FEM Simulation of 3D Deformable Solids: A practitioner's guide to theory, discretization and model reduction.

ACM SIGGRAPH 2012 Course

Part 2: Model Reduction

Jernej Barbič
Univ. of Southern California

About me

- Assistant professor in CS
  at Univ. of Southern California
  in Los Angeles

- Post-doc at MIT

- PhD, Carnegie Mellon University

About me

- Background:
  BSc Mathematics
  PhD Computer Science

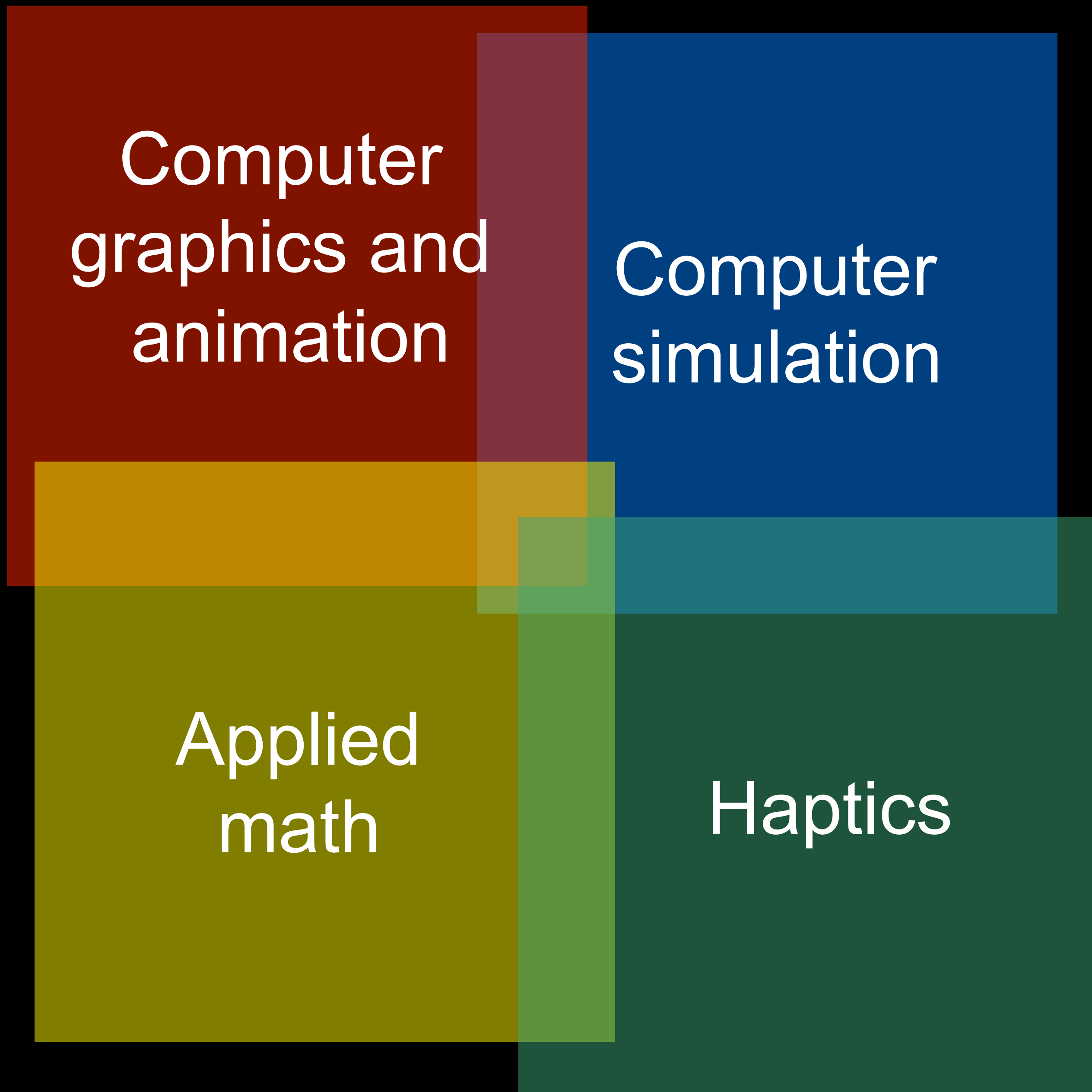
- Research interests:
  graphics, animation, real-time physics, control, sound, haptics

About me

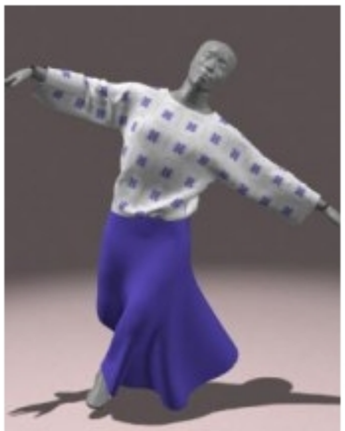
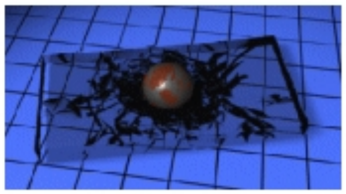
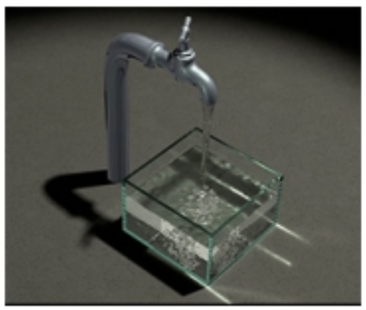
FROM mathematics,

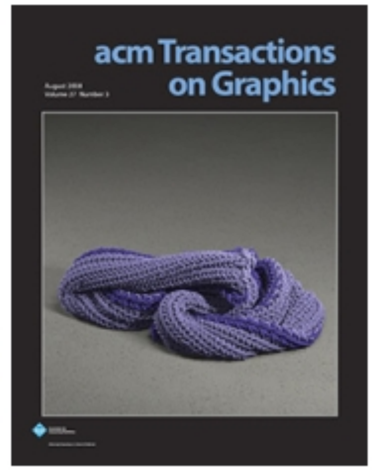
TO computer graphics,

TO mechanics.


