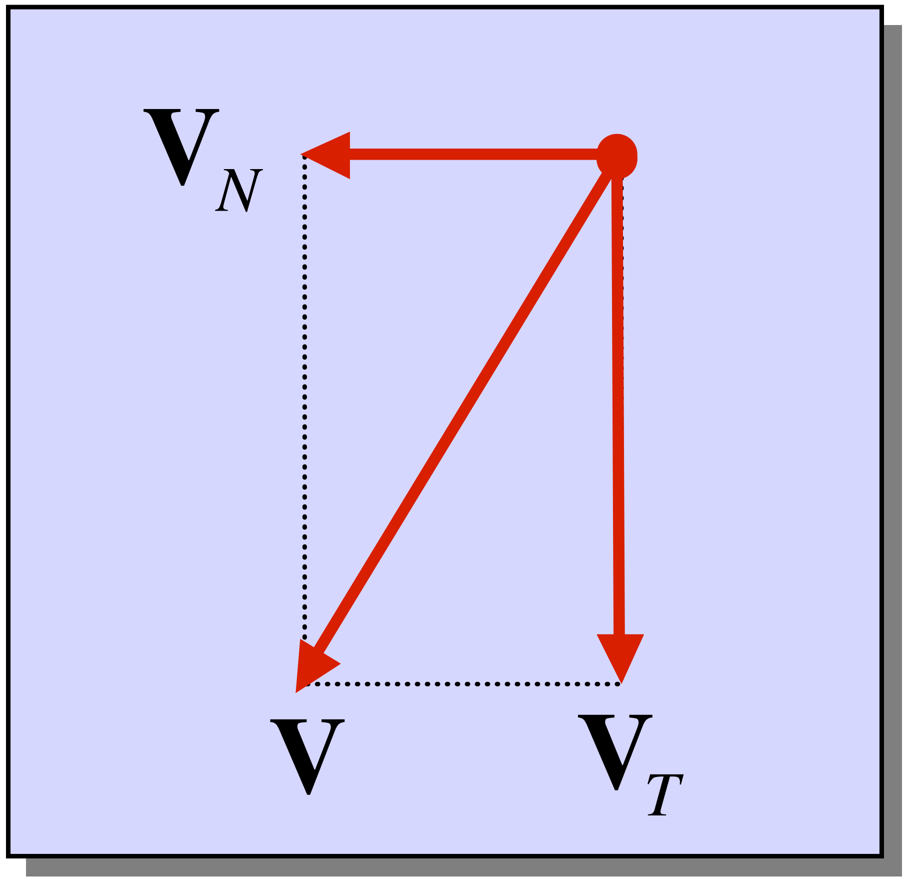
$$V_N = (N \cdot V)N$$
$$V_T = V - V_N$$

Collision Detection

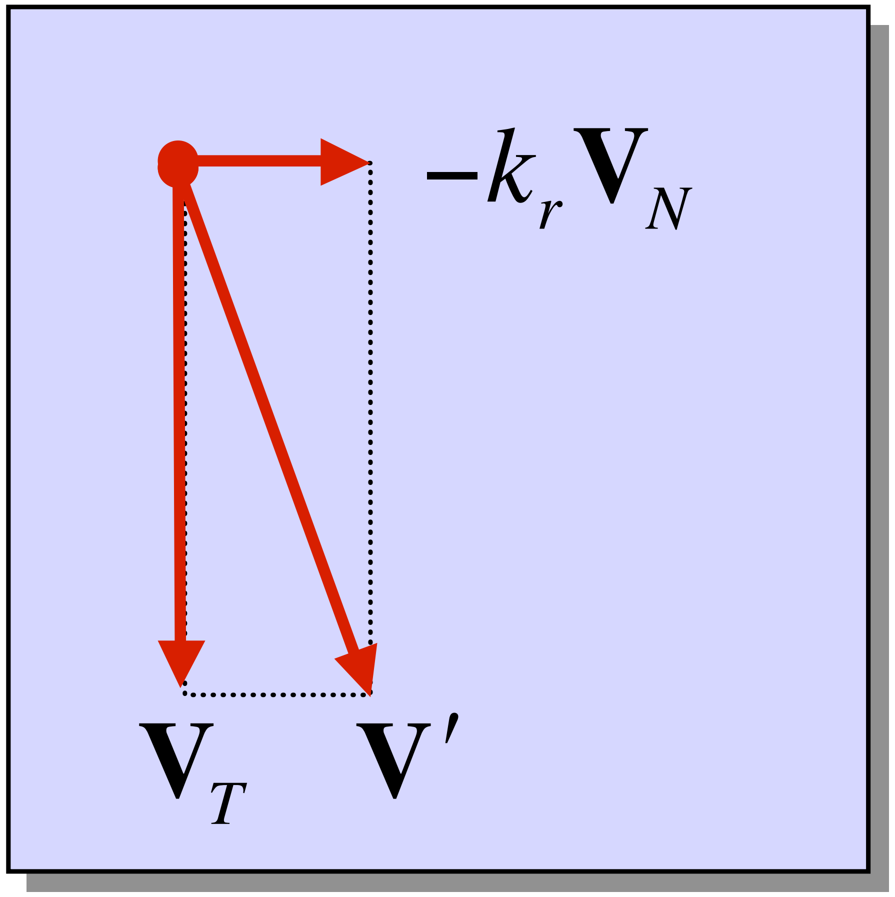


$(\mathbf{X}-\mathbf{P})\cdot\mathbf{N} < \varepsilon$	Within e of the wall
$\mathbf{N}\cdot\mathbf{V} < 0$	Heading in

Collision Response



before



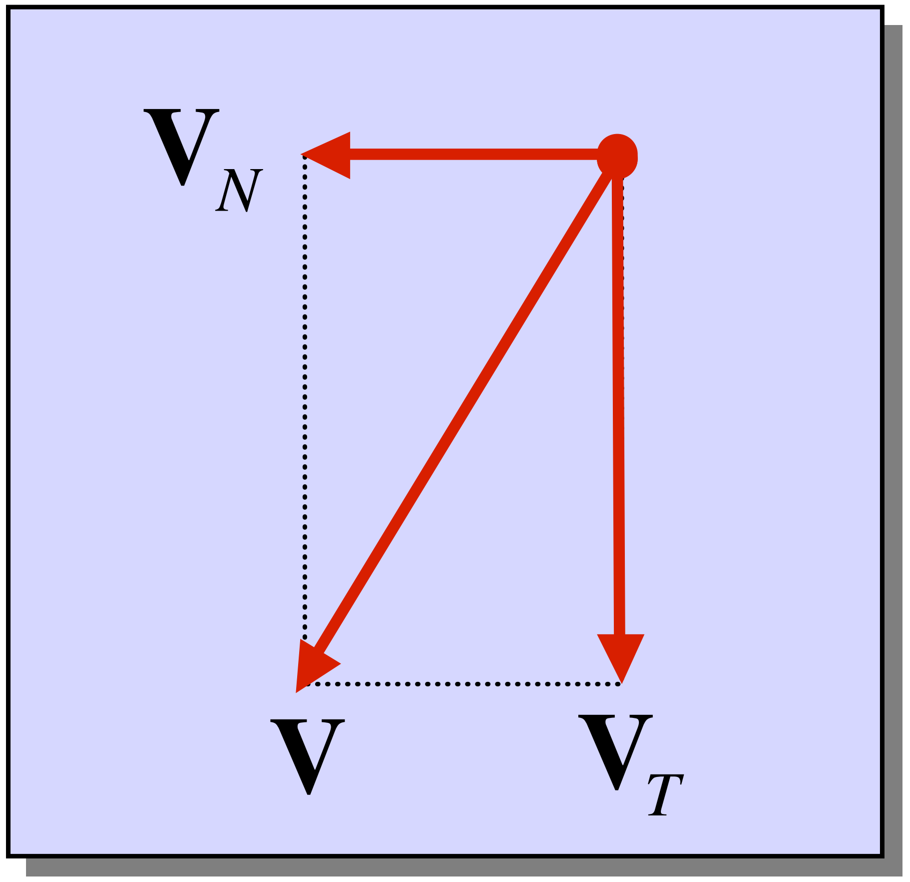
after

$$\mathbf{V}' = \mathbf{V}_T - k_r\mathbf{V}_N$$

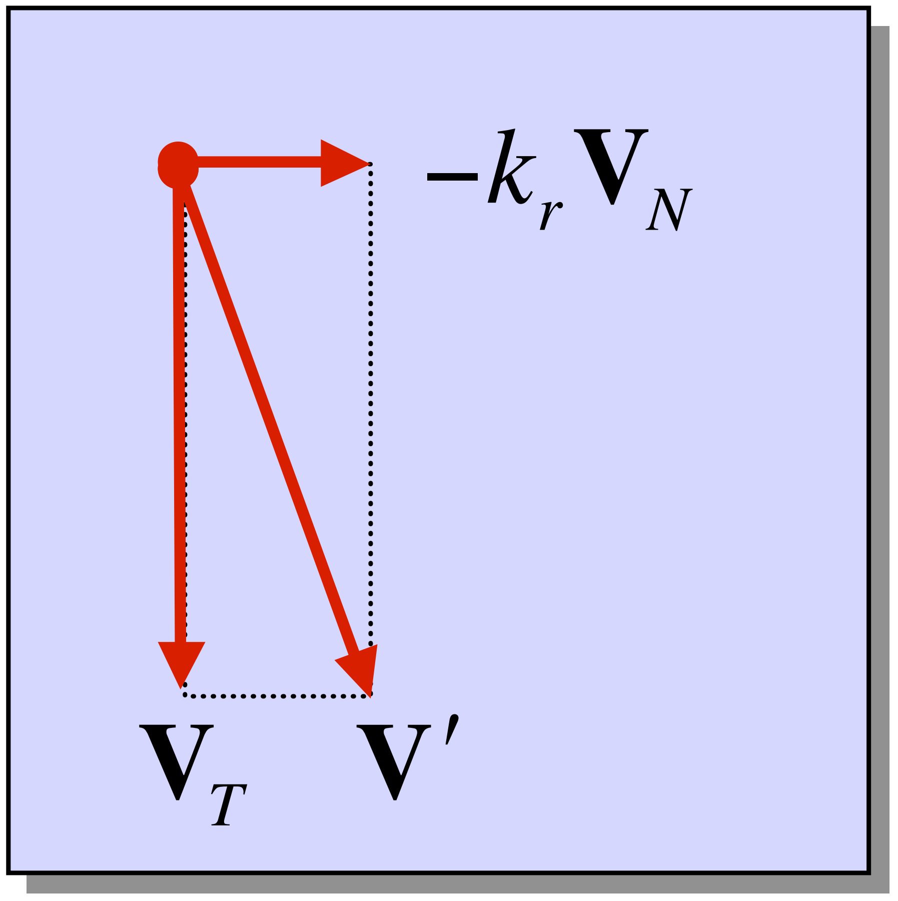
Collision Response

$$\mathbf{V}' = \mathbf{V}_T - k_r \mathbf{V}_N$$

Collision Response



before



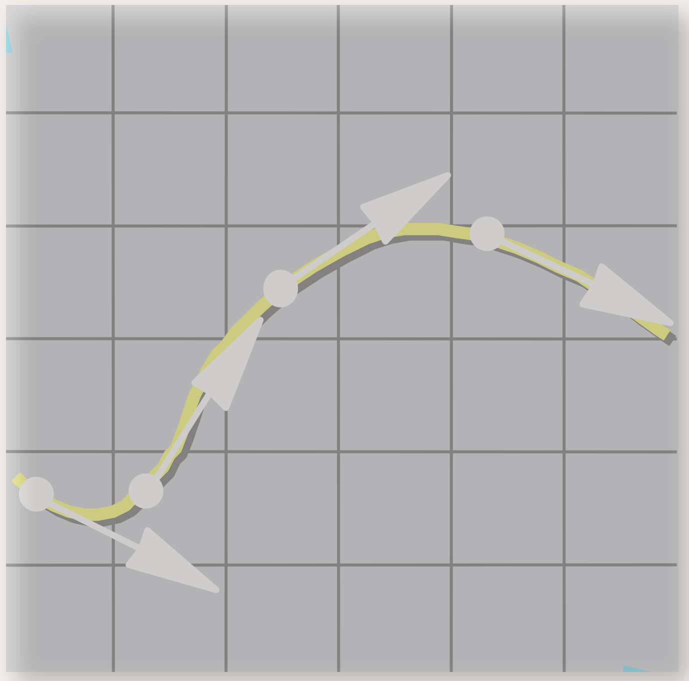
after

$$\mathbf{V}' = \mathbf{V}_T - k_r\mathbf{V}_N$$

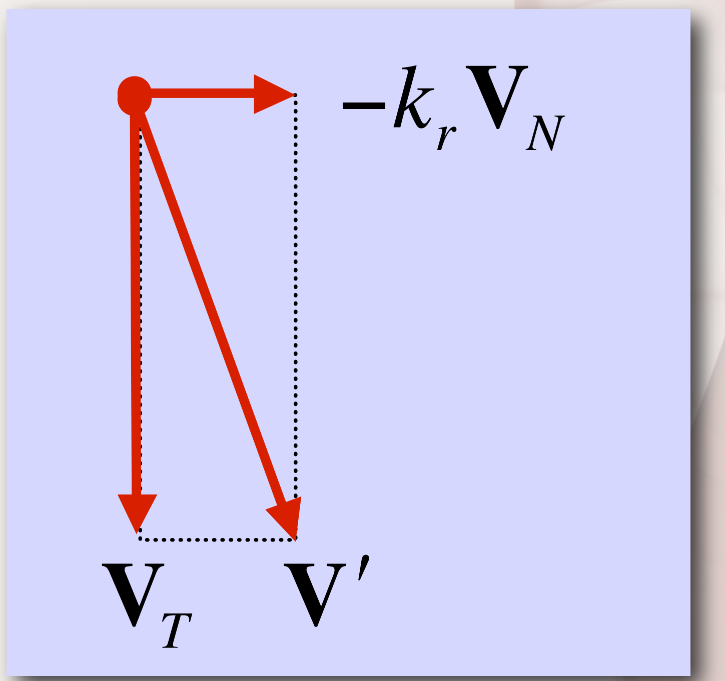
Summary

- Physics of a particle system
- Various forces acting on a particle
- Combining particles into a particle system
- Euler method for solving differential equations

