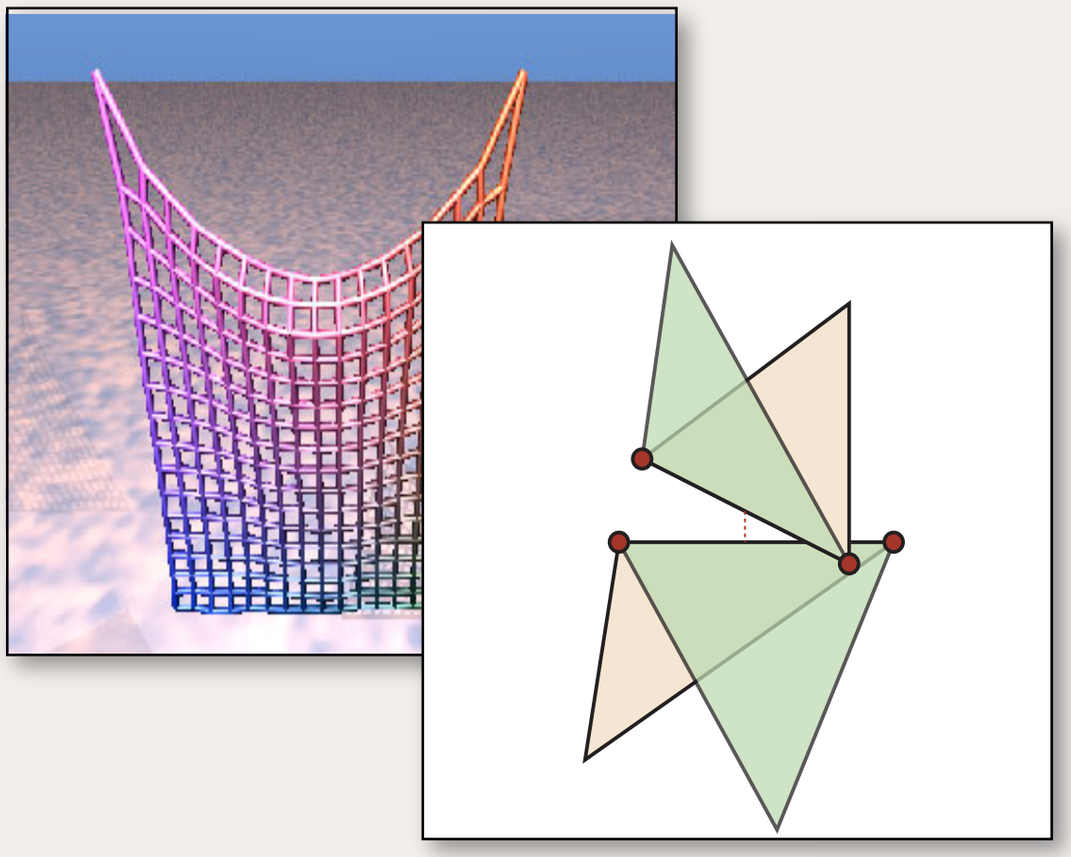
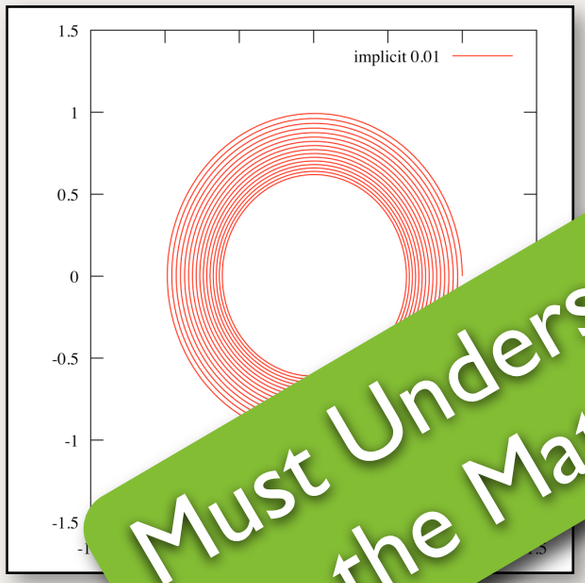
Implicit Integration and Cloth



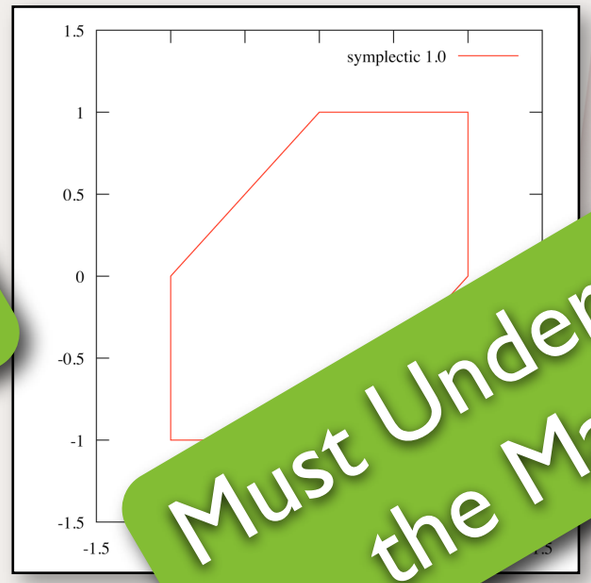
Adrien Treuille

Overview



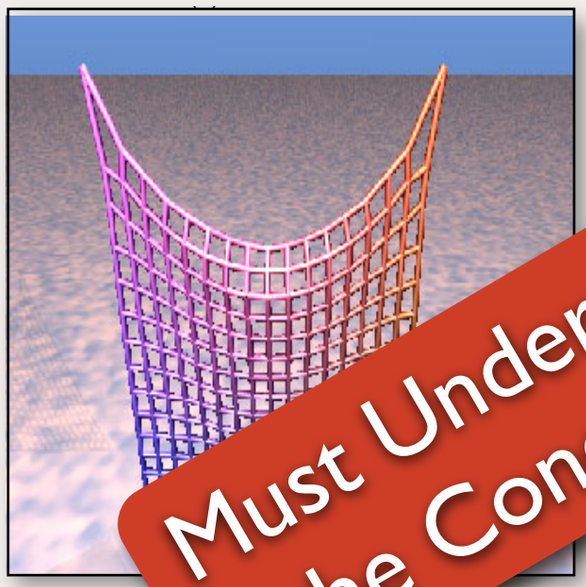
Must Understand the Math!

Impli gration



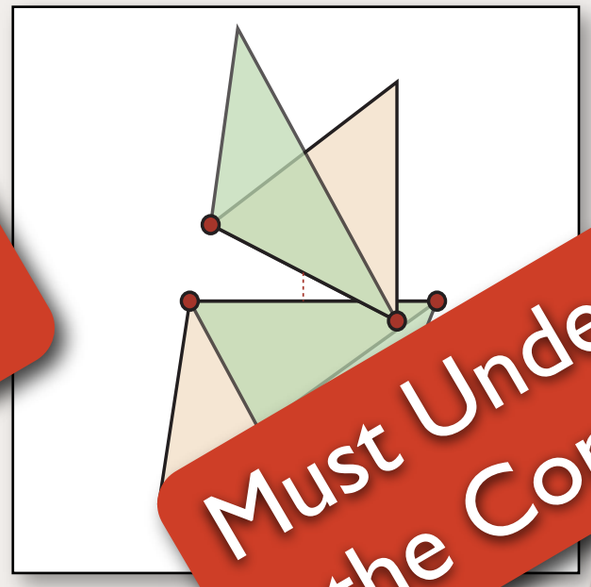
Must Understand the Math!

Symple egration



Must Understand the Concepts!

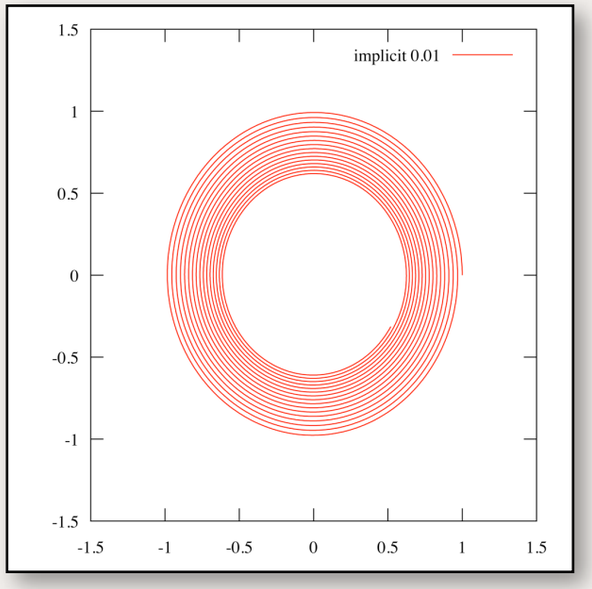
Cloth tion



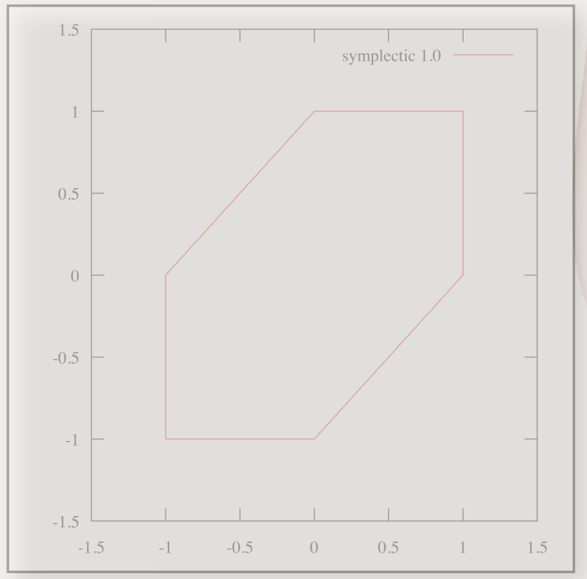
Must Understand the Concepts!

Cloth ons

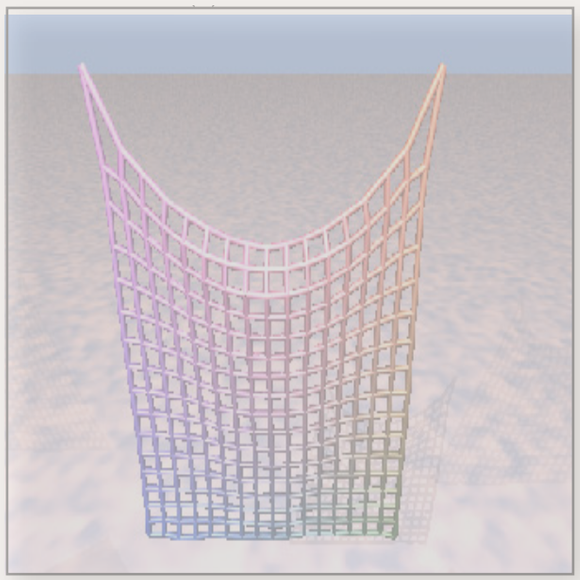
Overview



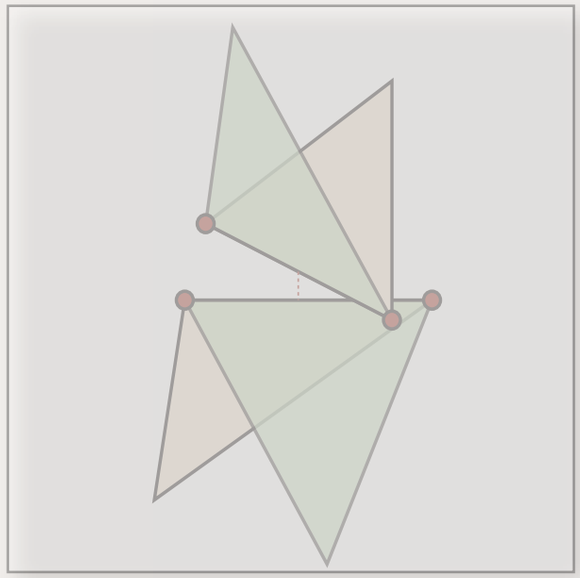
Implicit Integration



Symplectic Integration



Cloth Simulation



Cloth Collisions

Implicit Integration

Implicit Methods

David Baraff

PIXAR
ANIMATION STUDIOS

“Give me Stability
or
Give me Death”

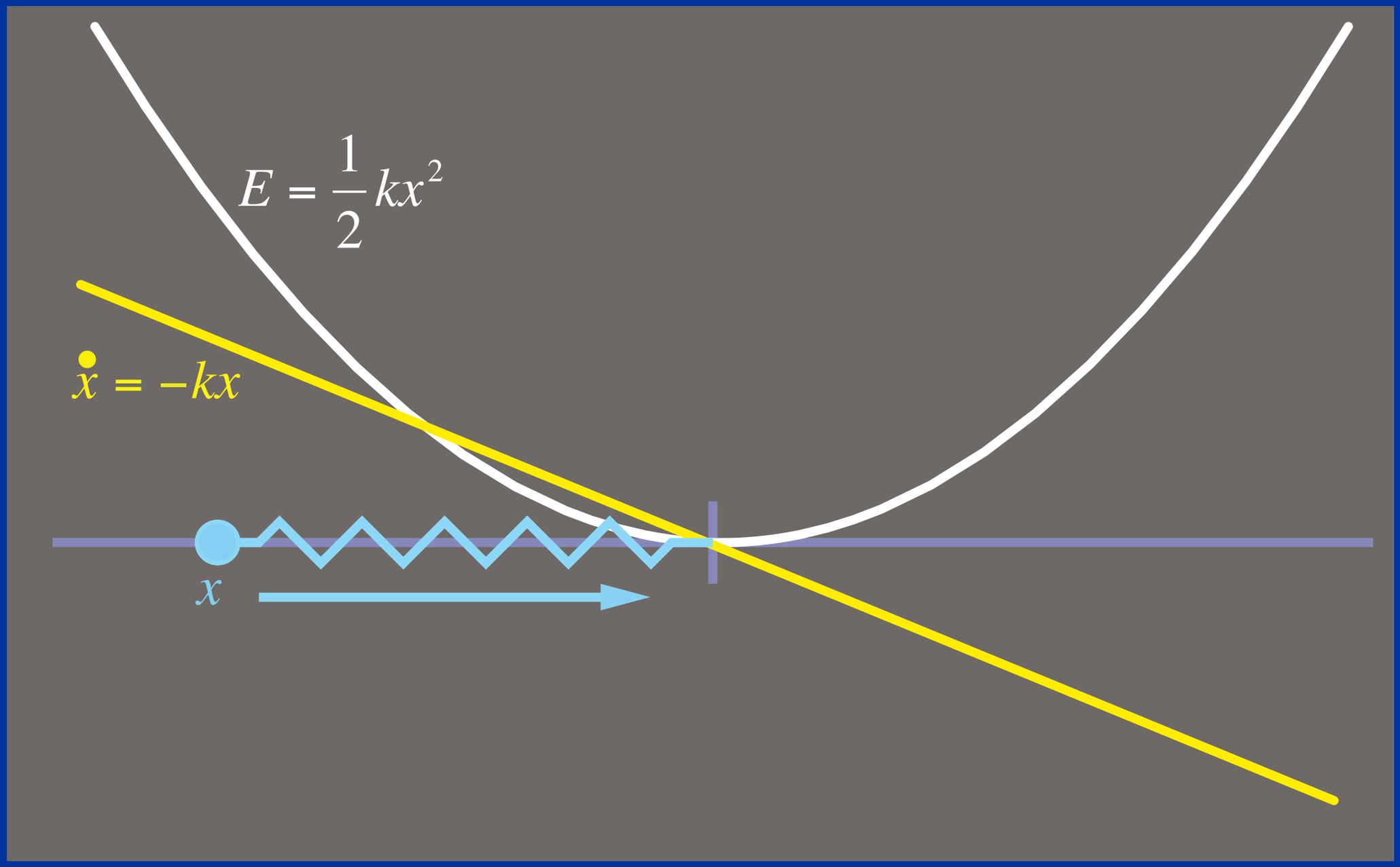
— Baraff’s other motto

stability is all stability is all stability is all

- If your step size is too big, your simulation blows up. It isn't pretty.
- Sometimes you have to make the step size so small that you never get anyplace.
- Nasty cases: cloth, constrained systems.

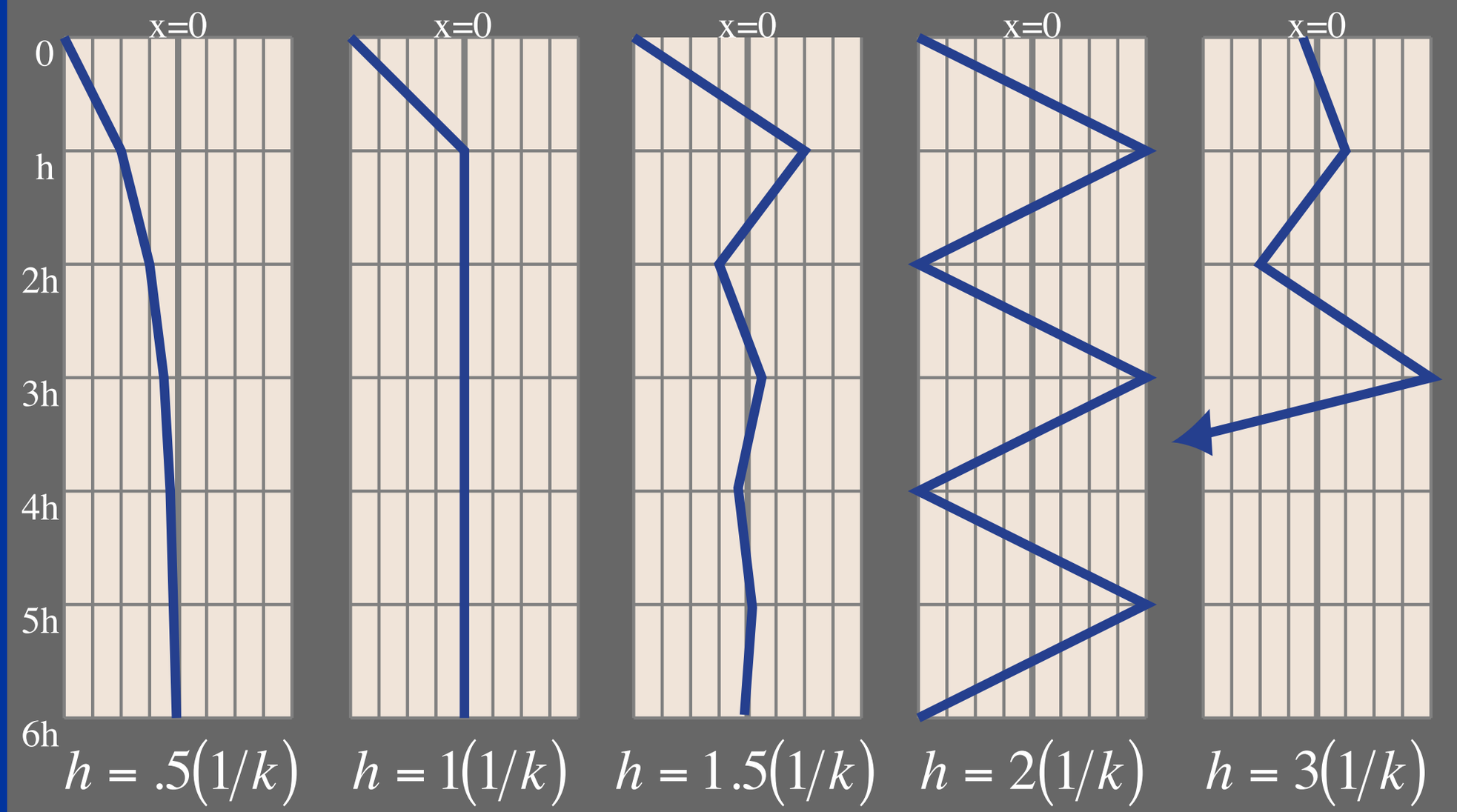
A very simple equation

A 1-D particle governed by $\dot{x} = -kx$ where k is a stiffness constant.



Euler's method has a speed limit

$$x = -kx \qquad \Delta x = -hkx$$



$h > 1/k$: oscillate. $h > 2/k$: explode!

Stiff Equations

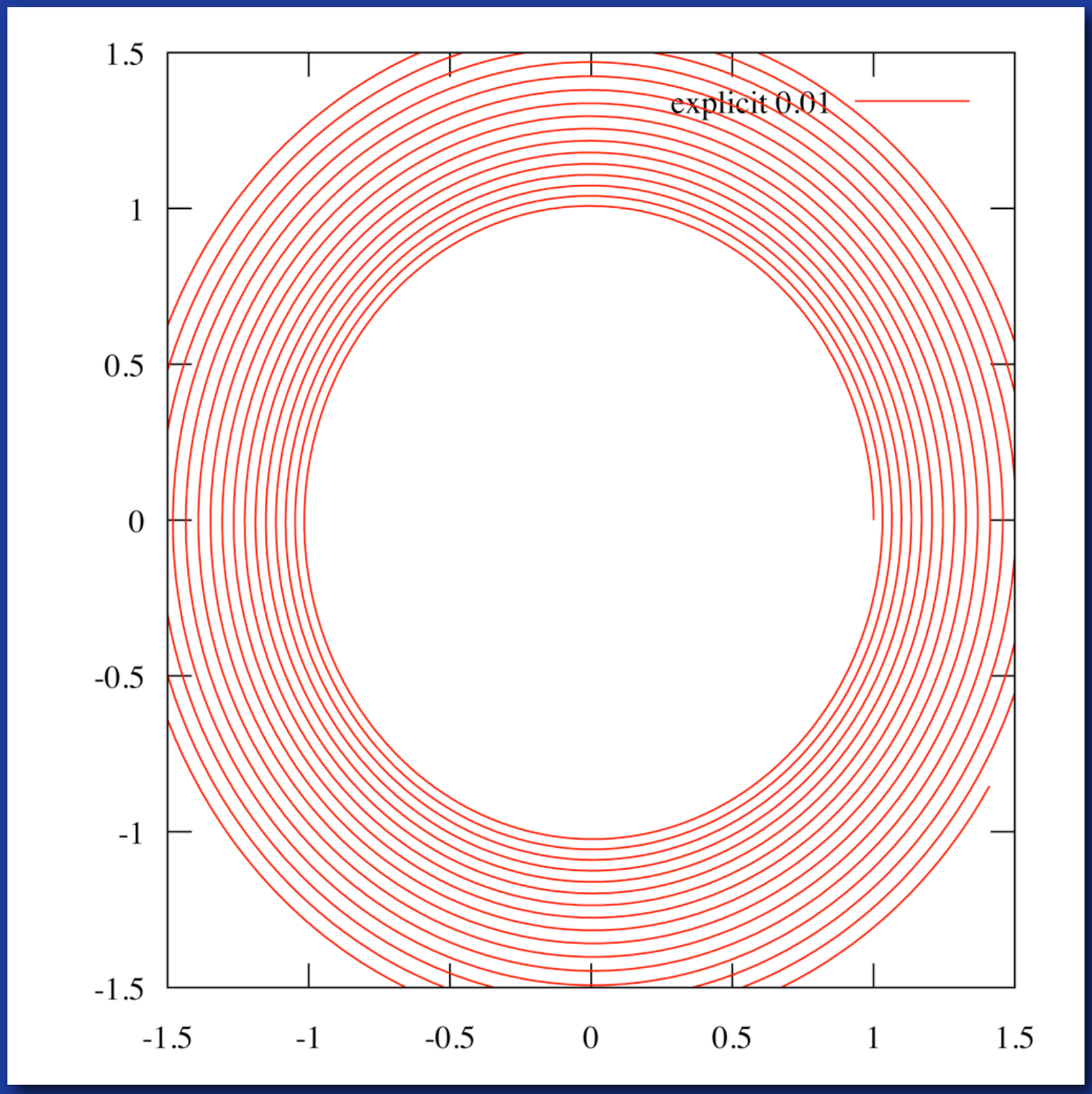
- In more complex systems, step size is limited by the largest k. One stiff spring can screw it up for everyone else.
- Systems that have some big k's mixed in are called stiff systems.