Computer graphics and animation
Computer simulation
Applied math
Haptics

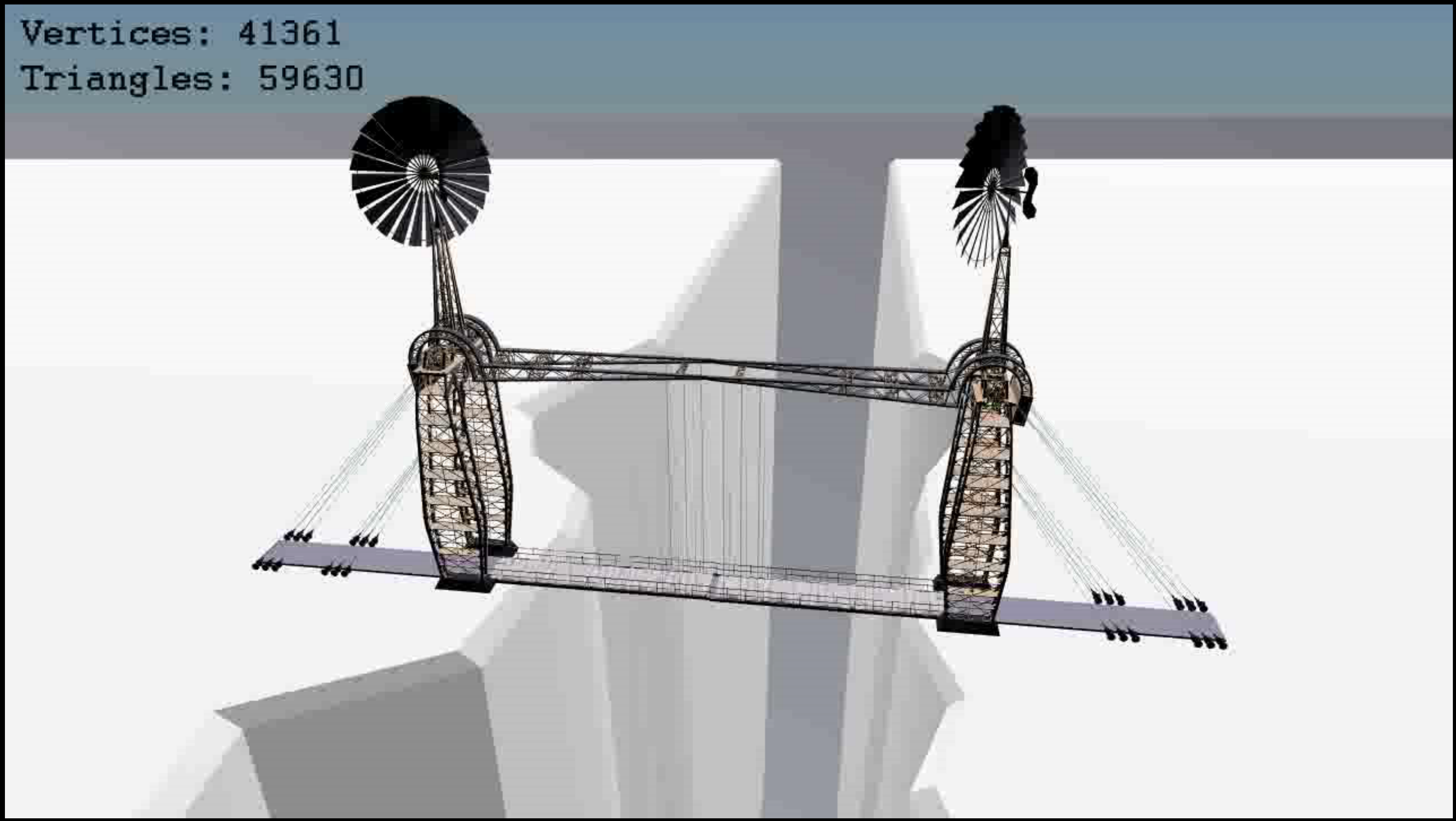
Physically Based Modeling

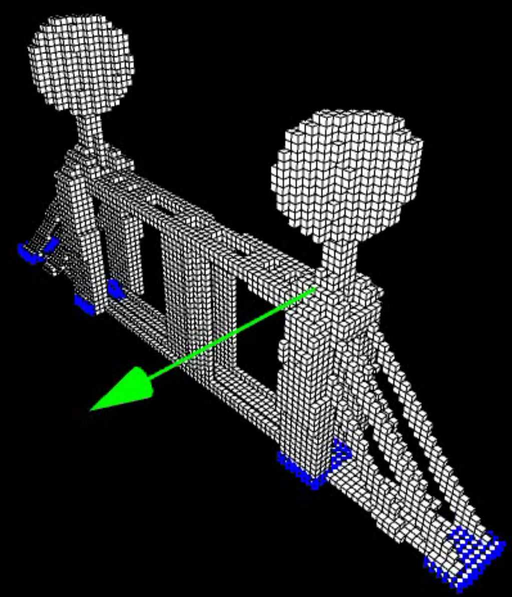
Anything one man can imagine,
other men can make real.
Jules Verne

Deformable object simulations



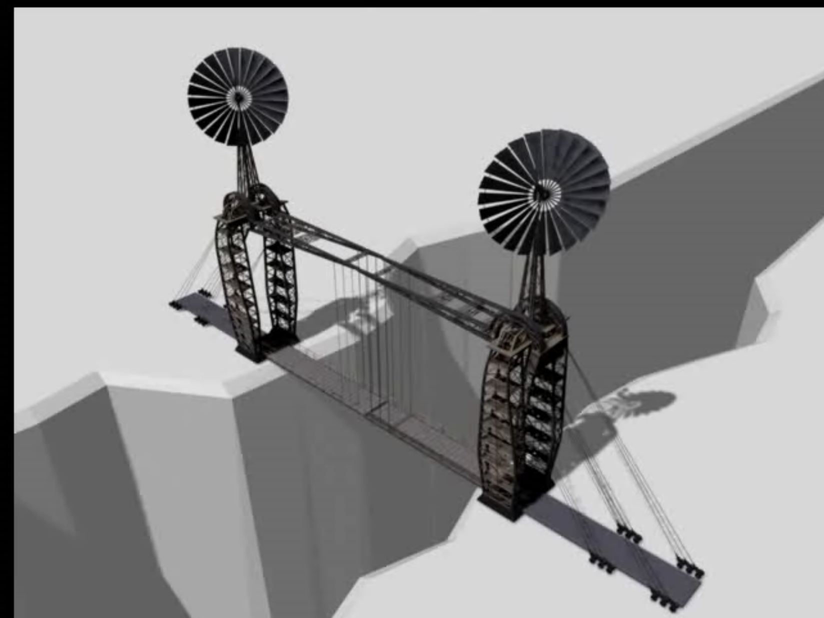
Task: Compute the dynamic deformations of the bridge under given external forces.