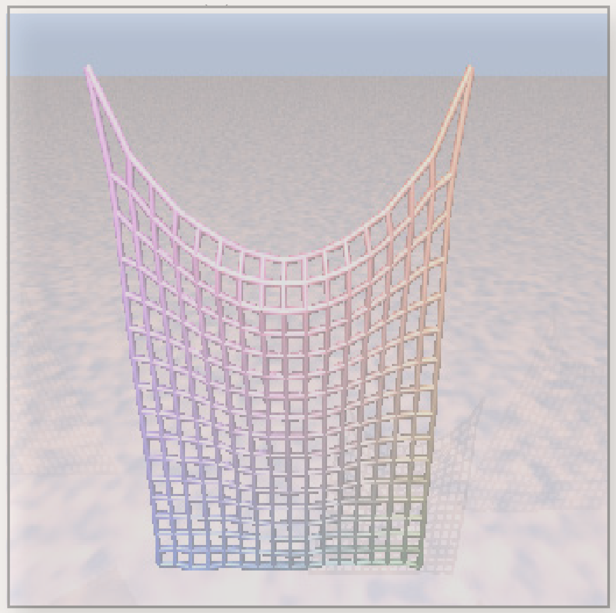
Overview



DiffEQ Review



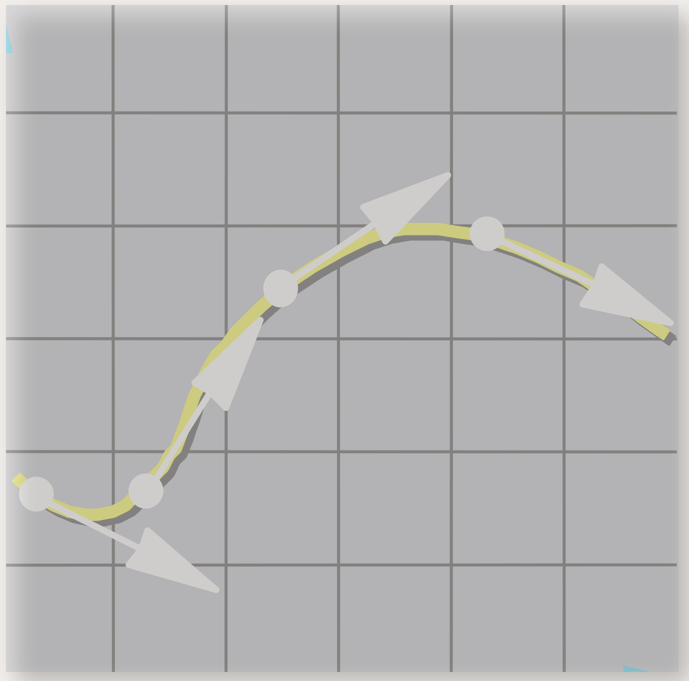
Particle Dynamics



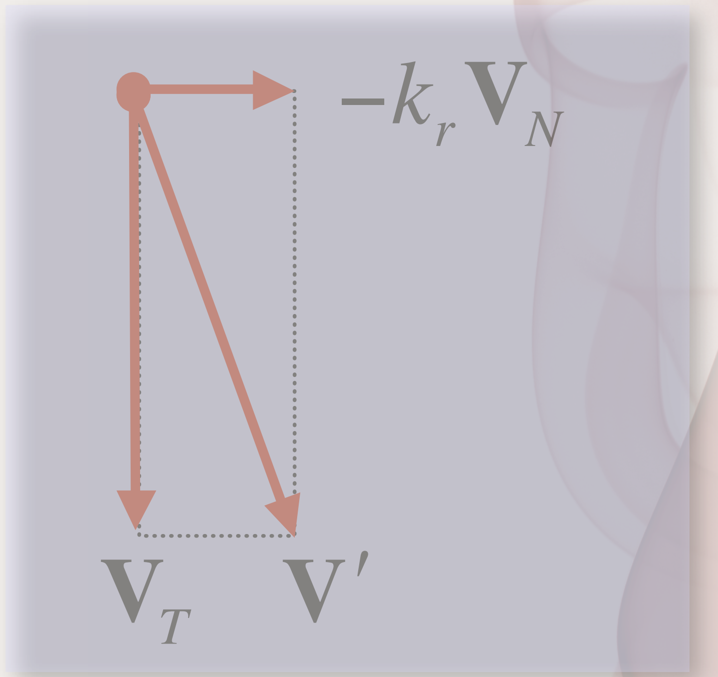
Cloth

Hair

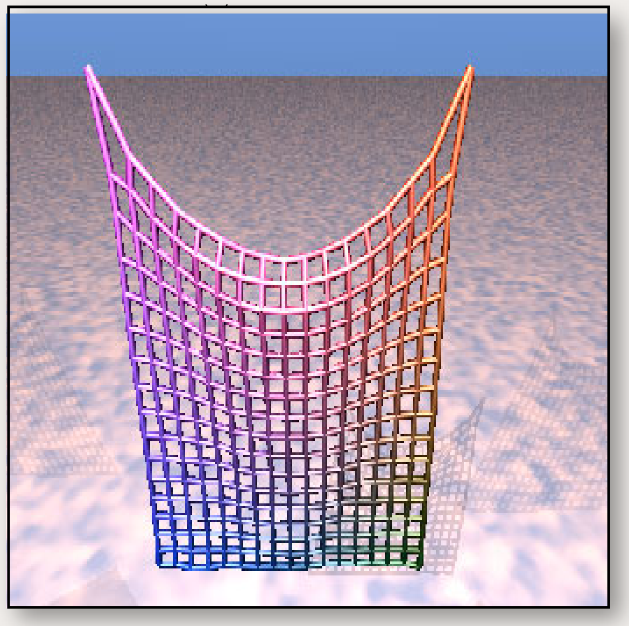
Overview



DiffEQ Review



Particle Dynamics

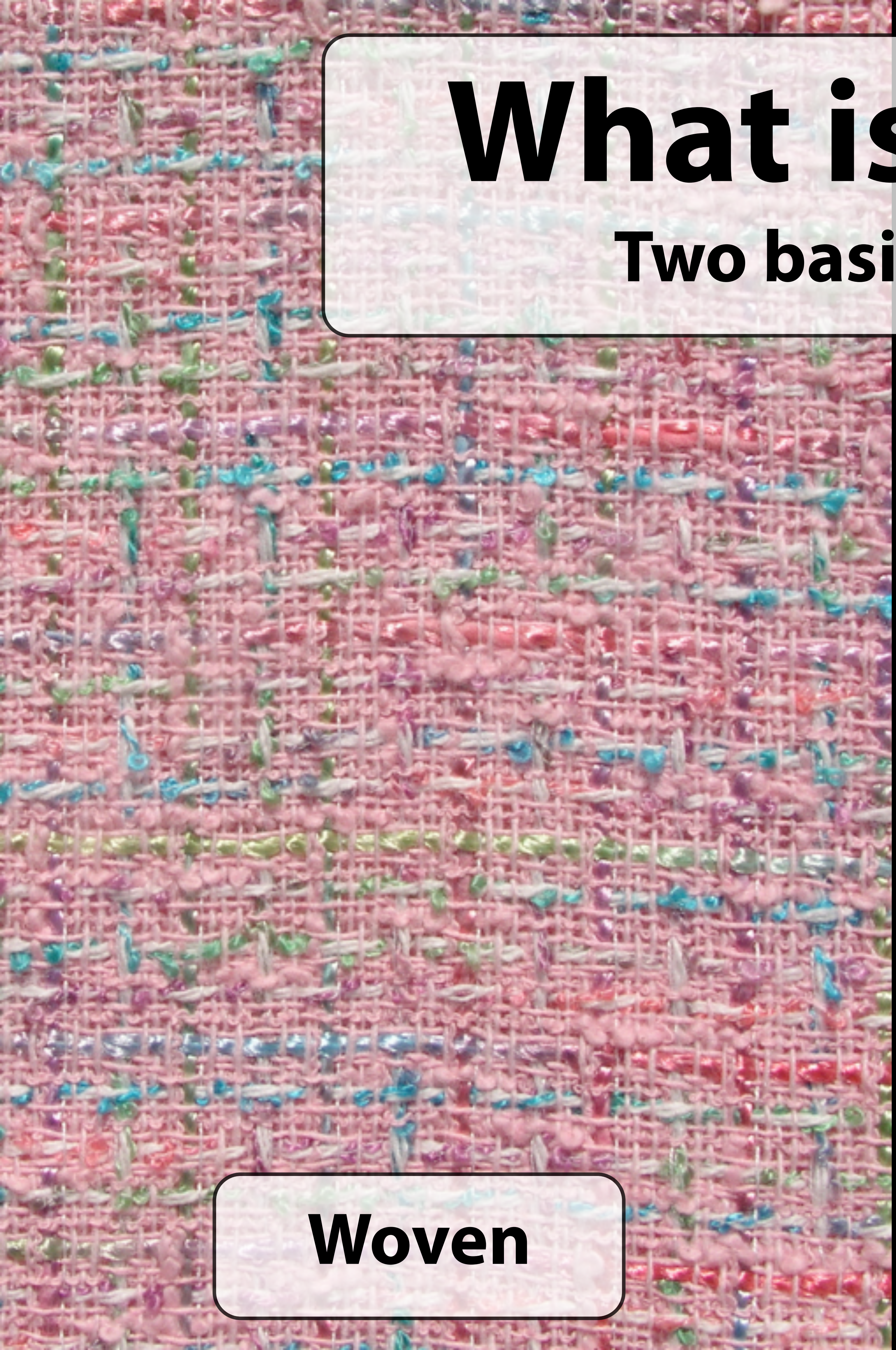


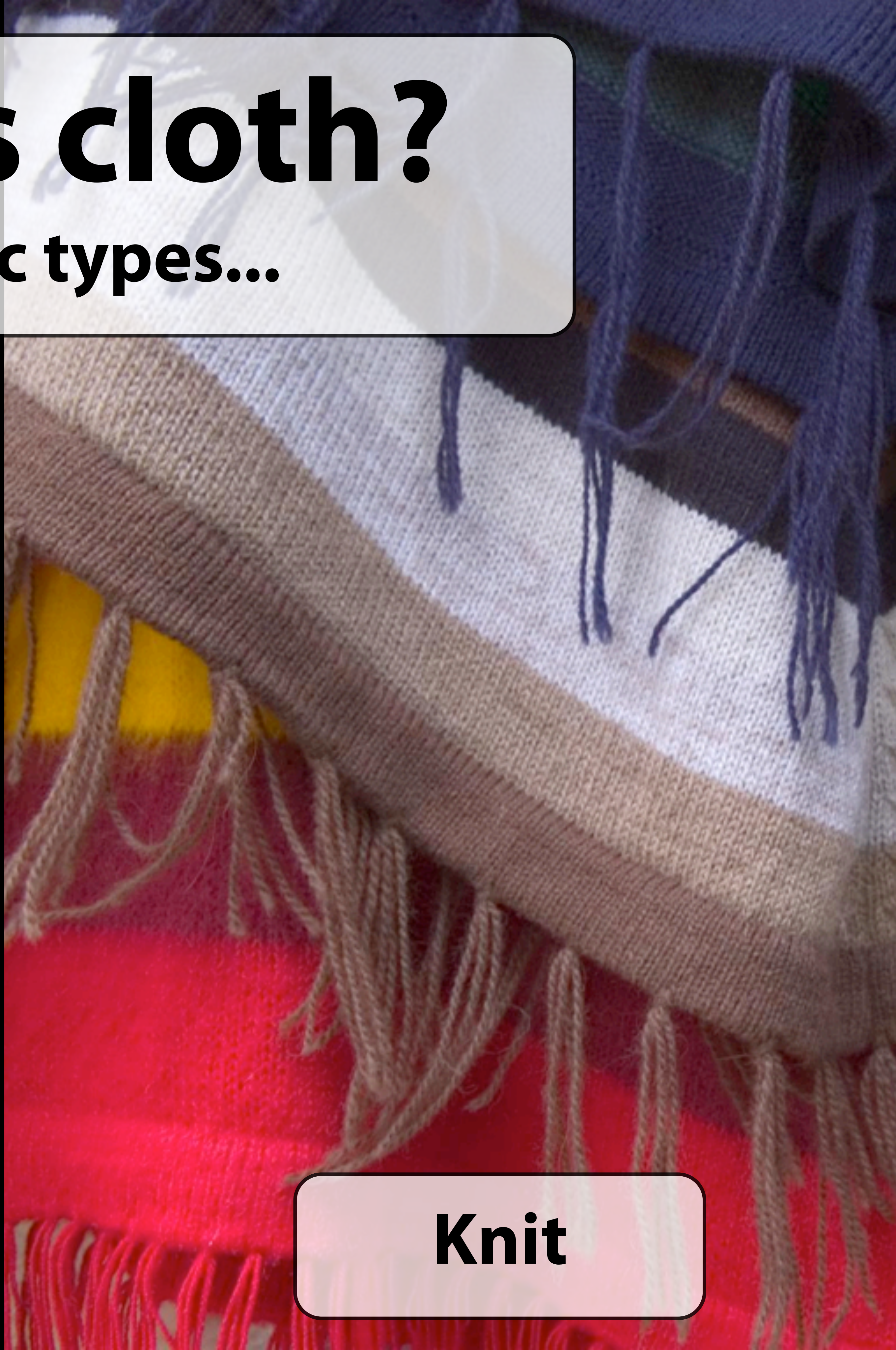
Cloth

Hair

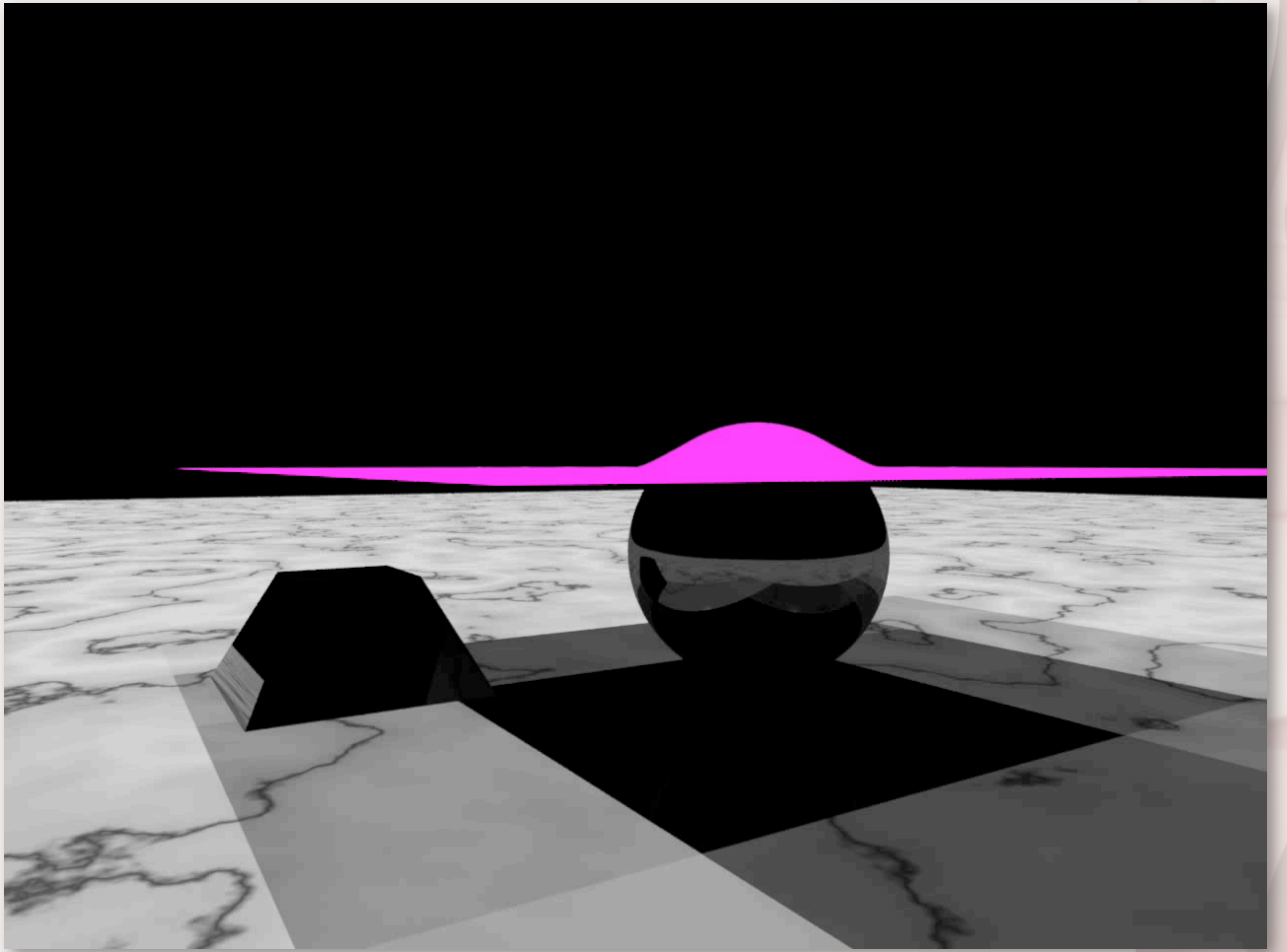
What is cloth?

Two basic types...





Woven Cloth



Bridson, R., Marino, S. and Fedkiw, R., "Simulation of Clothing with Folds and Wrinkles", ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA), edited by D. Breen and M. Lin, pp. 28-36, 2003.

Knit Cloth

Jonathan Kaldor, Doug L. James, and Steve Marschner. Simulating Knitted Cloth at the Yarn Level. SIGGRAPH 2008.

What is cloth?

- 2 basic types: woven and knit
- We'll restrict to woven
  - Warp vs. weft

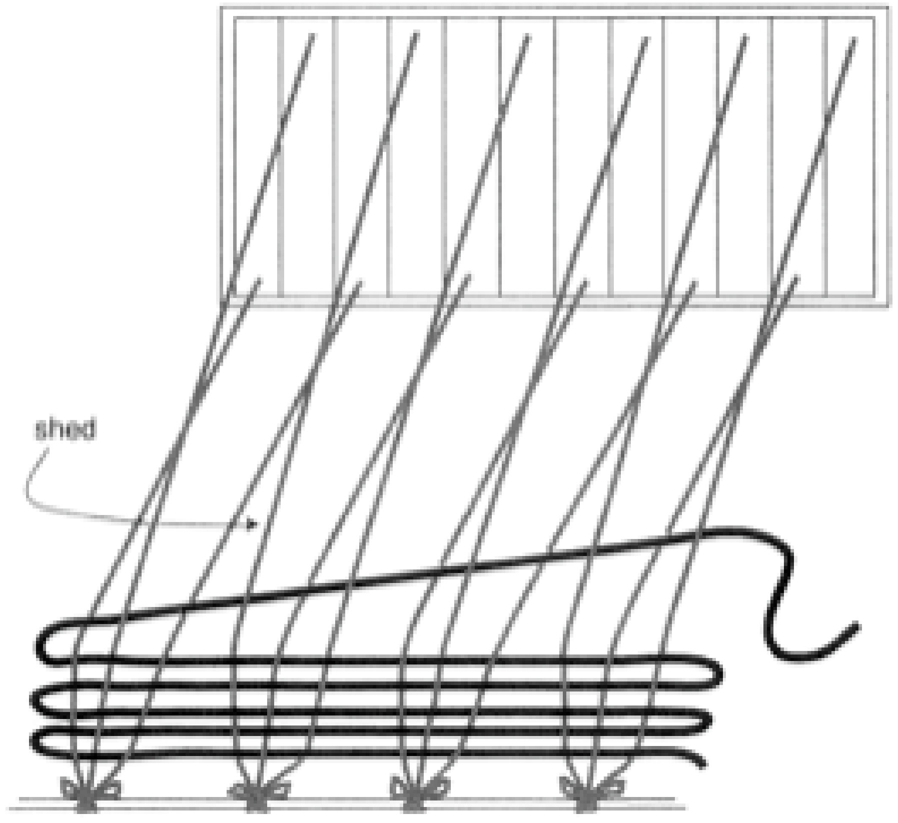
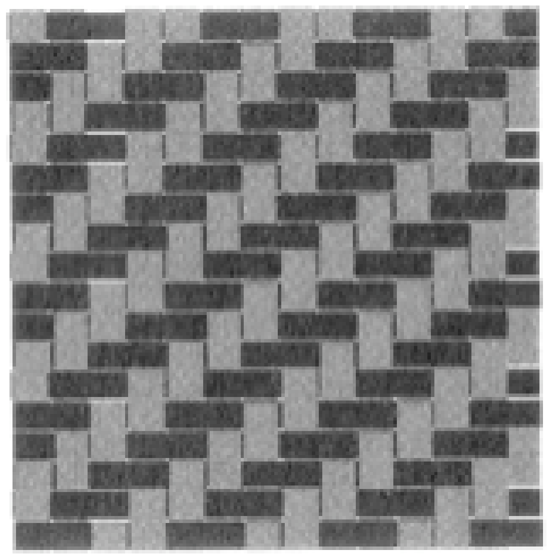
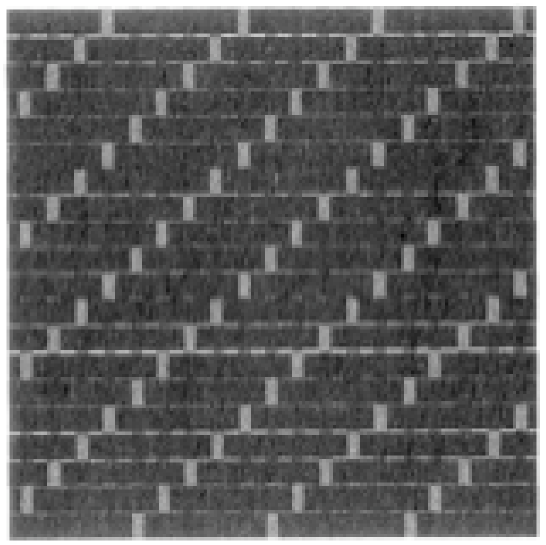


Figure 1.8. The weaving process.

House, Breen [2000]

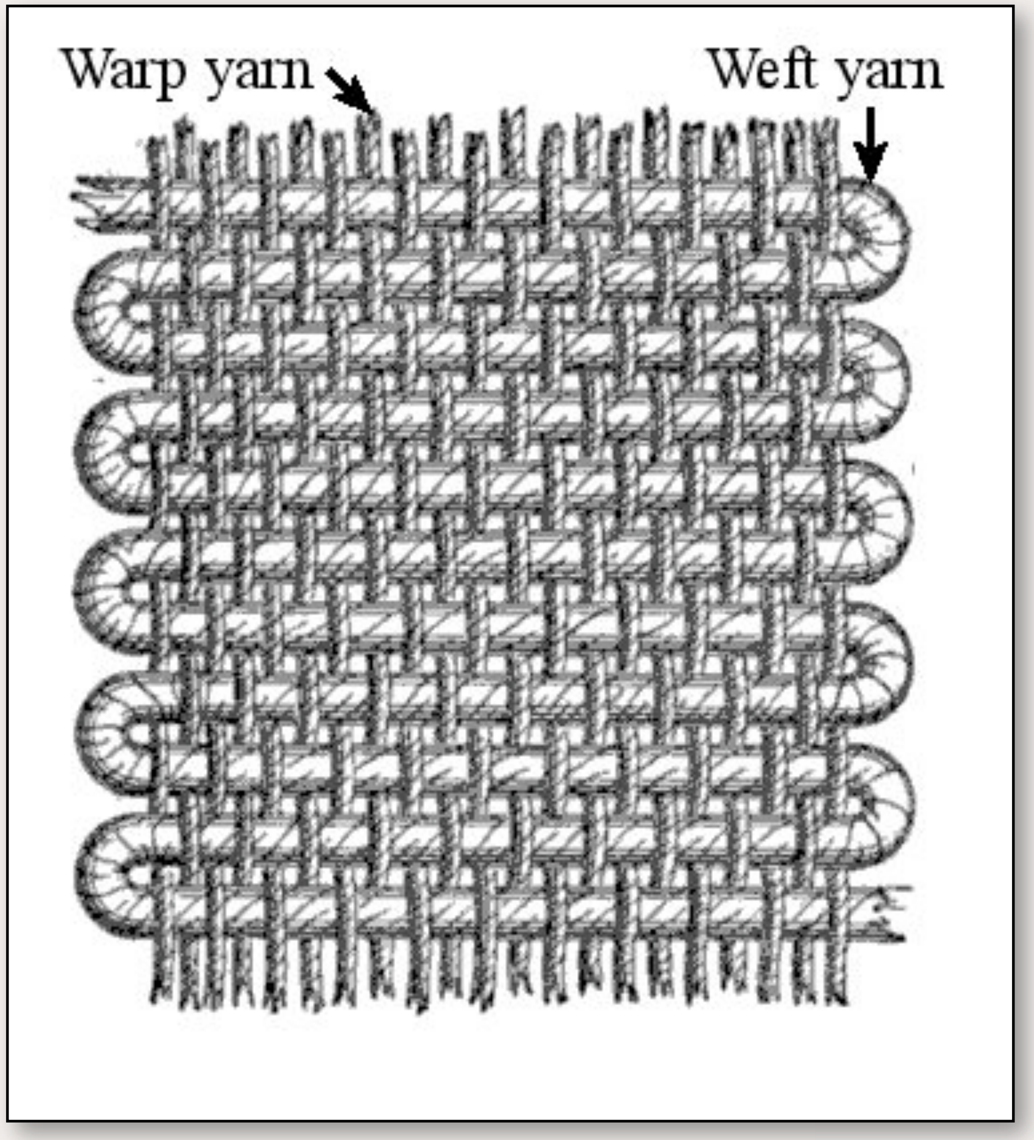


b) twill

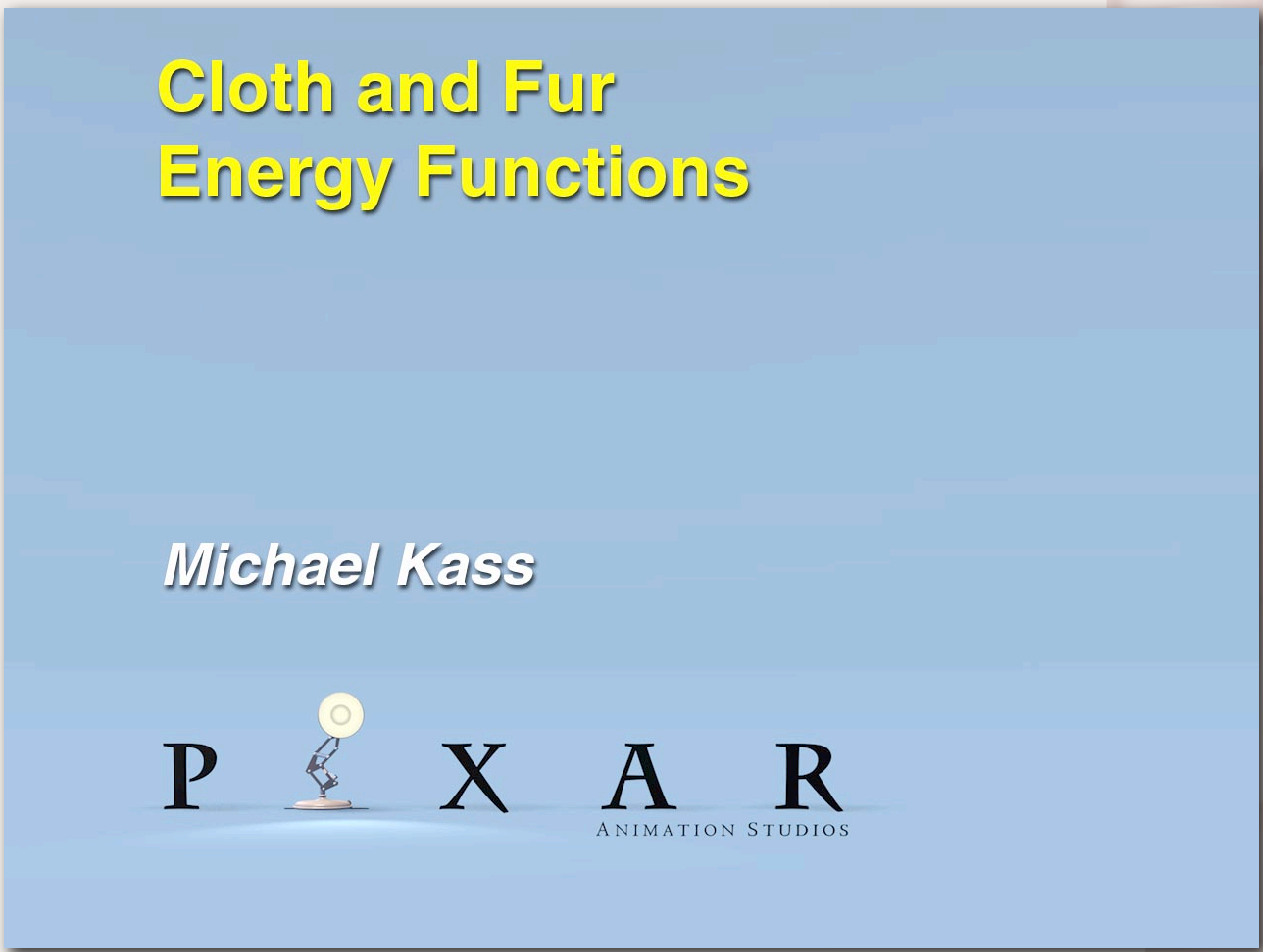


c) satin

Warp and Weft



source: Wikipedia

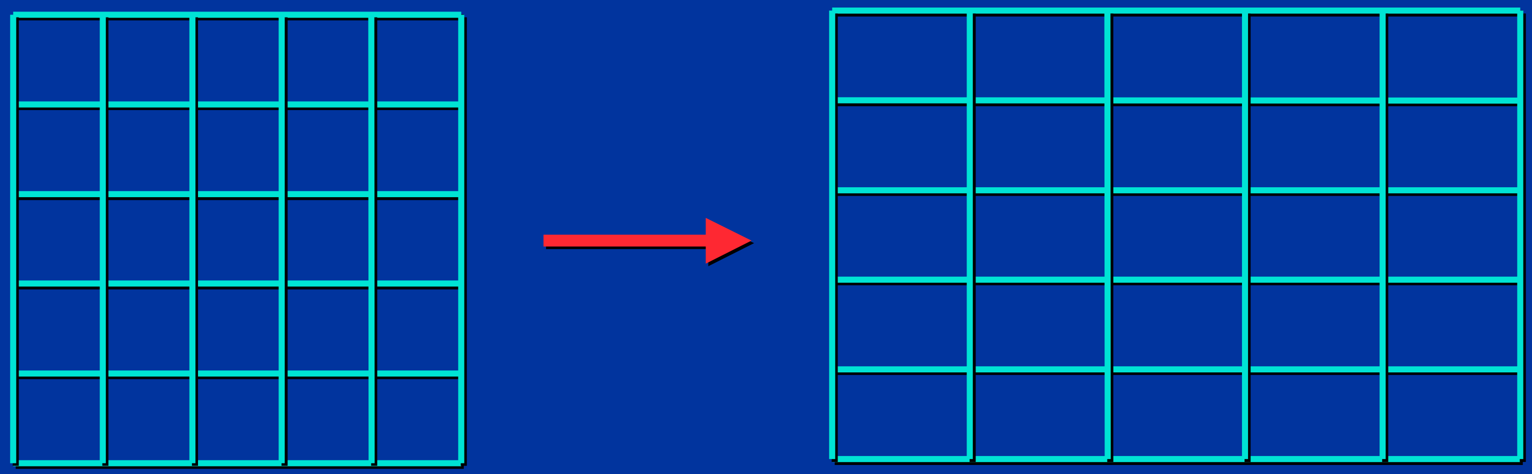


Cloth and Fur Energy Functions

Michael Kass

PIXAR
Animation Studios

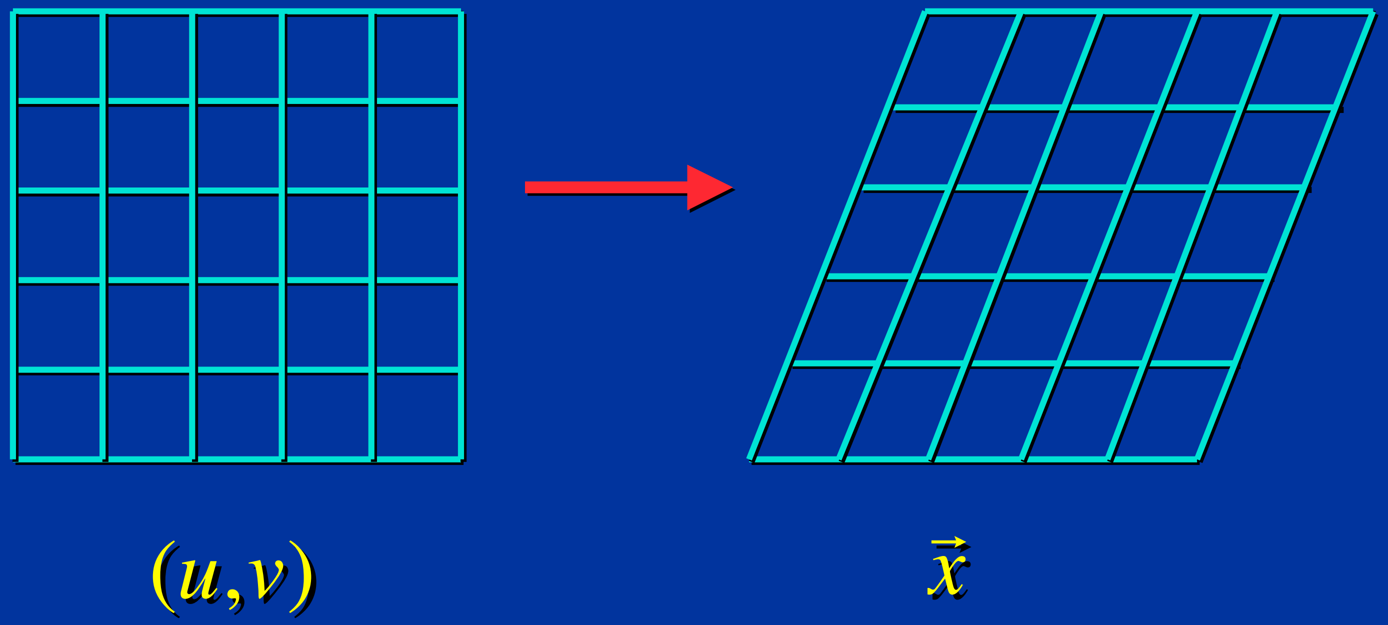
Stretch (Continuum Version)