Accuracy

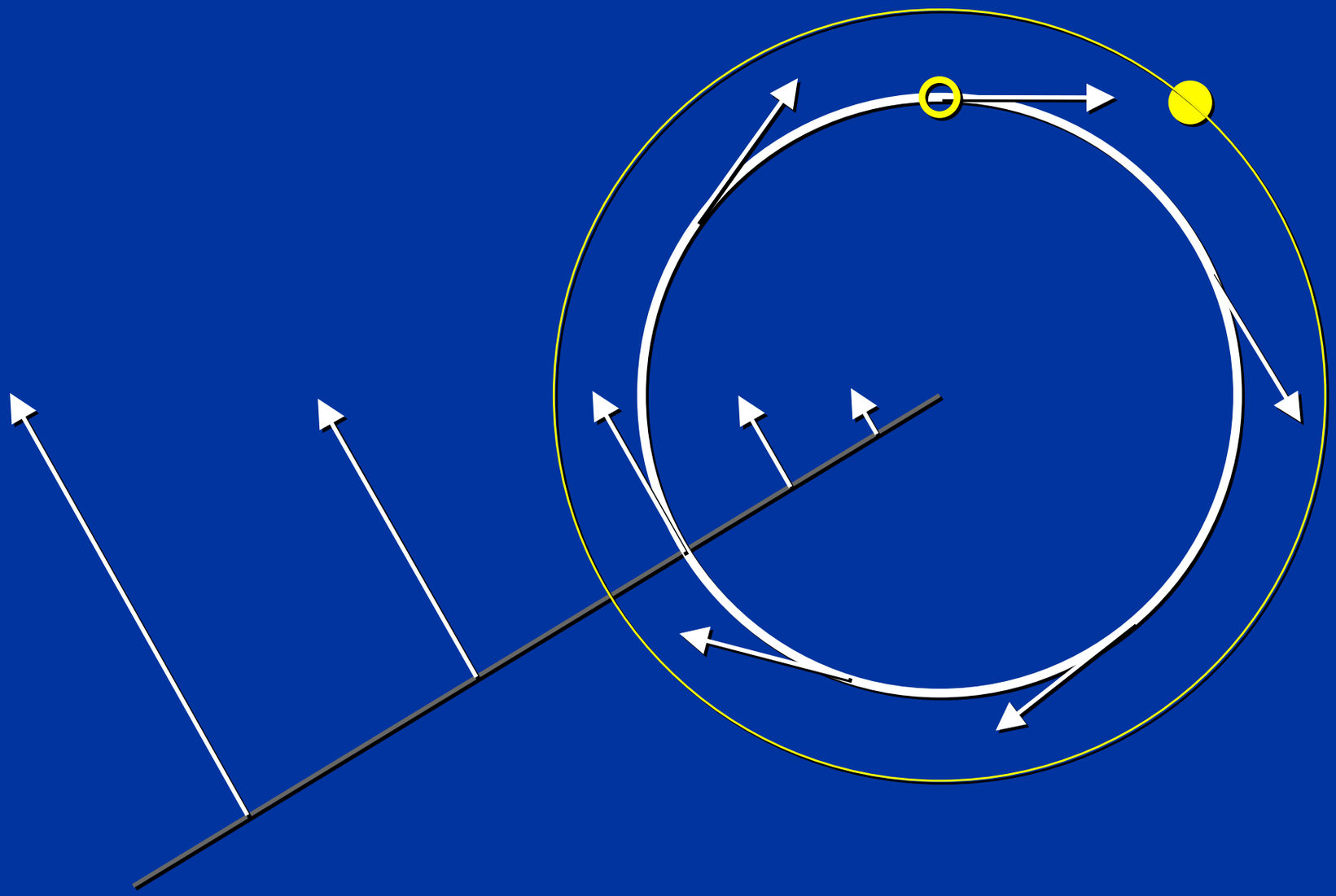
Consider the equation:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{x}$$

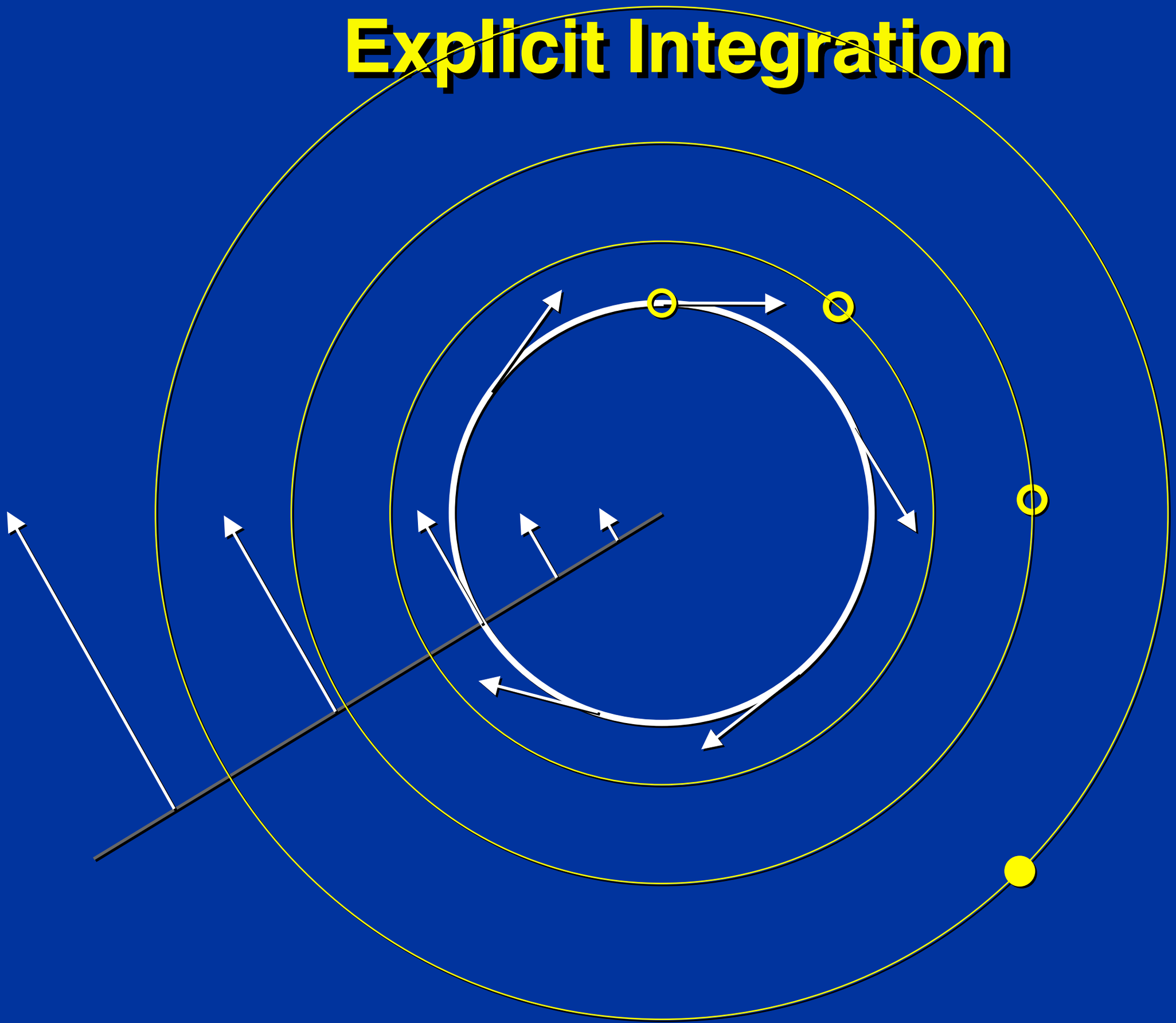
What do the integral curves look like?


explicit 0.01
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

Explicit Integration



Explicit Integration



(Explicit) Euler Method

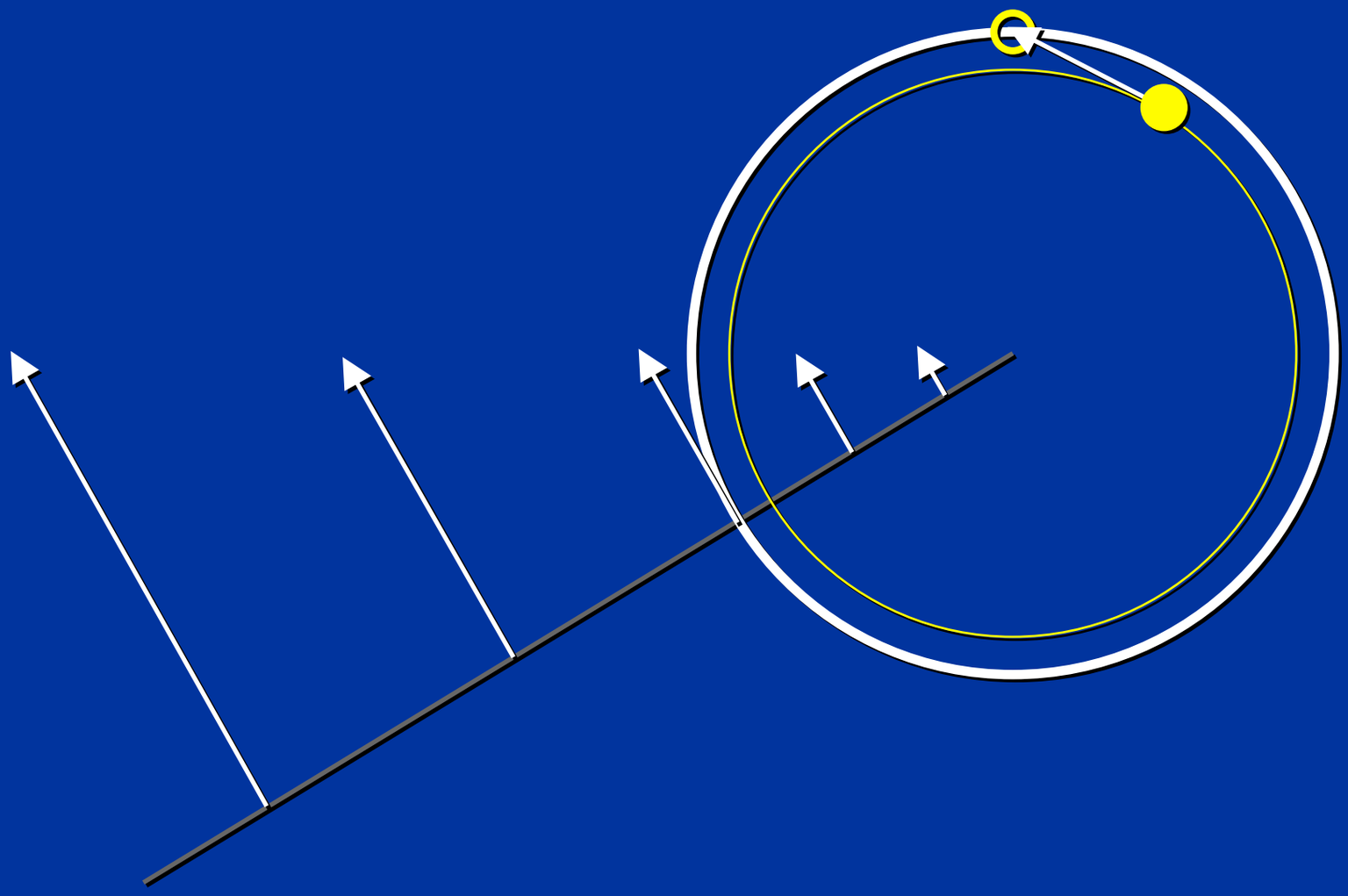
$$x(t_0 + h) = x(t_0) + h\dot{x}(t_0)$$

Implicit Euler Method

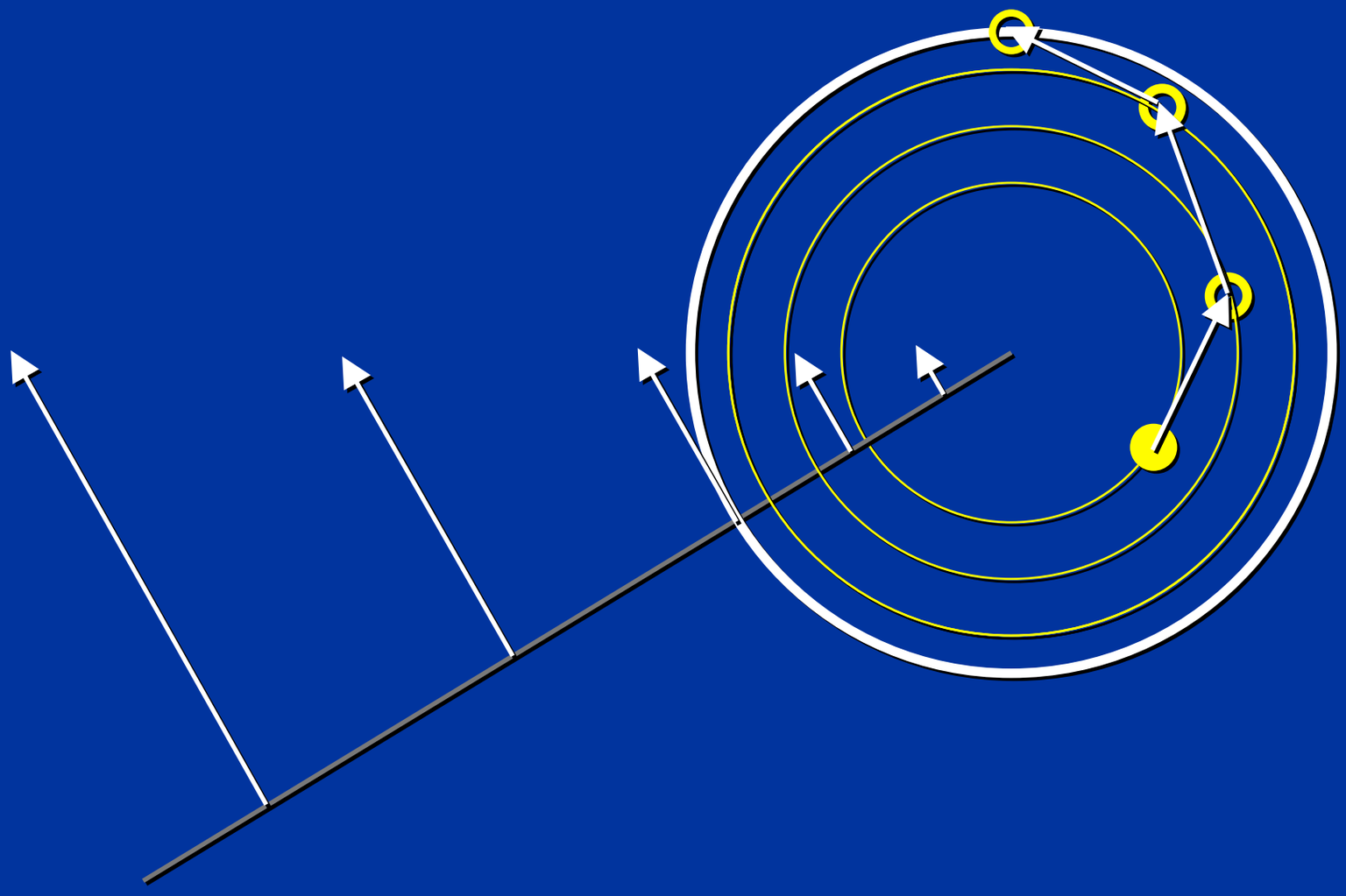
$$x(t_0 + h) = x(t_0) + h\dot{x}(t_0)$$

$$x(t_0 + h) = x(t_0) + h\dot{x}(t_0 + \Delta t)$$

Implicit Integration



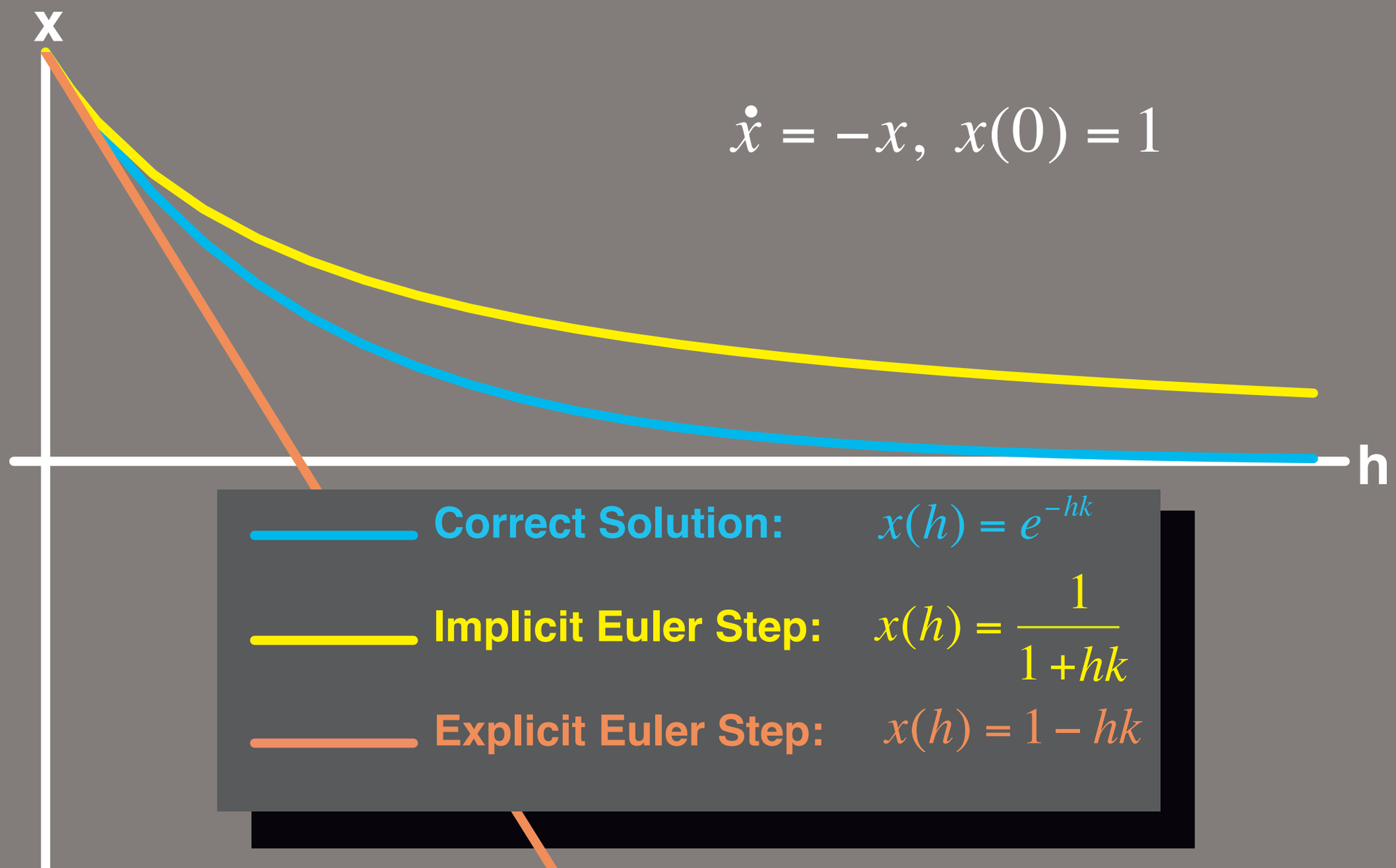
Implicit Integration



Implicit Euler

$$\begin{cases} \dot{x} = -x \\ x(0) = 1 \end{cases}$$

One Step: Implicit vs. Explicit



Large Systems

$$\frac{d}{dt}\mathbf{X}(t) = \dot{\mathbf{X}}(t) = f\left(\mathbf{X}(t)\right)$$

$$\Delta\mathbf{X}(t_0) = h\dot{\mathbf{X}}(t_0 + \Delta t) = h f\left(\mathbf{X}(t_0 + \Delta t)\right)$$

$$= h f\left(\mathbf{X}(t_0) + \Delta\mathbf{X}(t_0)\right)$$

(Linearized) Implicit Integration

$$\dot{\mathbf{X}}(t) = f\left(\mathbf{X}(t)\right)$$

$$\Delta\mathbf{X} = h\, f\left(\mathbf{X}_0 + \Delta\mathbf{X}\right)$$

$$\Delta\mathbf{X} = h\left(f\left(\mathbf{X}_0\right) + \left(\frac{\partial f}{\partial \mathbf{X}}\right)\Delta\mathbf{X}\right)$$

Single-Step Implicit Euler Method

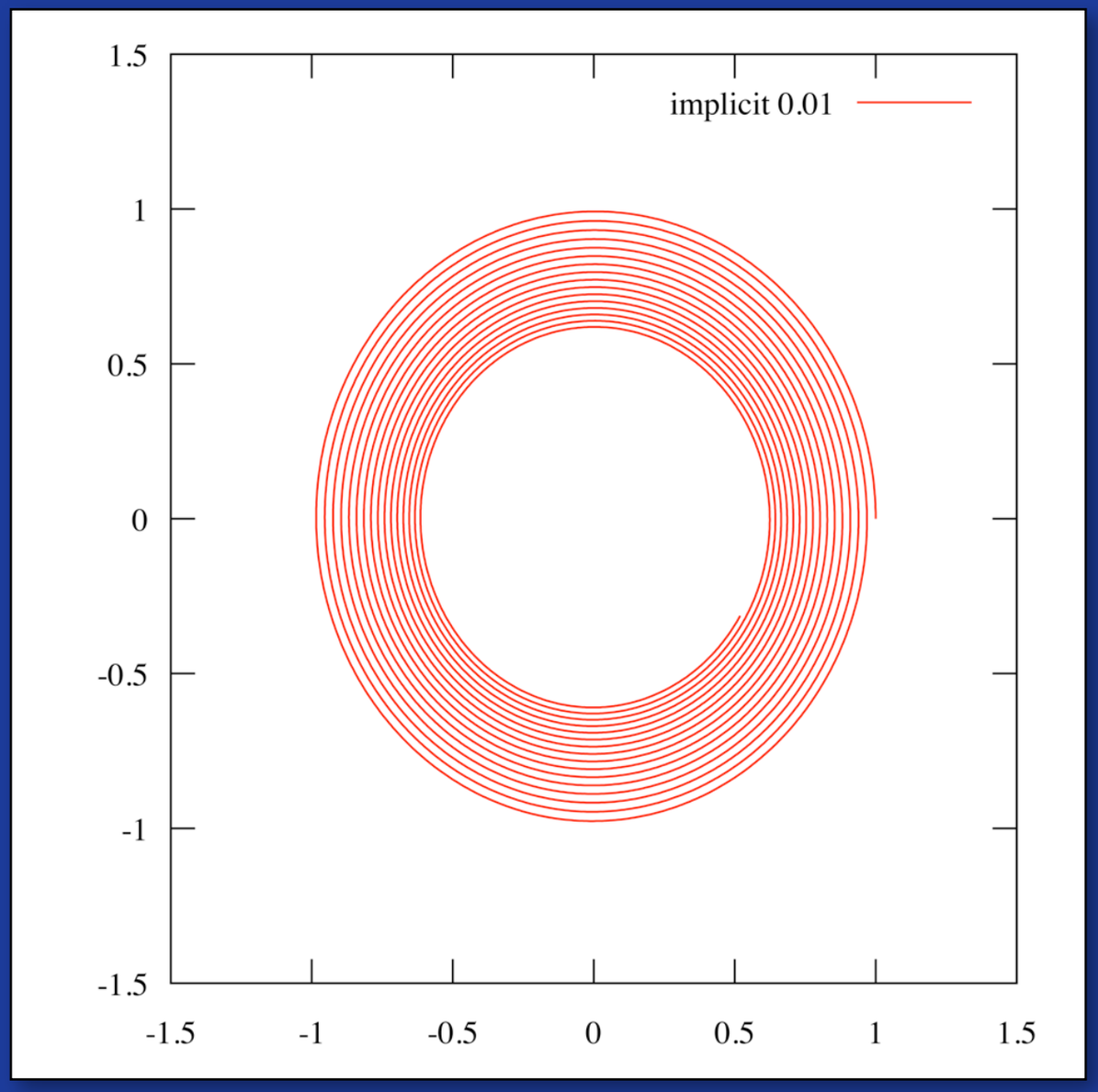
$$\Delta \mathbf{X} = h\left(f(\mathbf{X}_0) + \left(\frac{\partial f}{\partial \mathbf{X}} \right) \Delta \mathbf{X} \right)$$

$$\underbrace{\left(\mathbf{I} - h \frac{\partial}{\partial \mathbf{X}} \left(\dot{\mathbf{X}}(t_0) \right) \right)}_{n \times n \text{ sparse matrix}} \Delta \mathbf{X} = h \dot{\mathbf{X}}(t_0)$$

Solving Large Systems

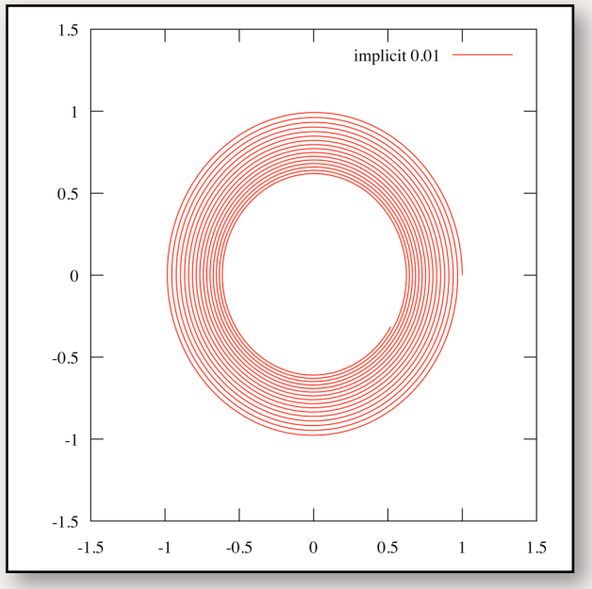
- Matrix structure reflects force-coupling: (i,j)th entry exists iff f_i depends on $\mathbf{X}_j$
- Conjugate gradient a good first choice
- Is this a lot of work?

Long Term Effect

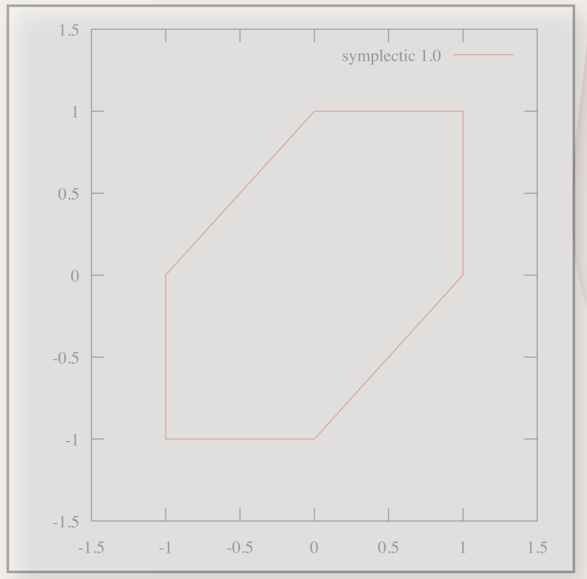


Implicit integration causes this rotational system to spiral inwards.