Deformable objects are computationally challenging



FEM Simulation

Computation time: 9.5 hours



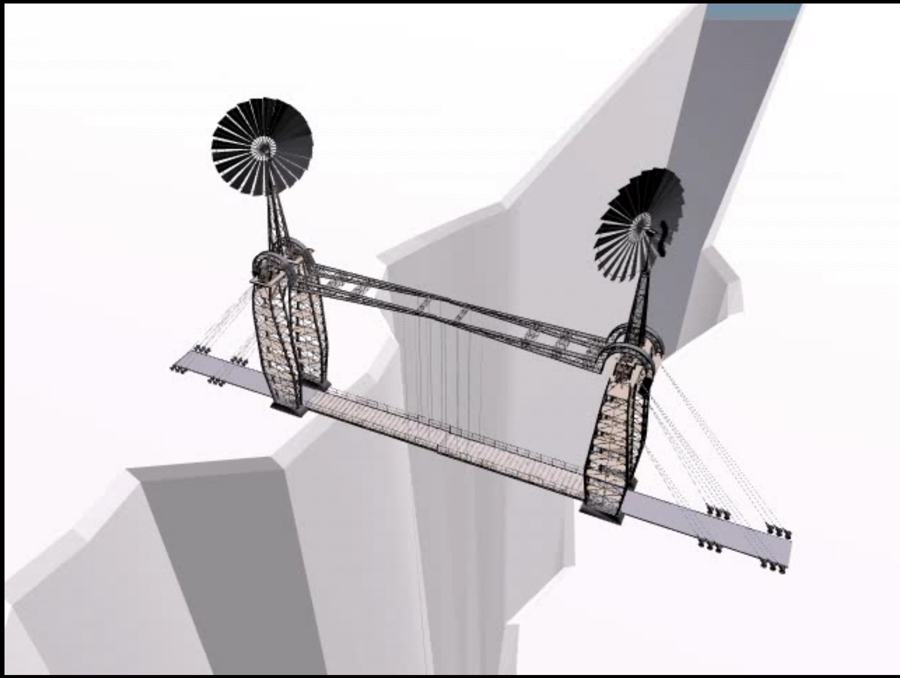
Rendered triangle mesh

Non-interactive simulation

35,000 DOFs, 2000 timesteps
Blue vertices = fixed

Real-time deformable objects

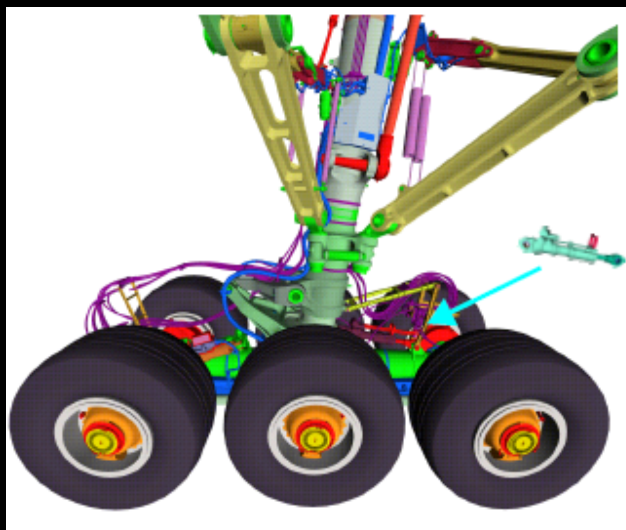
- 30 Hz for graphics
- 1000 Hz for force feedback
- 44100 Hz for sound
- Difficult !!!



[Barbic and James, SIGGRAPH 2005]

Real-time simulation
65 microsec / timestep
Speedup: 108,000x

Applications



[Source: Boeing]

Can the components of this Boeing 777 landing gear be assembled?

[Source: UW Dept of Surgery]

Surgery simulation
(artist illustration)

[Source: Crytek (Far Cry)]

Make bridges deformable?

Outline

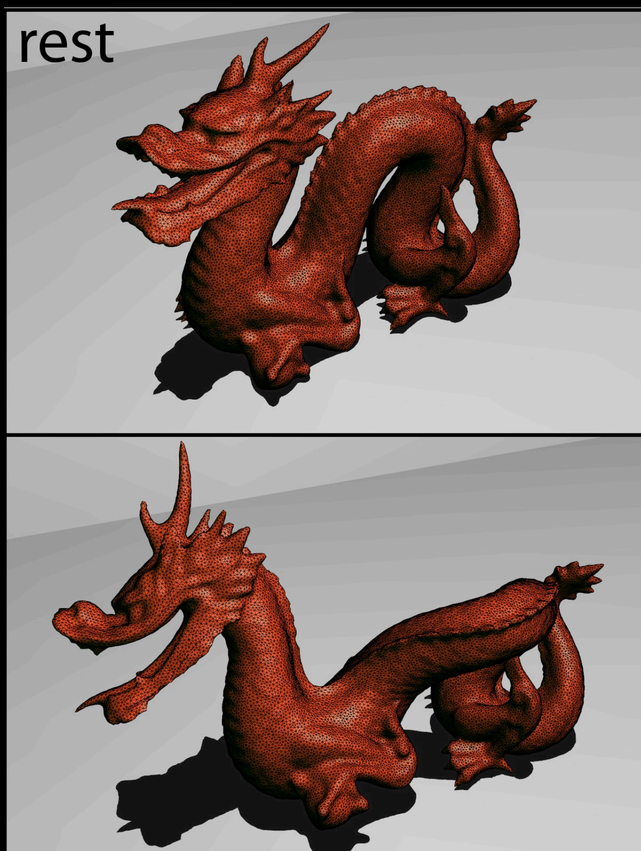
- Vega FEM
- Introduction to Model Reduction
- Linear Modal Analysis
- Model Reduction of Nonlinear Deformations
- Applications of Model Reduction

Jurij Vega (1754-1802)
Slovenian mathematician,
physicist and artillery officer

Vega FEM:

A free physics library to simulate
3D nonlinear deformable objects

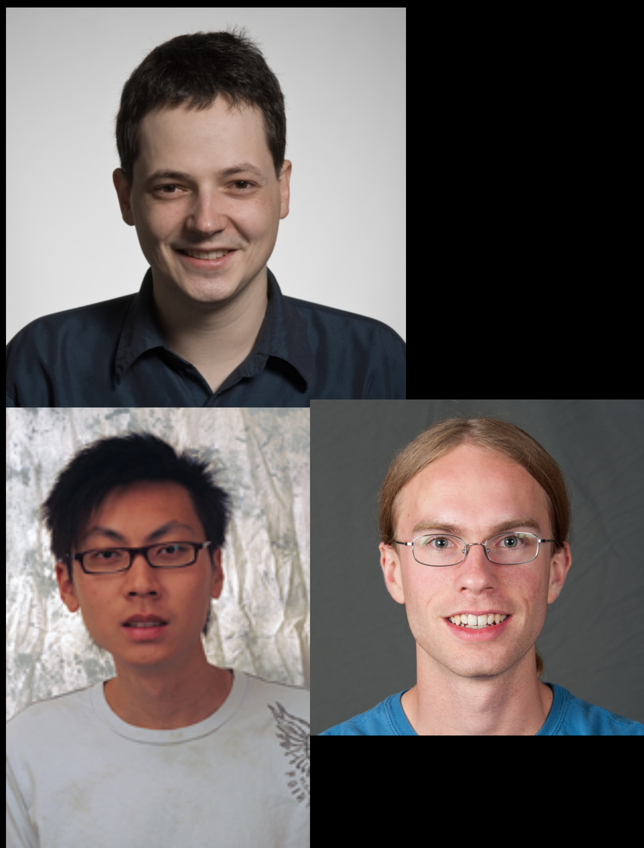
Vega



- Free and open source (BSD license), both for academia and industry
- 50,000 lines of C/C++ code
- No required external dependencies
- Released Aug 6, 2012

http://www.jernejbarbic.com/vega

Authors of Vega

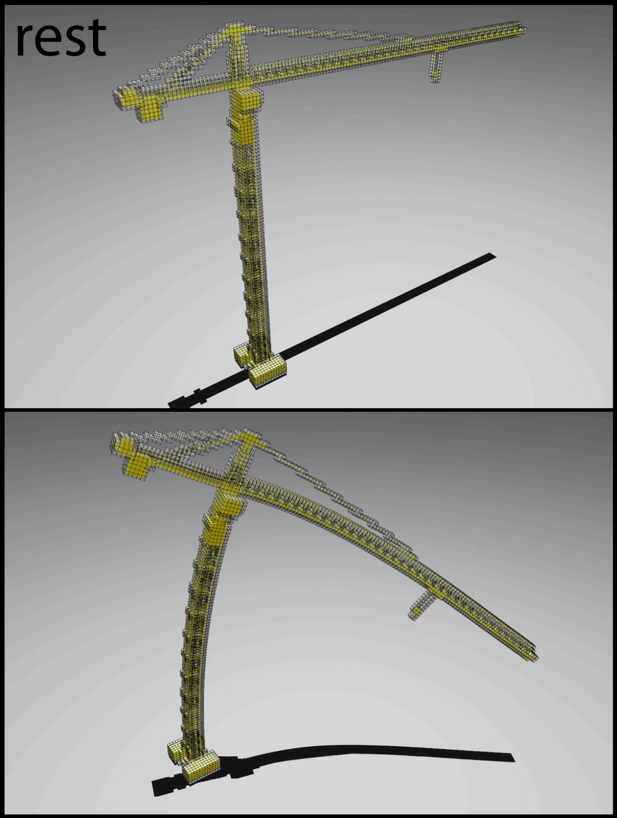


- Jernej Barbic
  (8 years of development)
- Fun Shing Sin
- Daniel Schroeder

http://www.jernejbarbic.com/vega

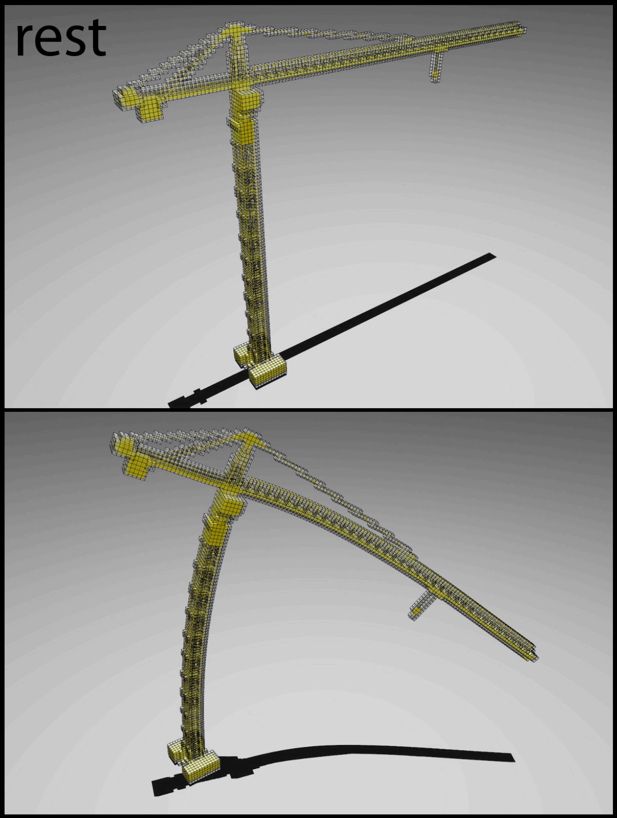
Deformable Models in Vega

- Linear FEM [Shabana 1990]
- Co-rotational linear FEM [Mueller and Gross 2004]
  Also with exact stiffness matrix [Barbic 2012] [Chao et al. 2010]
- Invertible FEM [Irving et al. 20
  [Teran et al. 2005]
- Saint-Venant Kirchhoff FEM
- Mass-spring Systems



Deformable Models in Vega

- All models provide internal elastic forces, AND tangent stiffness matrices, in ANY deformed configuration