(u,v) $\qquad\qquad\qquad \vec{x}$

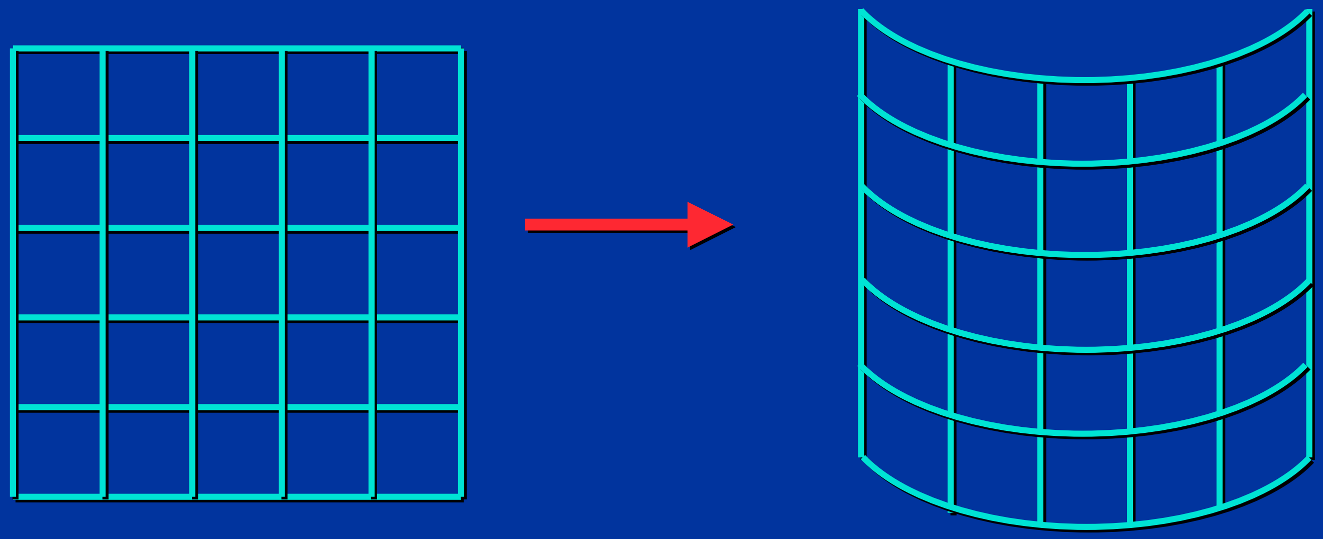
$$S_u = \left\| \frac{\partial \vec{x}}{\partial u} \right\| - 1 \qquad\qquad E = \tfrac{1}{2} k \int (S_u^2 + S_v^2)\, du\, dv$$

Shear (Continuum Version)



$$\theta = \cos^{-1}\left(\widehat{\frac{\partial\vec{x}}{\partial u}} \bullet \widehat{\frac{\partial\vec{x}}{\partial v}}\right) \qquad E = \tfrac{1}{2}k\int \theta^2\, du\, dv$$

Bend (Continuum Version)

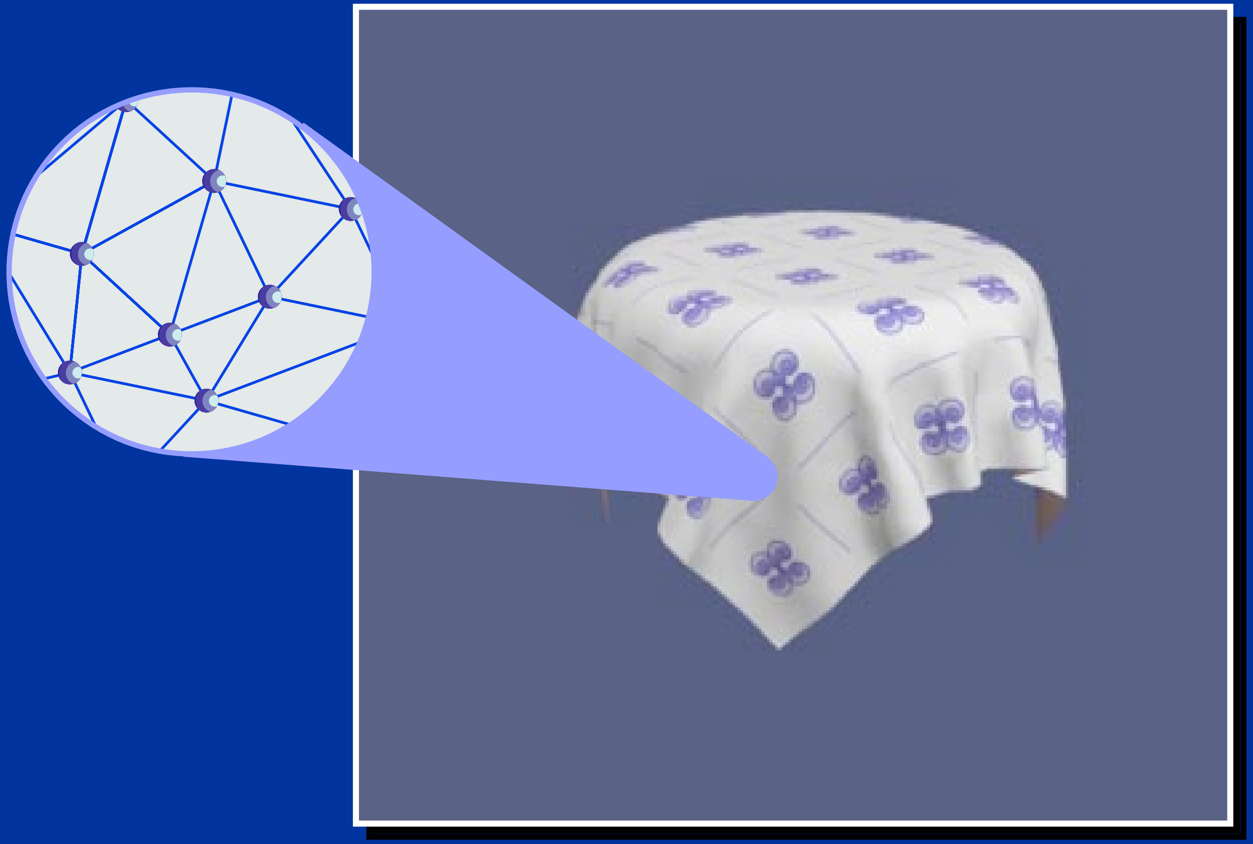


$$E = \tfrac{1}{2} k \int (\kappa_u^2 + \kappa_v^2)\, du\, dv$$

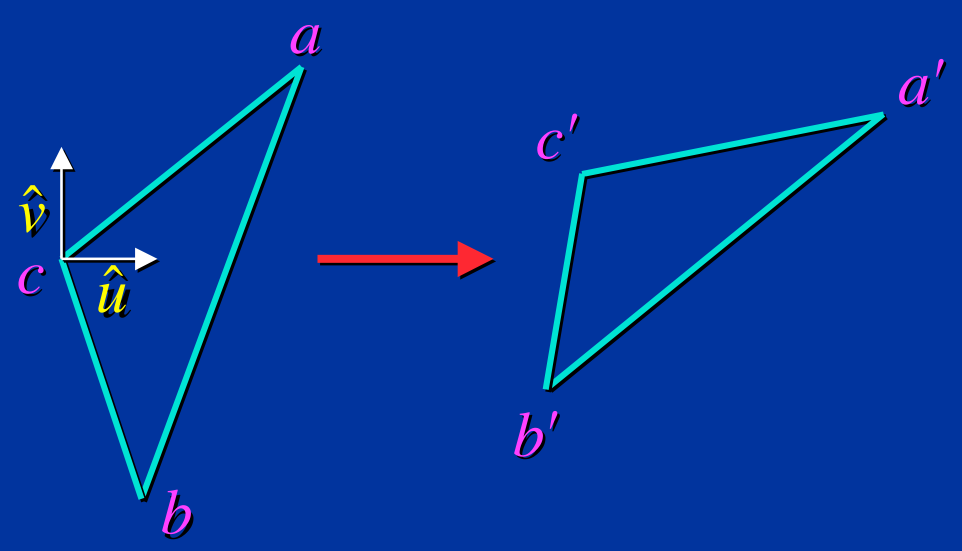
Resitence To...

- Stretching
- Shearing
- Bending

Discretization

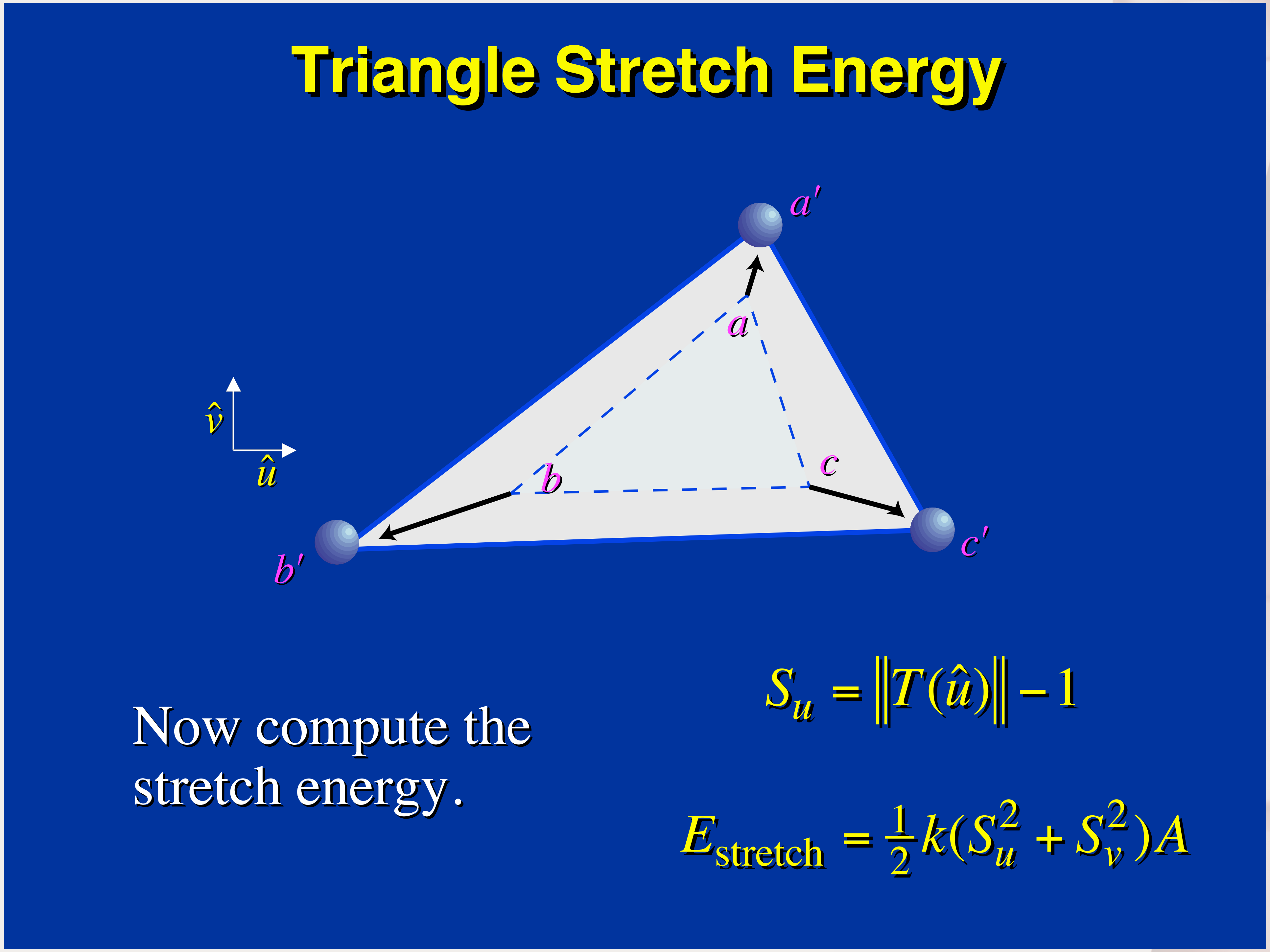


Triangle Energy

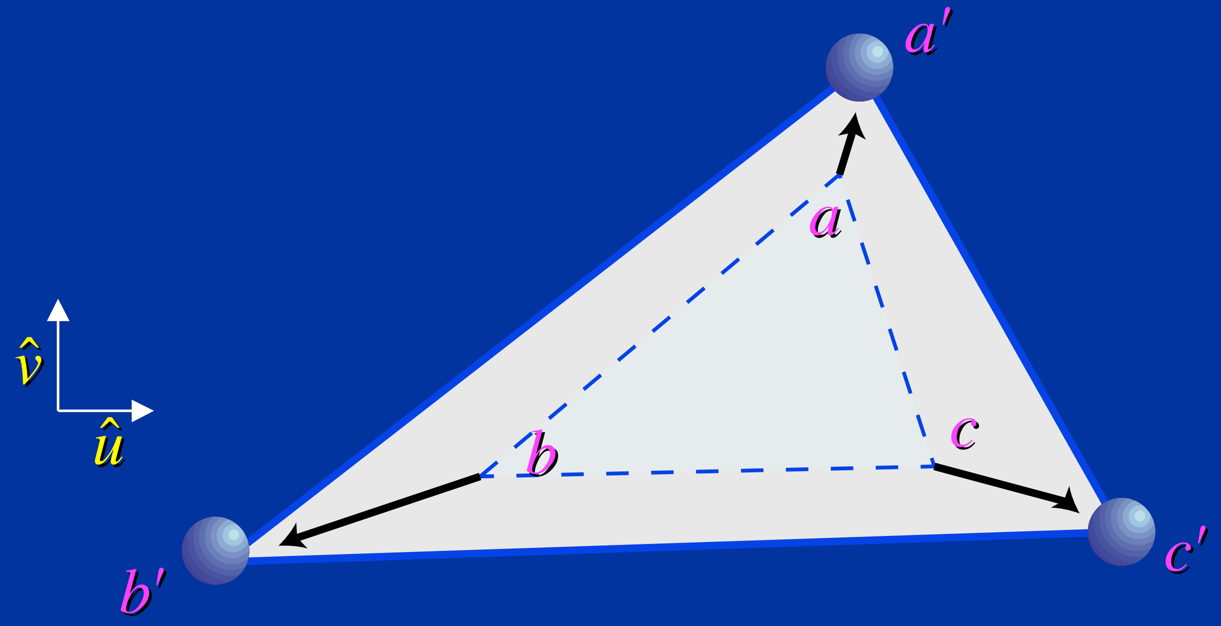


First, compute the affine transformation *T* that maps:

$$T : a \rightarrow c'$$
$$b \rightarrow b'$$
$$c \rightarrow c'$$



Triangle Stretch Energy

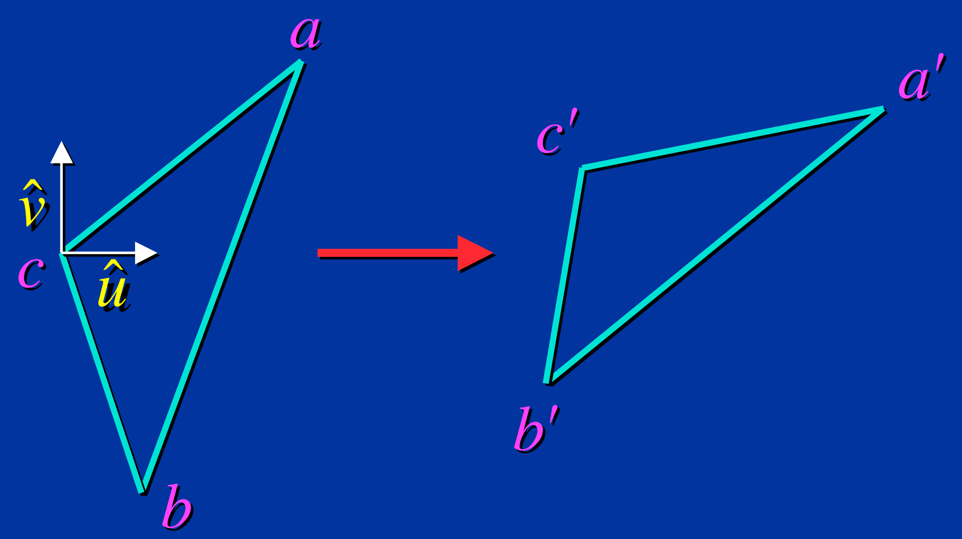


Now compute the stretch energy.

$$S_u = \left\| T(\hat{u}) \right\| - 1$$

$$E_{\text{stretch}} = \tfrac{1}{2} k (S_u^2 + S_v^2) A$$

Triangle Shear Energy

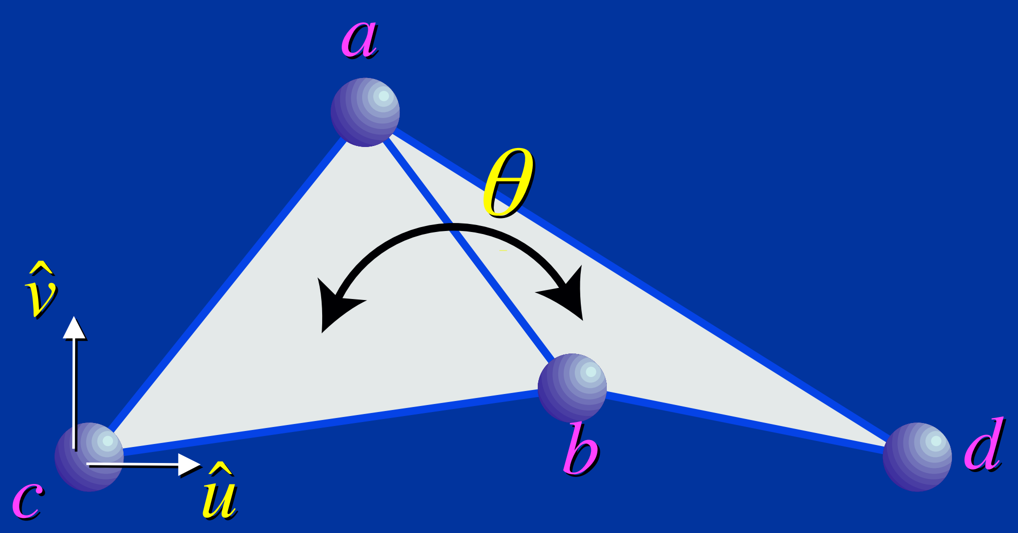


Next compute the shear energy.