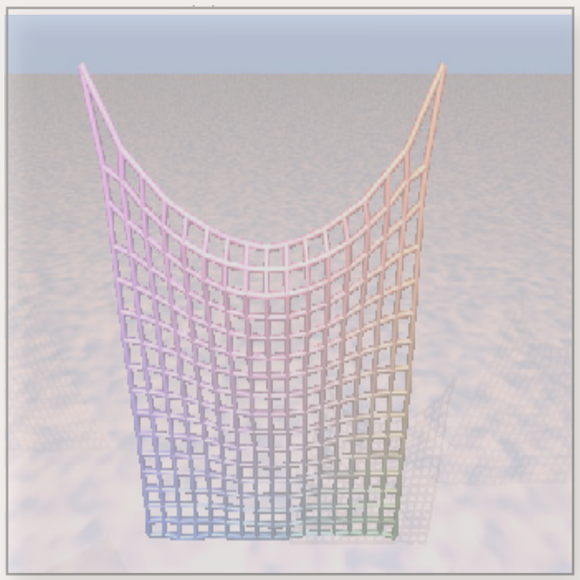
Overview



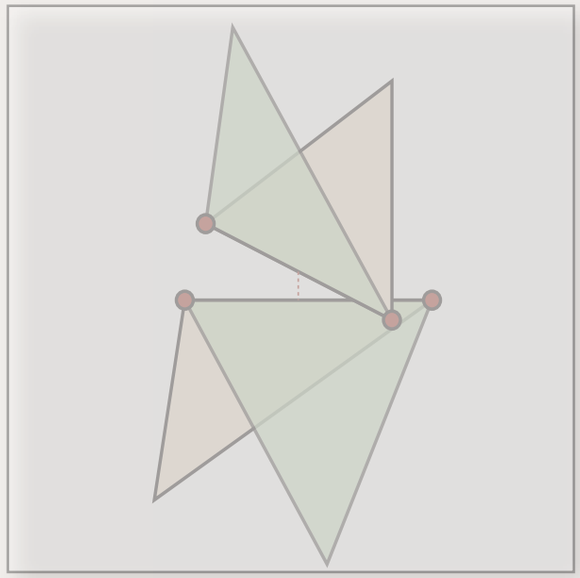
Implicit Integration



Symplectic Integration

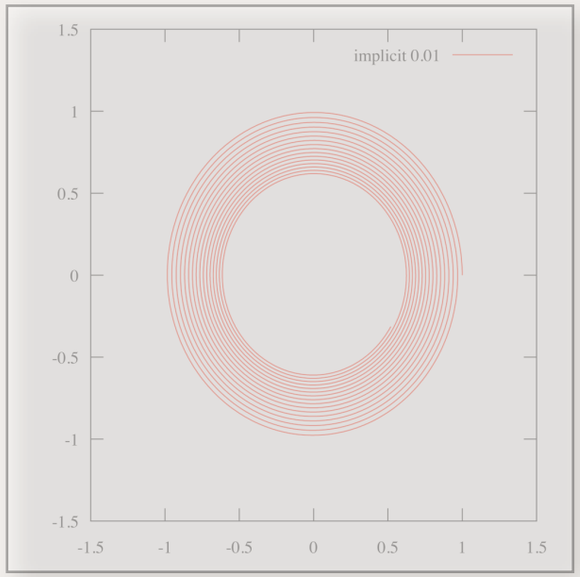


Cloth Simulation

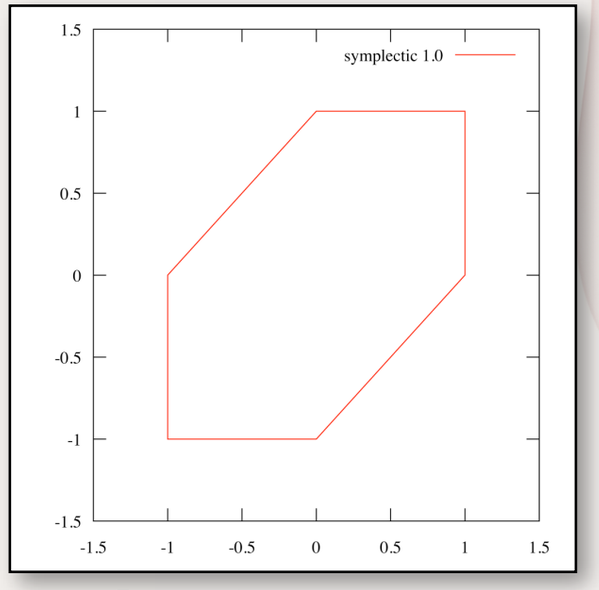


Cloth Collisions

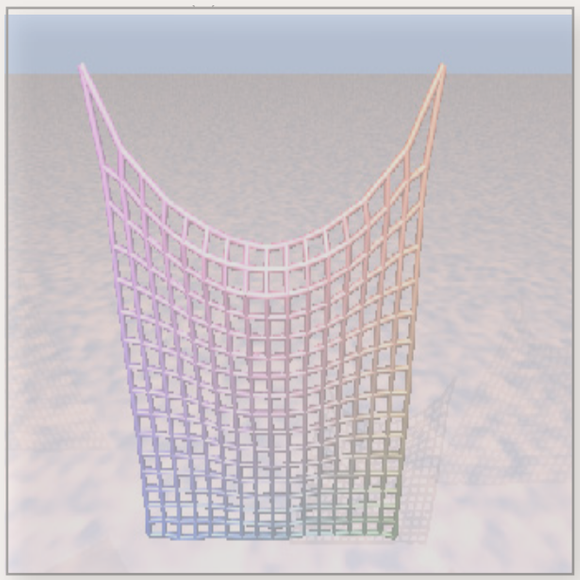
Overview



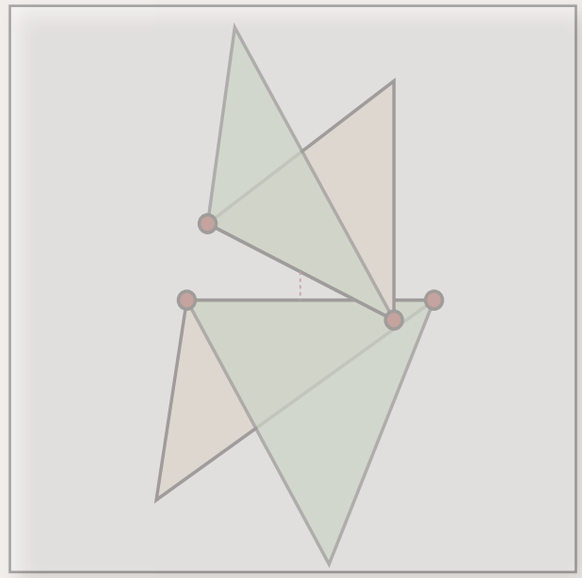
Implicit Integration



Symplectic Integration



Cloth Simulation



Cloth Collisions

Questions

- **Consider the system:**

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{x}$$

- **What would happen if you solved x_1 explicitly and x_2 implicitly?**

Symplectic

- **Consider the system:**

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{x}$$

- **or equivalently:**

$$\dot{x} = -y$$

$$\dot{y} = x$$

- **We want to solve x *explicitly* and y *implicitly*:**

$$x_{i+1} = x_i - h \cdot y_i \qquad \textit{(explicit)}$$

$$y_{i+1} = y_i + h \cdot x_{i+1} \qquad \textit{(implicit)}$$

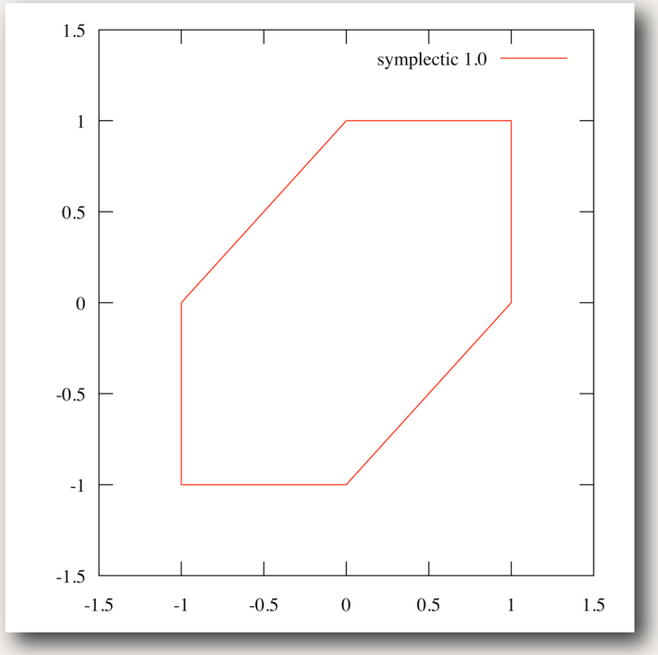
Long Term Evolution

(h = 1.0)

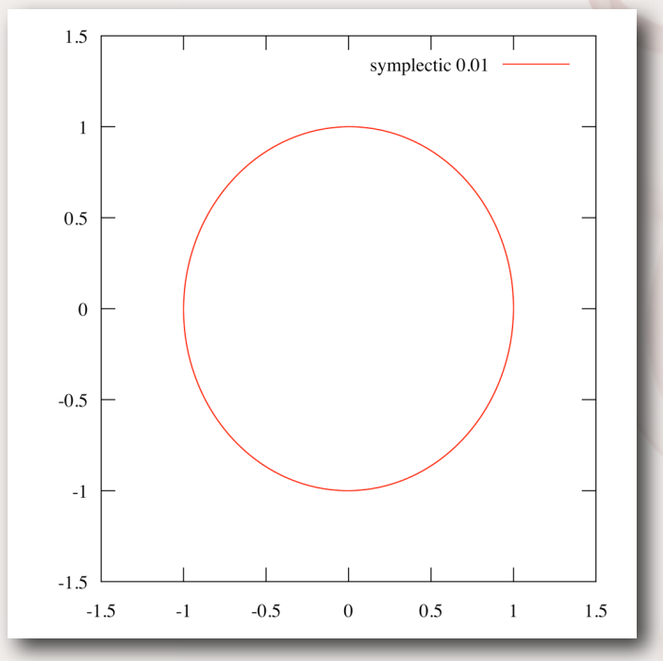
```
x =  1.0;
y =  0.0;
while(true):
  print x, y;
  x -= y;
  y += x;
```

```
 1.0  0.0
 1.0  1.0
 0.0  1.0
-1.0  0.0
-1.0 -1.0
 0.0 -1.0
 1.0  0.0
 1.0  1.0
 0.0  1.0
-1.0  0.0
-1.0 -1.0
 0.0 -1.0
 1.0  0.0
```

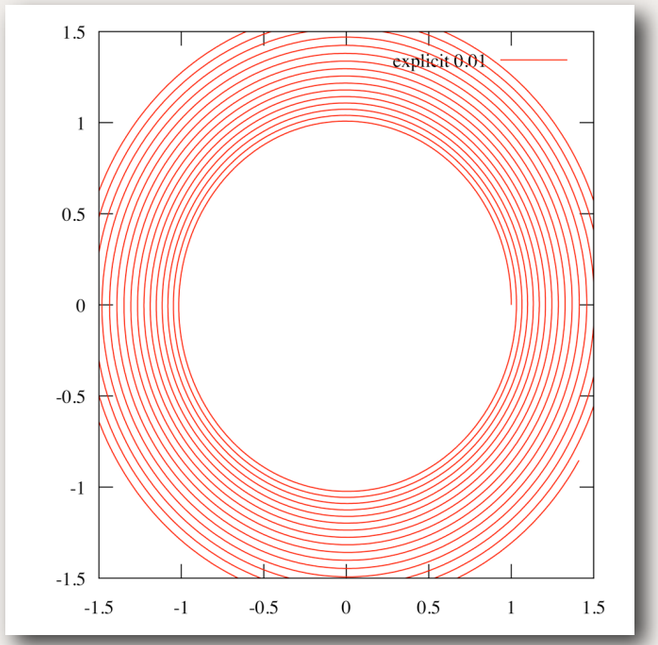
Long Term Evolution



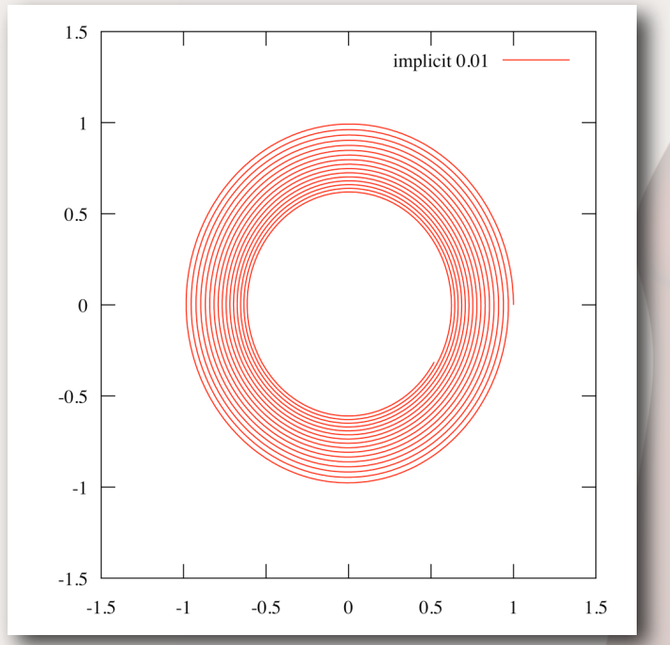
Symplectic 1.0



Symplectic 0.01



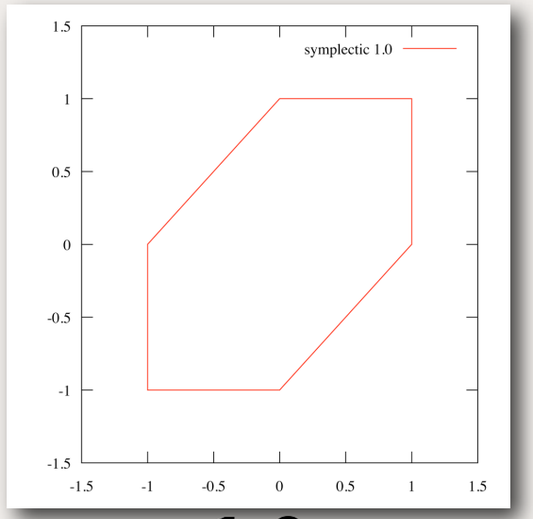
Explicit 0.01



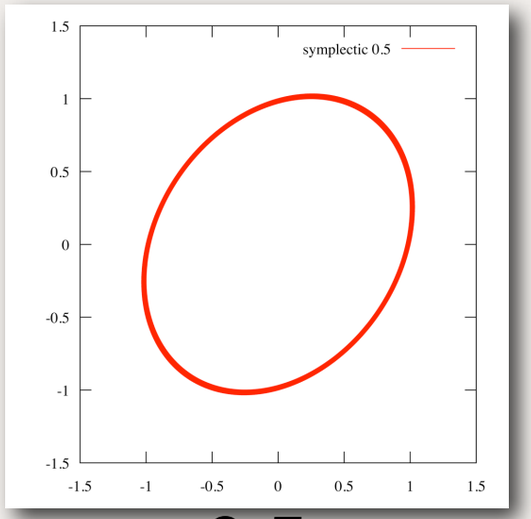
Implicit 0.01

Long Term Evolution

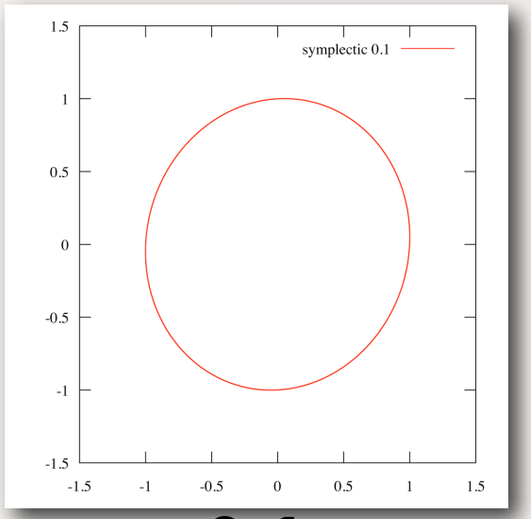
Decreasing Timestep:



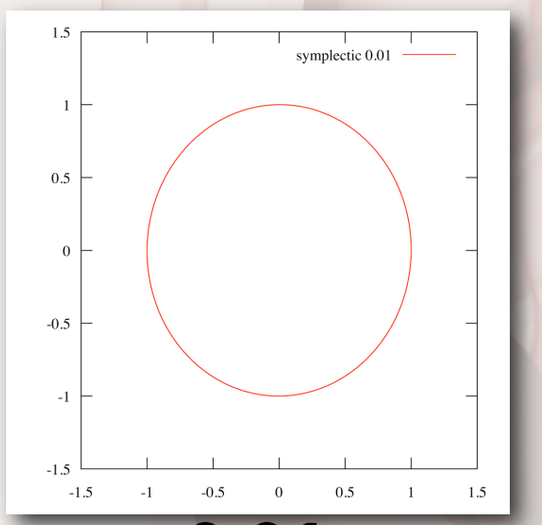
1.0



0.5

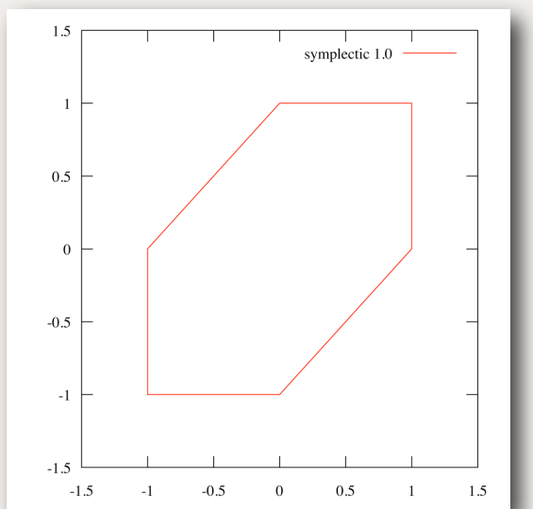


0.1

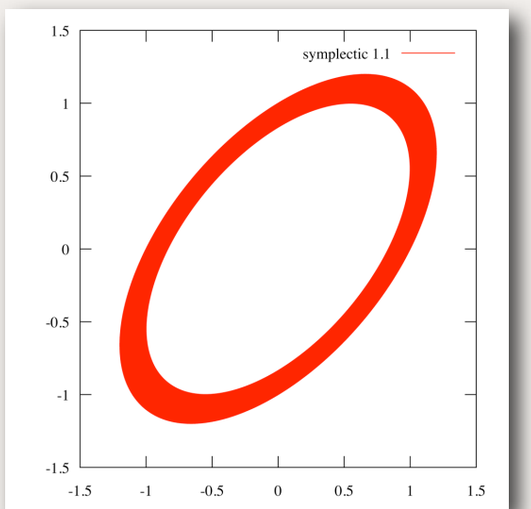


0.01

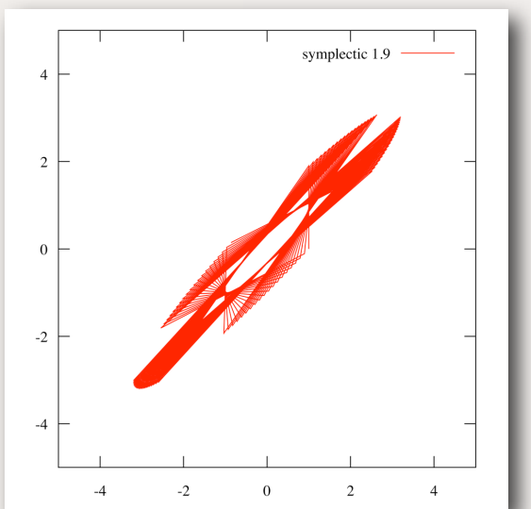
Increasing Timestep:



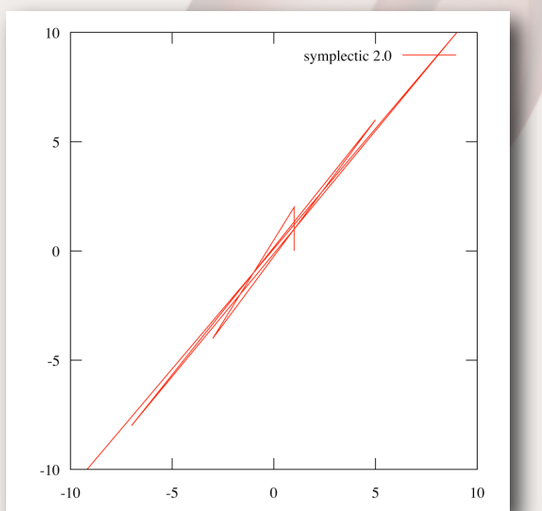
1.0



1.1



1.9



2.0

Why?

- **The symplectic integrator:**

$$\begin{aligned} x_{i+1} &= x_i - h y_i \\ y_{i+1} &= y_i + h x_{i+1} \end{aligned}$$

- **Can be rewritten:**

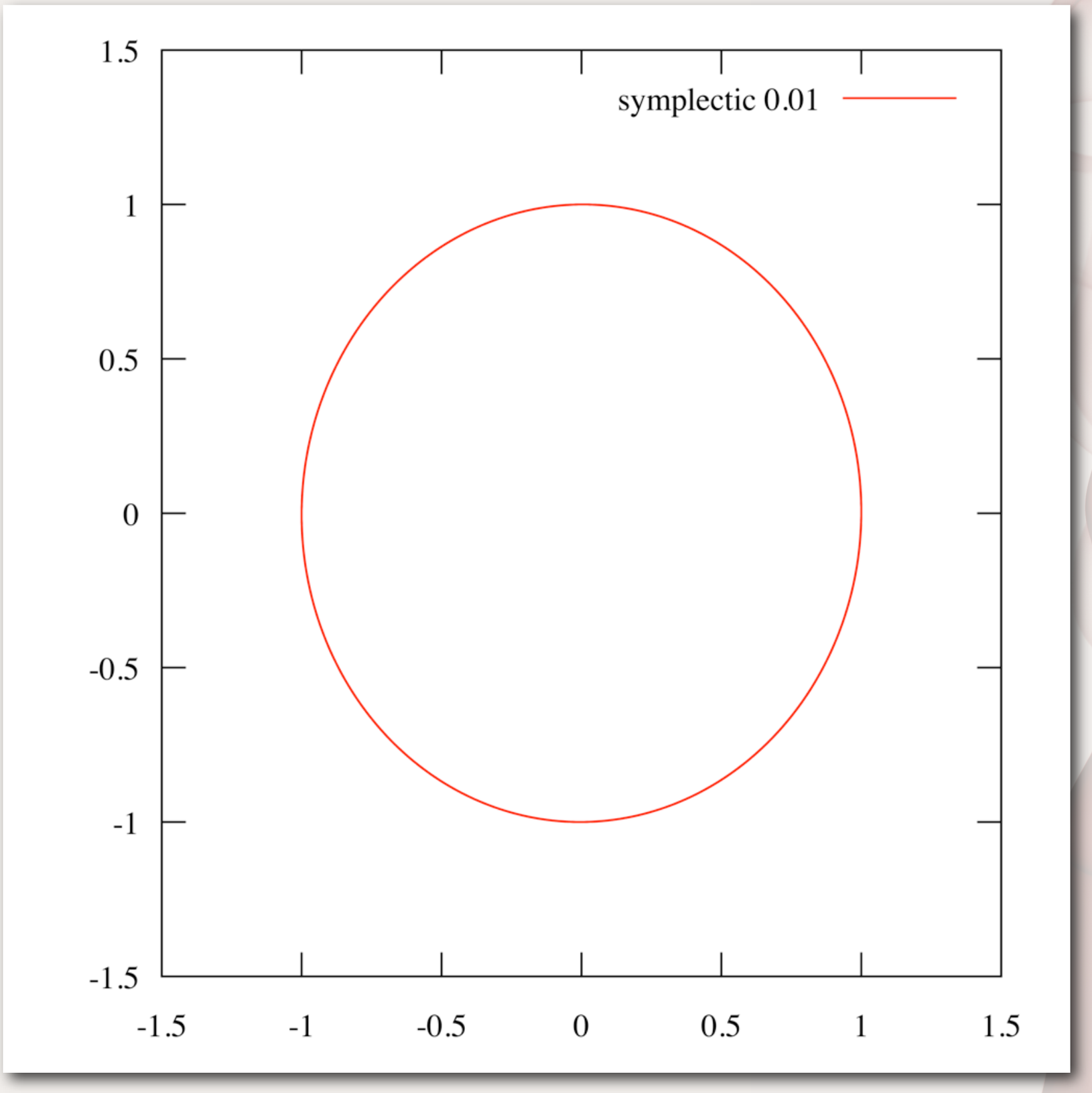
$$\mathbf{x}_{i+1} = \begin{bmatrix} 1 & -h \\ h & 1-h^2 \end{bmatrix} \mathbf{x}_i$$

- **Which implies:**

$$\mathbf{x}_i = \begin{bmatrix} 1 & -h \\ h & 1-h^2 \end{bmatrix}^i \mathbf{x}_0$$

- **But:**

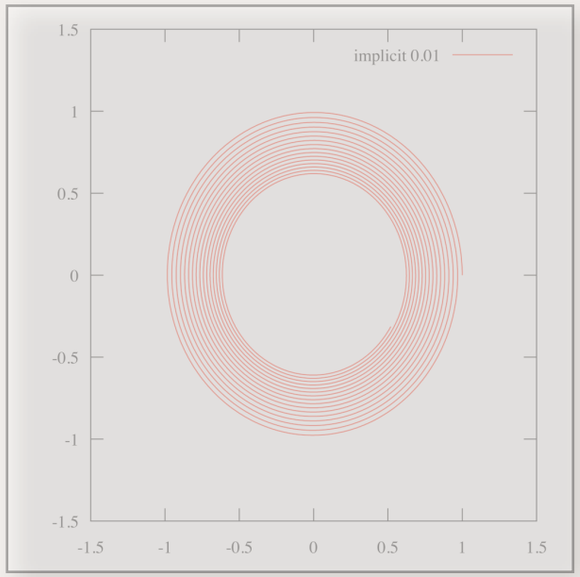
$$\left\| \mathrm{eig}\left(\begin{bmatrix} 1 & -h \\ h & 1-h^2 \end{bmatrix} \right) \right\| = 1 \qquad \text{if } h < 2$$


symplectic 0.01
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

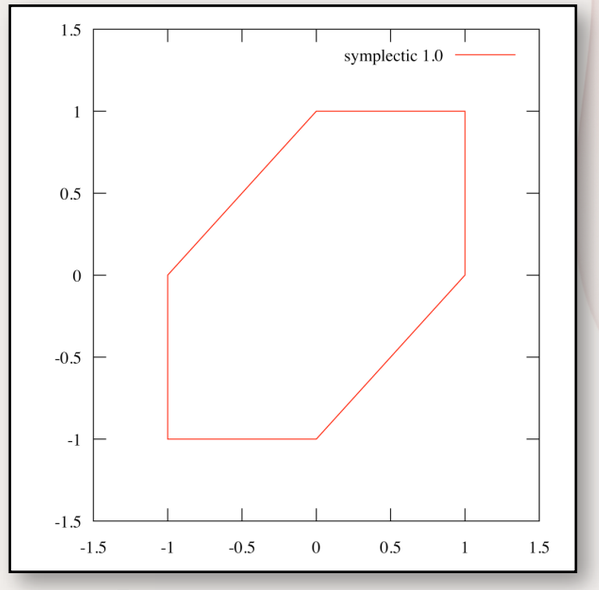
Symplectic

- This is not general:
  - Hamiltonian systems
  - Preserves Area
- Why did we learn this?
- Numerical Integration is subtle!
  - Small changes can have profound long-term effects.