- All models include support for multi-core CPU computing

- All models support non-homogeneous material properties

Integrators in Vega

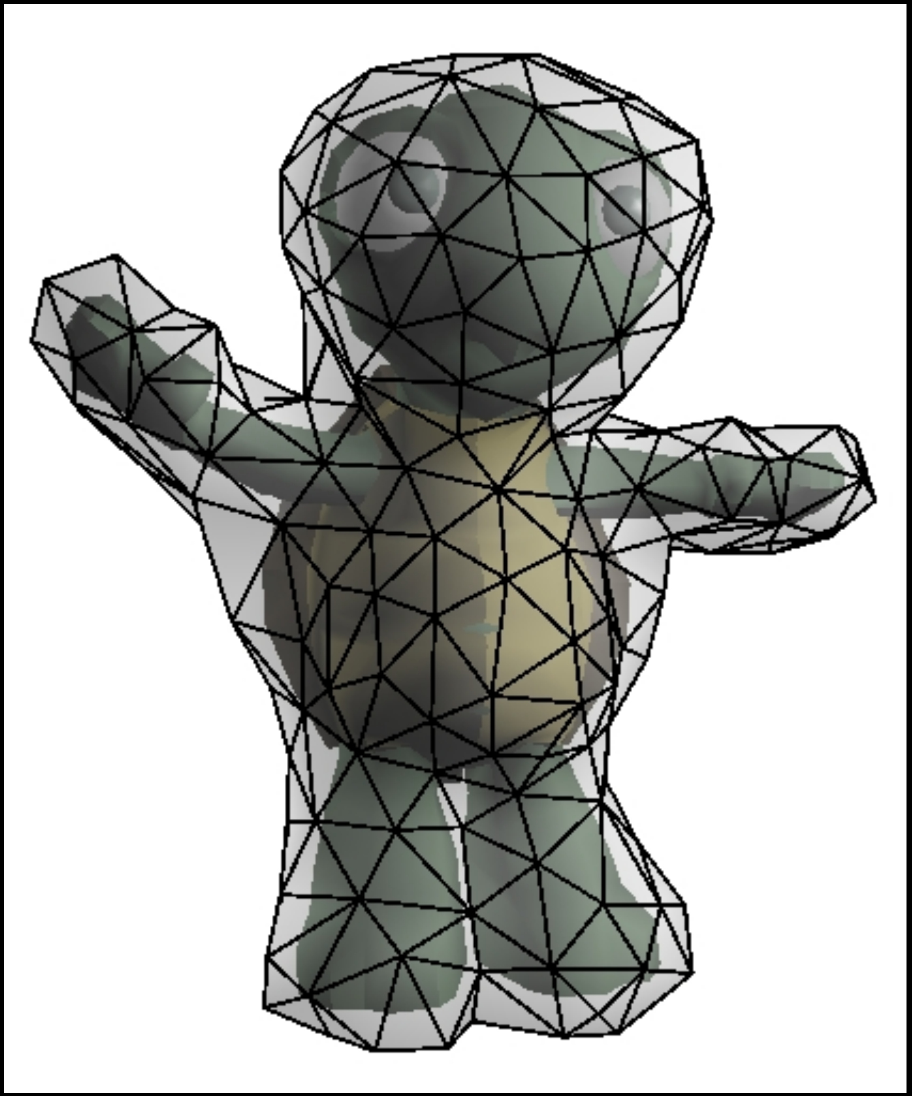
- Implicit Newmark [Wriggers 2002]
- Central differences [Wriggers 2002]
- Implicit Backward Euler [Baraff and Witkin 1998]
- Symplectic Euler
- others can be added easily

Vega is modular

- All deformable models can be used **independently** of each other, and of the integrators
- All integrators can be used **independently** of each other and of the deformable models

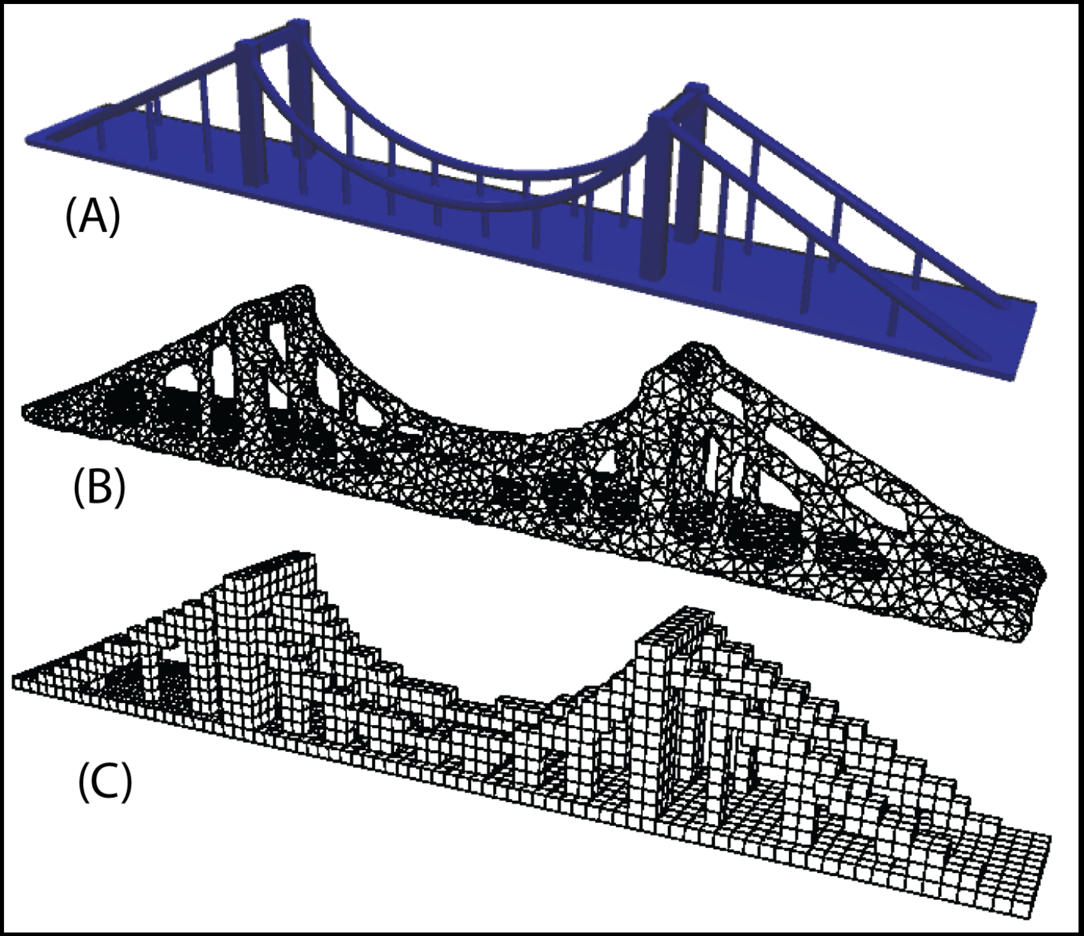
Materials in Vega

- Linear materials
- Neo-Hookean
- Mooney-Rivlin
- Arbitrary isotropic nonlinear materials easily supported