$$\theta = \cos^{-1}(T(\hat{u}) \bullet T(\hat{v}))$$

$$E_{\text{shear}} = \frac{1}{2} k \theta^2 A$$

Triangle Bend Energy

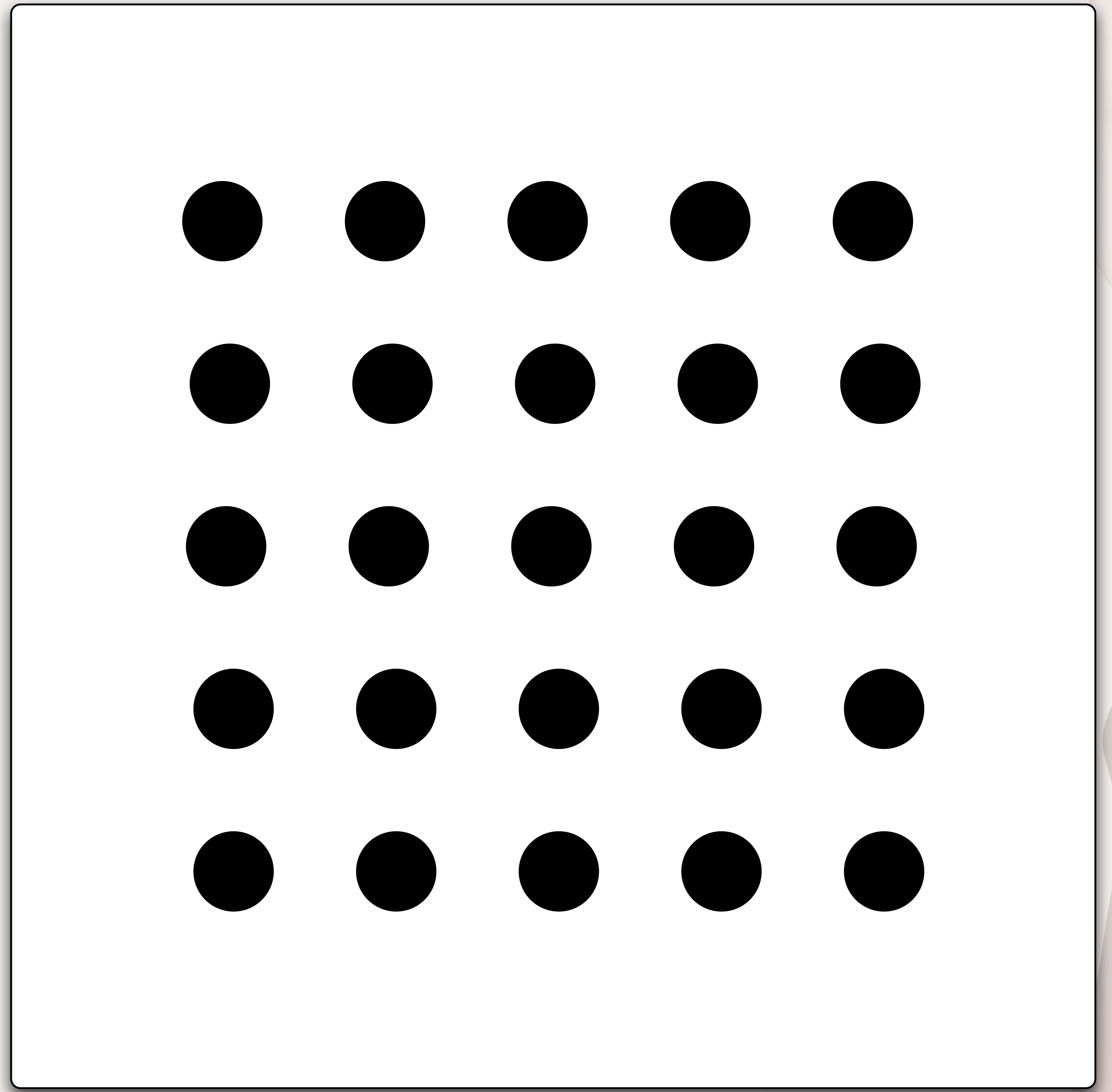


$$\kappa = \frac{\theta}{l_{perp}}$$

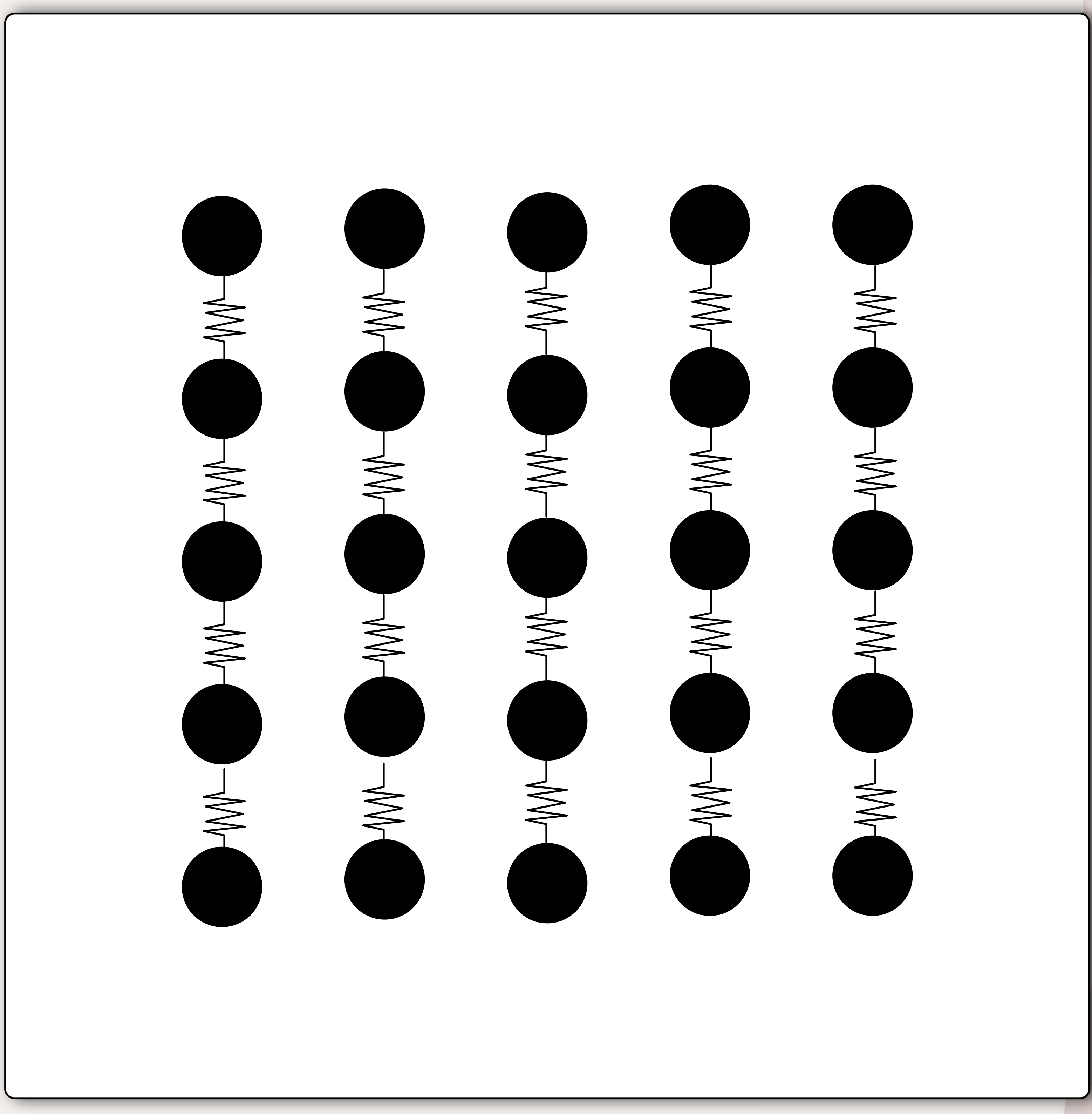
Finally compute the bend energy.

$$E_{\text{bend}} = \frac{k}{2}(\kappa^2)A$$

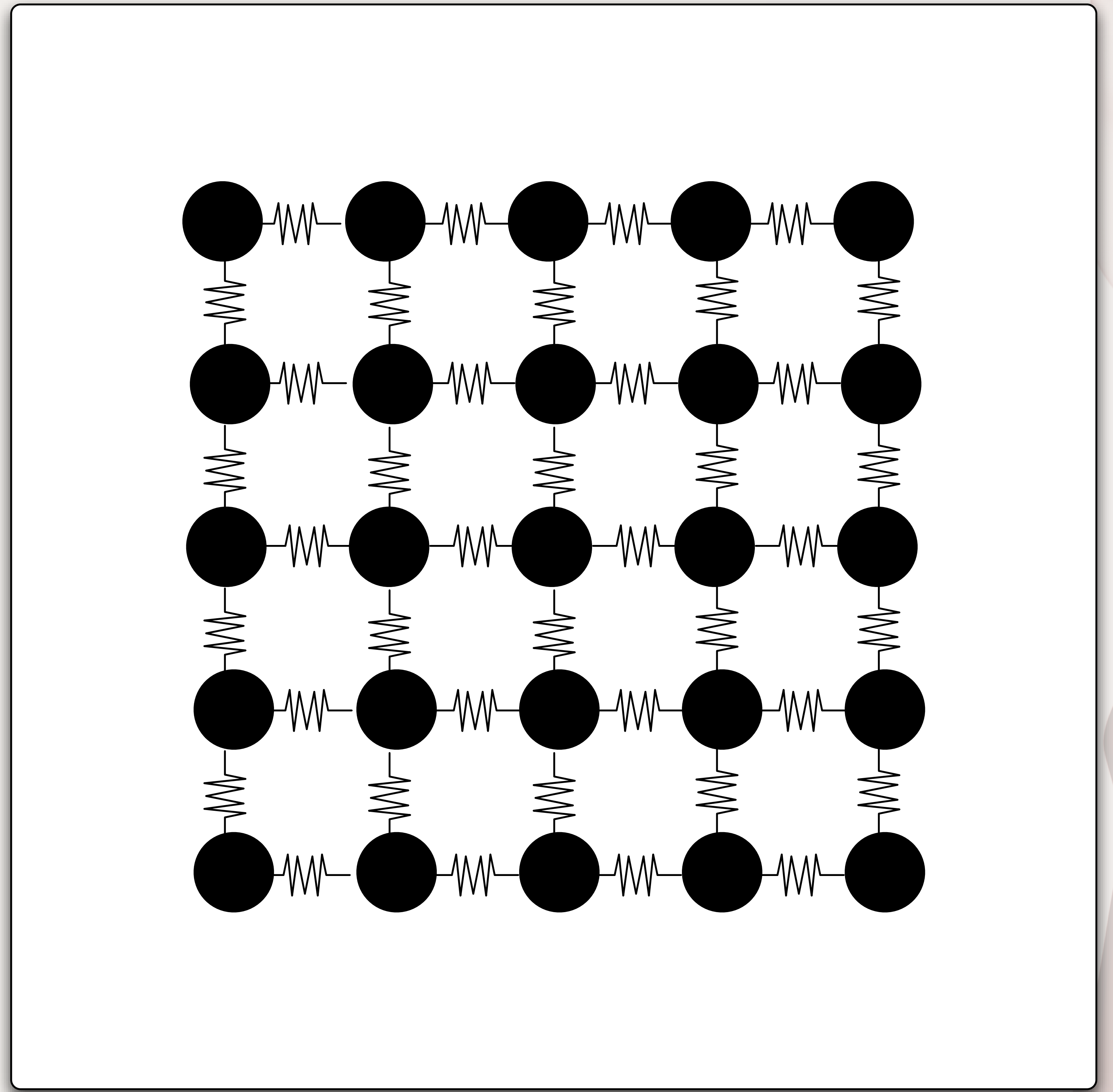
Basic Model



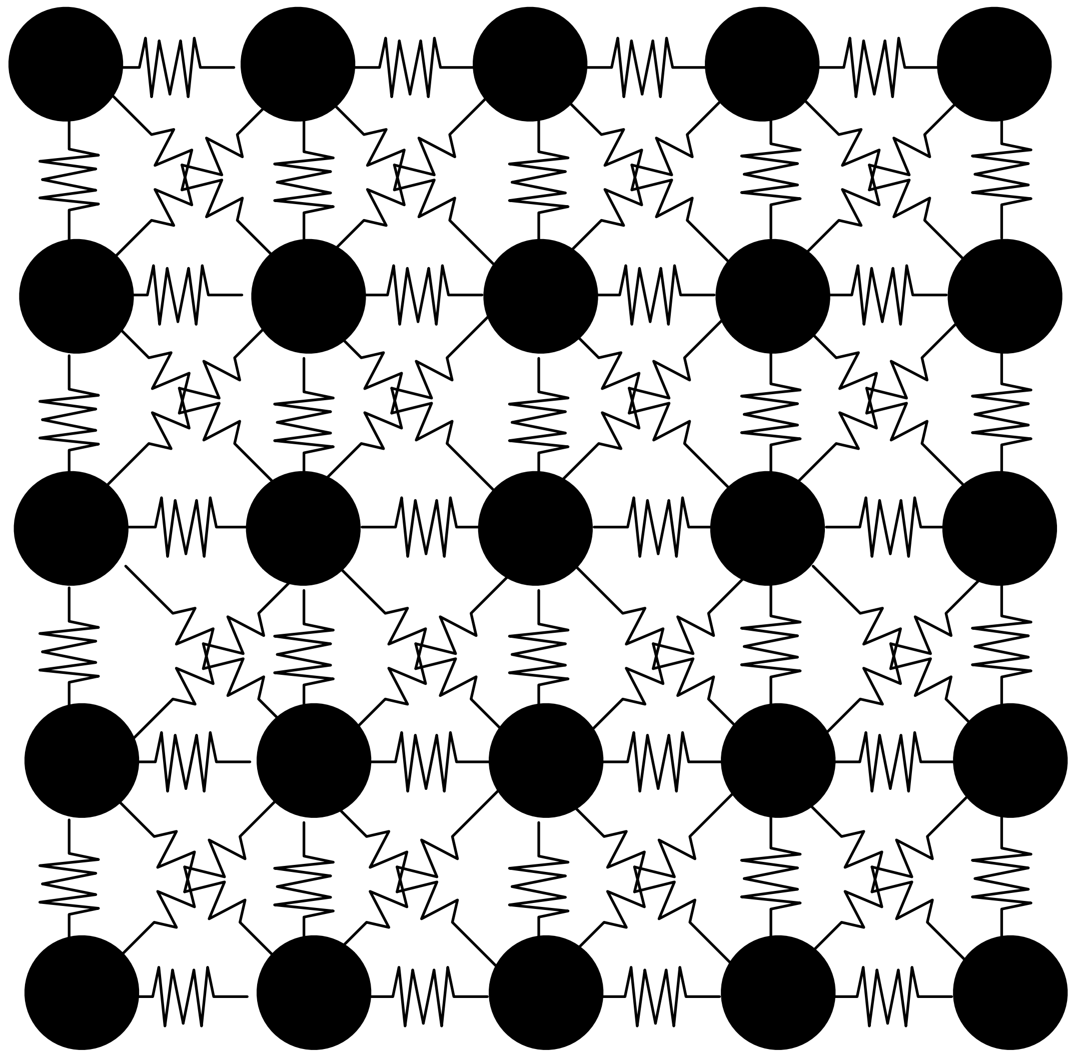
Warp Srings



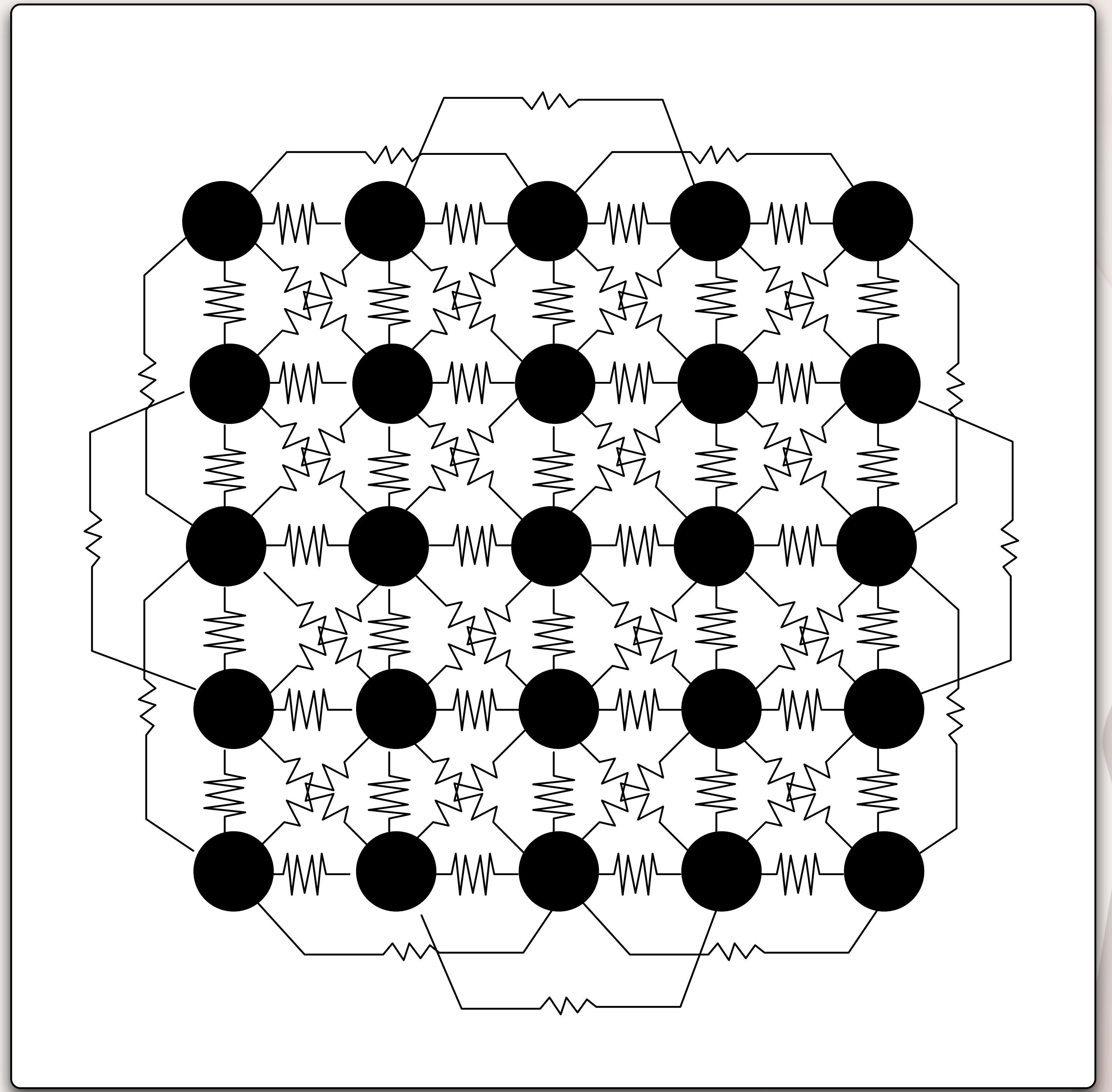
Weft Springs



Shear Springs



Bend Springs



Parameters

- Given stretch, shear, and bending constants...
- How would you make a wrinkly t-shirt, thick cloth, or non-uniform cloth?

Creating Clothes

- How could we create the 3D model the clothes for a character?

Non-flat Cloth

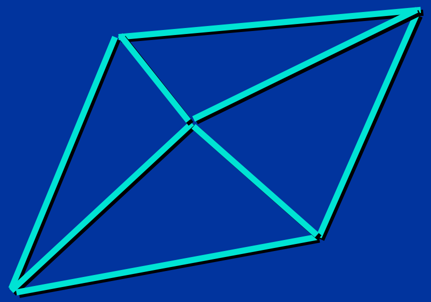
Non-flat cloth is strange stuff:

A baseball with no seams?

Wrinkles give strength?

Clothing cut out of a volume?

Convexities that pop?



Even 4 Triangles are over-constrained:
16 rest angles, 8 rest lengths.
24 constraints on 15 dofs.
Must be consistent!

Rest Mesh Options

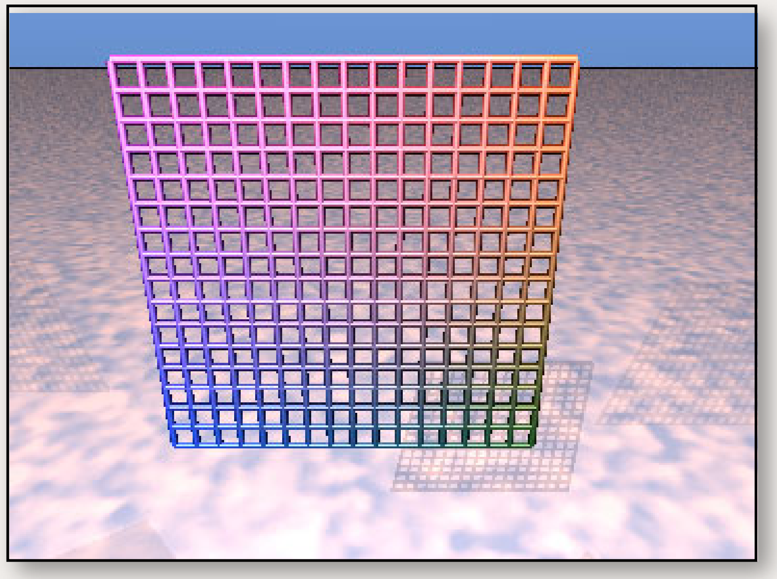
Model in 3D

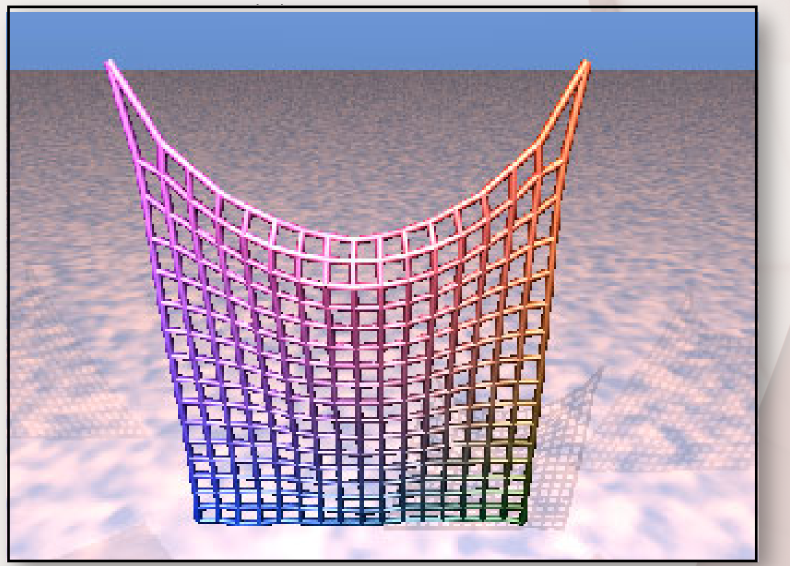
- Clothing already on characters.
- Can directly craft desired 3D shape.
- Annotate warp/weft directions.
- Clothing probably will not locally flatten.

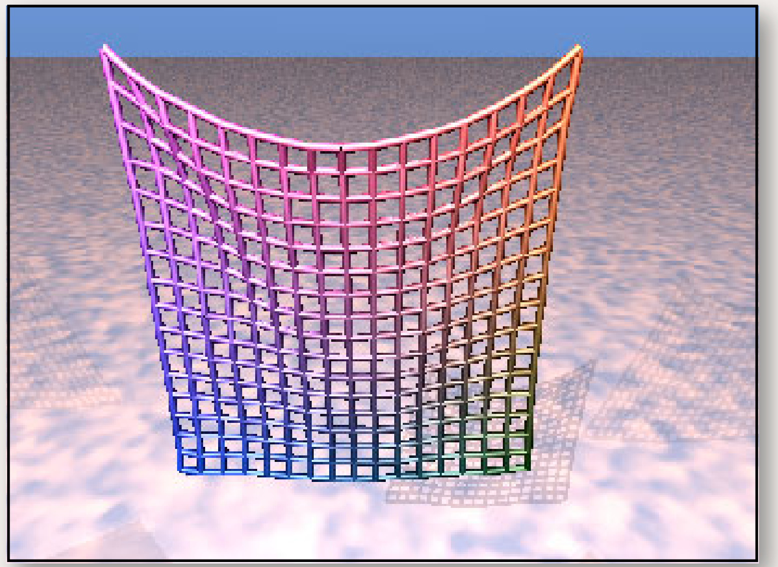
Model in 2D

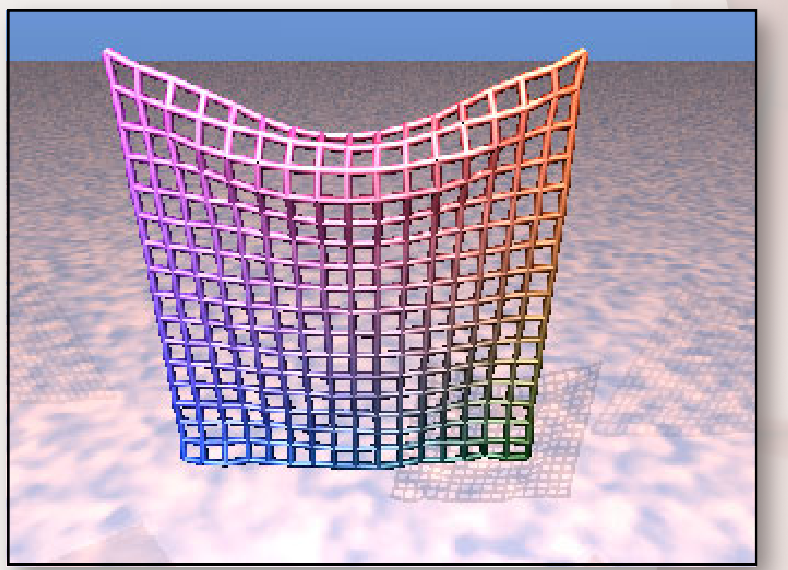
- Must put clothing on characters
- Hire a tailor to get the pattern right.
- Sew parts together.
- Clothing guaranteed to flatten locally.
- Greater realism.