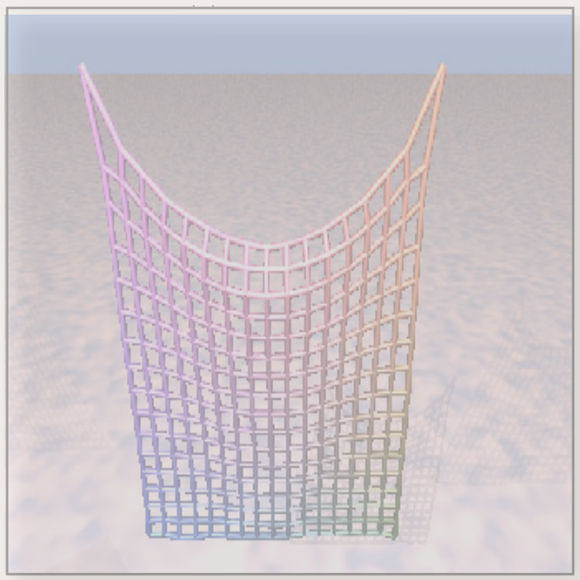
Overview



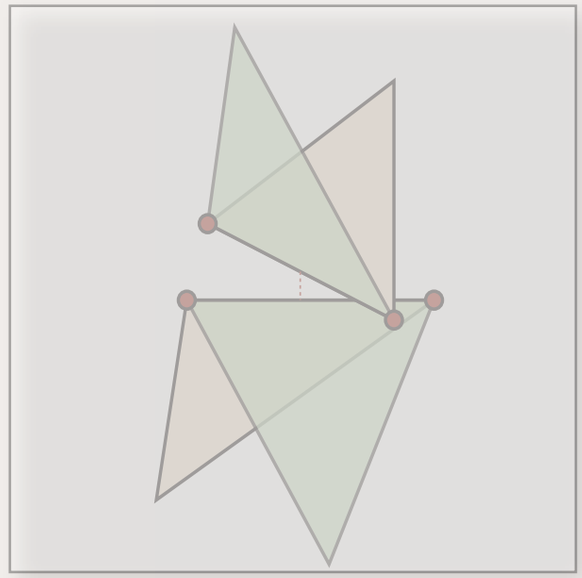
Implicit Integration



Symplectic Integration

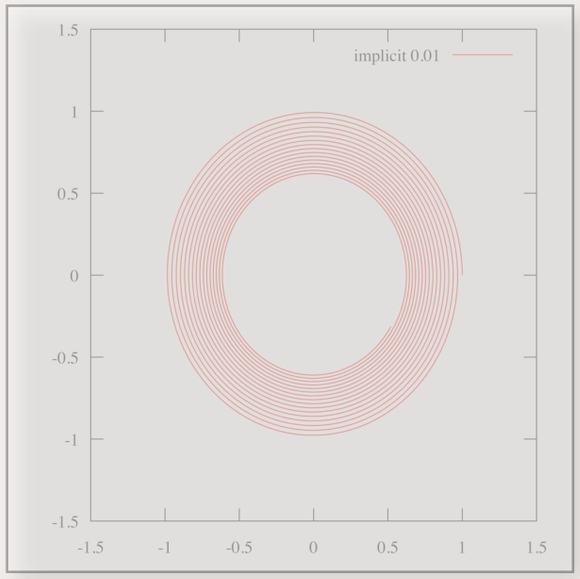


Cloth Simulation

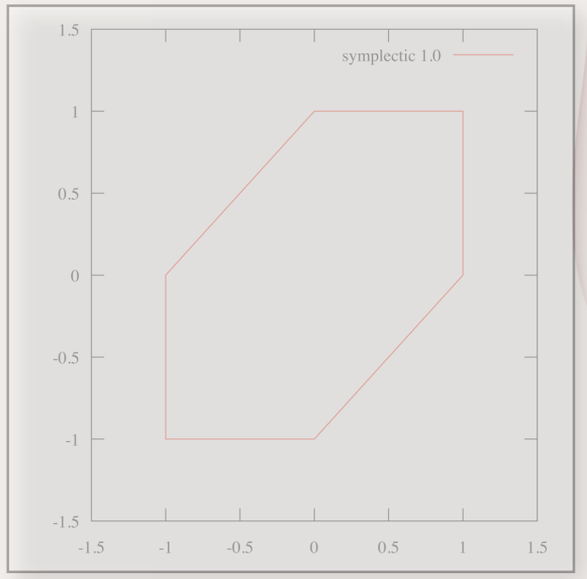


Cloth Collisions

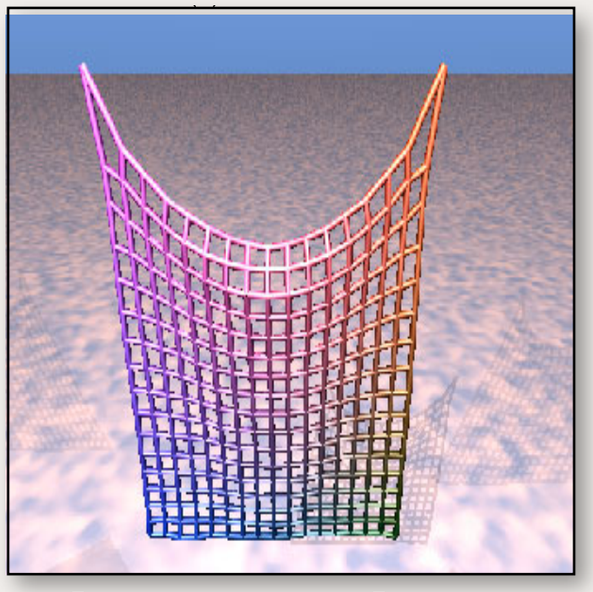
Overview



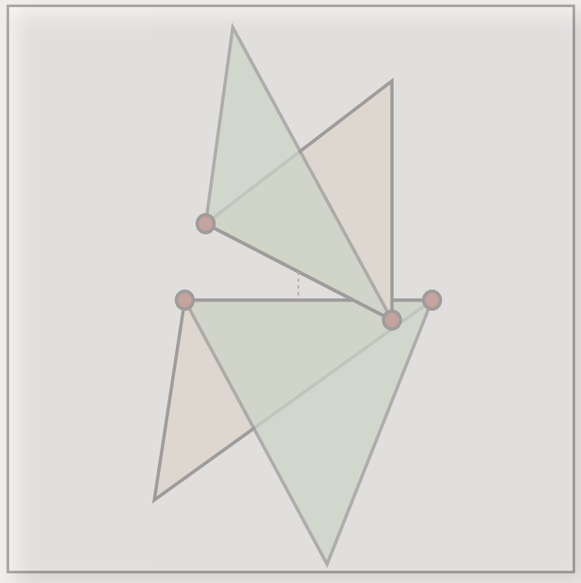
Implicit Integration



Symplectic Integration



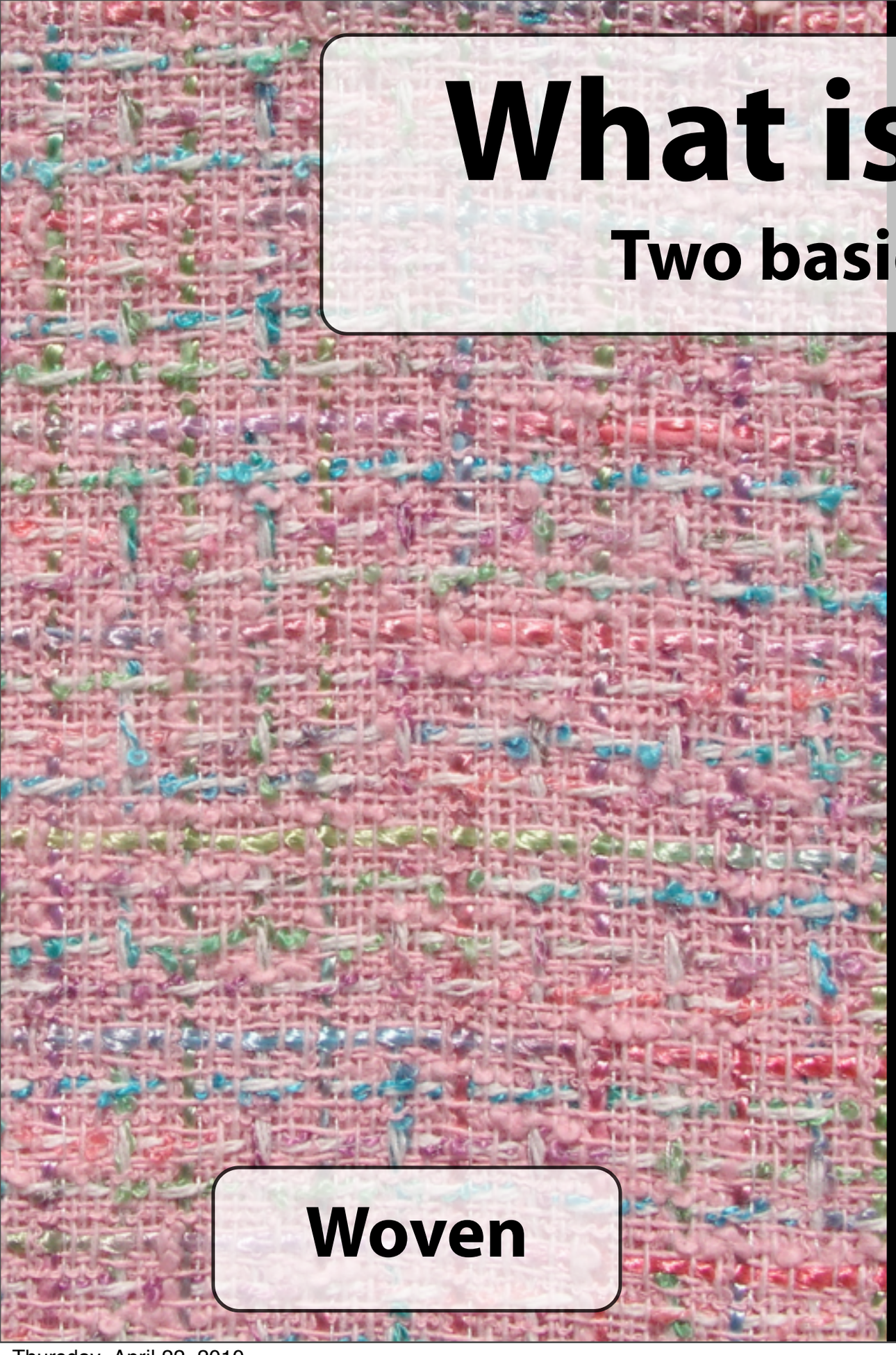
Cloth Simulation



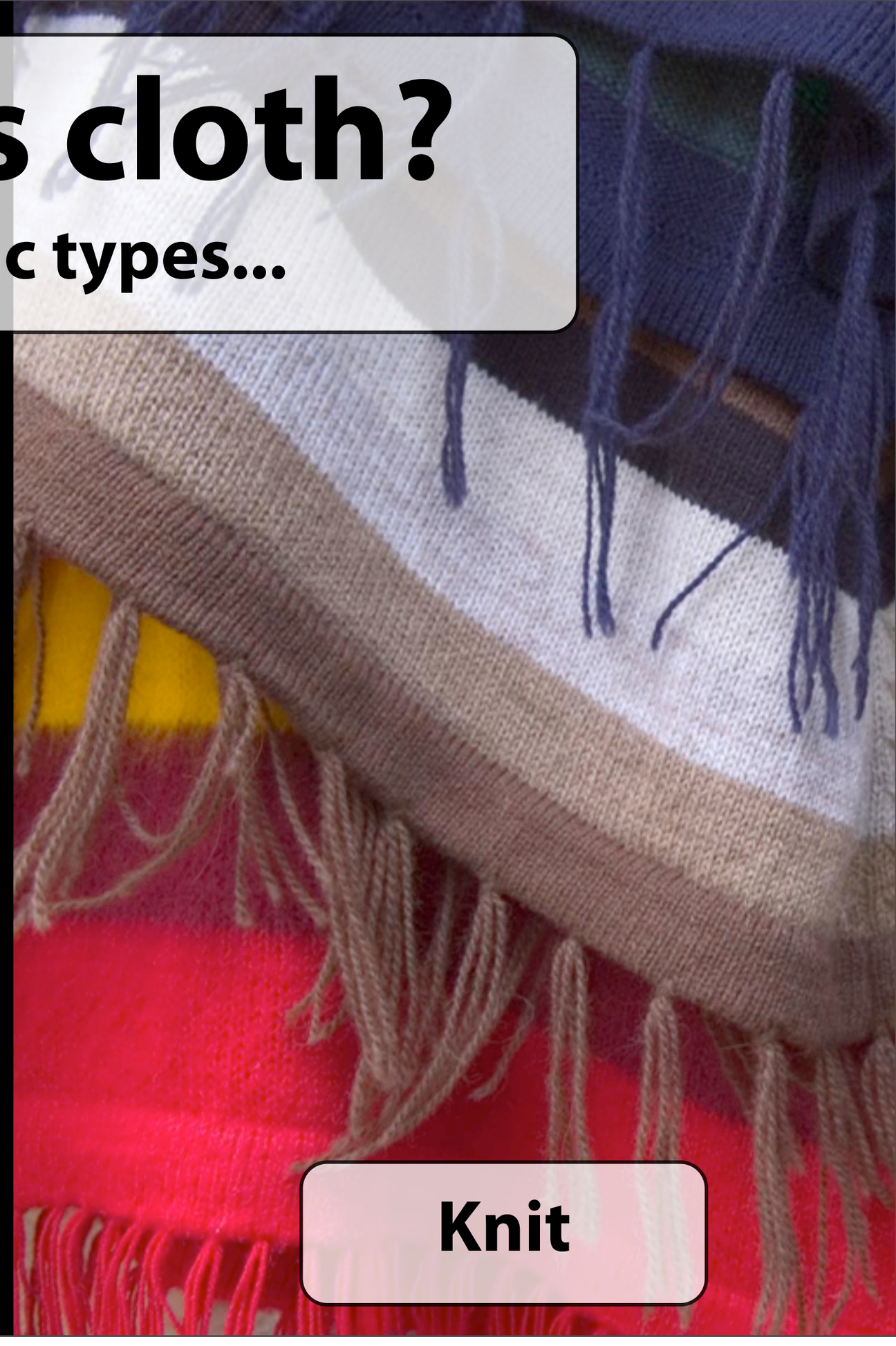
Cloth Collisions

What is cloth?

Two basic types...

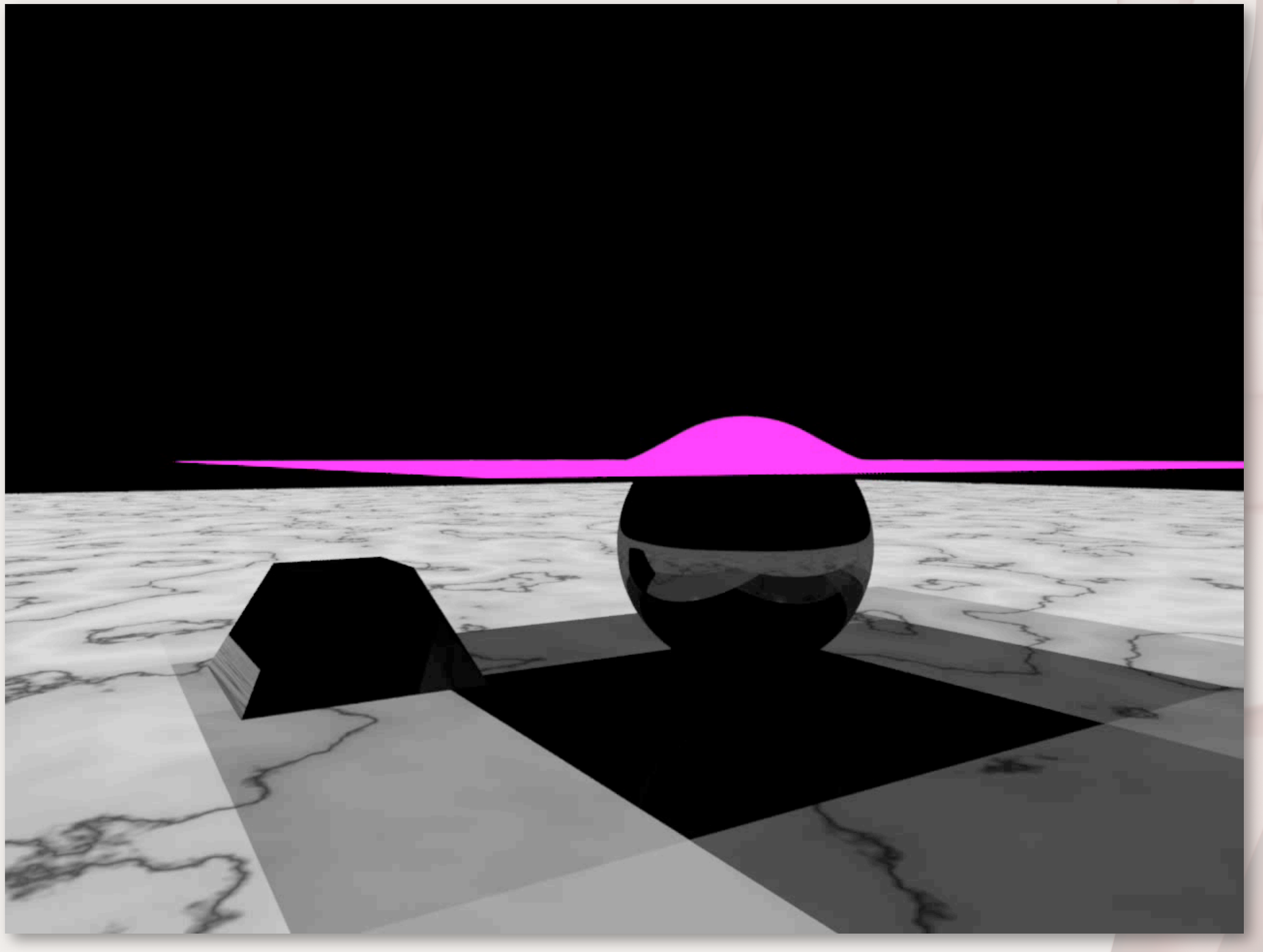


Woven



Knit

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?

Cloth Animation

Christopher Twigg
March 4, 2003

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

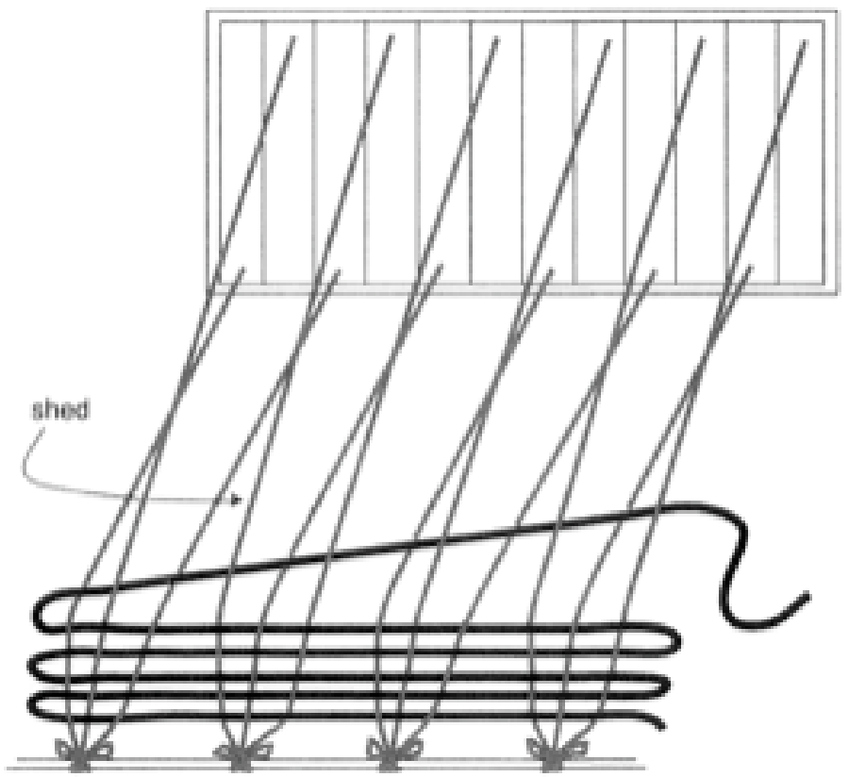
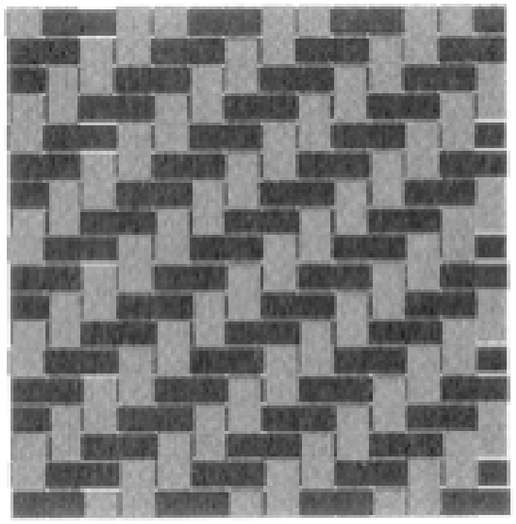
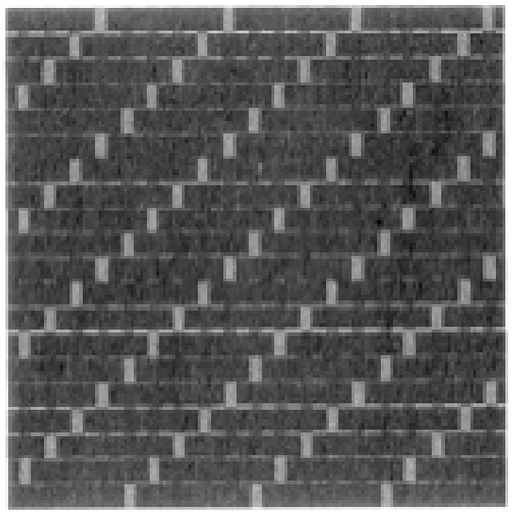


Figure 1.8. The weaving process.

House, Breen [2000]

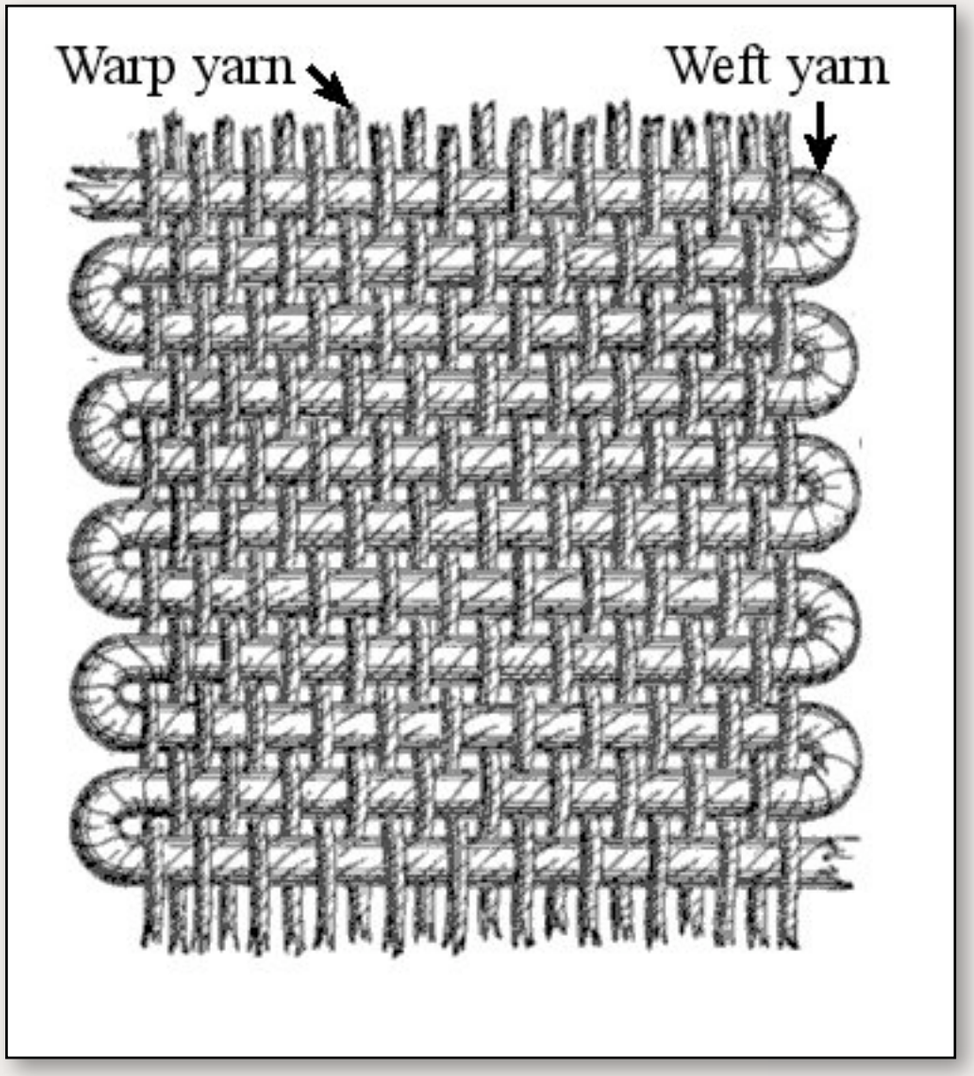


b) twill



c) satin

Warp and Weft



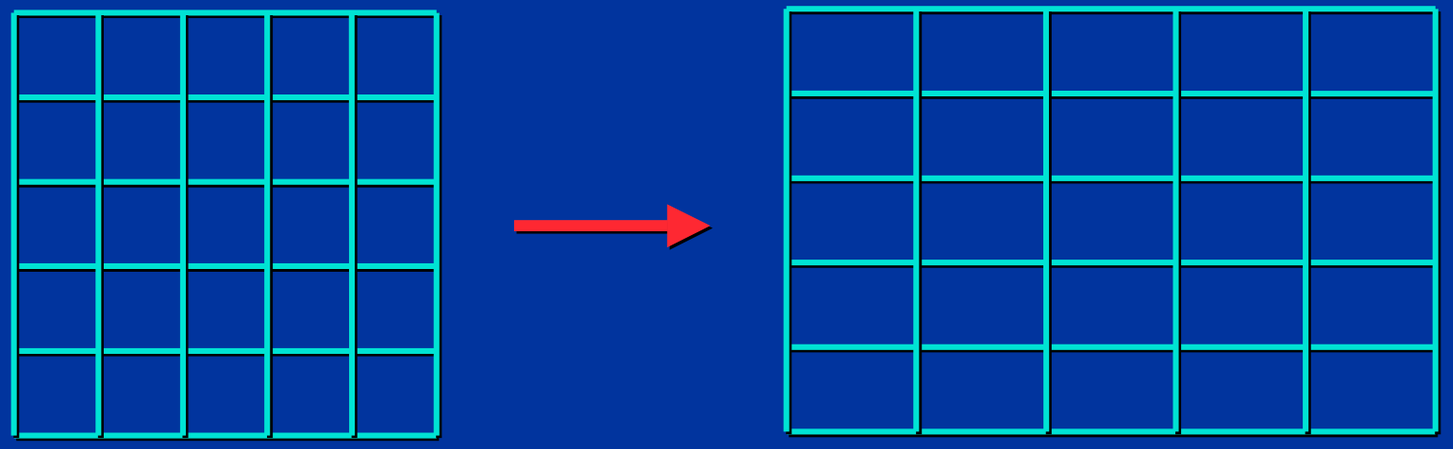
source: Wikipedia

Cloth and Fur Energy Functions

Michael Kass

PIXAR
ANIMATION STUDIOS

Stretch (Continuum Version)

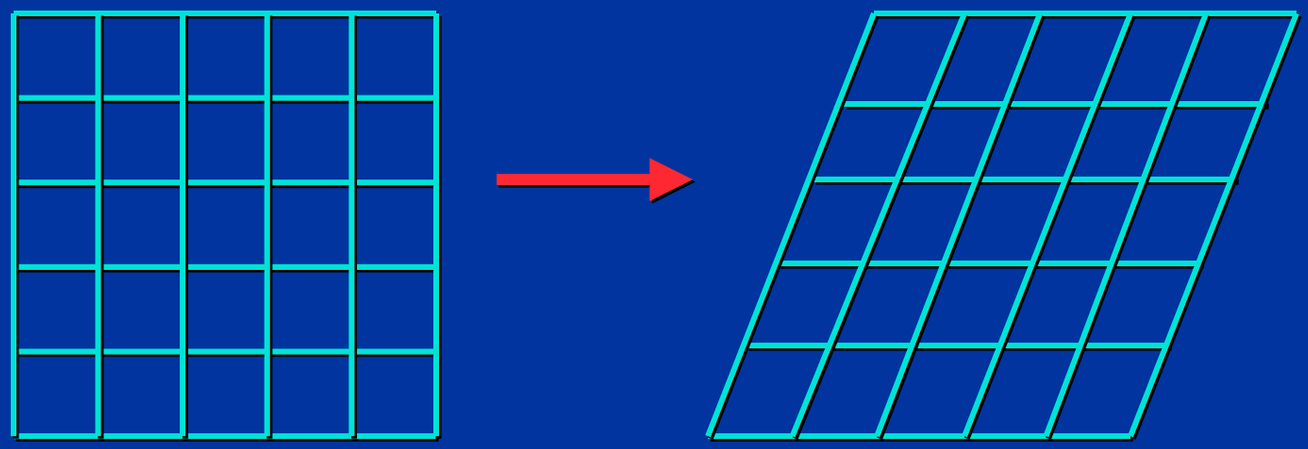


(u,v) $\vec{x}$

$$S_u = \left\| \frac{\partial \vec{x}}{\partial u} \right\| - 1$$

$$E = \frac{1}{2} k \int (S_u^2 + S_v^2) \, du \, dv$$

Shear (Continuum Version)

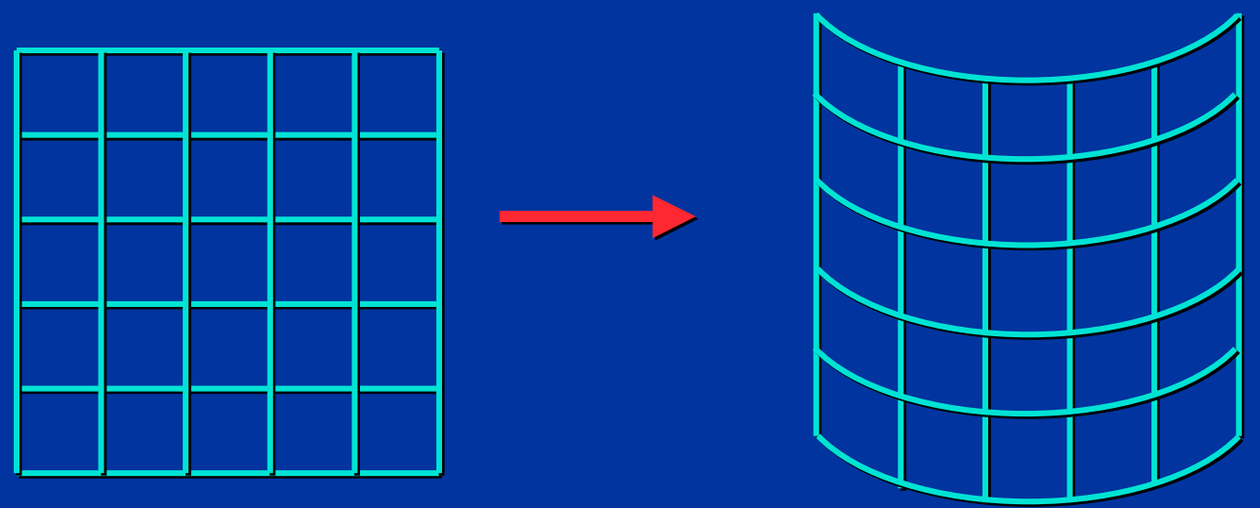


(u,v) $\vec{x}$

$$\theta = \cos^{-1}\left(\frac{\widehat{\partial \vec{x}}}{\partial u} \bullet \frac{\widehat{\partial \vec{x}}}{\partial v}\right)$$

$$E = \frac{1}{2} k \int \theta^2 \, du \, dv$$

Bend (Continuum Version)