Elements in Vega:

- Tetrahedral
- Cubic



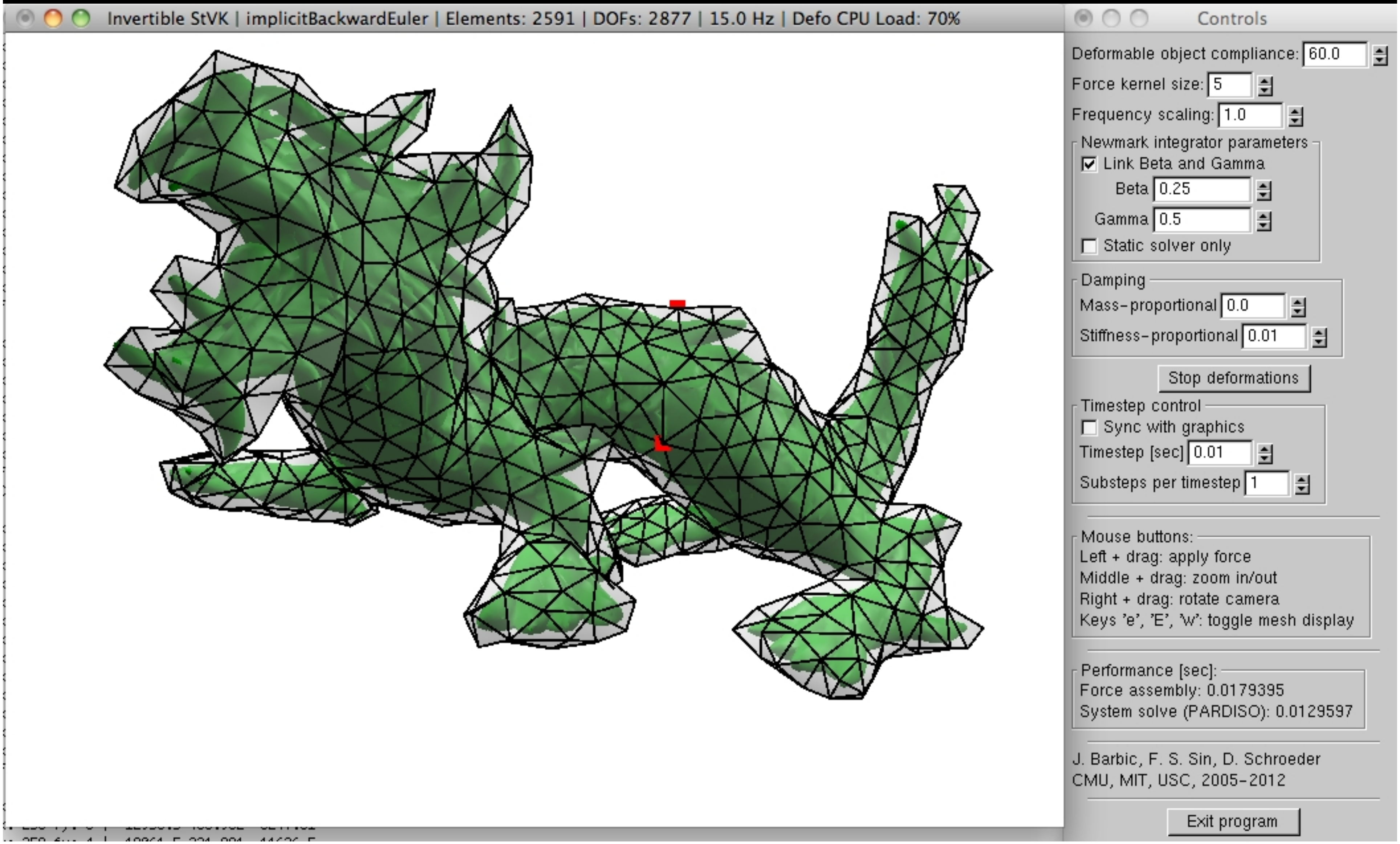
Sparse Linear Solvers:

- Jacobi-preconditioned Conjugate Gradients (iterative solver)
  "without the agonizing pain" [Shewchuk 1994]

- PARDISO (direct solver)

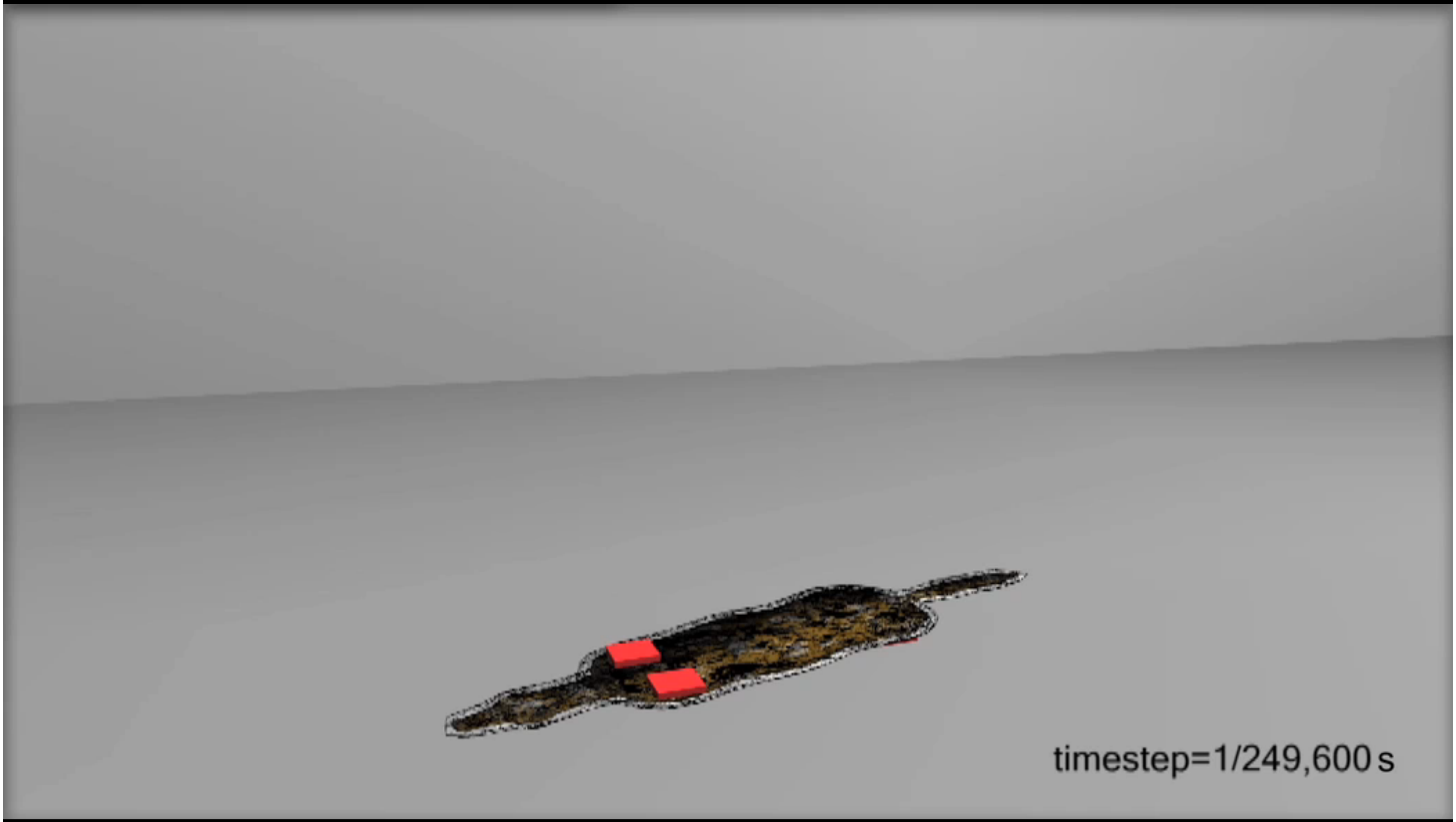
- SPOOLES (direct solver)