Springs vs. Constraints

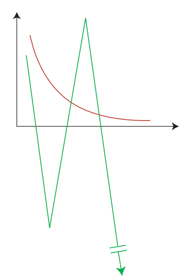

Before Simulation


Only Springs

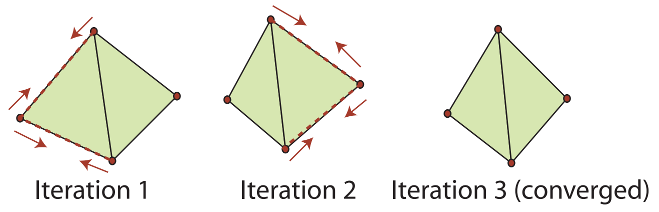

Stretch Constraints


Stretch+Shear Constraints

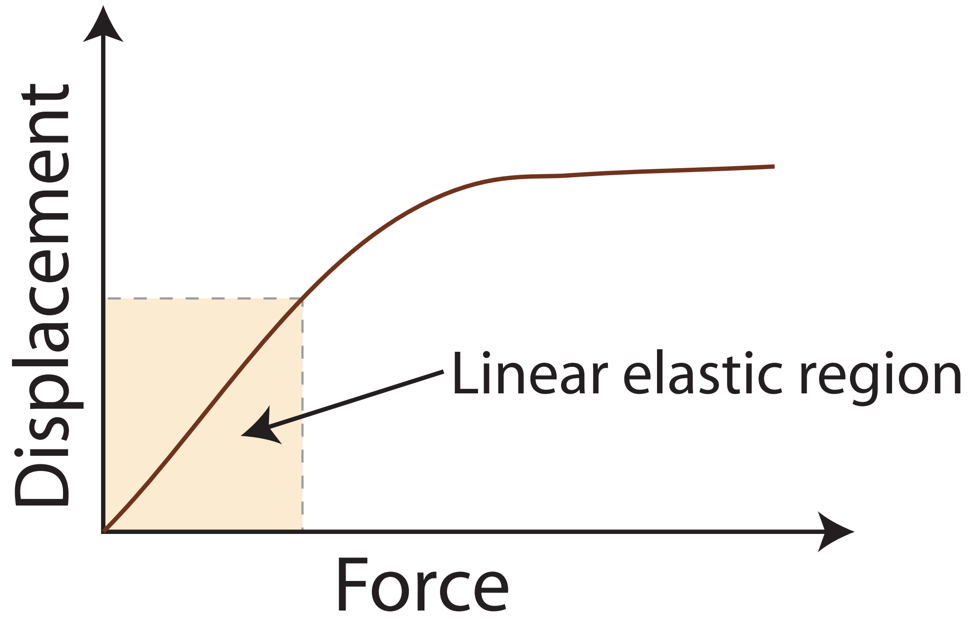
Source: Xavier Provot
Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior



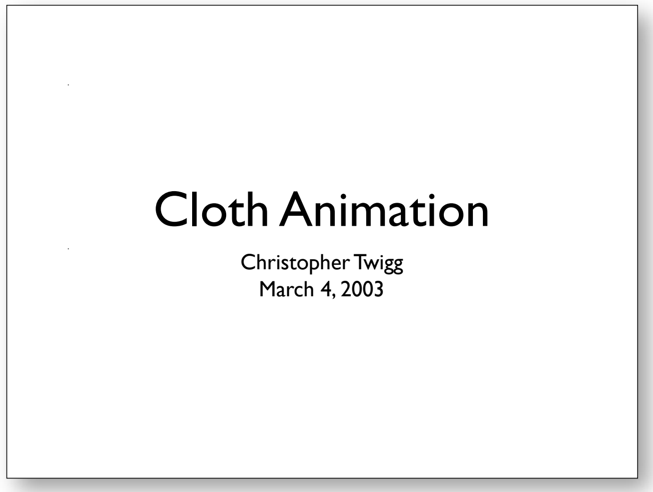
Avoiding stiffness (2)



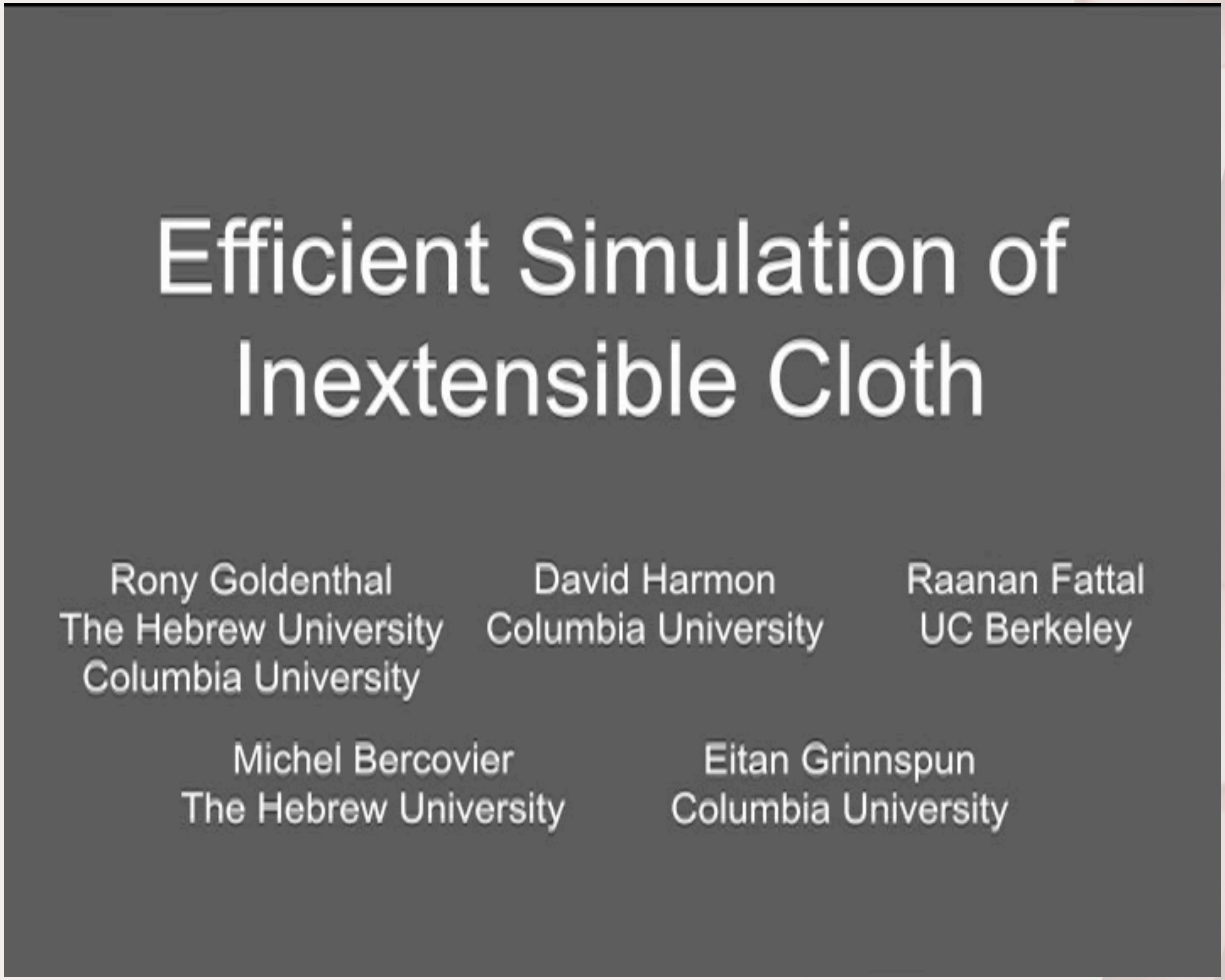
- Popular for interactive applications
- Justification
  - Biphasic spring model
  -
  -
  -
  -
  -
  -
  -
  - Plausible dynamics



From Desbrun, Meyer, Barr [2000]



Cloth Animation
Christopher Twigg
March 4, 2003



Efficient Simulation of Inextensible Cloth

Rony Goldenthal
The Hebrew University
Columbia University

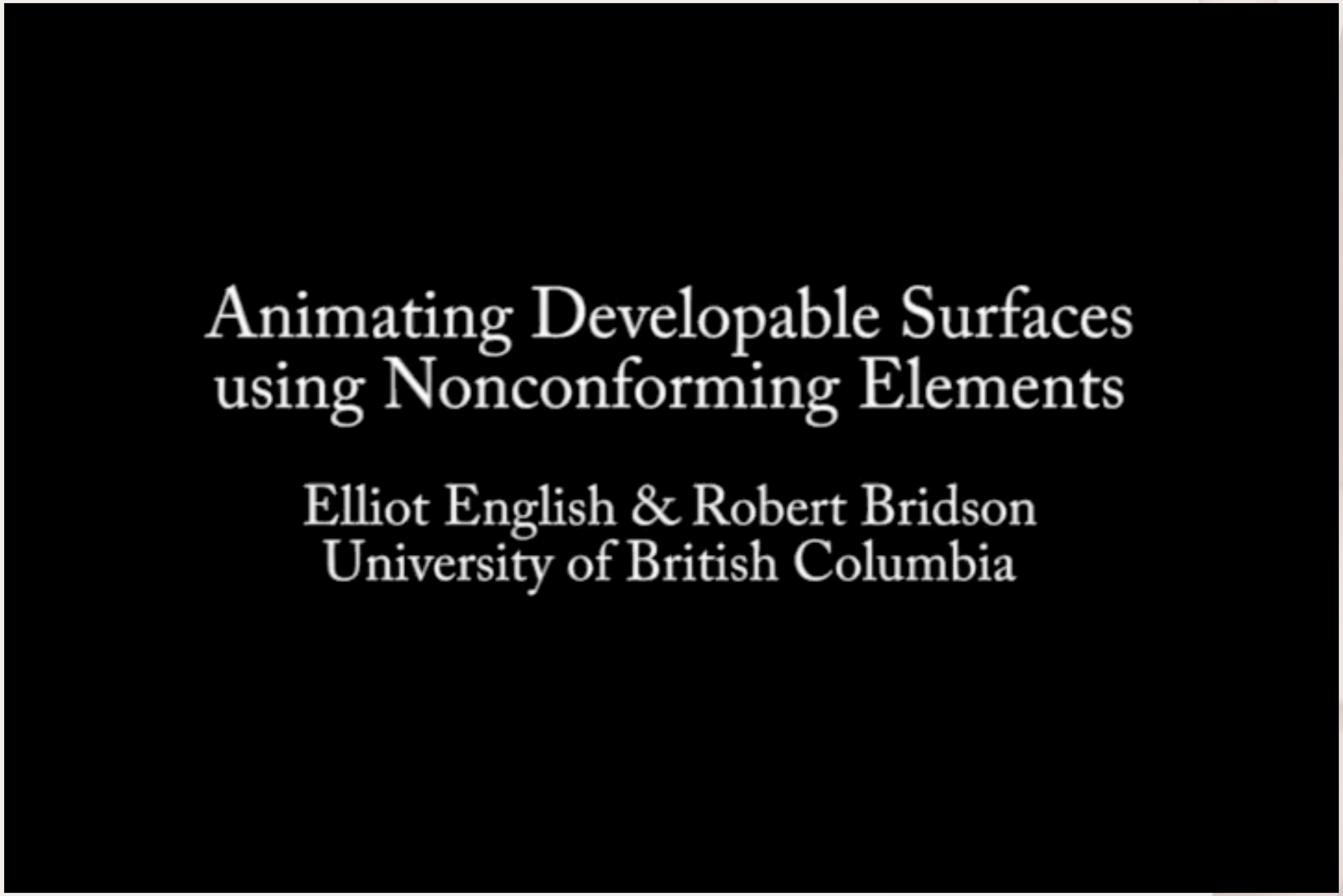
David Harmon
Columbia University

Raanan Fattal
UC Berkeley

Michel Bercovier
The Hebrew University

Eitan Grinnspun
Columbia University

Developable Surfaces



Animating Developable Surfaces using Nonconforming Elements

Elliot English & Robert Bridson
University of British Columbia

Developable Surfaces

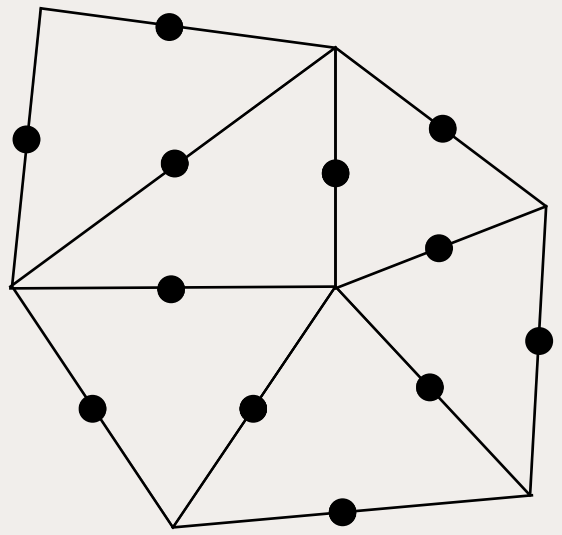
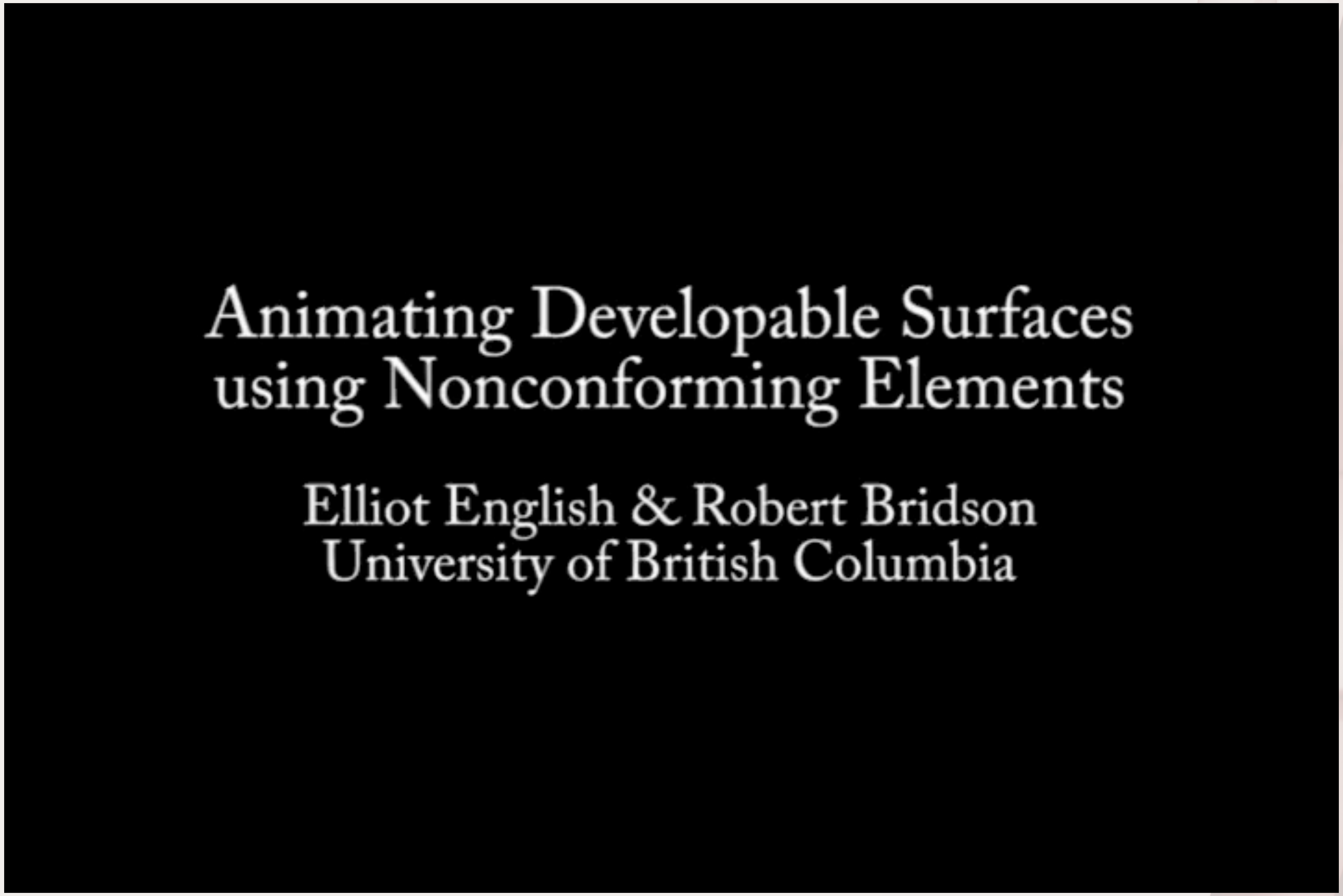


Figure 2: *Schematic of nonconforming variables, located at mid-points of edges between triangles. While continuous at these points, the surface may be discontinuous along the rest of each edge.*

Elliot English[*] University of British Columbia
Robert Bridson[†] University of British Columbia

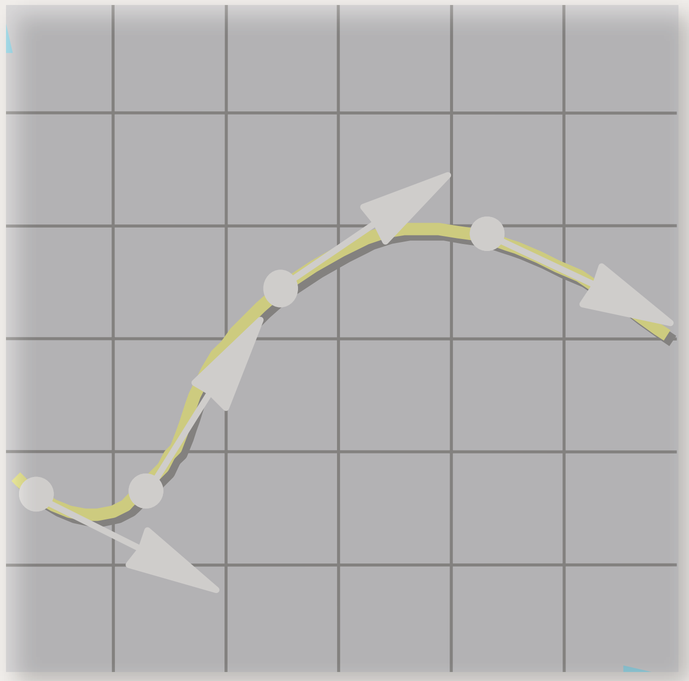
Developable Surfaces



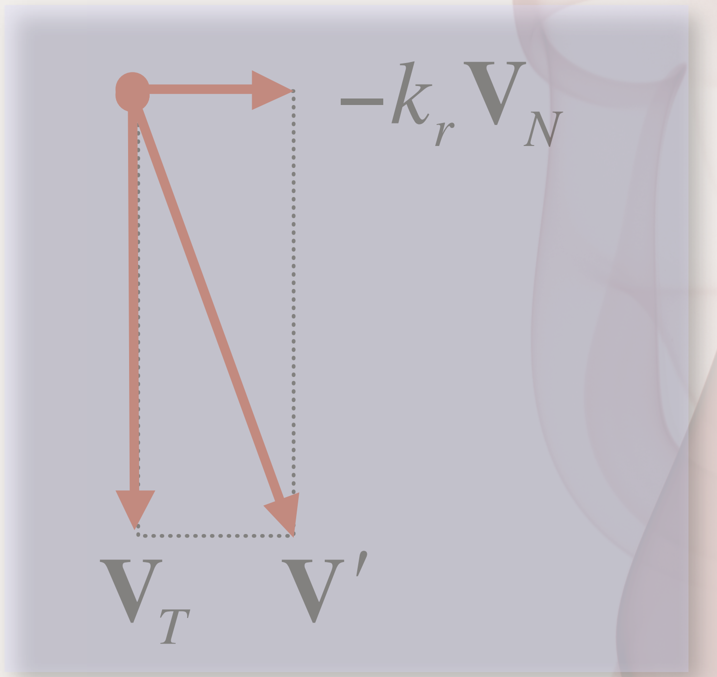
Animating Developable Surfaces using Nonconforming Elements