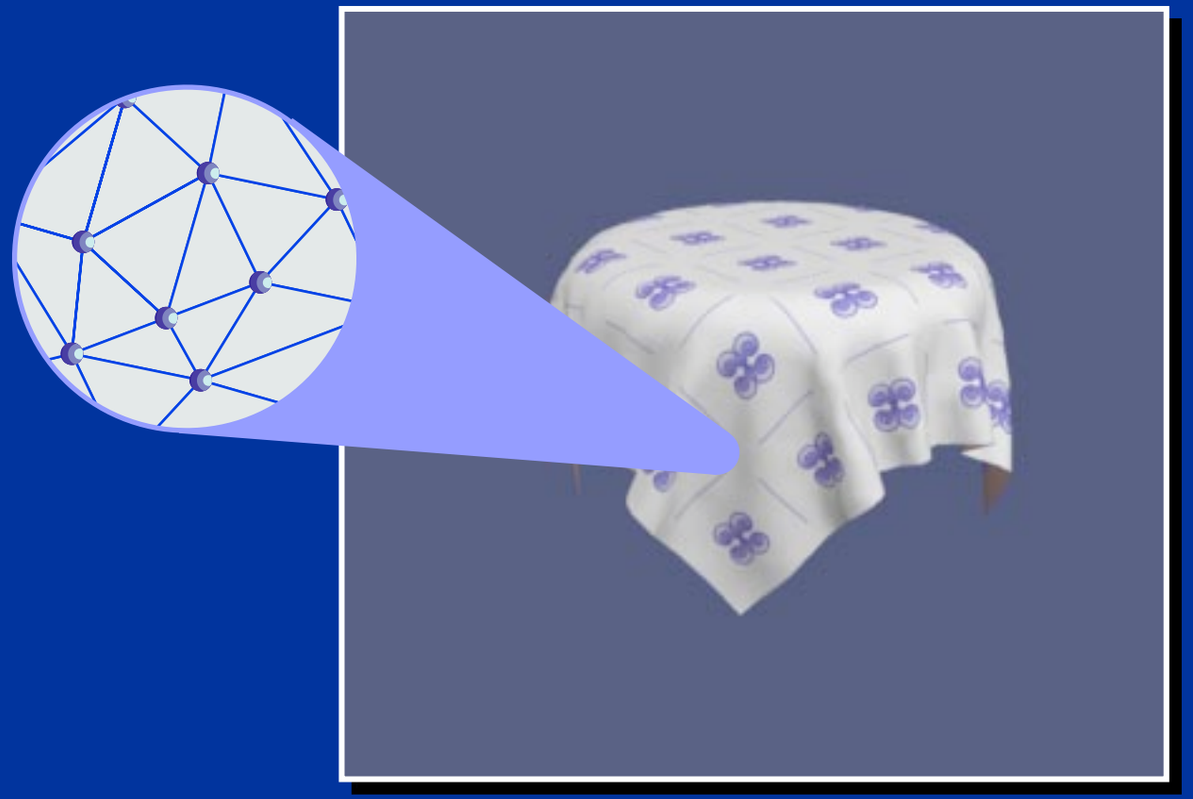
(u,v) $\vec{x}$

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

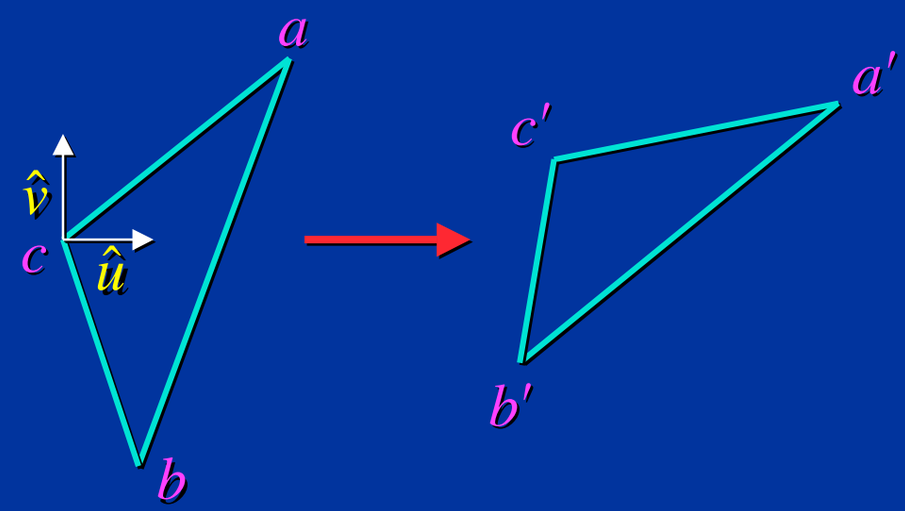
Resitence To...

- Stretching
- Shearing
- Bending

Discretization



Triangle Energy

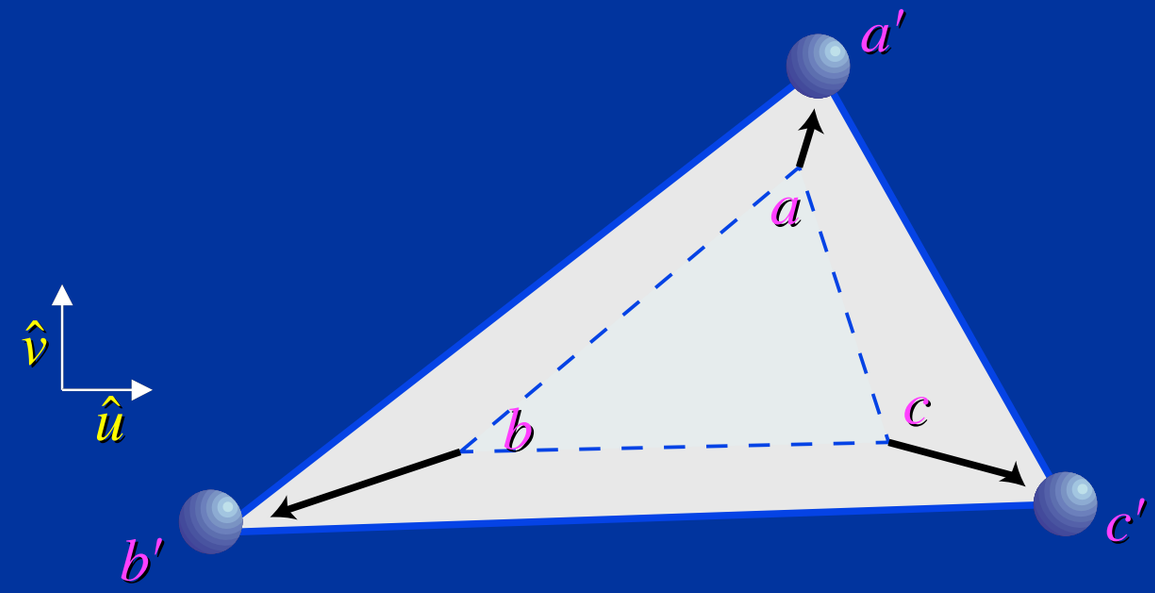


First, compute the affine transformation T that maps: $T : a \rightarrow c'$

$b \rightarrow b'$

$c \rightarrow c'$

Triangle Stretch Energy

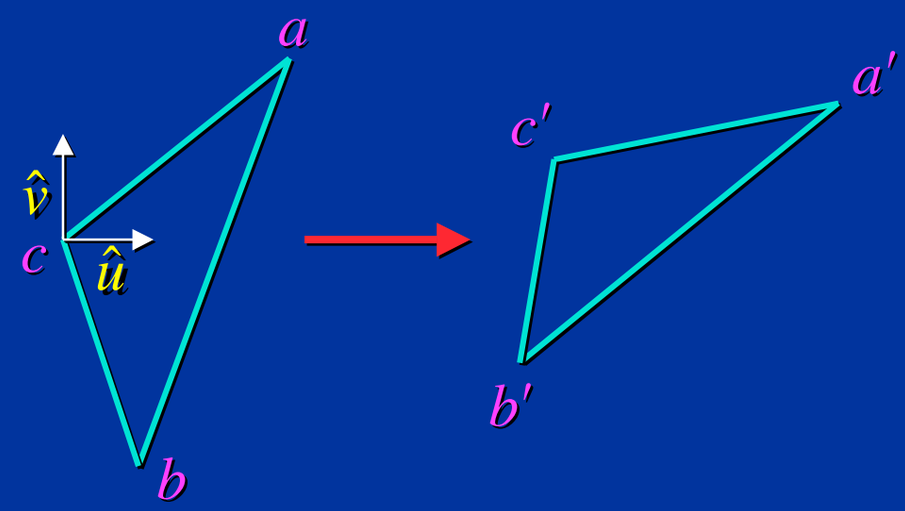


Now compute the stretch energy.

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

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

Triangle Shear Energy

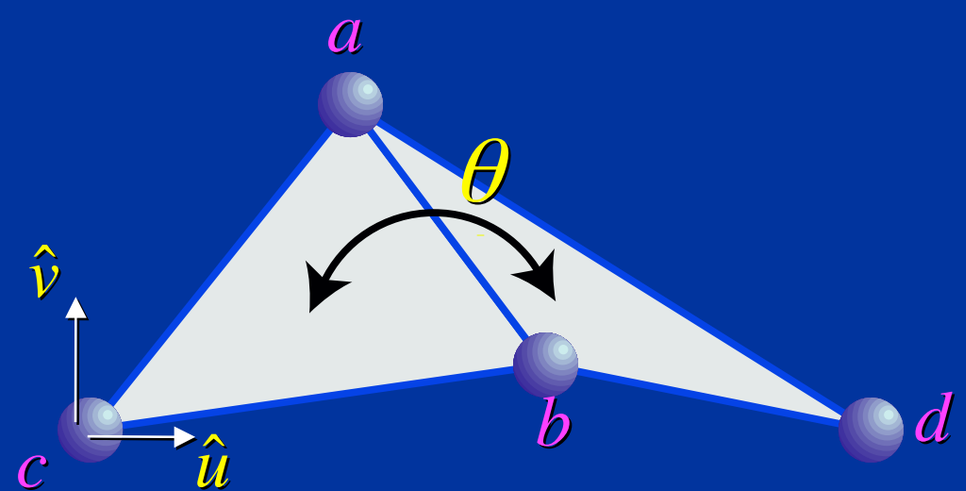


Next compute the shear energy.

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

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

Triangle Bend Energy

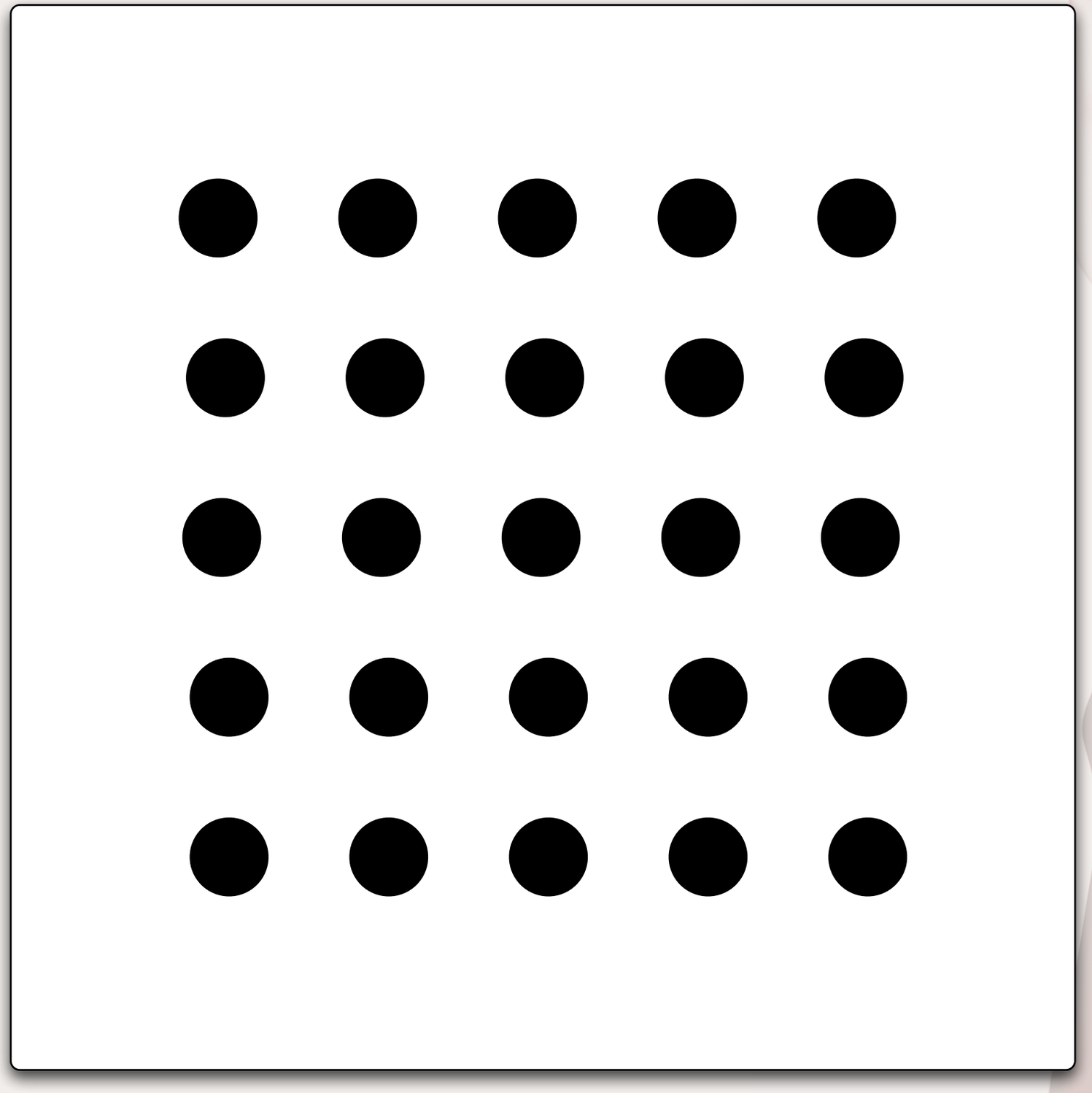


$$\kappa = \frac{\theta}{l_{perp}}$$

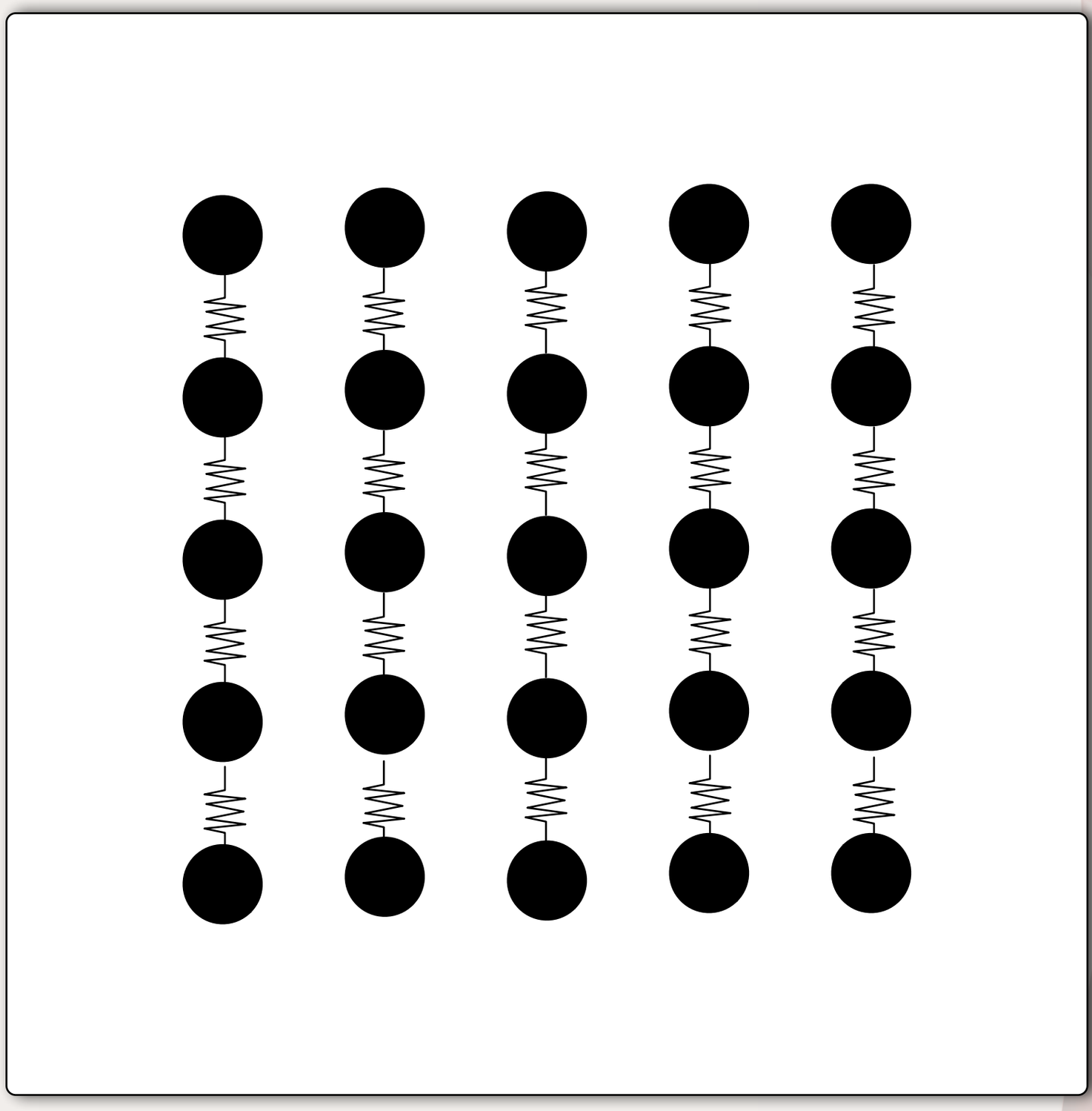
Finally compute the bend energy.

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

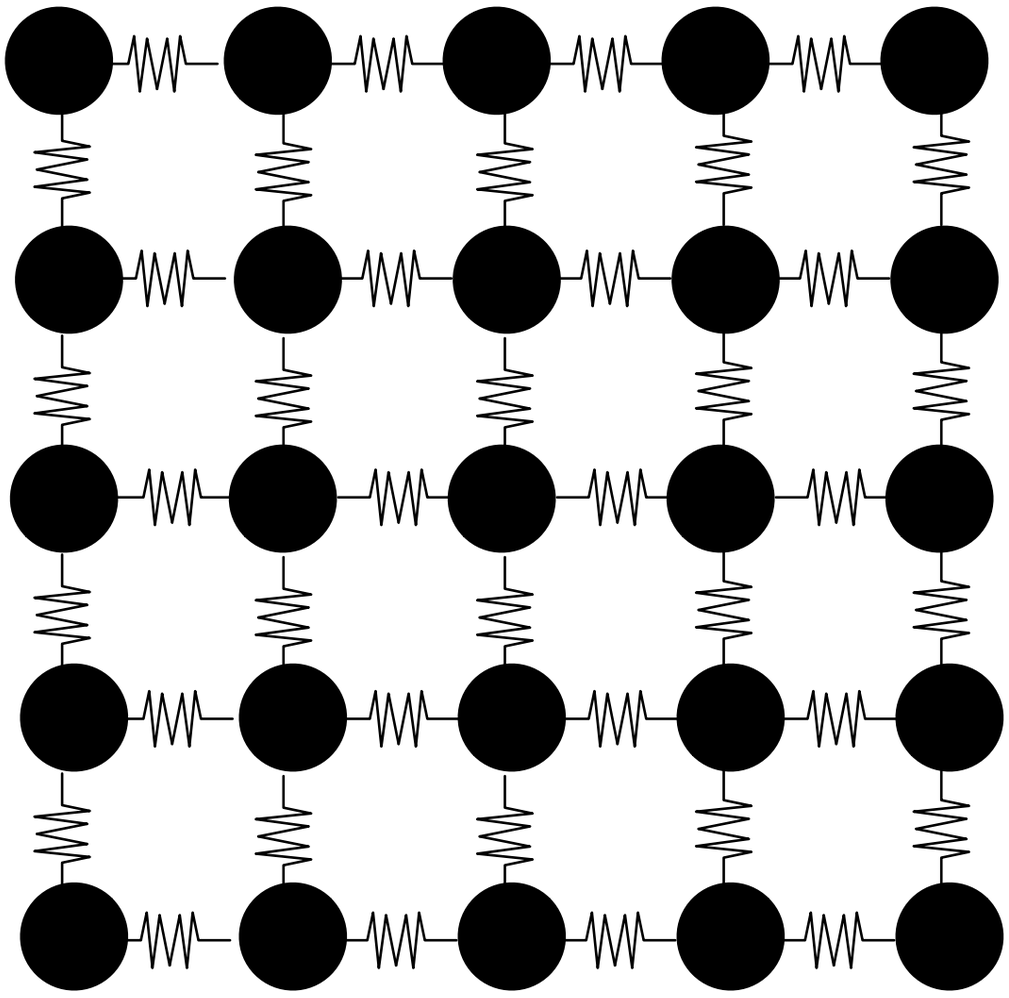
Basic Model



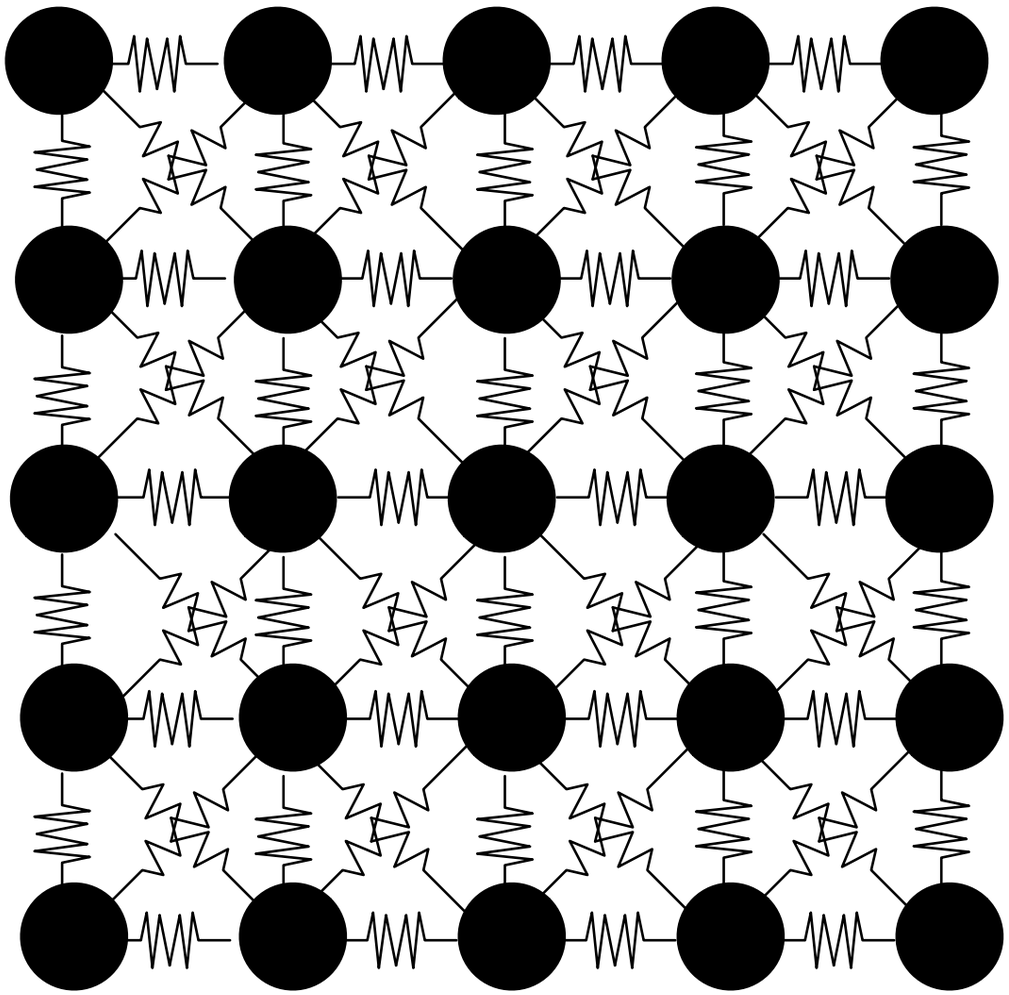
Warp Srings



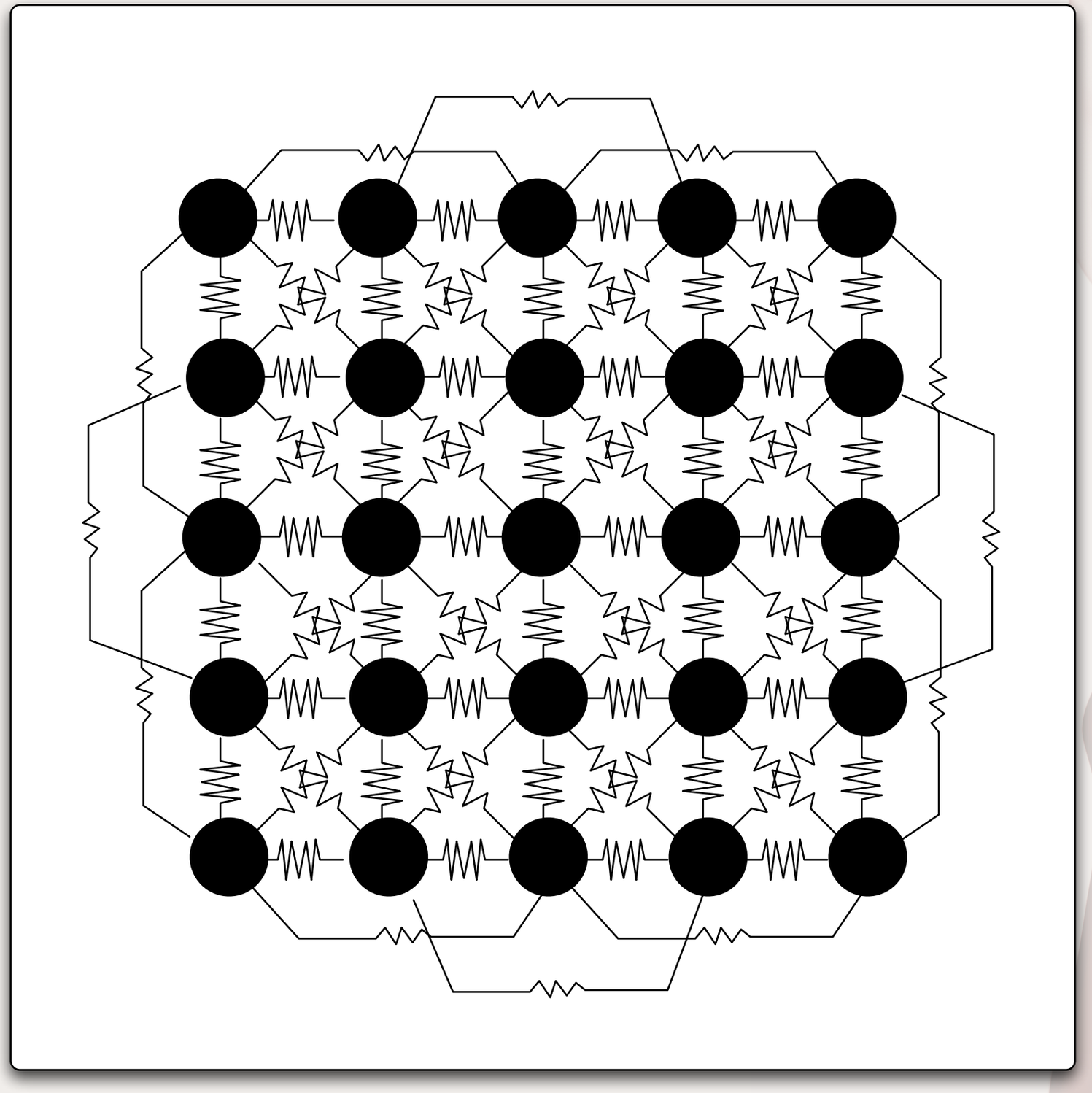
Weft Springs



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

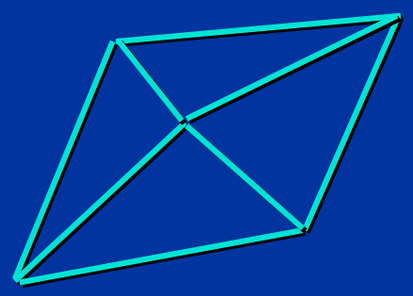
Non-flat cloth is strange stuff:

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

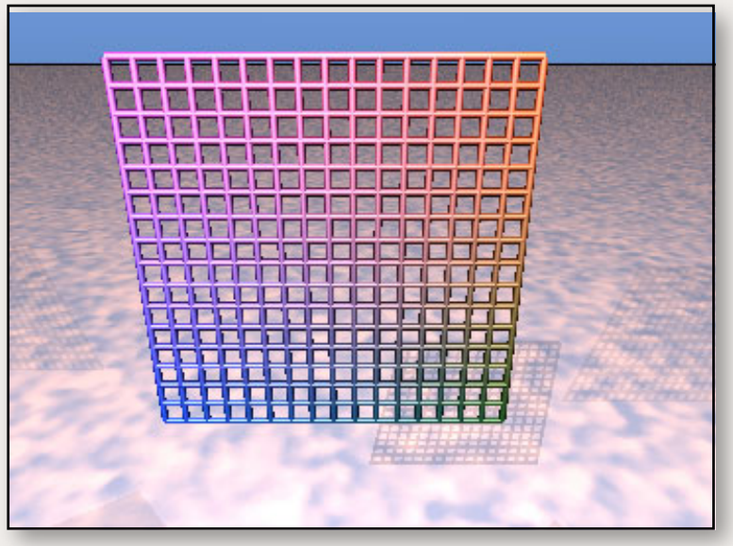
Model in 3D

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

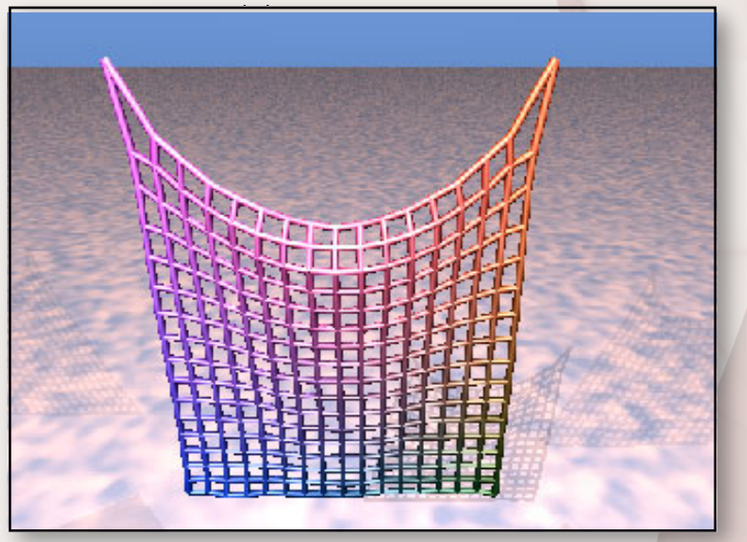
Model in 2D

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

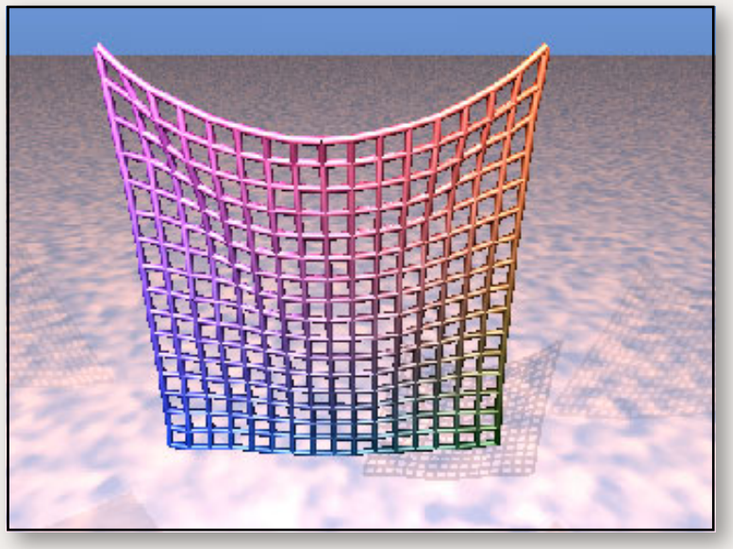
Springs vs. Constraints



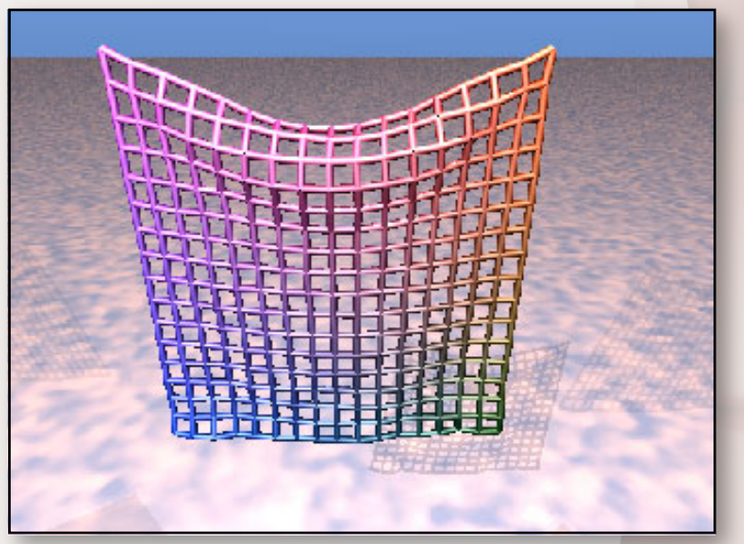
Before Simulation



Only Springs



Stretch Constraints

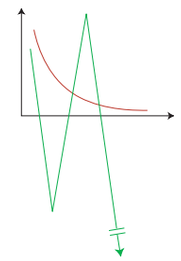


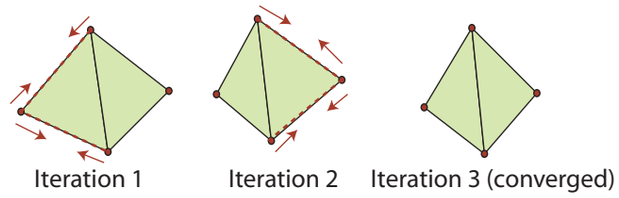
Stretch+Shear Constraints

Source: Xavier Provot

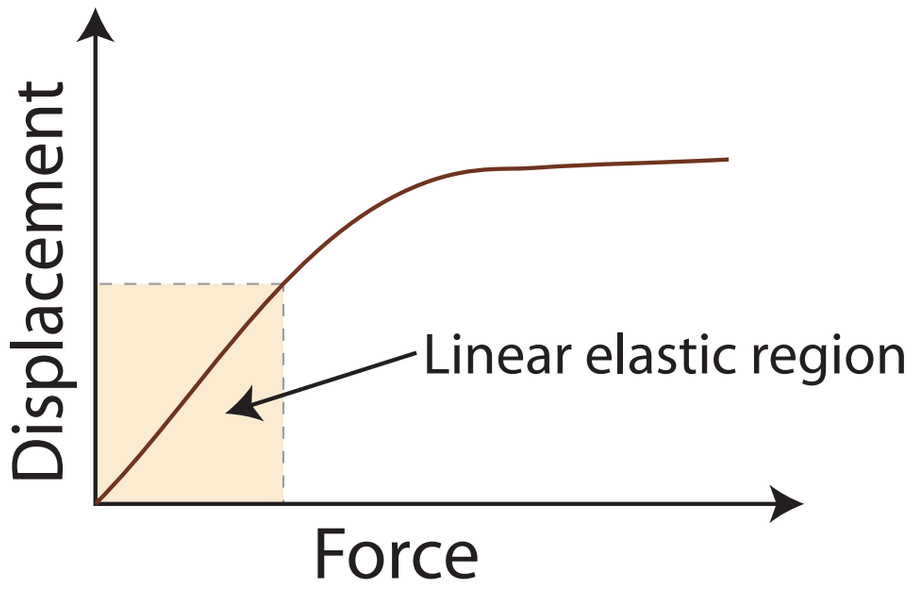
Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior

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

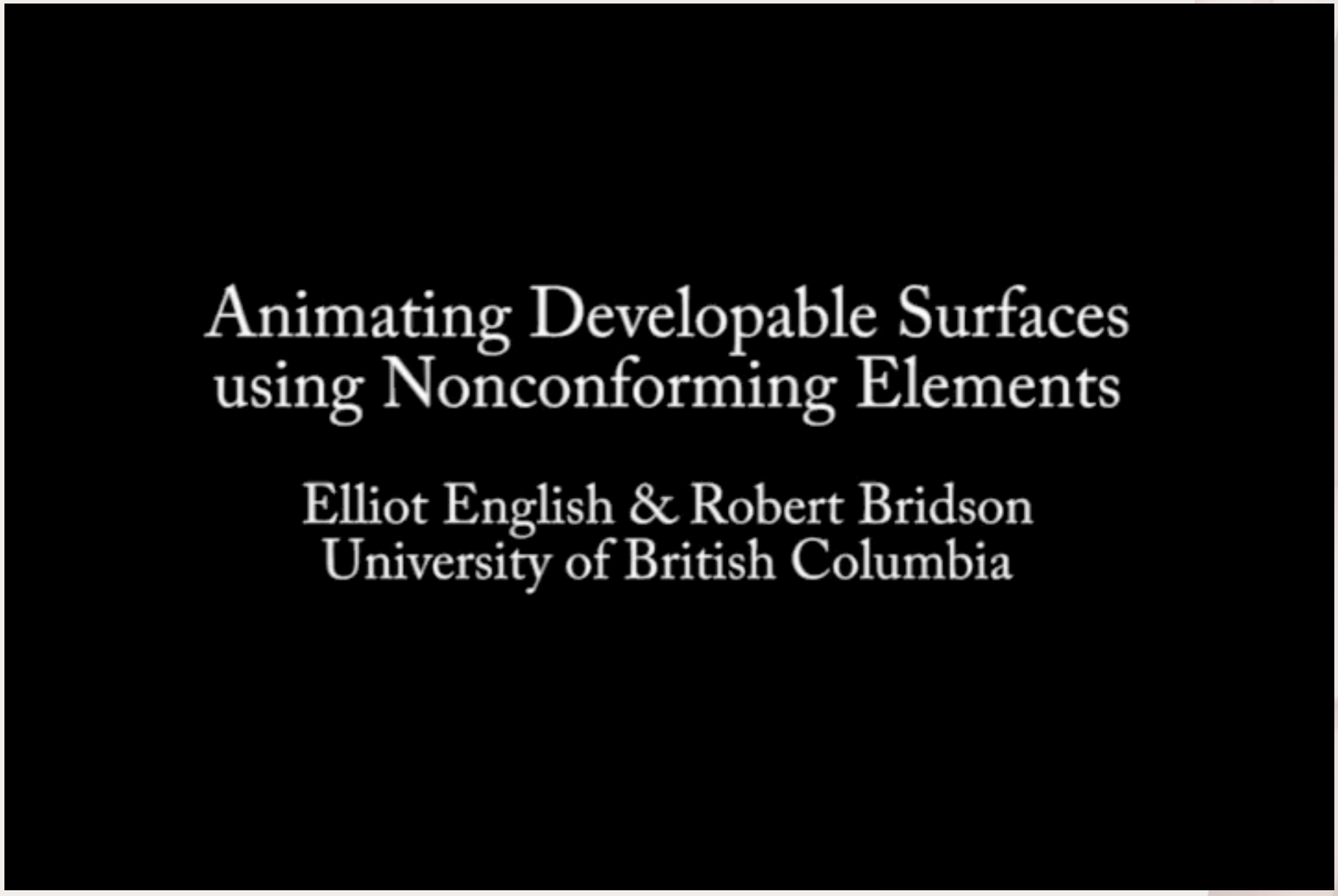
David Harmon
Columbia University

Raanan Fattal
UC Berkeley

Michel Bercovier
The Hebrew University

Eitan Grinnspun
Columbia University

Developable Surfaces



Developable Surfaces

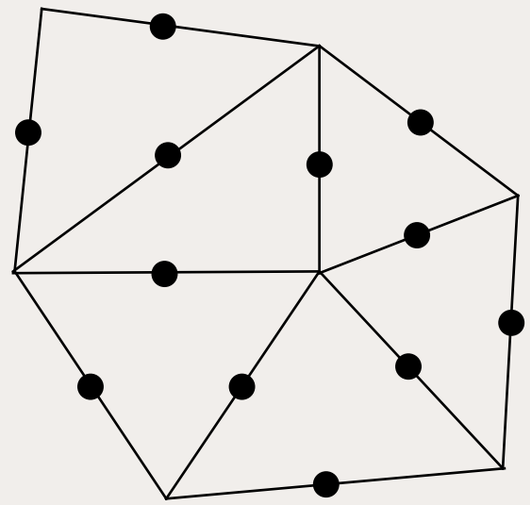


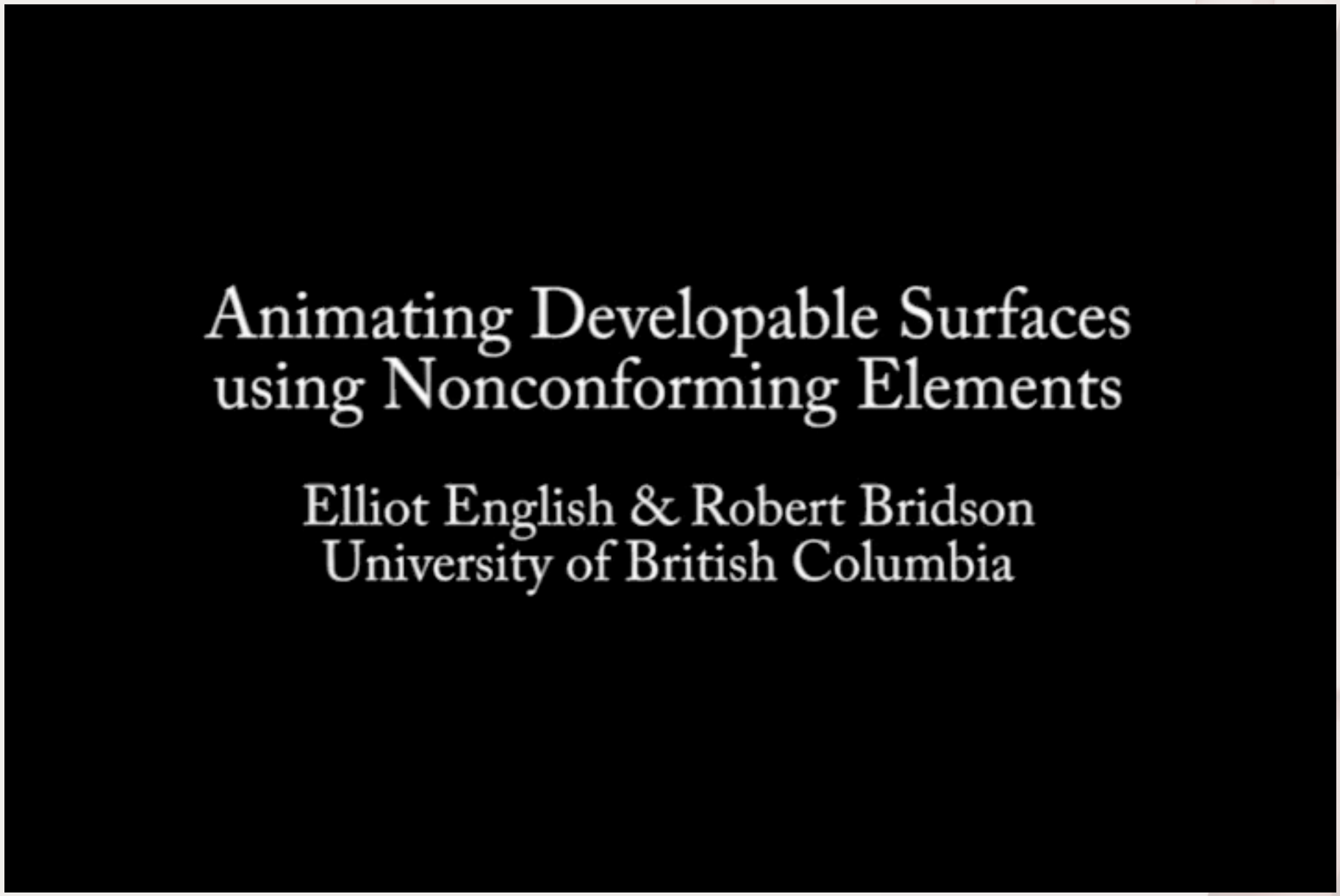
Figure 2: *Schematic of nonconforming variables, located at mid-points of edges between triangles. While continuous at these points, the surface may be discontinuous along the rest of each edge.*

Animating Developable Surfaces using Nonconforming Elements

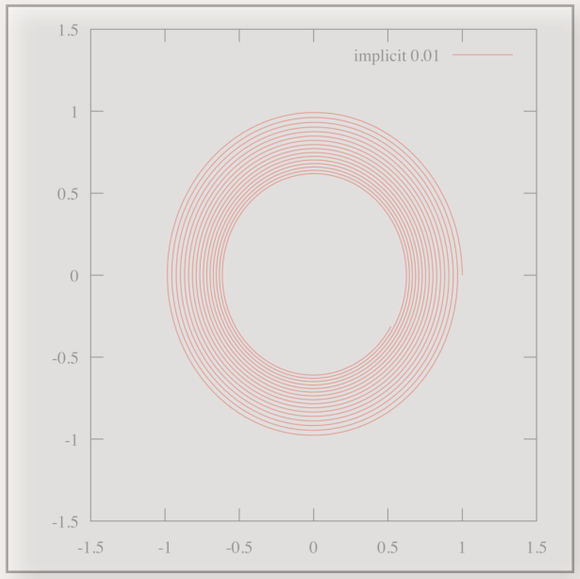
Elliot English*
University of British Columbia

Robert Bridson†
University of British Columbia

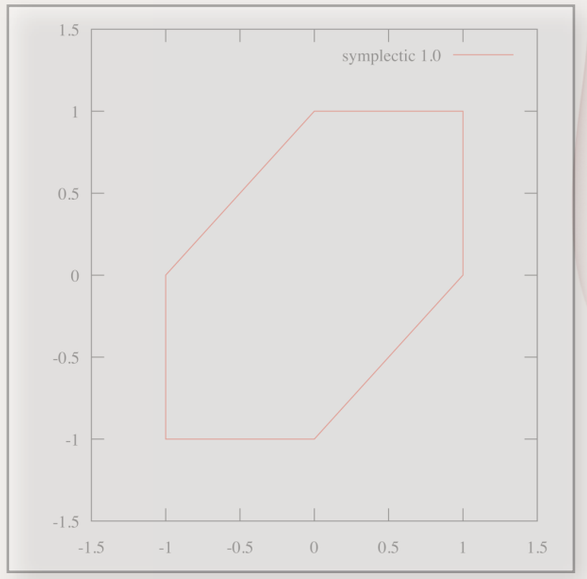
Developable Surfaces



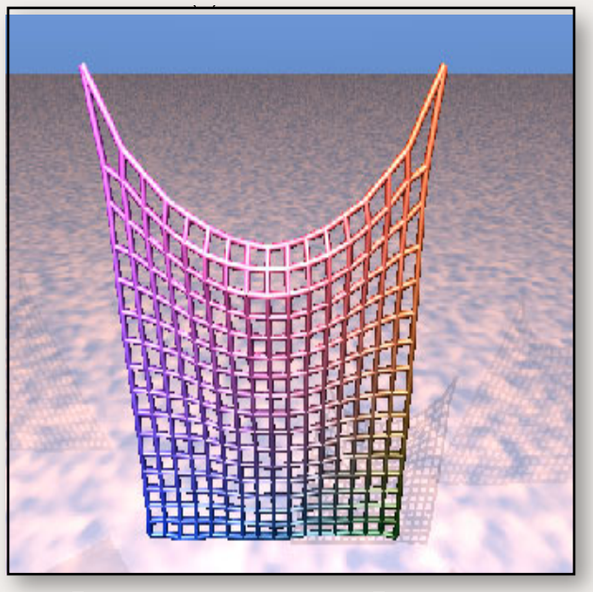
Overview



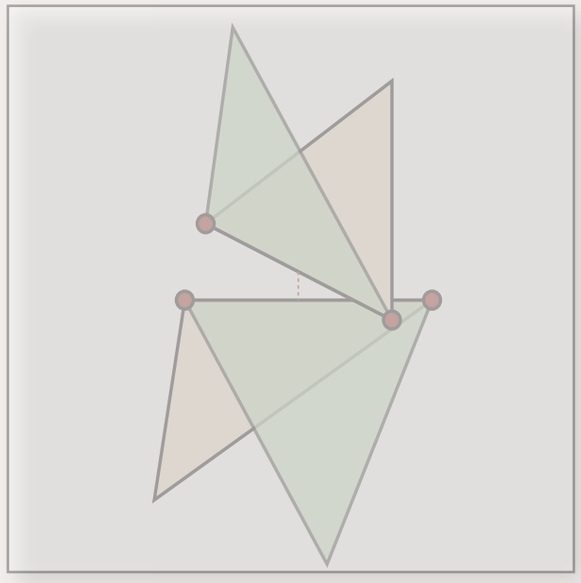
Implicit Integration



Symplectic Integration

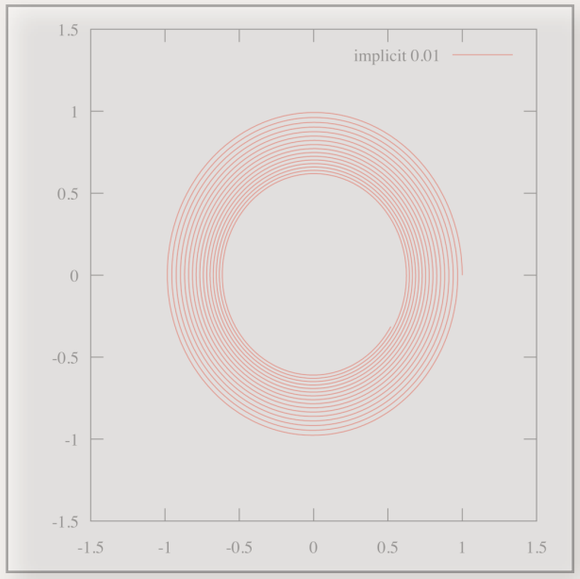


Cloth Simulation

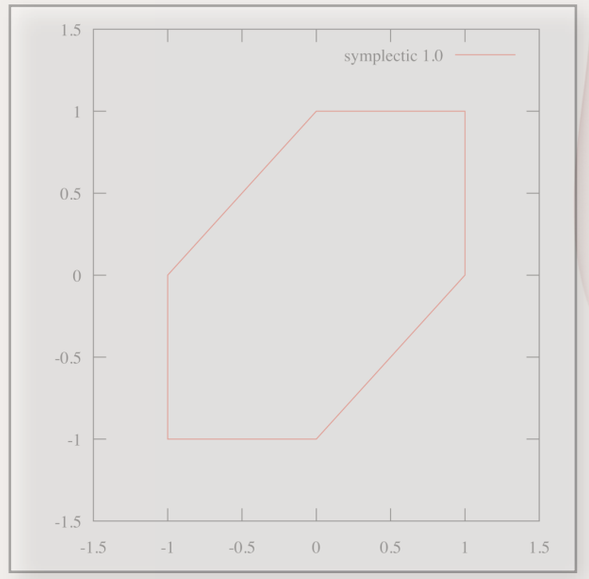


Cloth Collisions

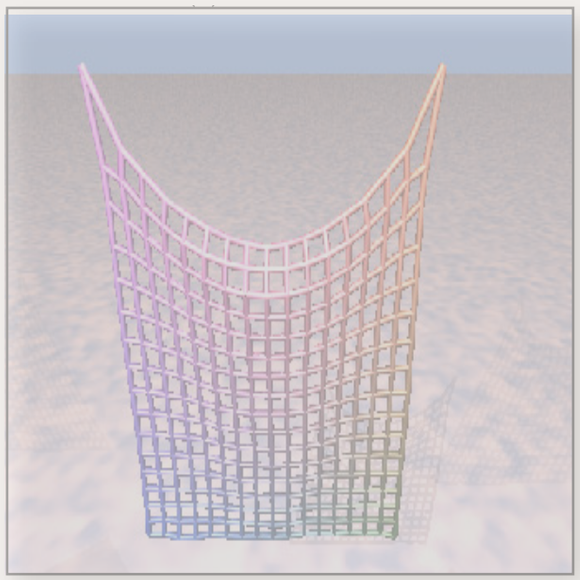
Overview



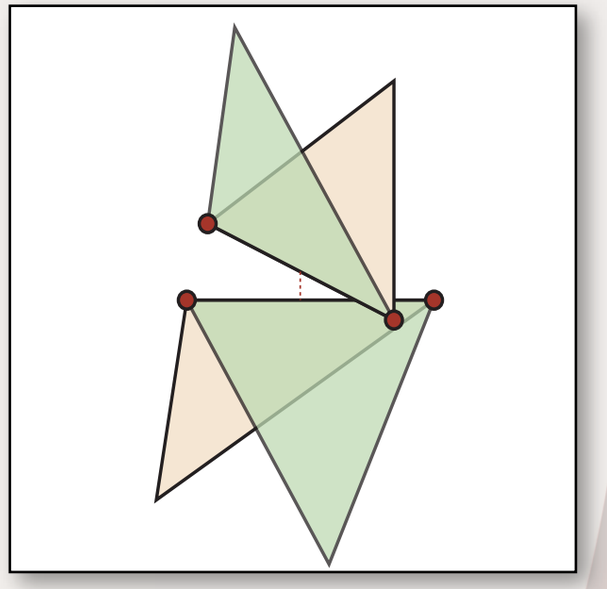
Implicit Integration



Symplectic Integration



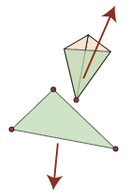
Cloth Simulation



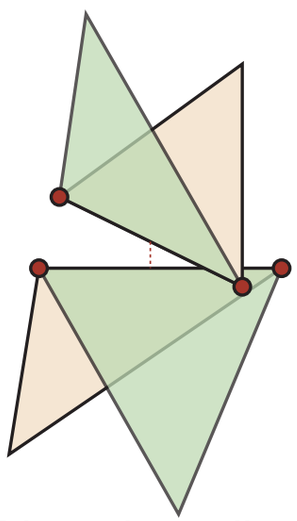
Cloth Collisions

Collision Detection

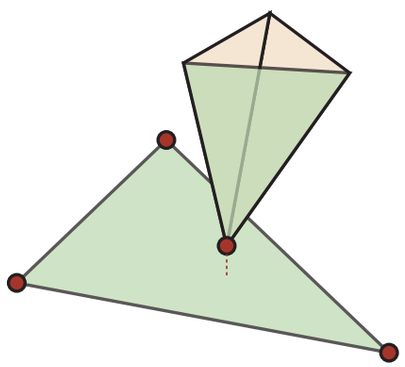
Collision detection



- Generally need to do triangle-triangle collision checks:



Edge-edge collision

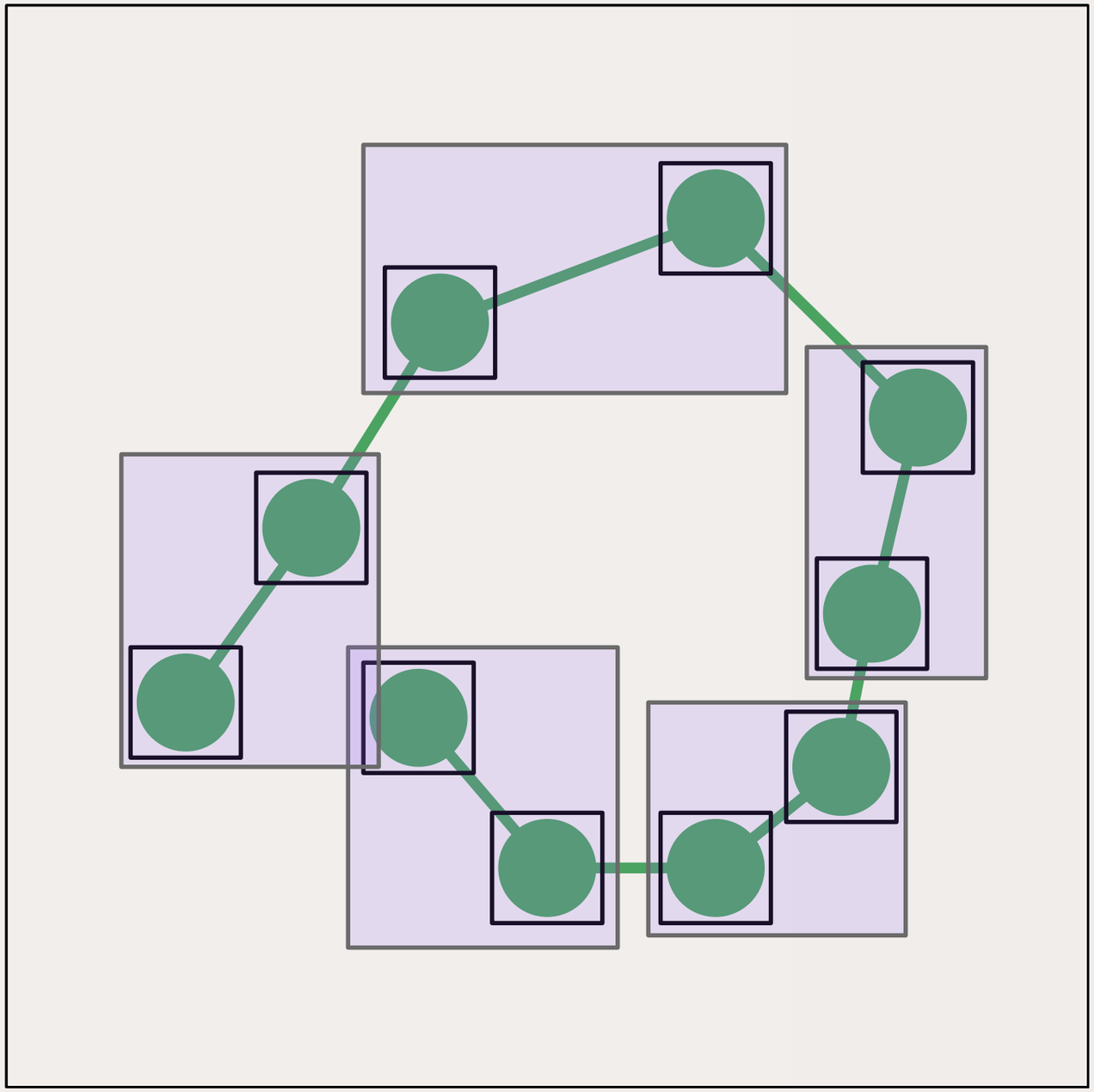


Point-face collision

Cloth Animation

Christopher Twigg
March 4, 2003

Cloth Collision Detection



Curvature Optimization

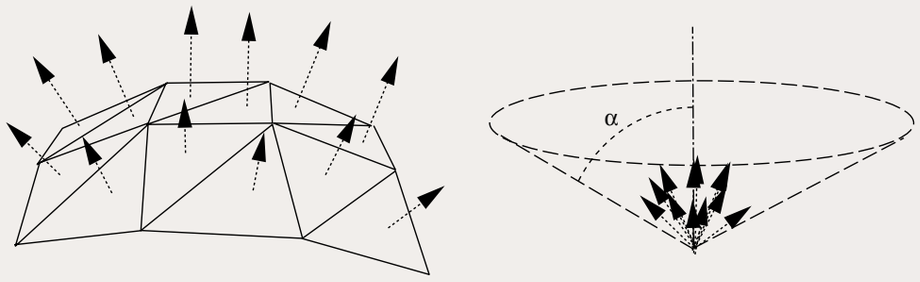


Figure 1: Cone including normals to triangles of a zone of the cloth surface.

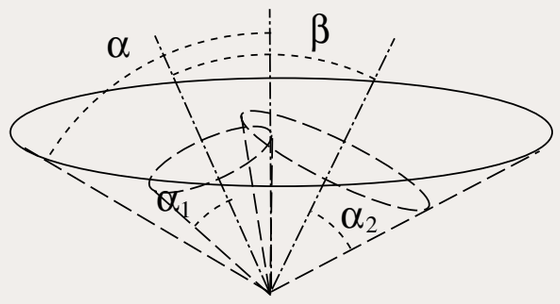
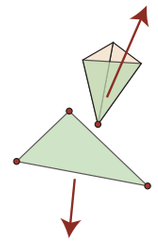


Figure 2: Cone (angle α) enclosing its two "descendant" cones in the hierarchical tree (angles α_1 and α_2).

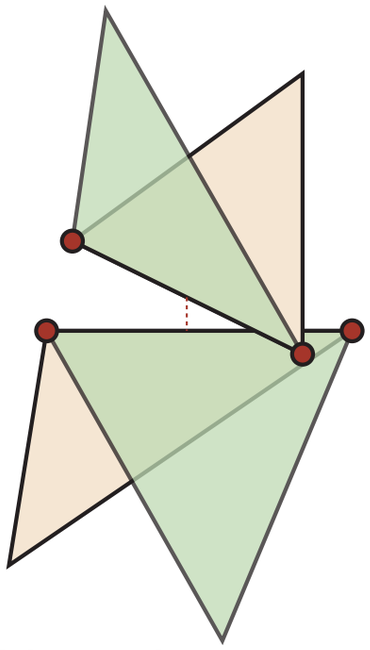
Collision and self-collision handling in cloth model dedicated to design garments

Xavier PROVOT
Institut National de Recherche en Informatique et Automatique (INRIA)
B.P. 105, 78153 Le Chesnay Cedex, France
Xavier.Provot@inria.fr

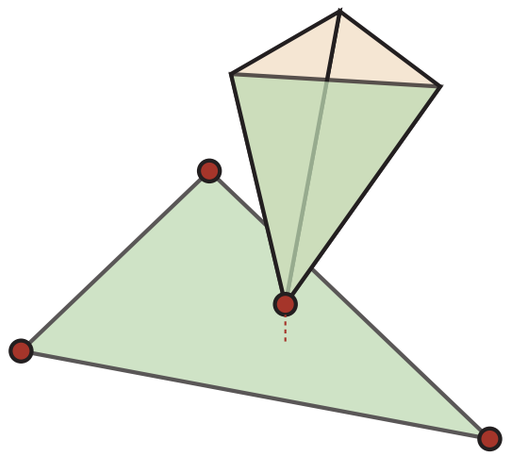
Collision detection



- We already covered this for deformable bodies
- Many of the same methods work, especially acceleration methods
- Generally need to do triangle-triangle collision checks:

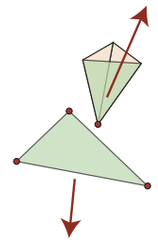


Edge-edge collision



Point-face collision

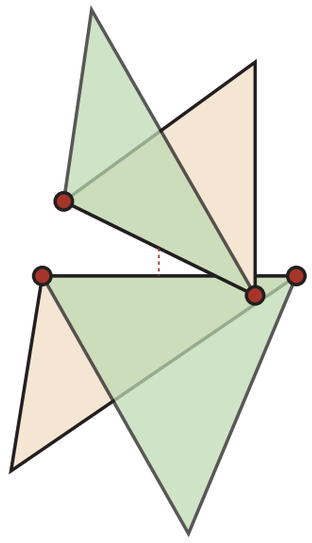
Robust collision detection

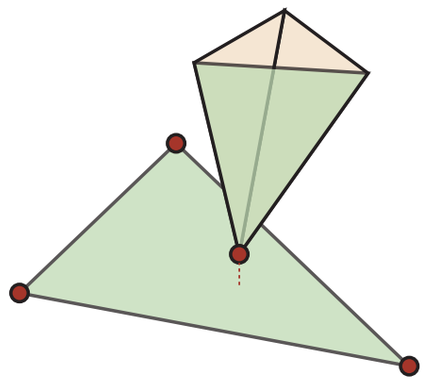


If triangles are moving too fast, they may pass through each other in a single timestep.

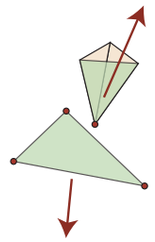
We can prevent this by checking for *any* collisions during the timestep (Provot [1997])

Note first that both point-face and edge-edge collisions occur when the appropriate 4 points are *coplanar*

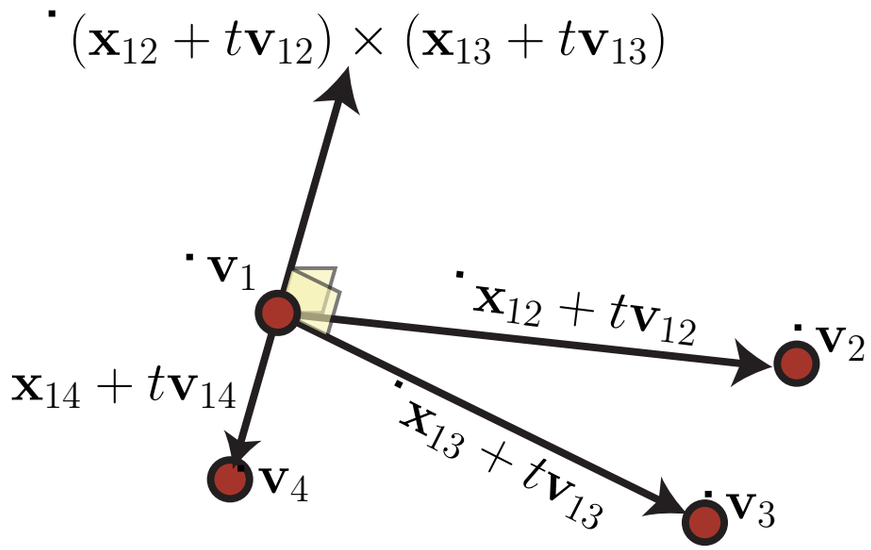




Robust collision detection (2)



Detecting time of coplanarity - assume linear velocity throughout timestep:

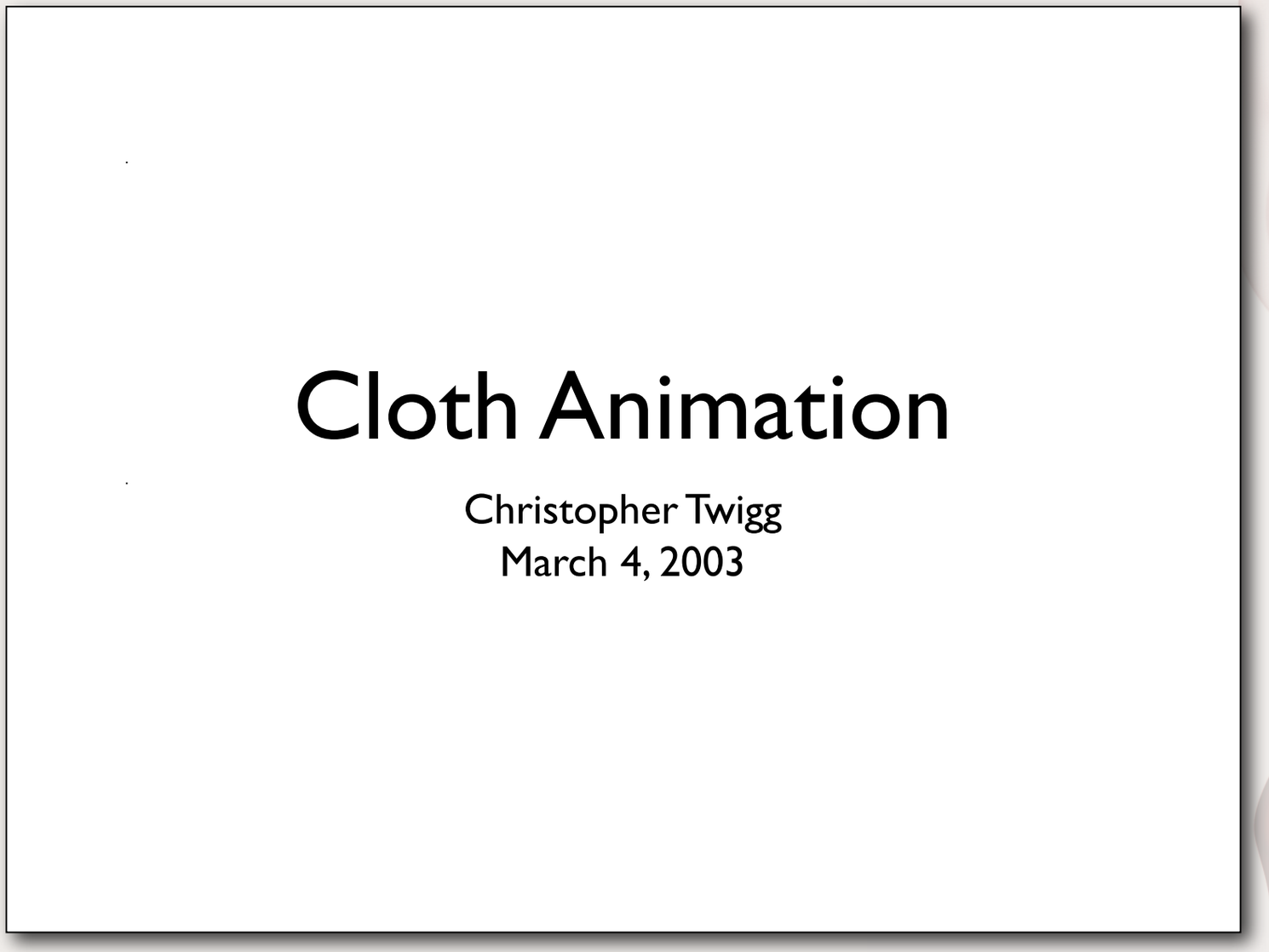


So the problem reduces to finding roots of the cubic equation

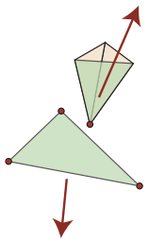
$$\big((\mathbf{x}_{12} + t\mathbf{v}_{12}) \times (\mathbf{x}_{13} + t\mathbf{v}_{13})\big) \cdot (\mathbf{x}_{14} + t\mathbf{v}_{14})$$

Once we have these roots, we can plug back in and test for triangle adjacency.

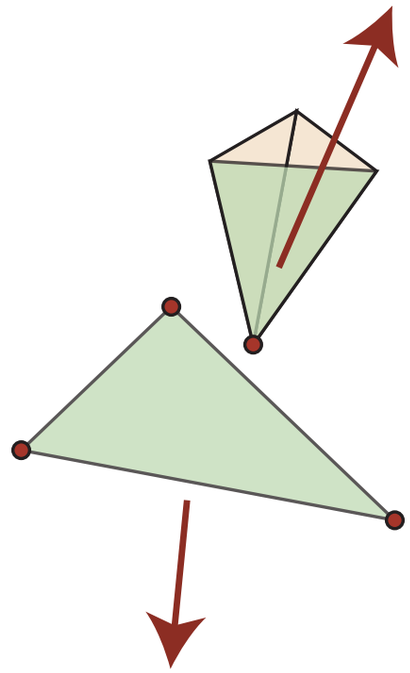
Collision Response



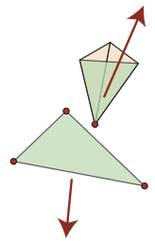
Collision response



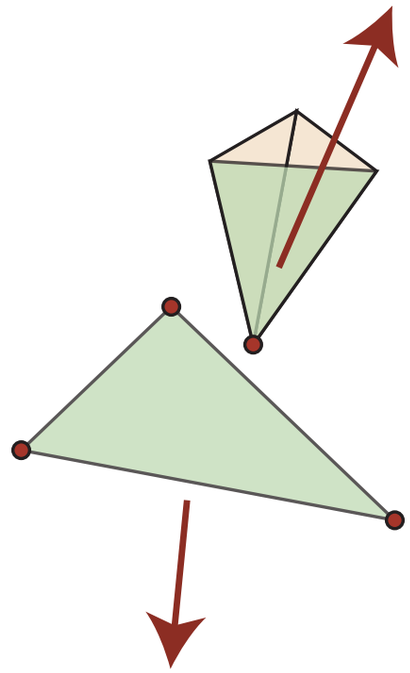
- 4 basic options:
  - Constraint-based
  - Penalty forces
  - Impulse-based
  - Rigid body dynamics (will explain)



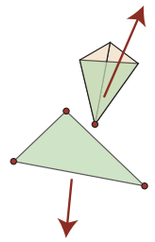
Collision response



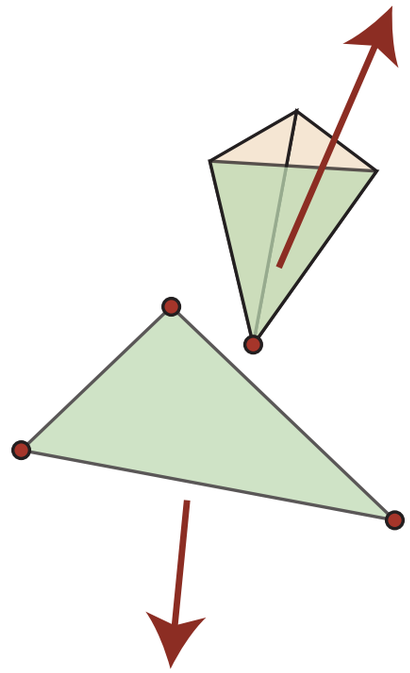
- 4 basic options:
  - Constraint-based
  - Penalty forces
  - Impulse-based
  - Rigid body dynamics (will explain)



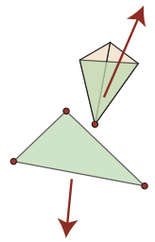
Collision response



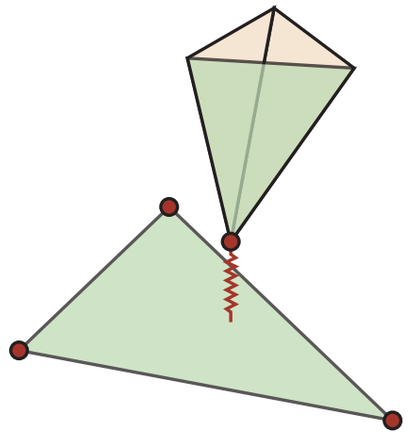
- 4 basic options:
  - Constraint-based
  - Penalty forces
  - Impulse-based
  - Rigid body dynamics (will explain)



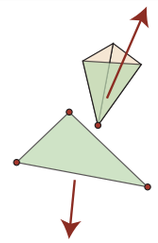
Penalty forces



- Apply a spring force that keeps particles away from each other
- Benefits:
  - Easy to fit into an existing simulator
  - Works with all kinds of collisions (use barycentric coordinates to distribute responses among vertices)
- Drawbacks:
  - Hard to tune: if force is too weak, it will sometimes fail; if force is too strong, it will cause the particles to "float" and "wiggle"

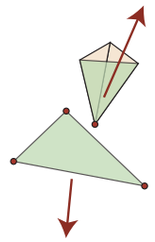


Penalty forces (2)

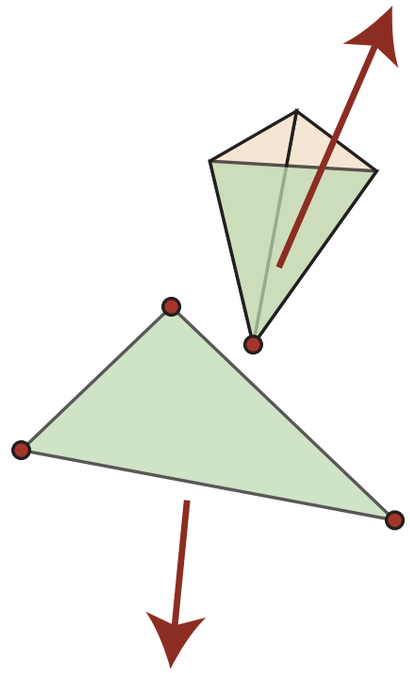


- In general, penalty forces are not inelastic (springs store energy)
- Can be made less elastic by limiting force when particles are moving away
- Some kind of additional damping may be needed to control deformation rate along surface

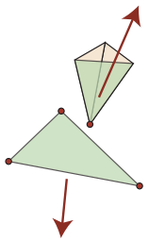
Collision response



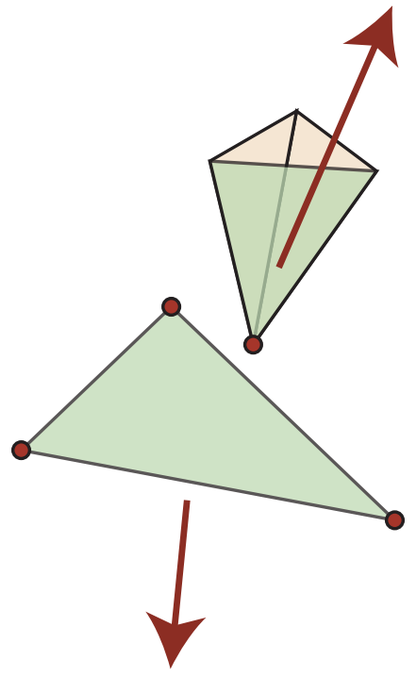
- 4 basic options:
  - Constraint-based
  - Penalty forces
  - Impulse-based
  - Rigid body dynamics (will explain)



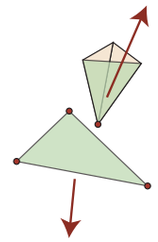
Collision response



- 4 basic options:
  - Constraint-based
  - Penalty forces
  - Impulse-based
  - Rigid body dynamics (will explain)



Impulses

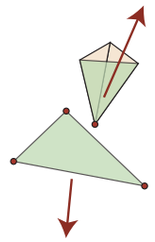


- "Instantaneous" change in momentum

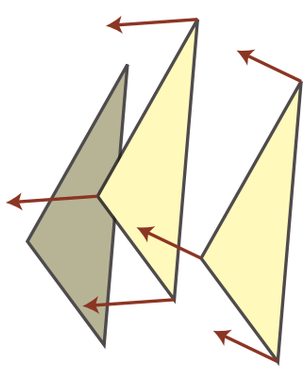
$$\mathbf{J} = \int_{t_i}^{t_f} \mathbf{F}\, dt = \mathbf{p}_f - \mathbf{p}_i$$

- Generally applied outside the simulator timestep
- Benefits
  - Correctly stops all collisions (no sloppy spring forces)
- Drawbacks
  - Can have poor numerical performance
  - Handles persistent contact poorly

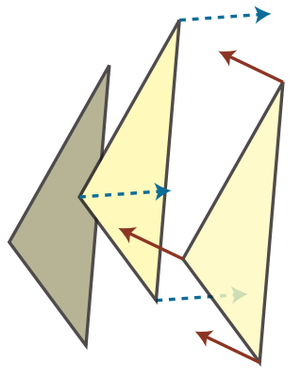
Impulses (2)



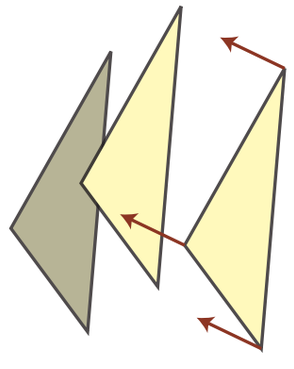
Iteration is generally necessary to remove all collisions.



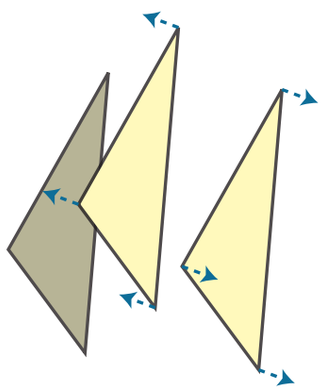
Initial state



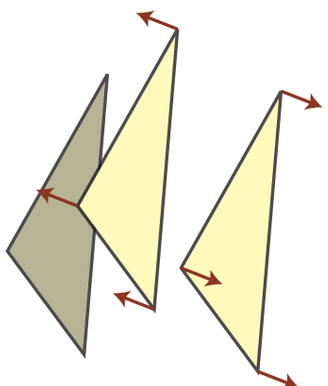
Apply impulse responses



Intermediate state



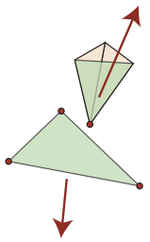
Apply impulse responses



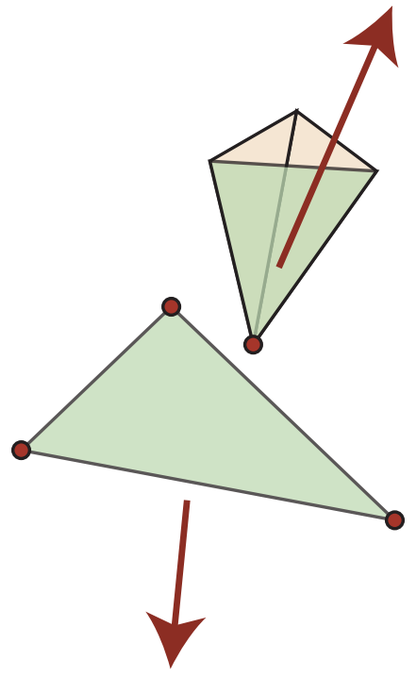
Final state

Convergence may be slow in some cases.