Demo Application Screenshot



Real-time Interaction

Inversion Handling



Vega: Nonlinear FEM Deformable Object Simulator

Funshing Sin[1], Daniel Schroeder[1,2], Jernej Barbič[1]

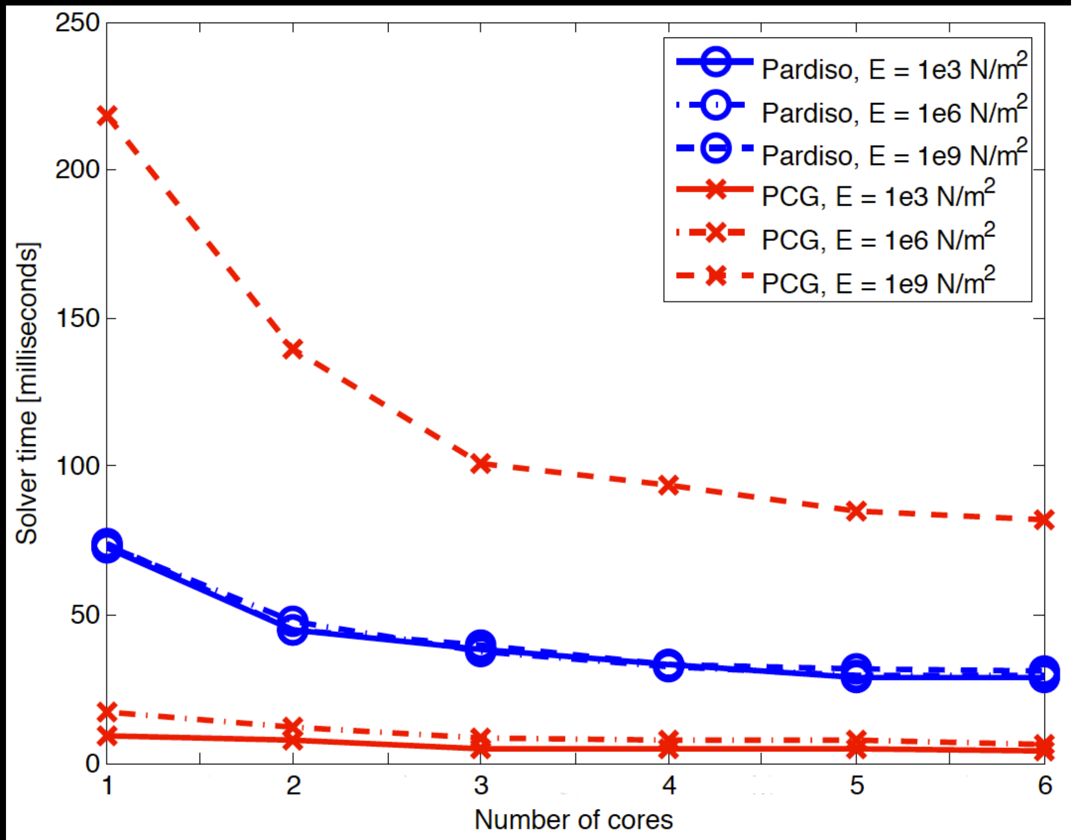
[1]University of Southern California, USA
[2]Carleton College, USA

Fun Shing Sin, Daniel Schroeder, Jernej Barbič:
Vega: Nonlinear FEM Deformable Object Simulator,
Computer Graphics Forum, to appear, 2012

http://www.jernejbarbic.com/vega

Direct Solver vs PCG

- Direct solver times are constant



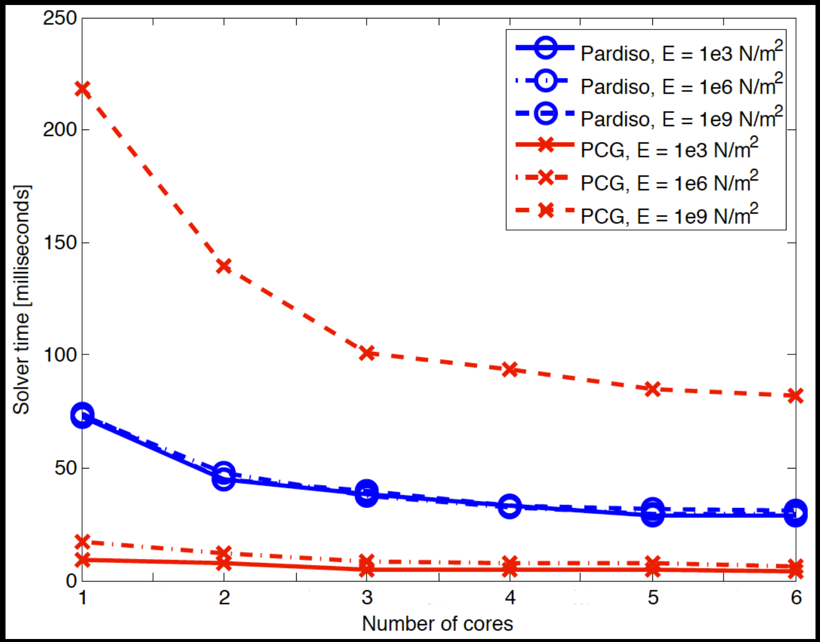
- PCG solver times depend on: material stiffness, convergence threshold