Elliot English & Robert Bridson
University of British Columbia

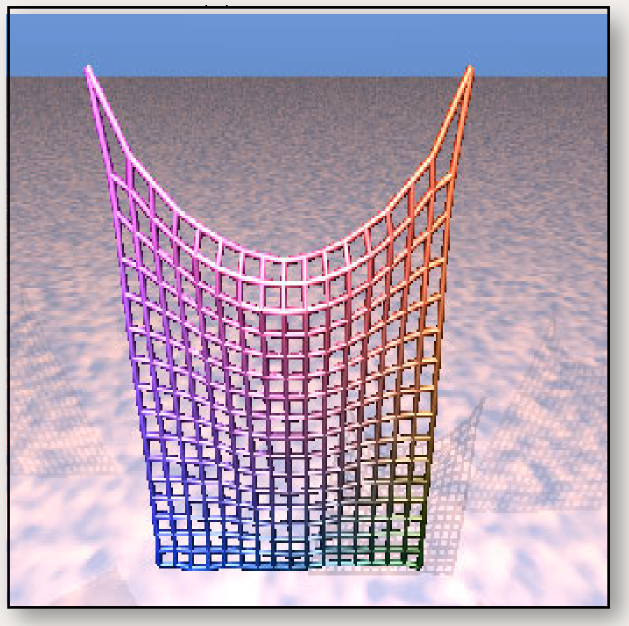
Overview



DiffEQ Review



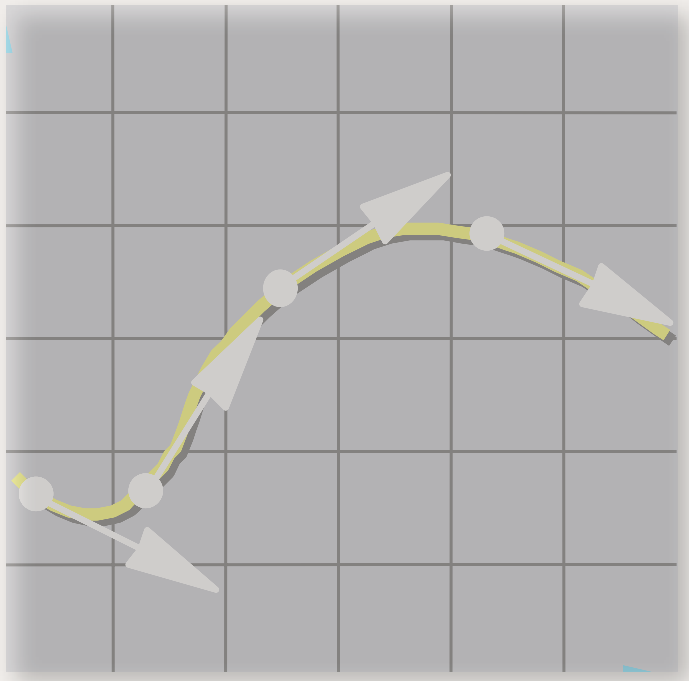
Particle Dynamics



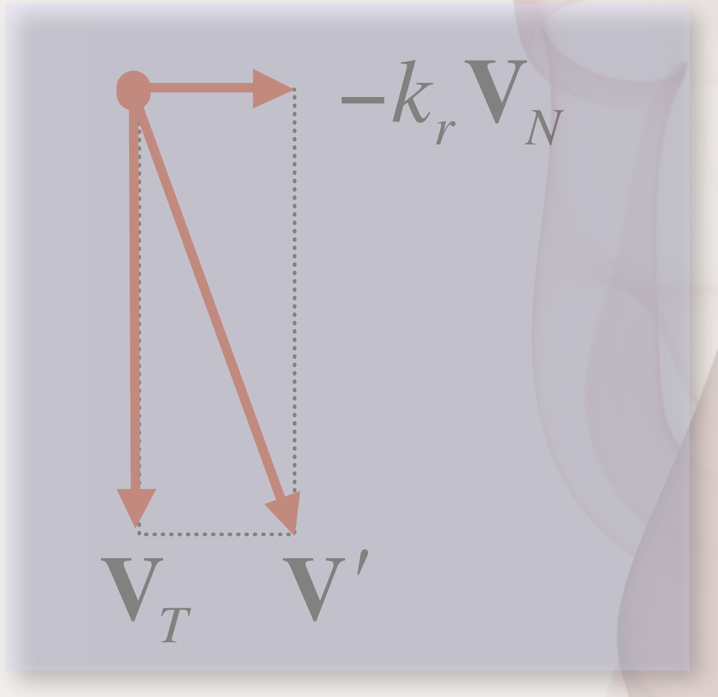
Cloth

Hair

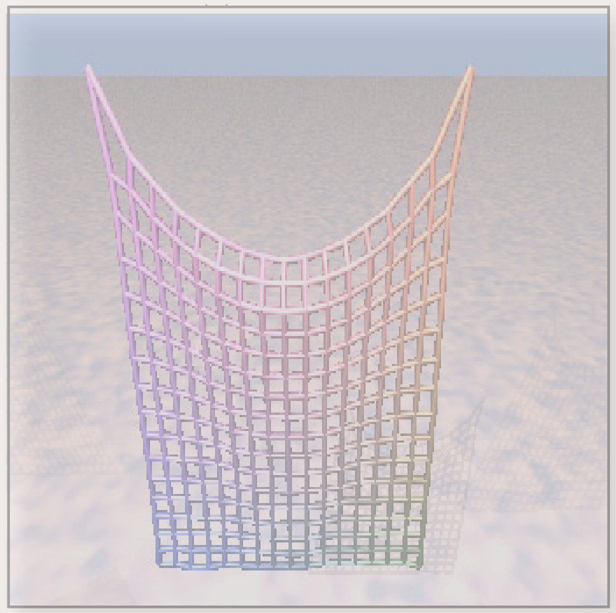
Overview



DiffEQ Review



Particle Dynamics



Cloth

Hair

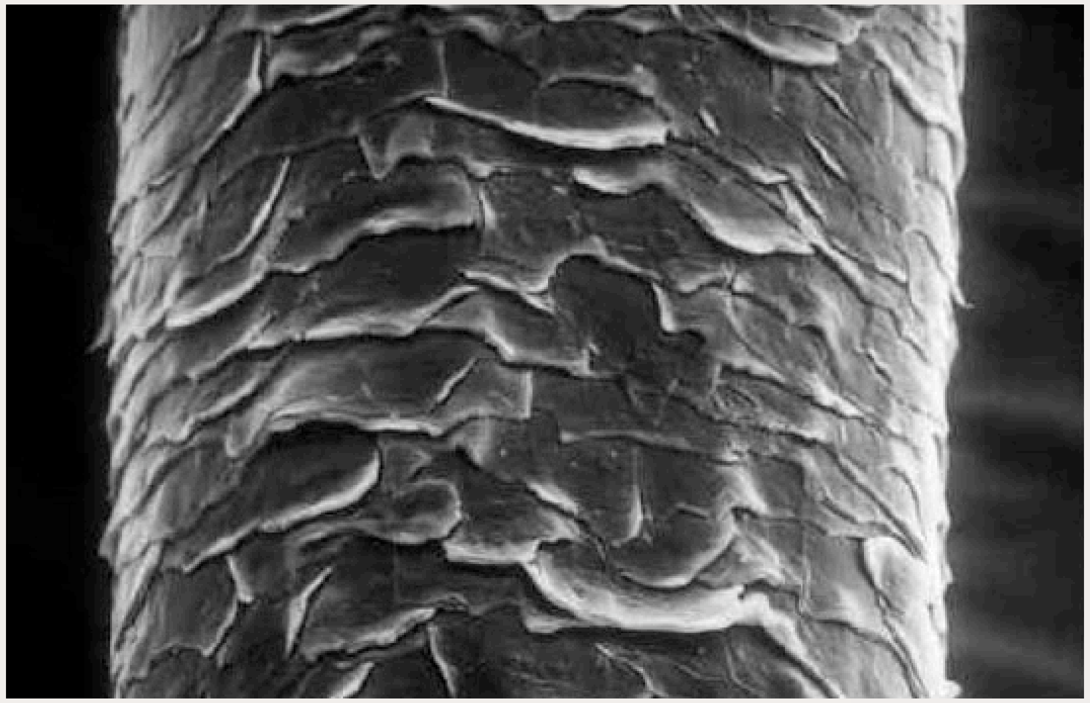
Real Hair: Curly

Short
curly hair

Real Hair: Straight

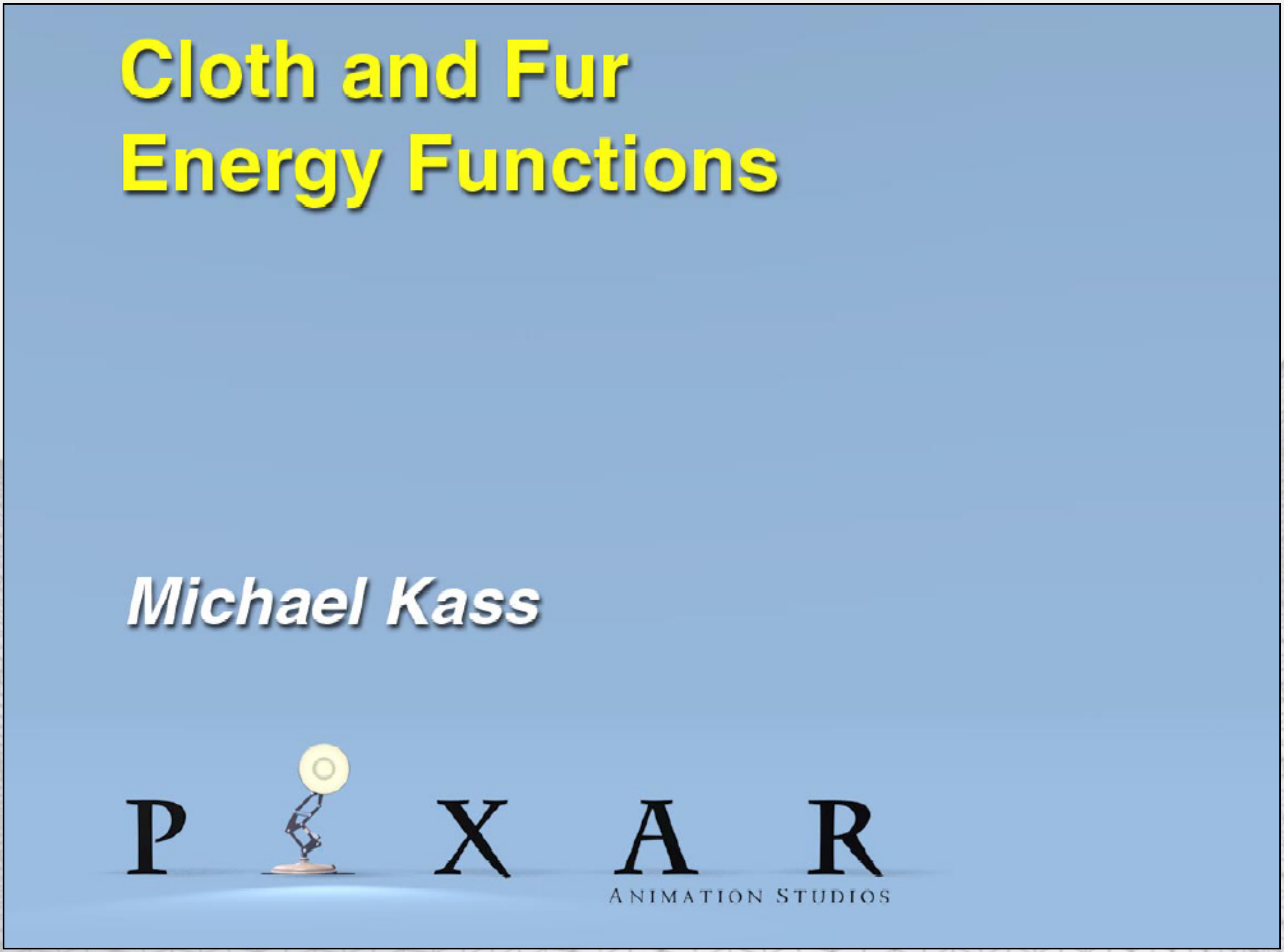
Long
smooth hair

Real Hair



- Typical human head has 150k-200k individual strands.
- Dynamics not well understood.
  - Subject still open to debate.

Recall...



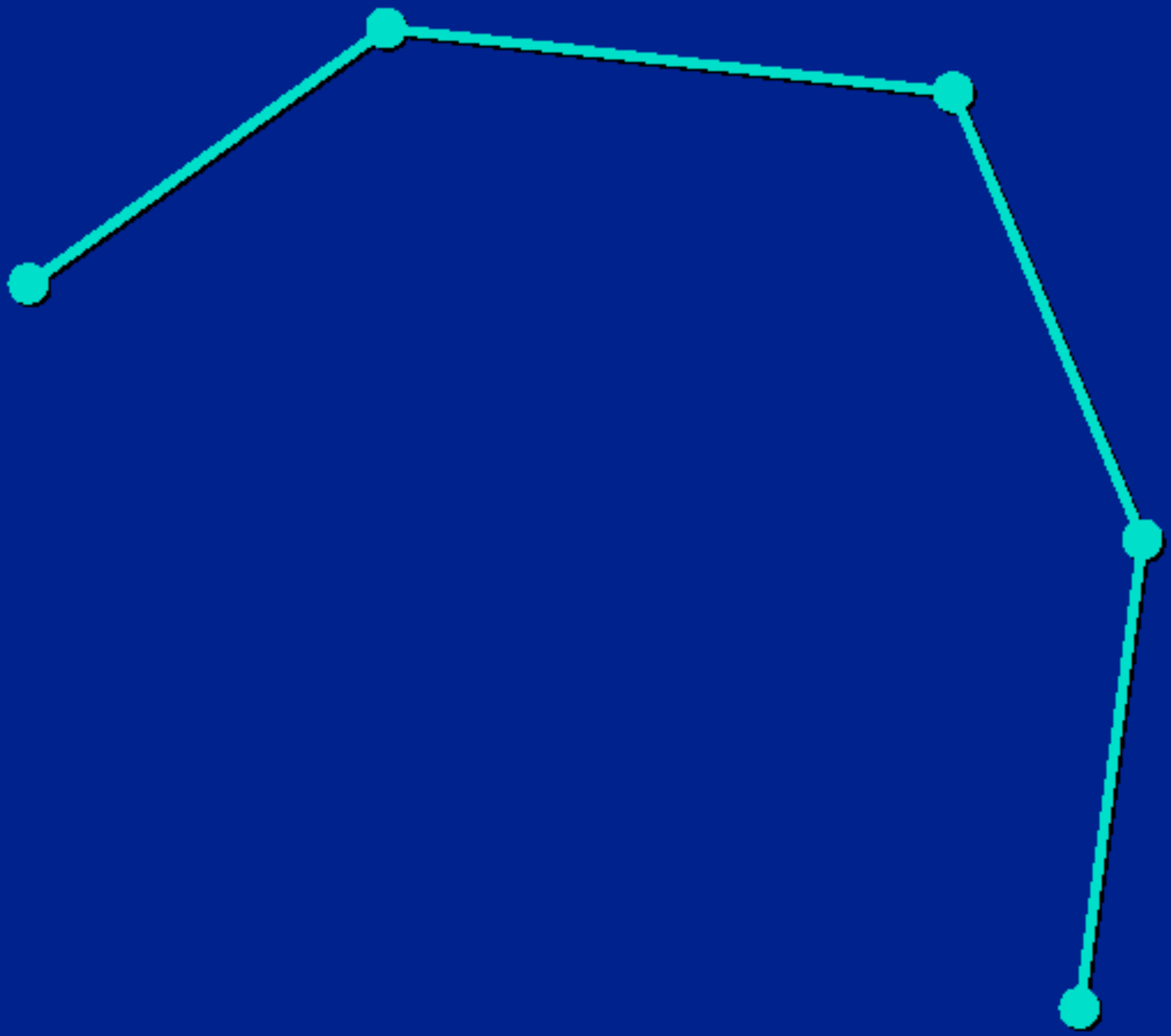
Cloth and Fur
Energy Functions

Michael Kass

PIXAR
ANIMATION STUDIOS

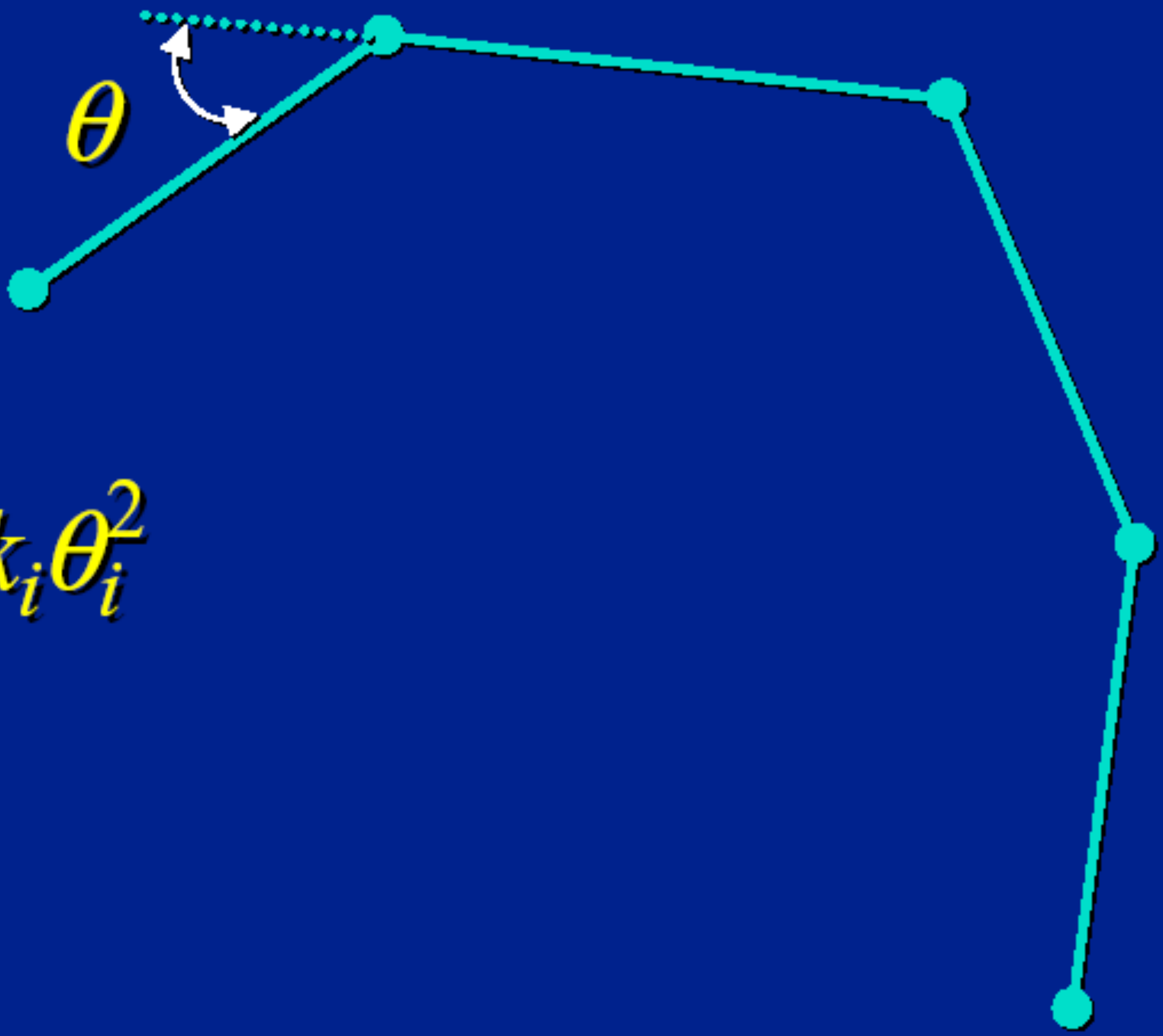
Hair Model

Limp hair: Just a set of springs.



Hair Model

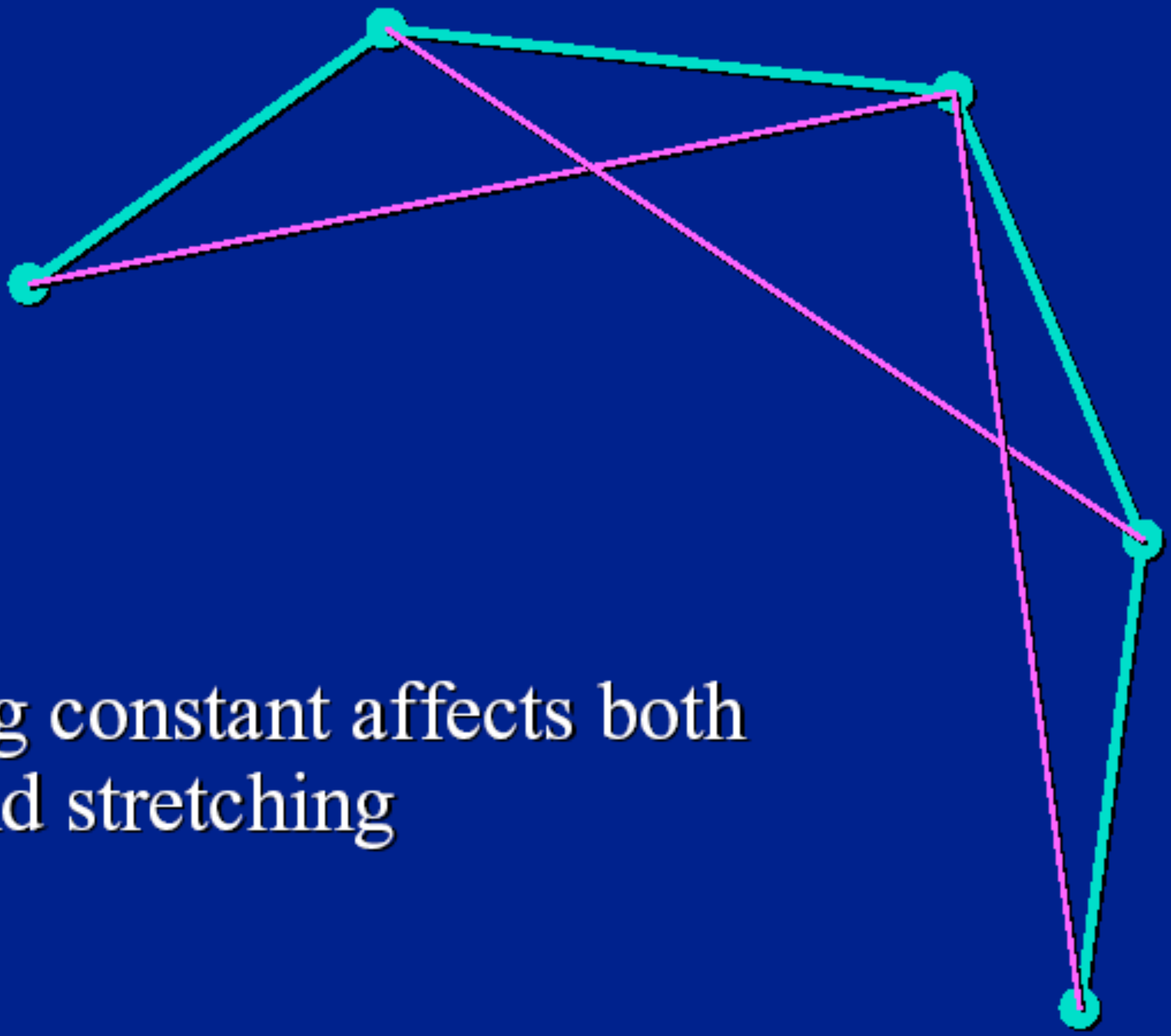
Add body: Angular Springs



$$E = \frac{1}{2}\sum_i k_i\theta_i^2$$

Hair Model

Alternative: More Linear Springs



Difficulty:

Each spring constant affects both bending and stretching

Problems

The linear spring model is very simple but has several problems:

- Not length preserving.
- No torsion forces (twist).

Problems

The linear spring model is very simple but has several problems:

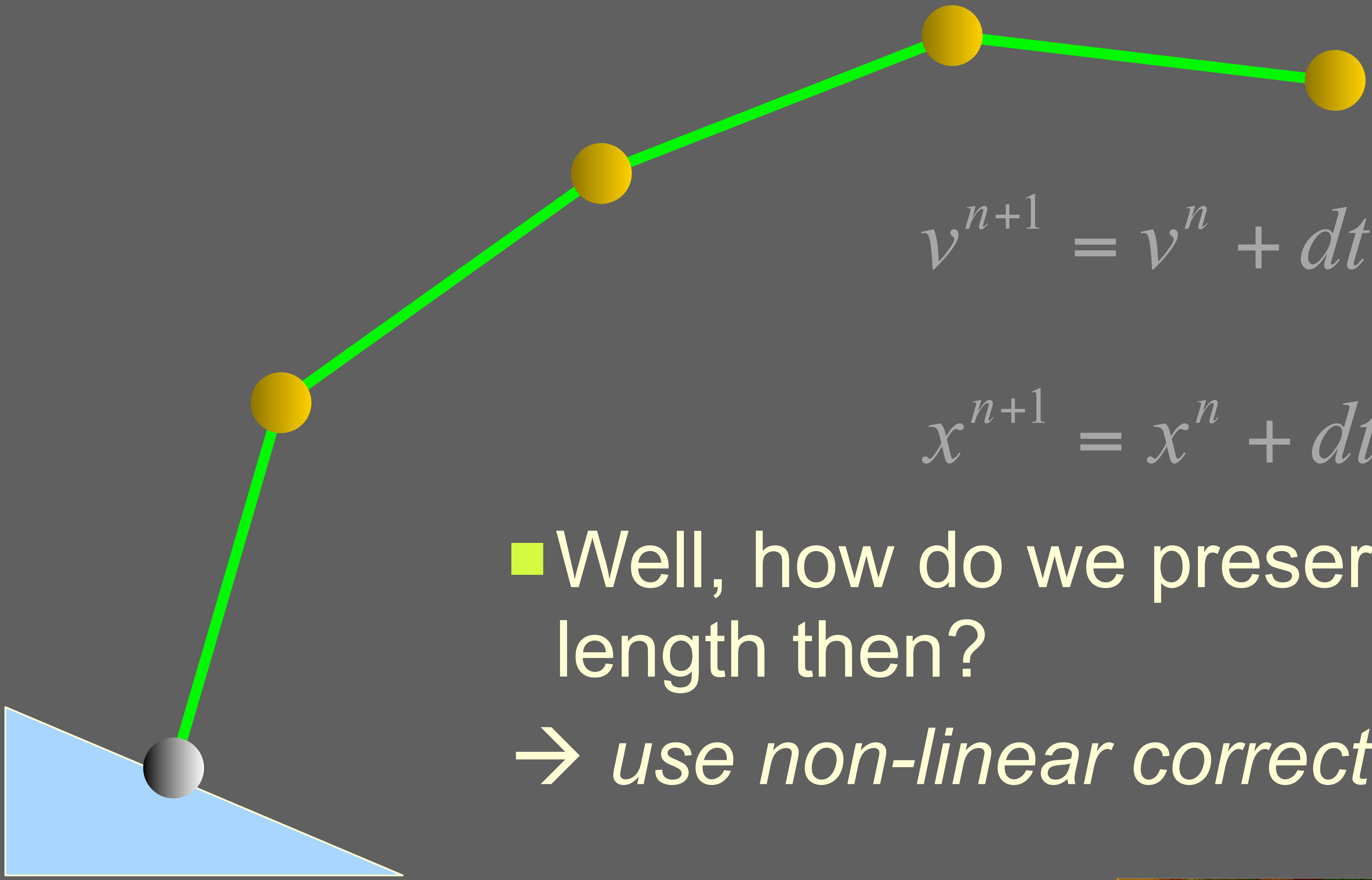
- Not length preserving.
- No torsion forces (twist).