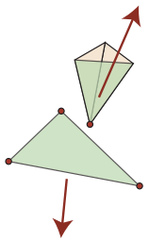
Collision response



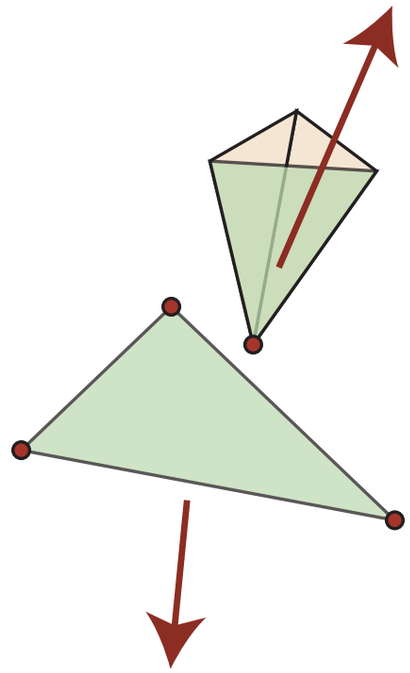
- 4 basic options:
  - Constraint-based
  - Penalty forces
  - Impulse-based
  - Rigid body dynamics (will explain)

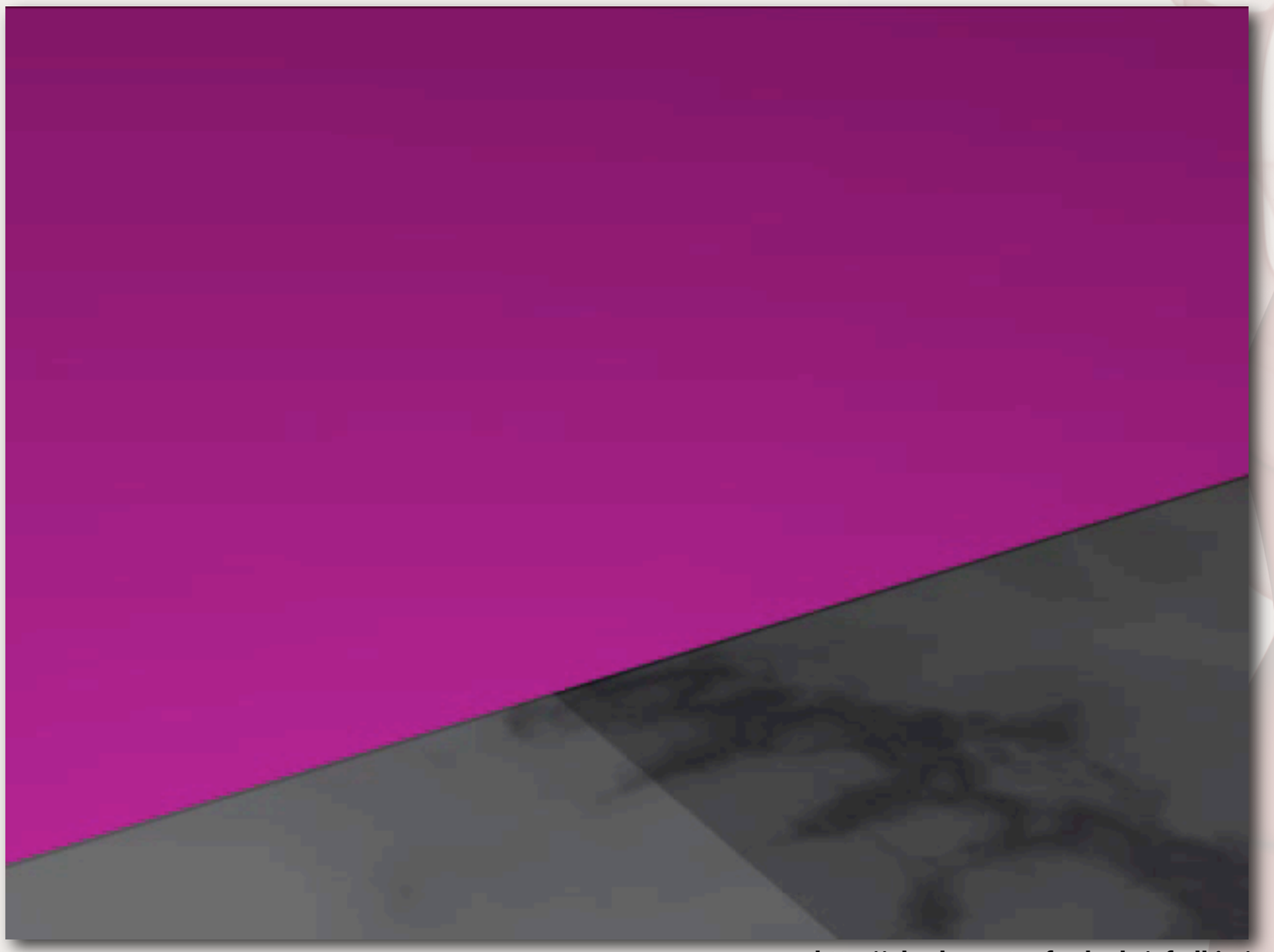


Collision response

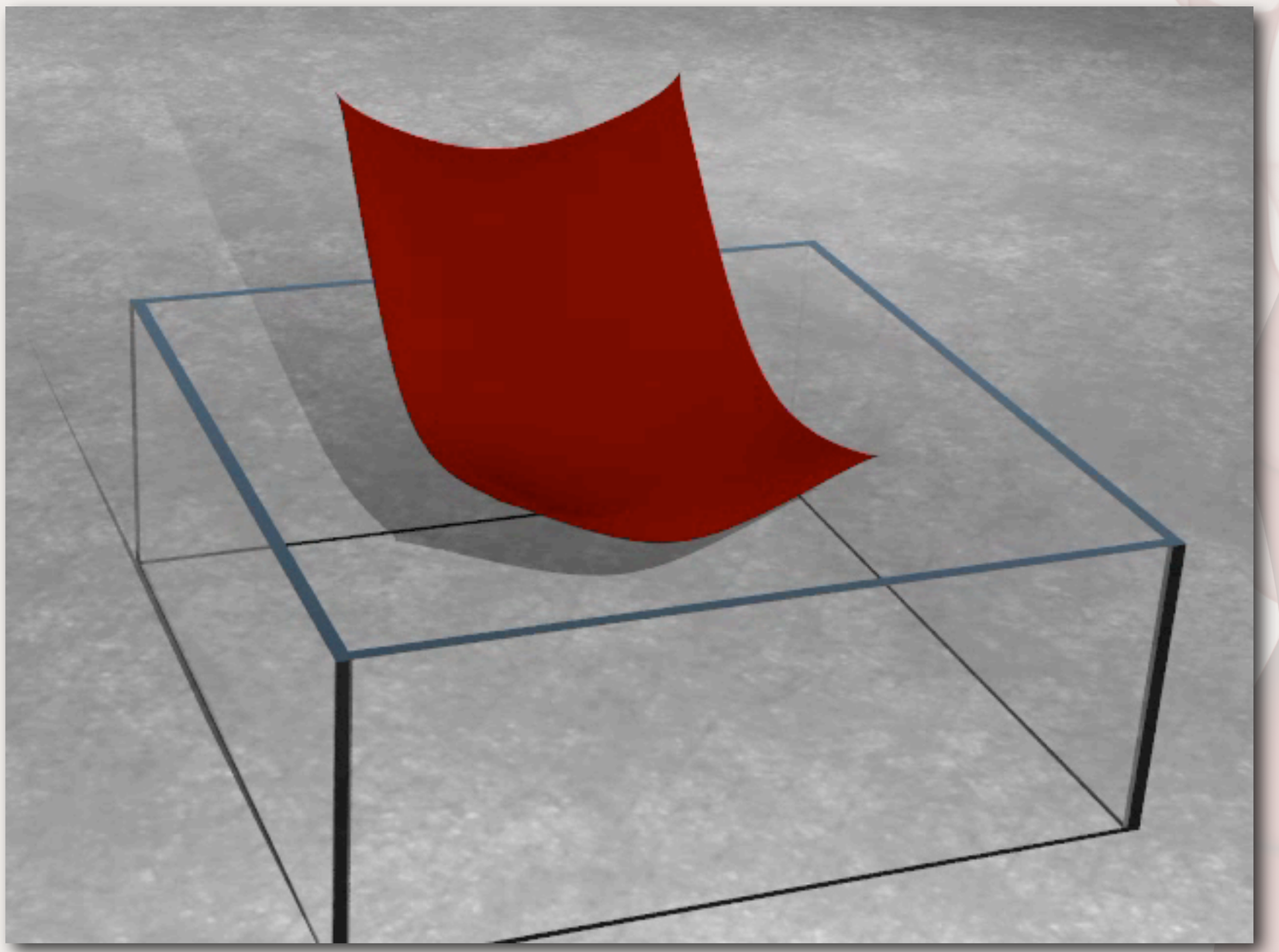


- 4 basic options:
  - Constraint-based
  - Penalty forces
  - Impulse-based
  - Rigid body dynamics (will explain)

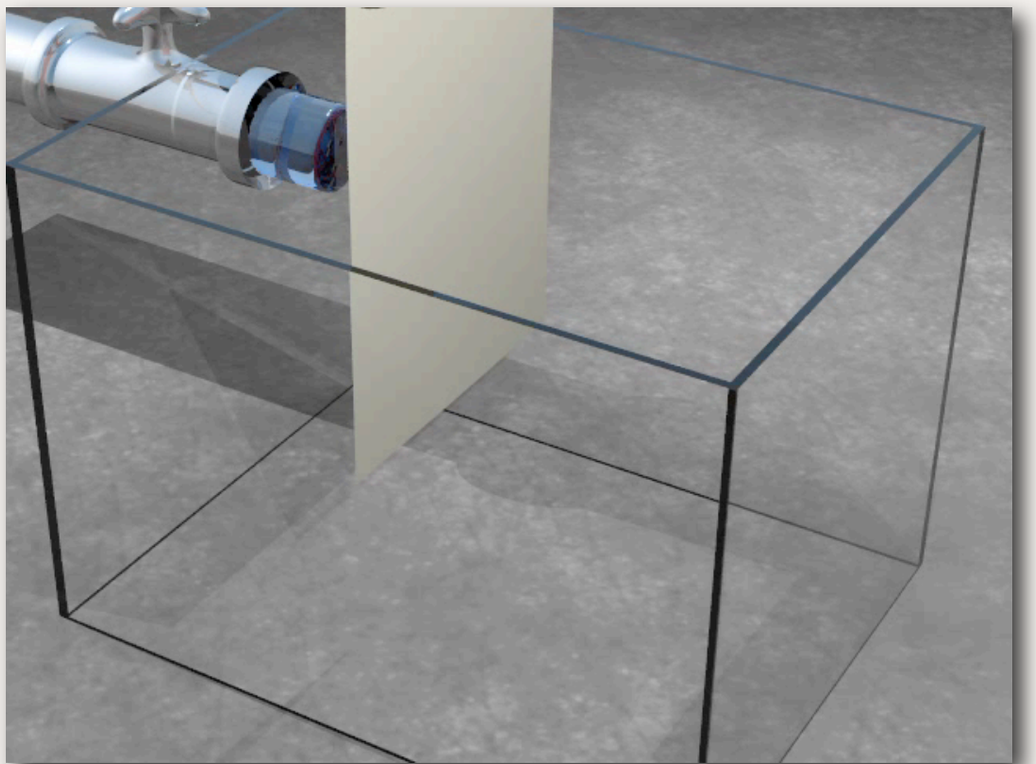




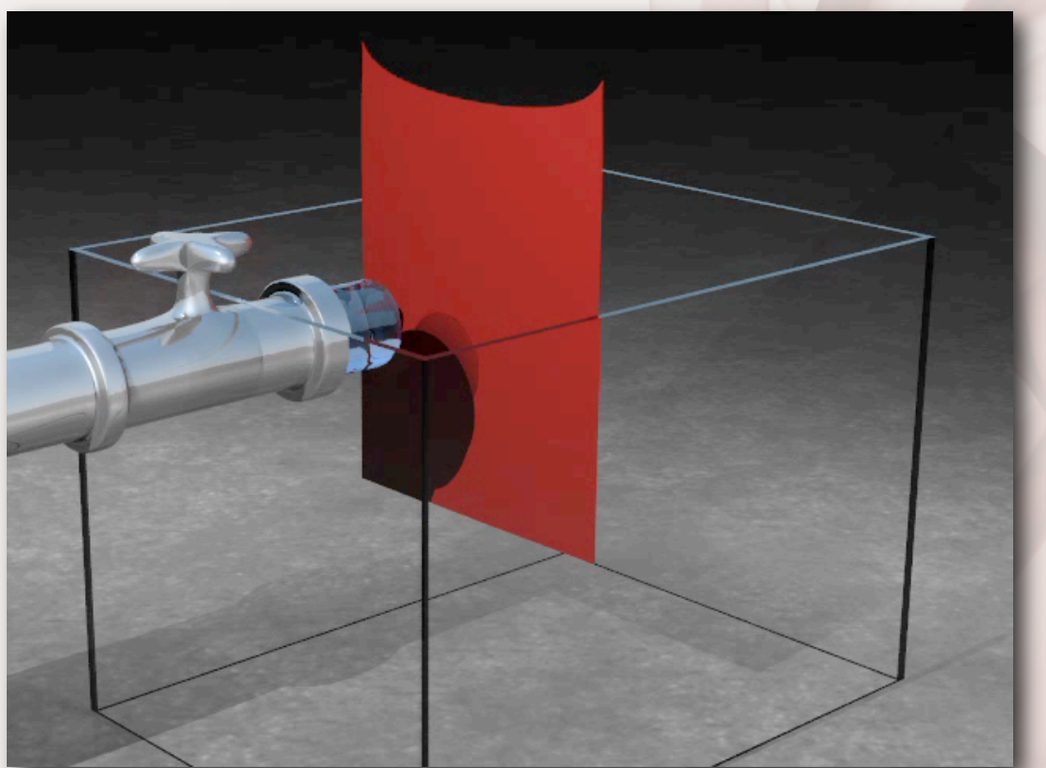
http://physbam.stanford.edu/~fedkiw/



http://physbam.stanford.edu/~fedkiw/

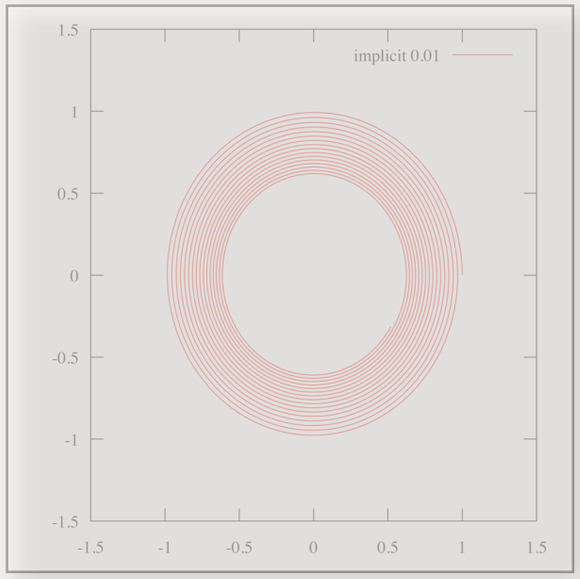


http://physbam.stanford.edu/~fedkiw/

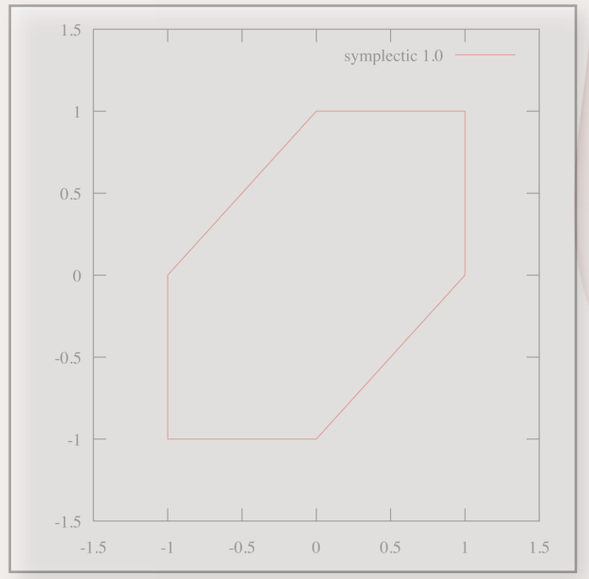


http://physbam.stanford.edu/~fedkiw/

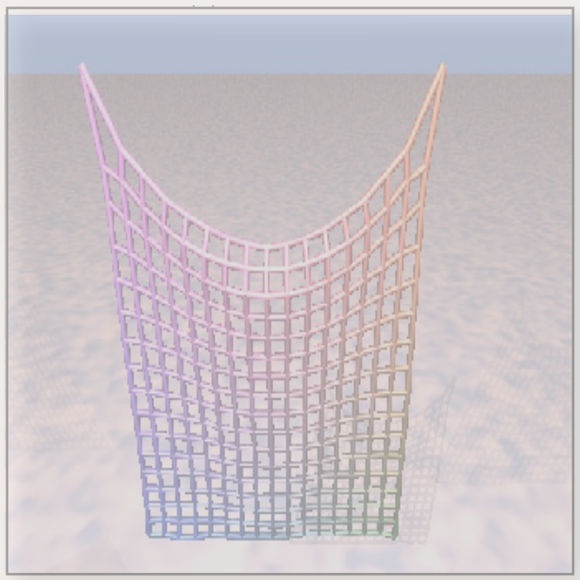
Overview



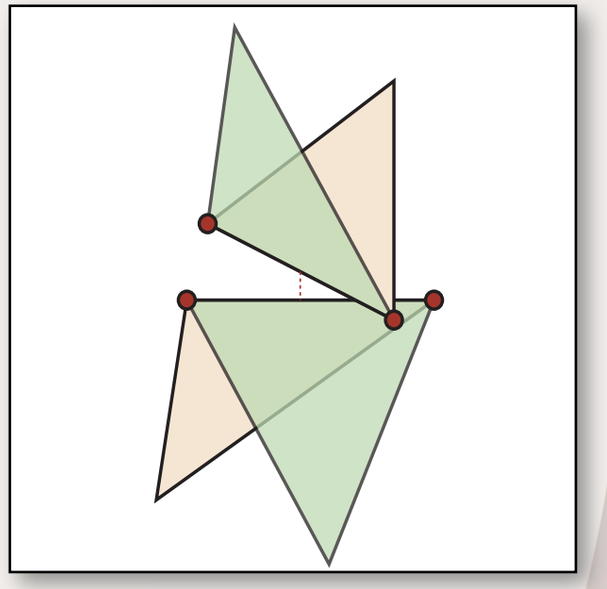
Implicit Integration



Symplectic Integration

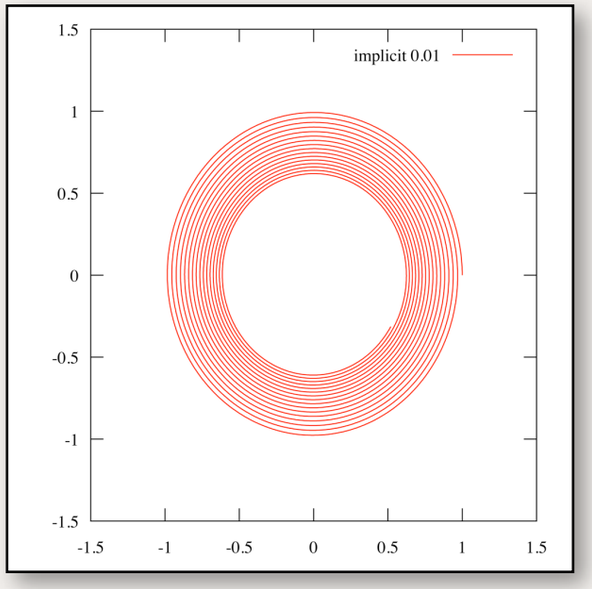


Cloth Simulation

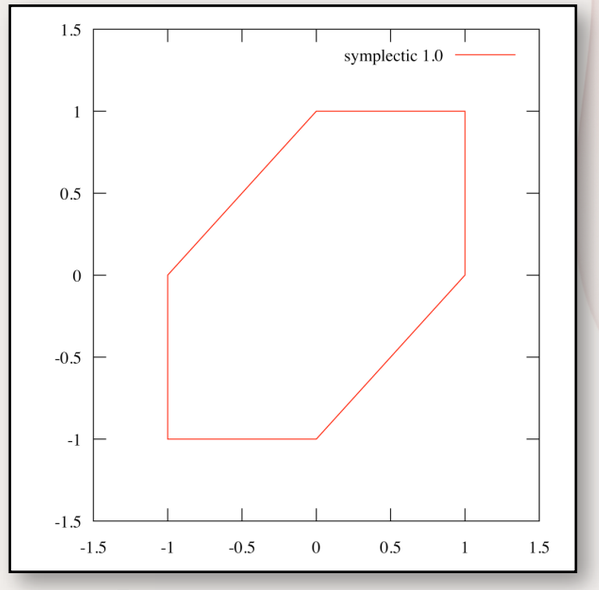


Cloth Collisions

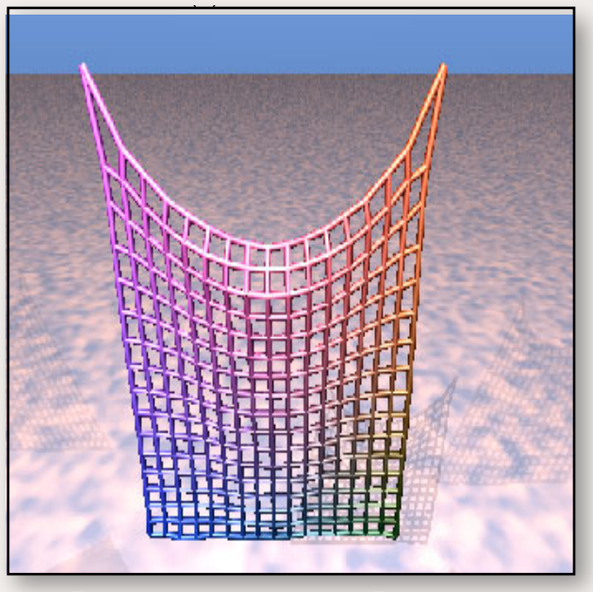
Overview



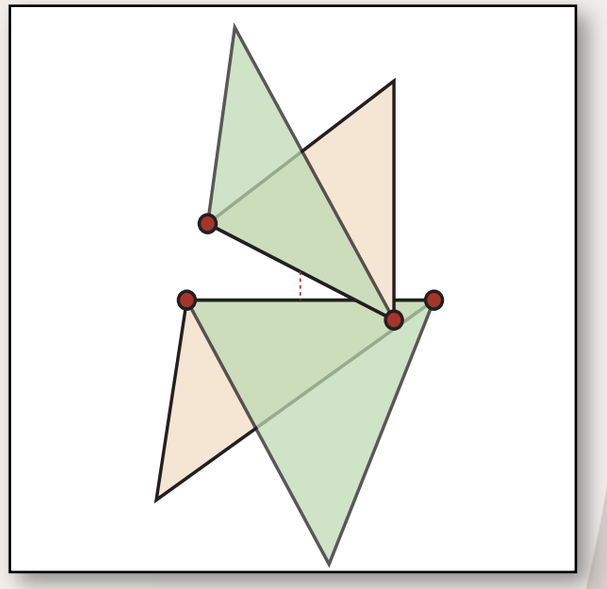
Implicit Integration



Symplectic Integration

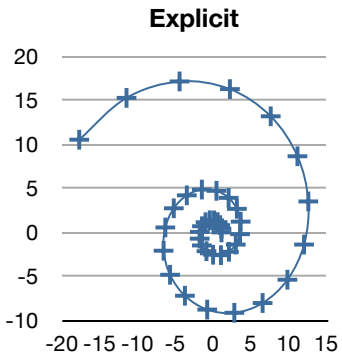
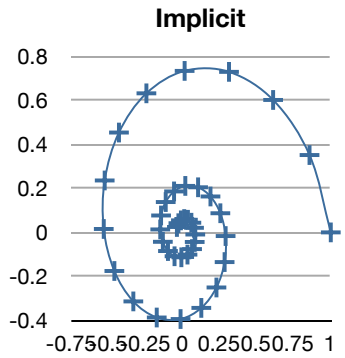
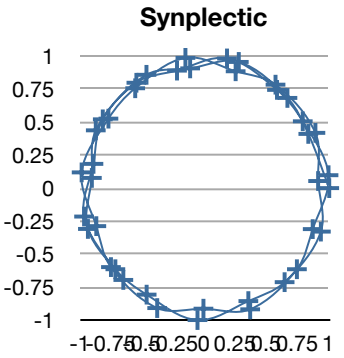


Cloth Simulation



Cloth Collisions

dt= 0.41

Explicit		Implicit		Symplectic	
x	y	x	y	x	y
1.0	0.0	1.0	0.0	1.0	0.0
1	0.41	0.8560910881	0.3509973461	0.8319	0.41
0.8319	0.82	0.6096928141	0.6009713999	0.6638	0.682158
0.4957	1.161079	0.3110132182	0.7284868194	0.27253044	0.954316
0.01965761	1.364316	0.0105587041	0.732815888	-0.11873912	0.9056329608
-0.53971195	1.3723756201	-0.248177219	0.6310632281	-0.470088588	0.8569499216
-1.102385954	1.1510937206	-0.433963824	0.4531380602	-0.821438056	0.5201603188
-1.57433438	0.6991154794	-0.530562819	0.2356073044	-0.896620049	0.1833707159
-1.860971726	0.0536383837	-0.53690764	0.0154751722	-0.971802043	-0.215068122
-1.882963464	-0.709360024	-0.46507359	-0.175205	-0.72026419	-0.613506959
-1.592125854	-1.481375044	-0.336648866	-0.313231034	-0.468726336	-0.805684757
-0.984762086	-2.134146644	-0.178258832	-0.386317156	-0.059602689	-0.997862555
-0.109761961	-2.537899099	-0.017009501	-0.393291051	0.3495209587	-0.854558962
0.9307766692	-2.582901503	0.1234824329	-0.342663253	0.6411356599	-0.711255369
1.9897662856	-2.201283069	0.2259861029	-0.250008951	0.9327503611	-0.328827721
2.8922923439	-1.385478892	0.2812171671	-0.134709913	0.9107743909	0.0535999274
3.4603386896	-0.199639031	0.2880303324	-0.016617476	0.8887984206	0.4180072798
3.5421906923	1.2190998318	0.2524128908	0.0868718088	0.5680084214	0.7824146323
3.0423597613	2.6713980156	0.185596652	0.1629664362	0.2472184221	0.8837741854
1.9470865749	3.9187655177	0.1016868531	0.204658046	-0.156686411	0.9851337385
0.3403927126	4.7170710134	0.0152187778	0.2108977448	-0.560591243	0.7552913287
-1.593606403	4.8566320256	-0.060995889	0.1858894305	-0.7760253	0.5254489189
-3.584825533	4.2032534004	-0.117464734	0.1377288897	-0.991459357	0.1189505826
-5.308159428	2.7334749317	-0.148902986	0.0766786653	-0.873564778	-0.287547754
-6.42888415	0.5571295664	-0.154388528	0.013379369	-0.755670199	-0.597372535
-6.657307272	-2.078712935	-0.136866766	-0.042736005	-0.383719299	-0.907197317
-5.805034968	-4.808208916	-0.102170194	-0.084625784	-0.011768399	-0.91202236
-3.833669313	-7.188273253	-0.057763567	-0.108308847	0.3641390366	-0.916847404
-0.886477279	-8.760077672	-0.011434757	-0.112997097	0.7400464722	-0.61342835
2.7051545665	-9.123533356	0.0298724877	-0.100749377	0.8671502839	-0.310009297
6.4458032425	-8.014419984	0.0609363345	-0.07576548	0.9942540956	0.0976348824
9.7317154358	-5.371640654	0.0787605353	-0.043473661	0.7870896803	0.5052790616
11.934088104	-1.381637326	0.0826853319	-0.009572674	0.5799252651	0.7430484203
12.500559408	3.511338797	0.0741461591	0.0208272507	0.1777899757	0.9808177789
11.060910501	8.6365681541	0.0561655563	0.0438551288	-0.224345314	0.8888362003
7.5199175576	13.171541459	0.0326897984	0.0572579461	-0.551055709	0.7968546217
2.1195855593	16.254707658	0.0078880579	0.0604920499	-0.877766103	0.4369705193
-4.544844581	17.123737737	-0.014479653	0.0545553922	-0.909371534	0.0770864169
-11.56557705	15.260351459	-0.0315447	0.0416220653	-0.940976965	-0.308714139
-17.82232115	10.518464868	-0.041614371	0.0245601733	-0.65622594	-0.694514695

Matrix		Inverse		Product	
1	0.41	0.8560910881	-0.350997346	1	0
-0.41	1	0.3509973461	0.8560910881	0	1