Hair simulation in Rhythm and Hues - The Chronicles of Narnia

Tae-Yong Kim
Rhythm and Hues Studios

Rhythm + Hues Studios

k Is infinity!



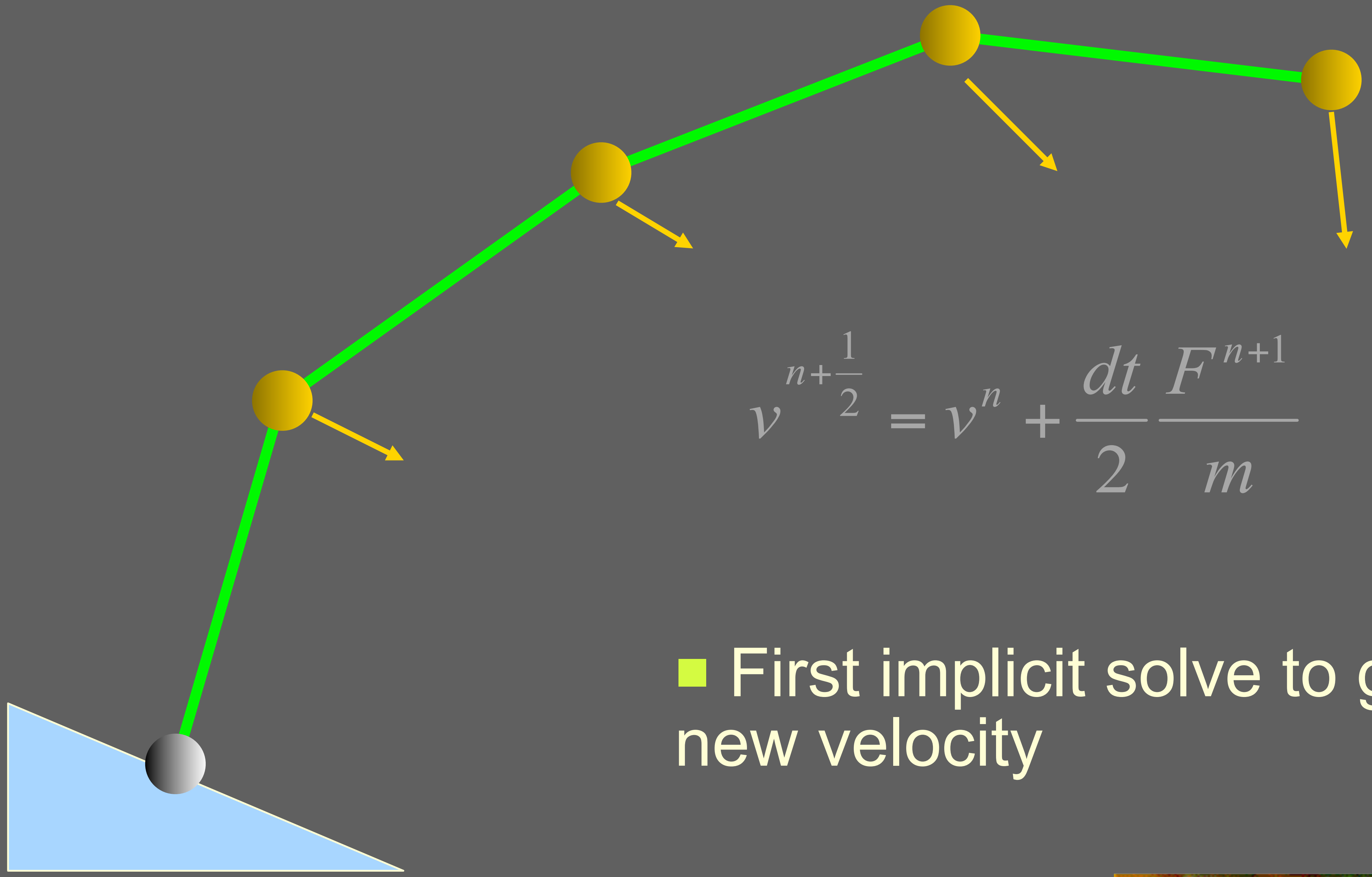
$$v^{n+1} = v^n + dt\frac{F}{m}$$

$$x^{n+1} = x^n + dtv^{n+1}$$

- Well, how do we preserve length then?

→ *use non-linear correction*

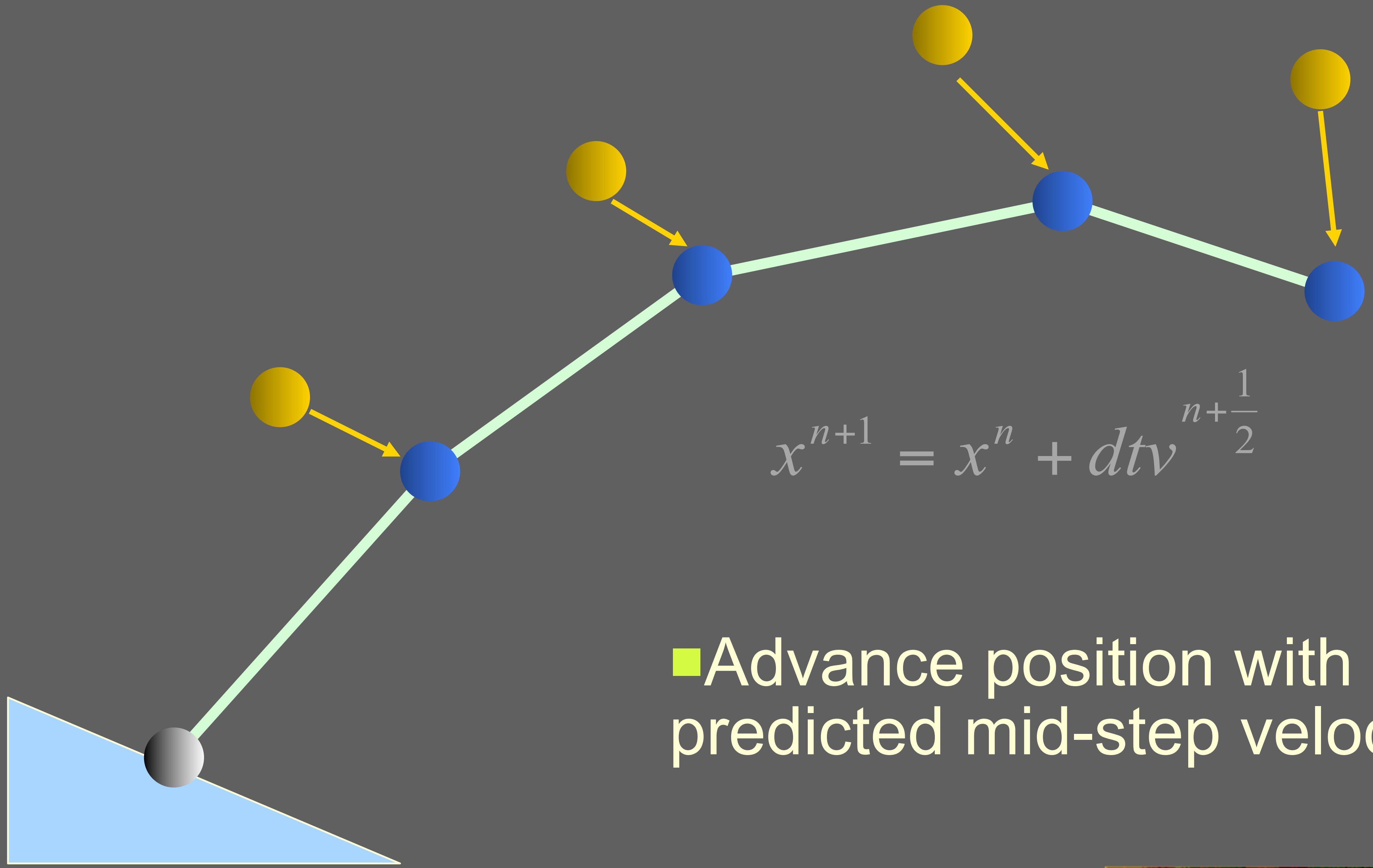
1.First pass-implicit integration



$$v^{n+\frac{1}{2}} = v^n + \frac{dt}{2}\frac{F^{n+1}}{m}$$

- First implicit solve to get new velocity

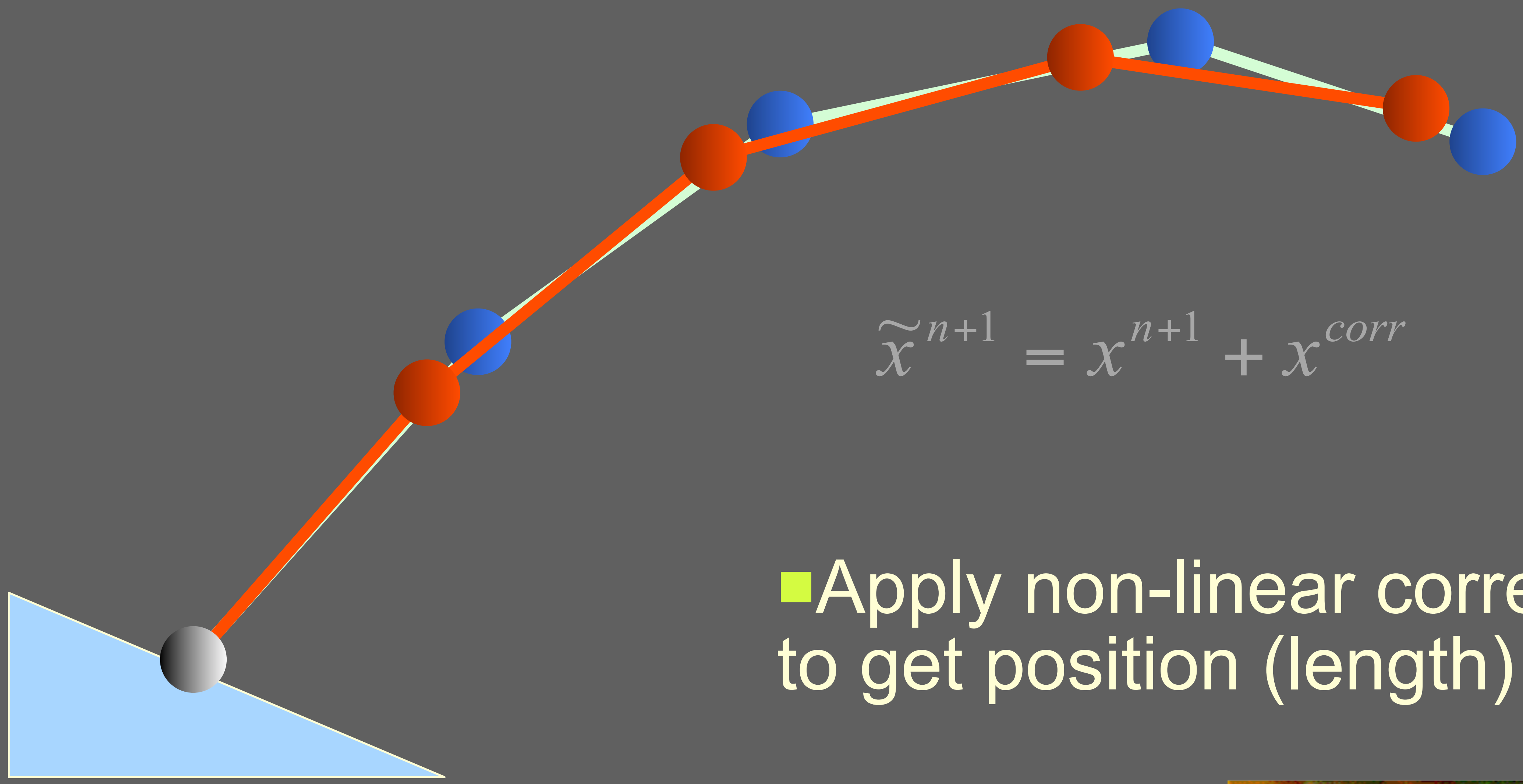
2.First pass-implicit integration



$$x^{n+1} = x^n + dt v^{n+\frac{1}{2}}$$

- Advance position with the predicted mid-step velocity

3.Non-linear Correction

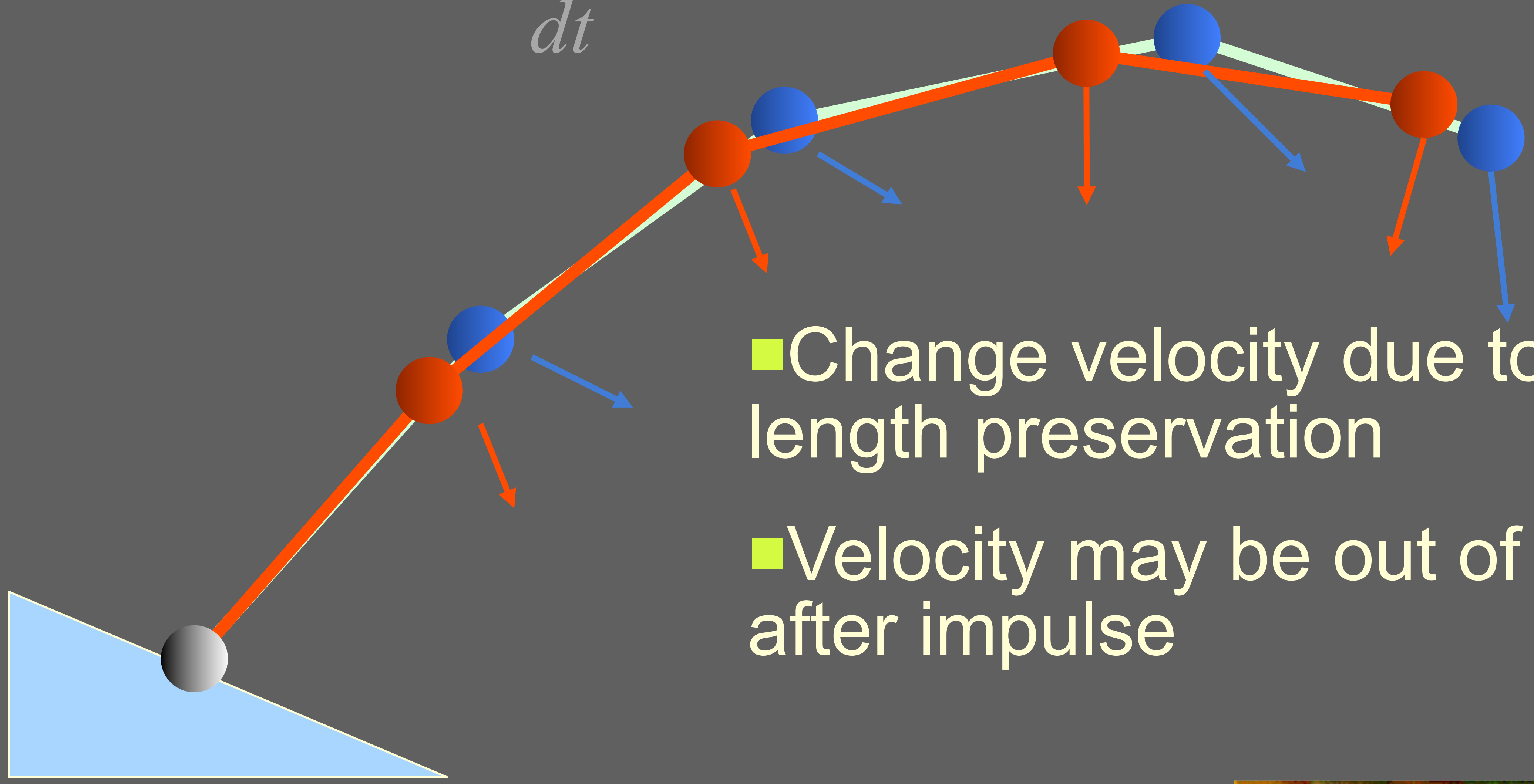


$$\tilde{x}^{n+1} = x^{n+1} + x^{corr}$$

- Apply non-linear corrector to get position (length) right

4.Impulse

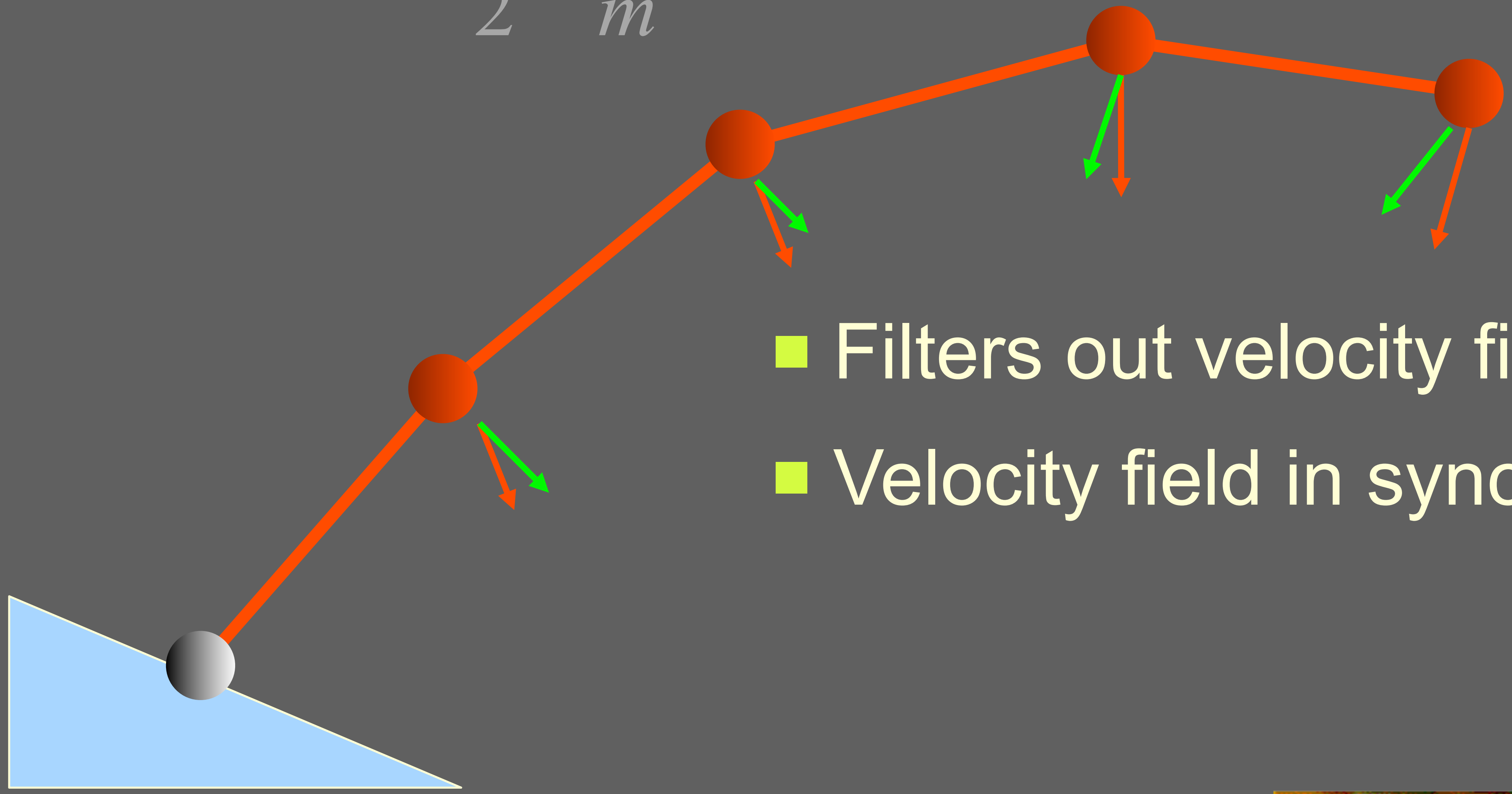
$$\tilde{v}^{n+\frac{1}{2}} = v^{n+\frac{1}{2}} + \frac{x^{corr}}{dt}$$



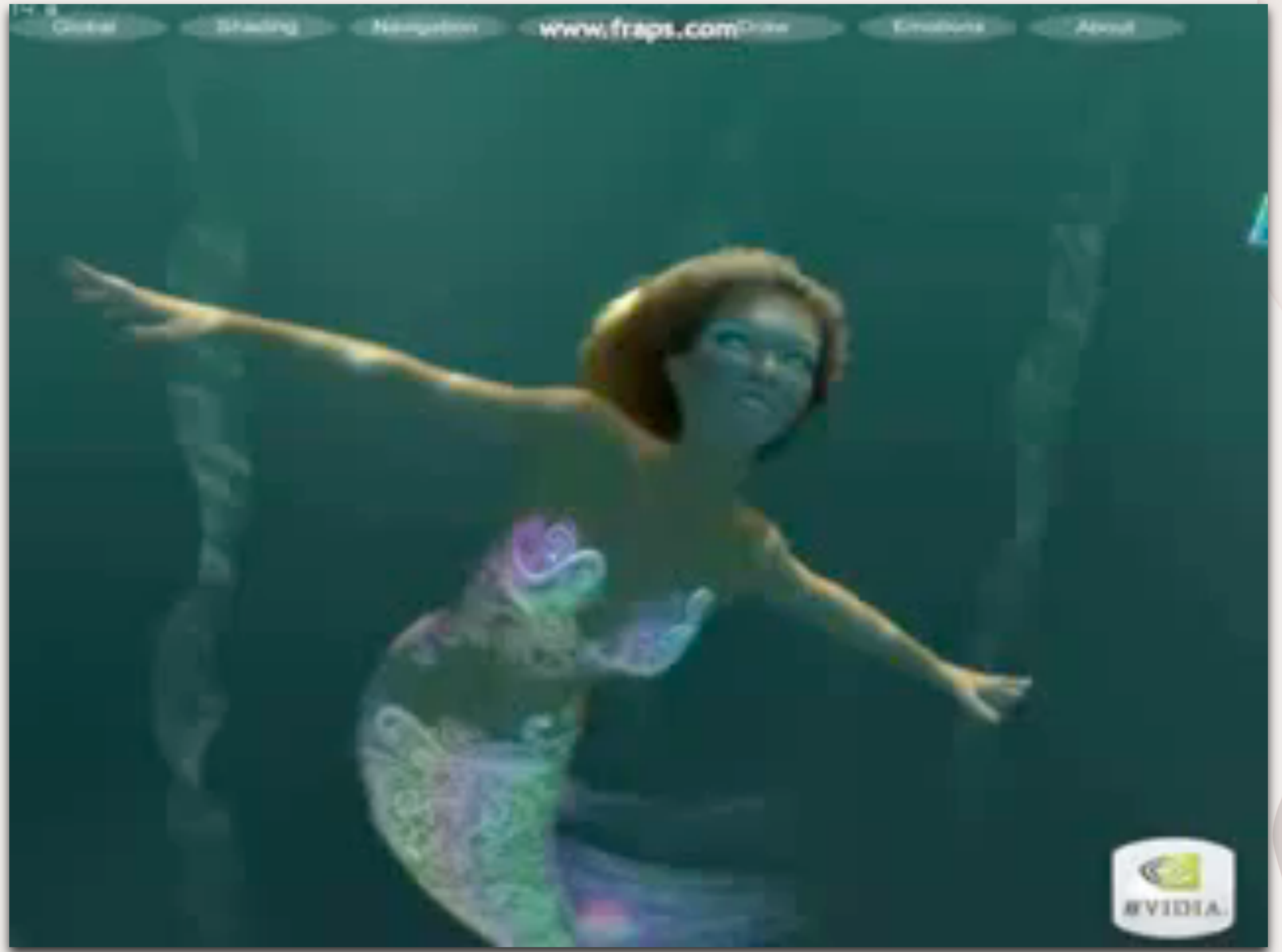
- Change velocity due to length preservation
- Velocity may be out of sync after impulse

5.Second implicit integration

$$v^{n+1} = \tilde{v}^{n+\frac{1}{2}} + \frac{dt}{2}\frac{F^{n+1}}{m}$$



- Filters out velocity field
- Velocity field in sync again



nVidia 'nalu' demo - http://www.youtube.com/watch?v=e0m0o6lbmeM

Problems

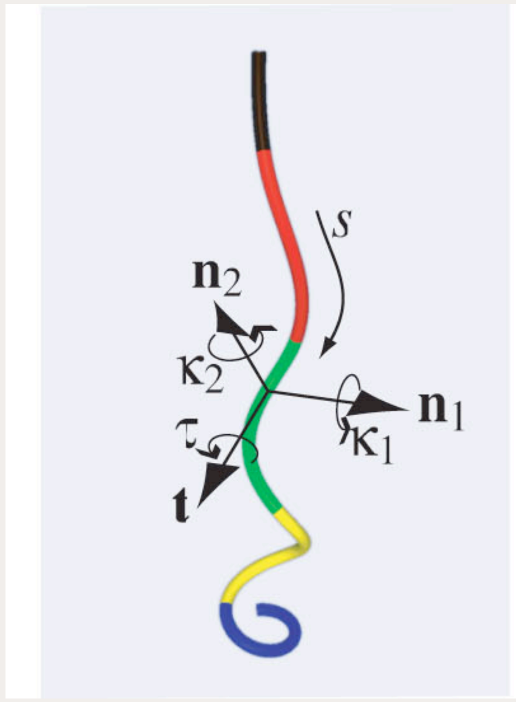
The linear spring model is very simple but has several problems:

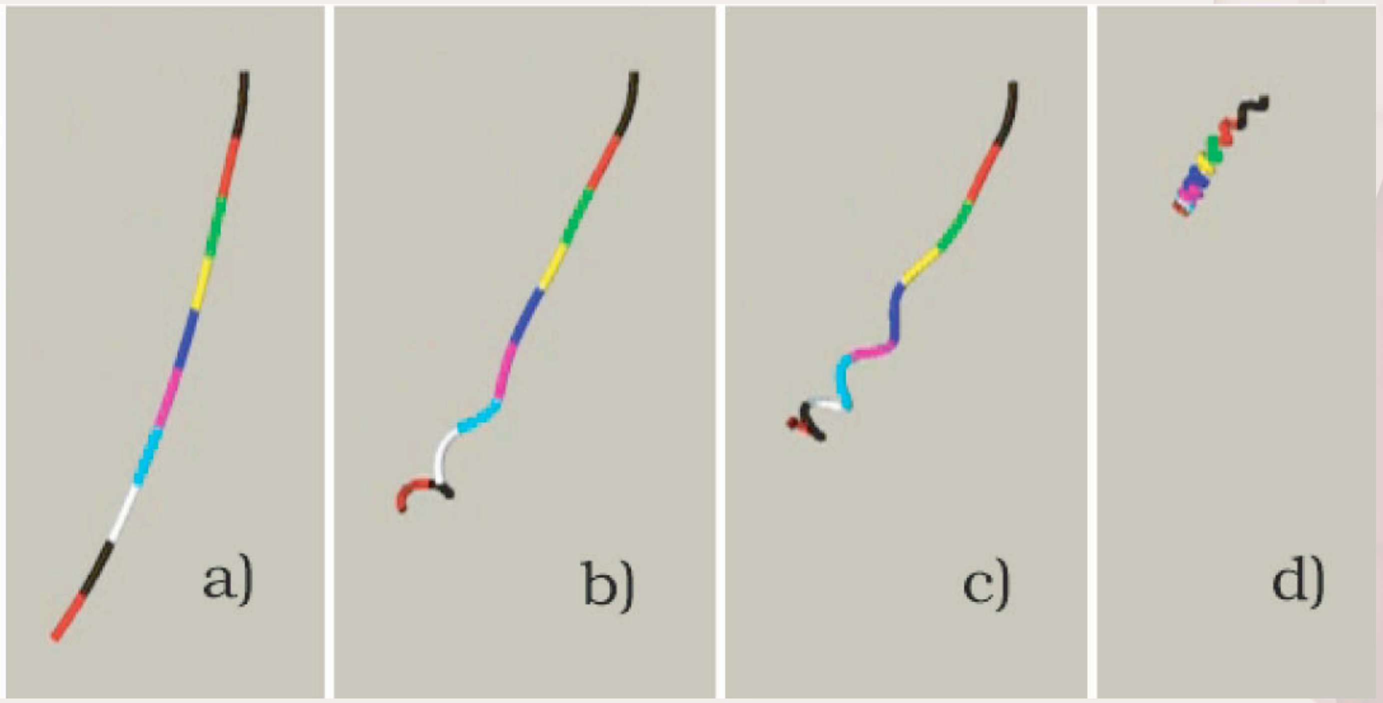
- Not length preserving.
- No torsion forces (twist).

Super Helices

Why just use straight rods?

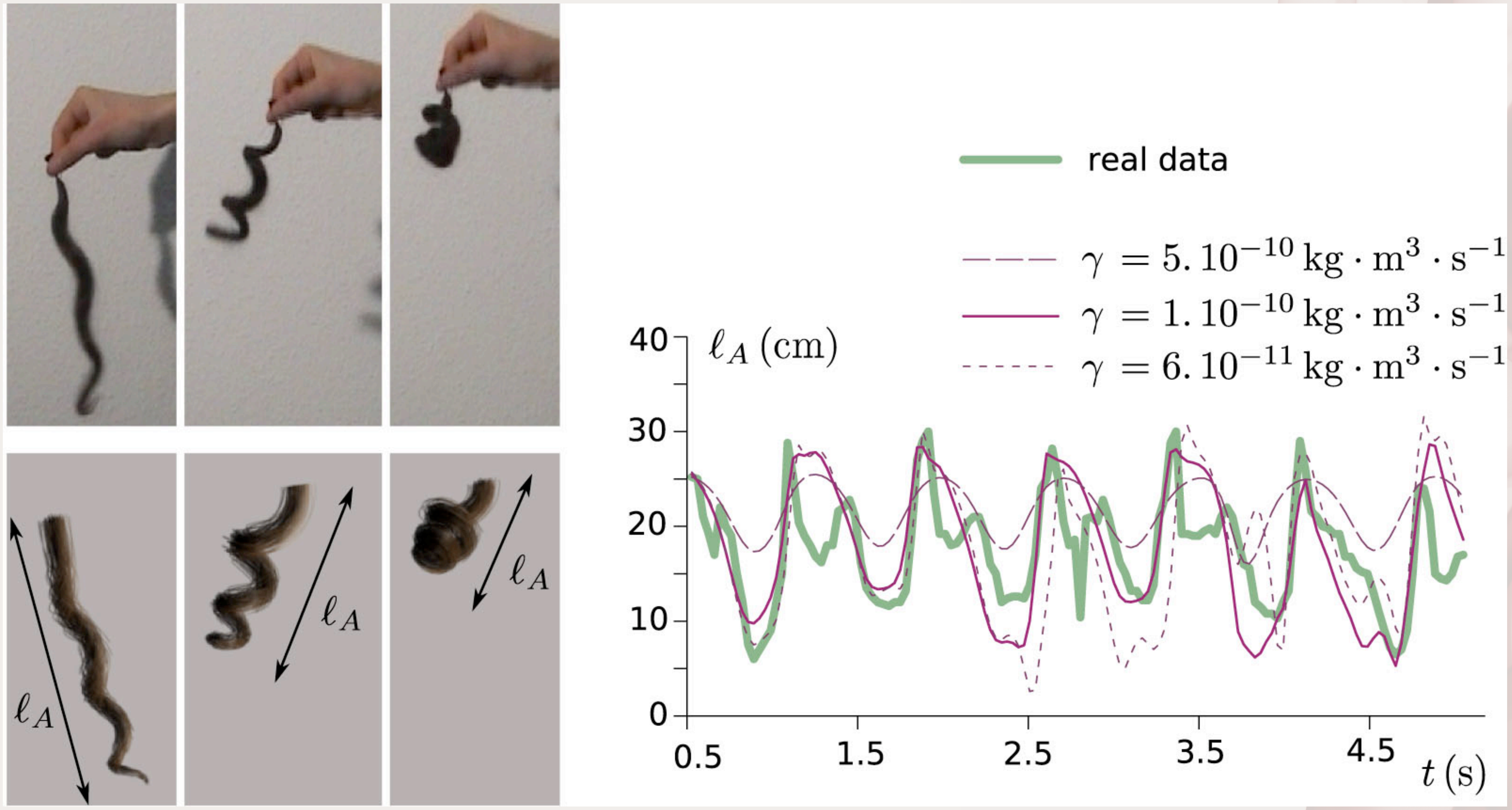
Super Helices



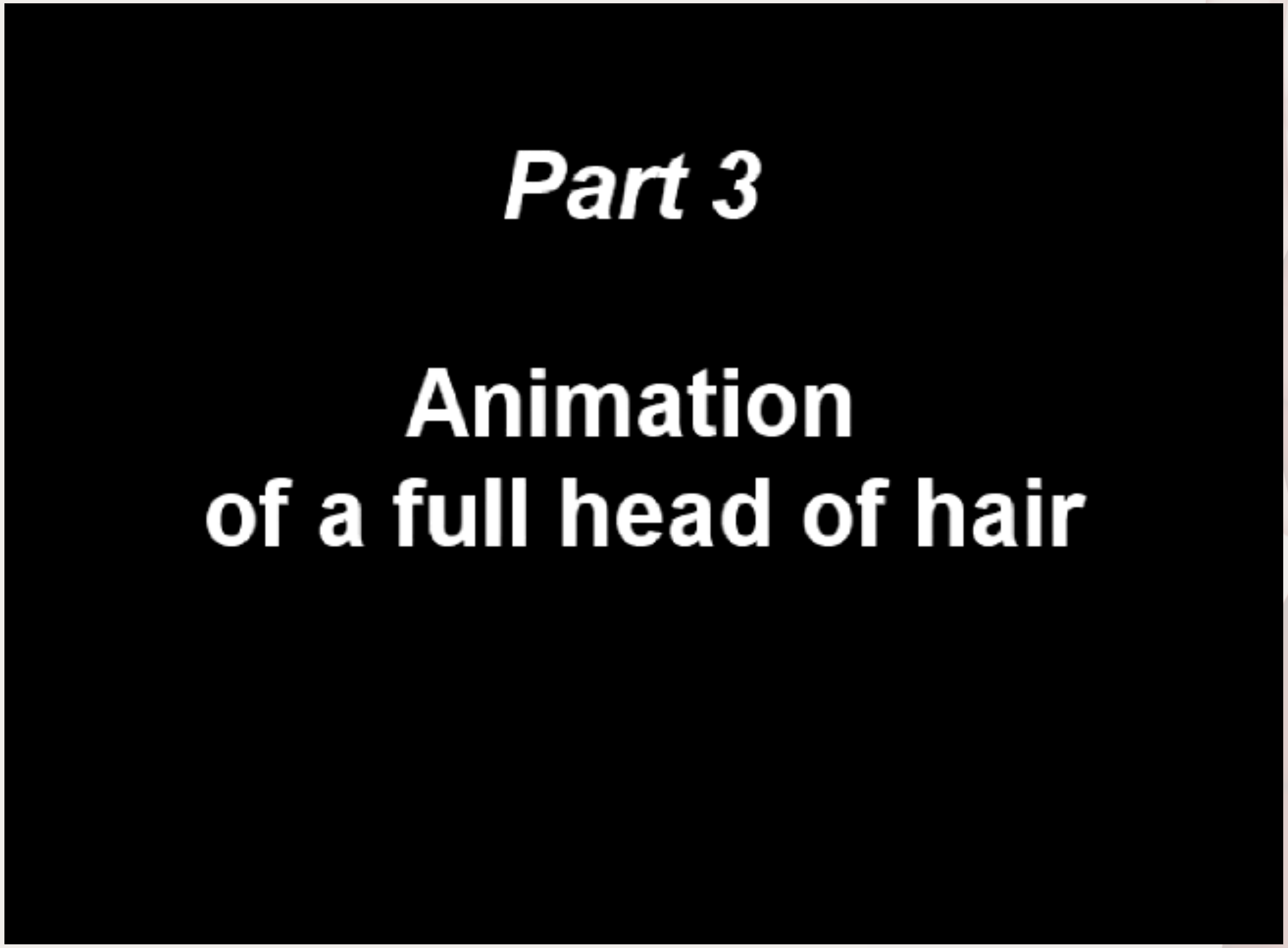


$$\mathbb{M}[s,\mathbf{q}]\cdot\ddot{\mathbf{q}}+\mathbb{K}\cdot(\mathbf{q}-\mathbf{q}^{\mathrm{n}})=\mathbf{A}[t,\mathbf{q},\dot{\mathbf{q}}]+\int_0^L \mathbf{J}_{iQ}[s,\mathbf{q},t]\cdot\mathbf{F}^{\mathrm{i}}(s,t)\,\mathrm{d}s.$$

Super Helices



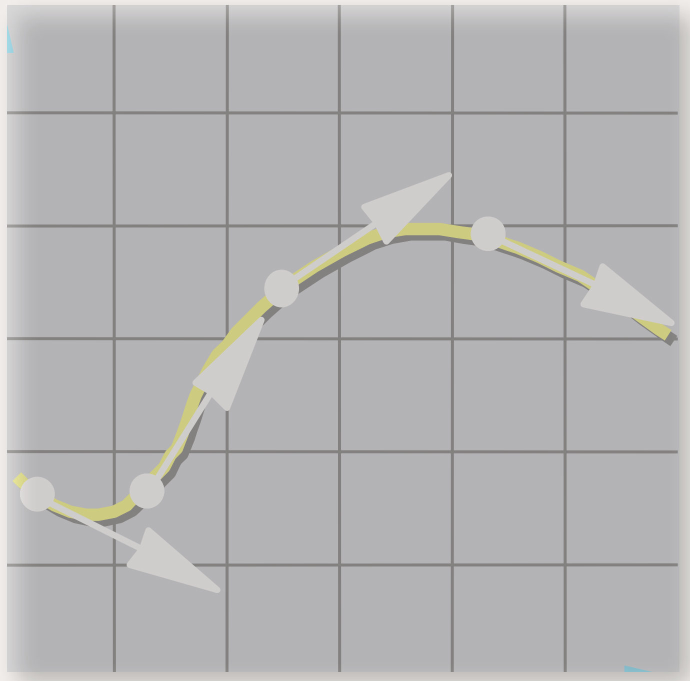
Super Helices



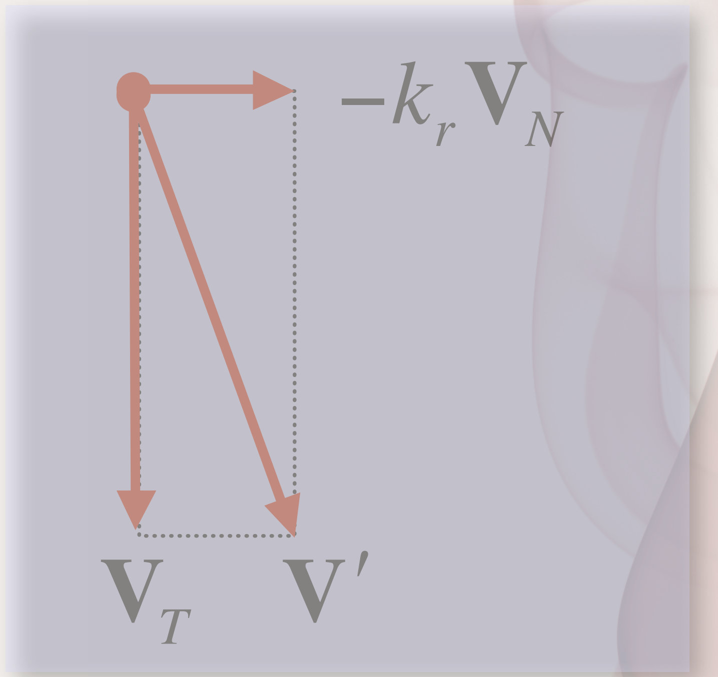
Part 3

Animation
of a full head of hair

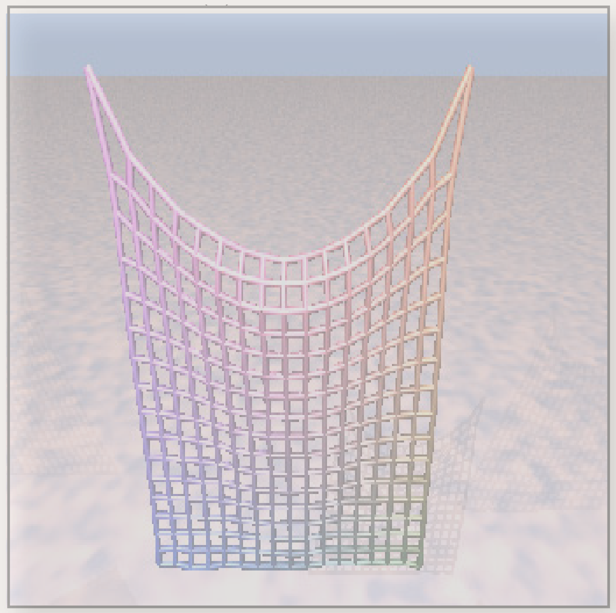
Overview



DiffEQ Review



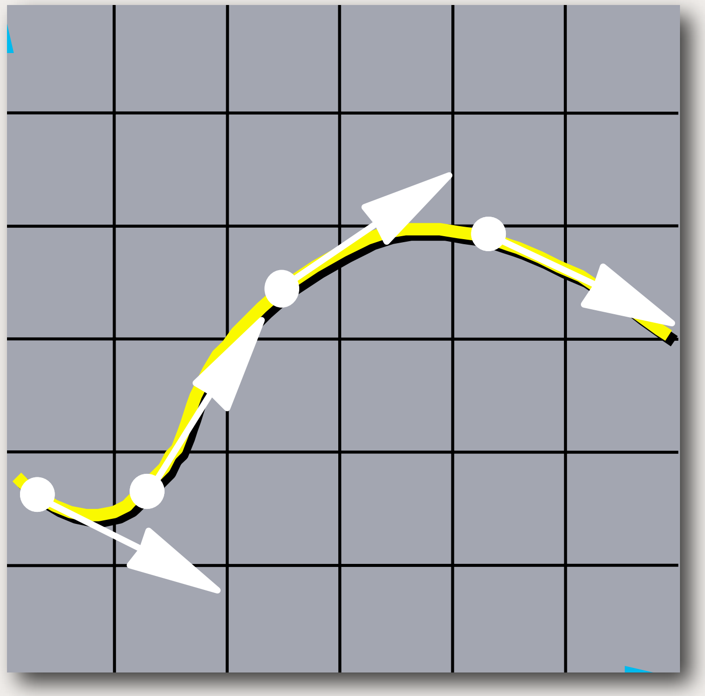
Particle Dynamics



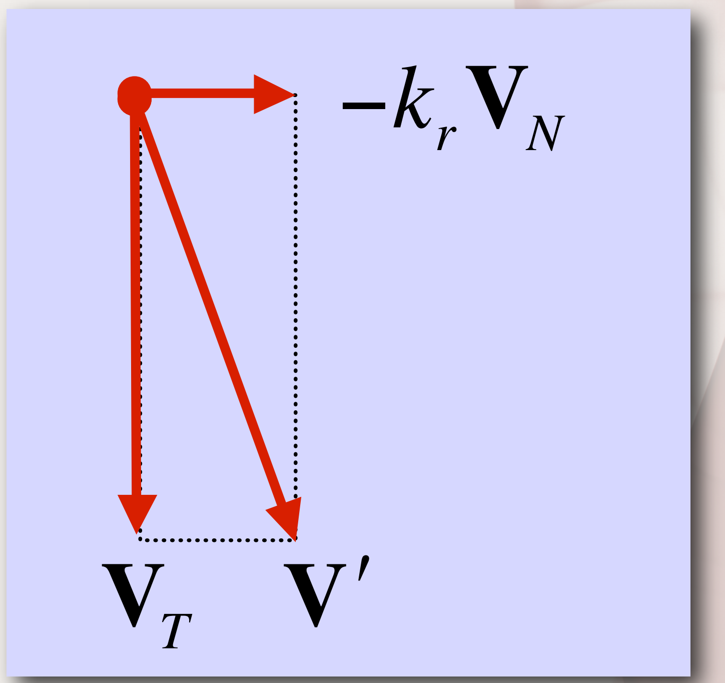
Cloth

Hair

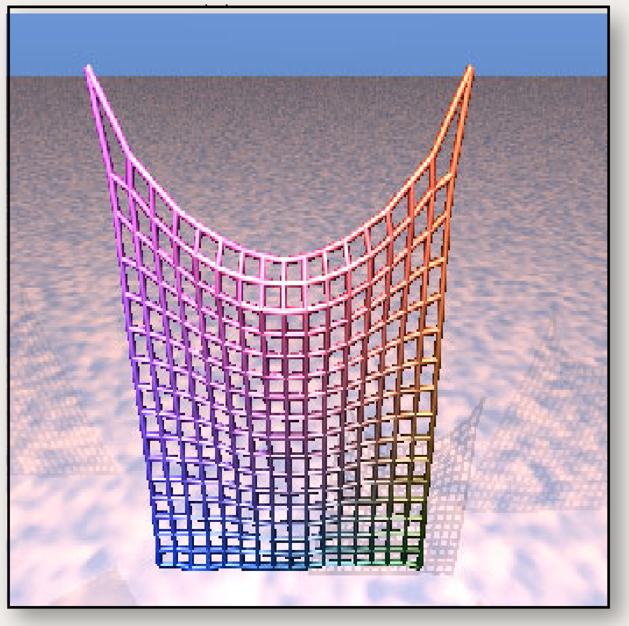
Overview



DiffEQ Review



Particle Dynamics



Cloth

Hair