Why PCG times depend on stiffness

- System matrix has the form:

$$A = k_1\, M + k_2\, K$$

(for some constants k_1 , k_2)



- K is much more poorly conditioned than M

- As material is made stiffer, k_2 grows, and the K term becomes dominant in A ⟹ A becomes more poorly conditioned ⟹ more CG iterations are needed

Limitations of Vega and Future Work

- Cutting / fracture
- Collisions must be handled externally
- Shells (cloth) and strands
- Model reduction
  (already released separately)

Vega Live Demo

Outline

- Vega FEM
- Introduction to Model Reduction
- Linear Modal Analysis
- Model Reduction of Nonlinear Deformations
- Applications of Model Reduction

Online Course Notes:

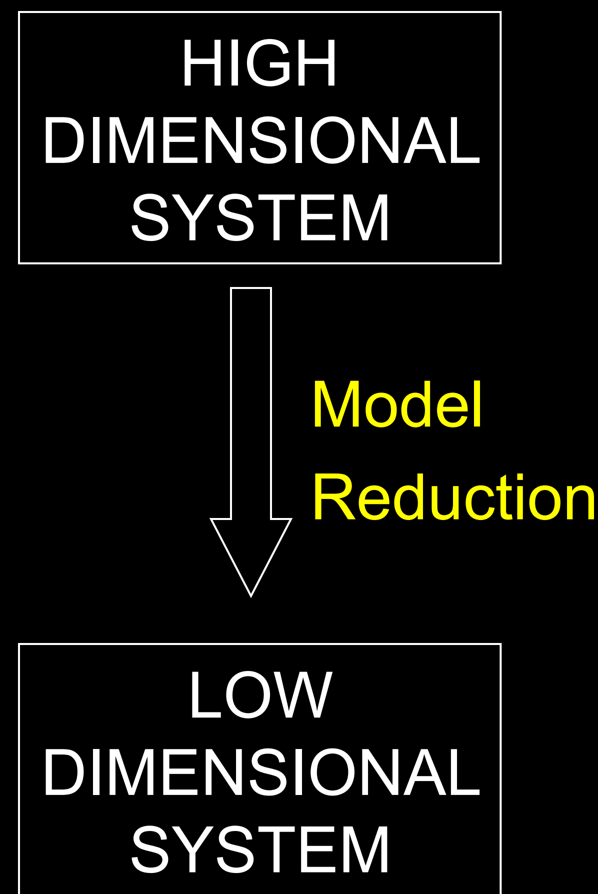
http://www.femdefo.org

(or, Google “Jernej Barbic”)

Model Reduction

- A technique to simplify simulations of systems described by Ordinary Differential Equations

- Project high-dimensional equations to low-dimensional equations



Model Reduction

+ Faster computation

+ Lower memory footprint

- Approximation only

Projection-based Model Reduction

A high-dimensional ODE: $\ddot{u} = F(u, \dot{u}, t)$

$$u = Uq$$

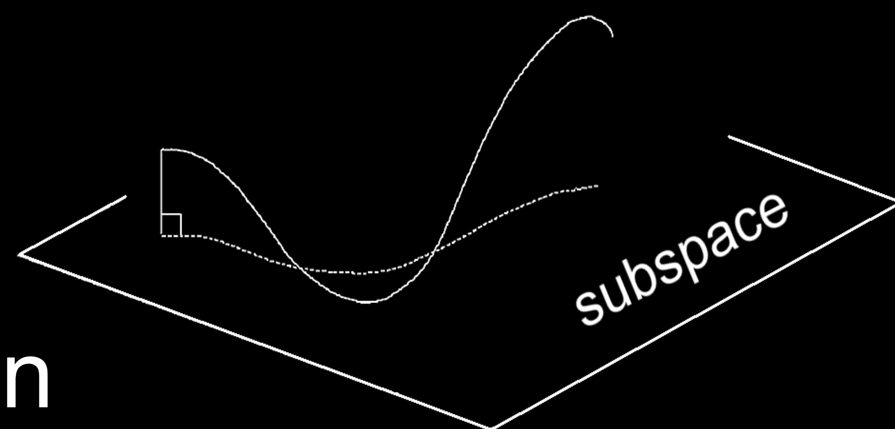
$$\Big\downarrow \quad \text{Pre-multiply with } U^T$$

Low-dimensional approximation: $\ddot{q} = U^T F(Uq, U\dot{q}, t)$

Elasticity, fluids, voltages, etc.

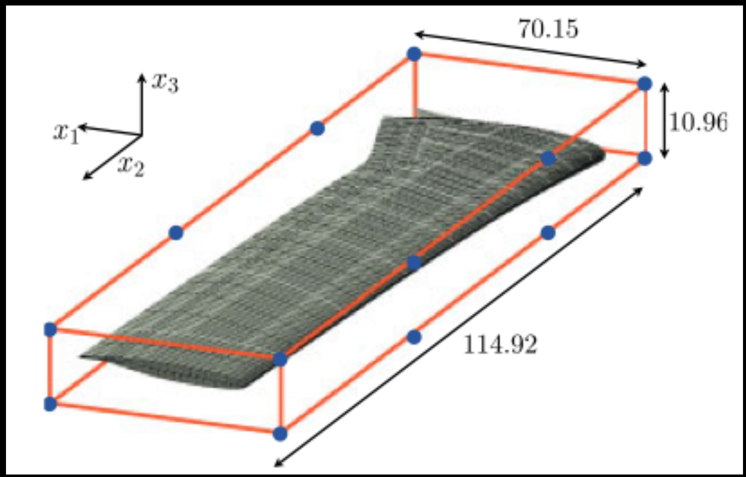
Other Names for Projection-Based Model Reduction

- “Principal Orthogonal Directions” method (POD)



- Subspace integration

Model Reduction Outside of Computer Graphics

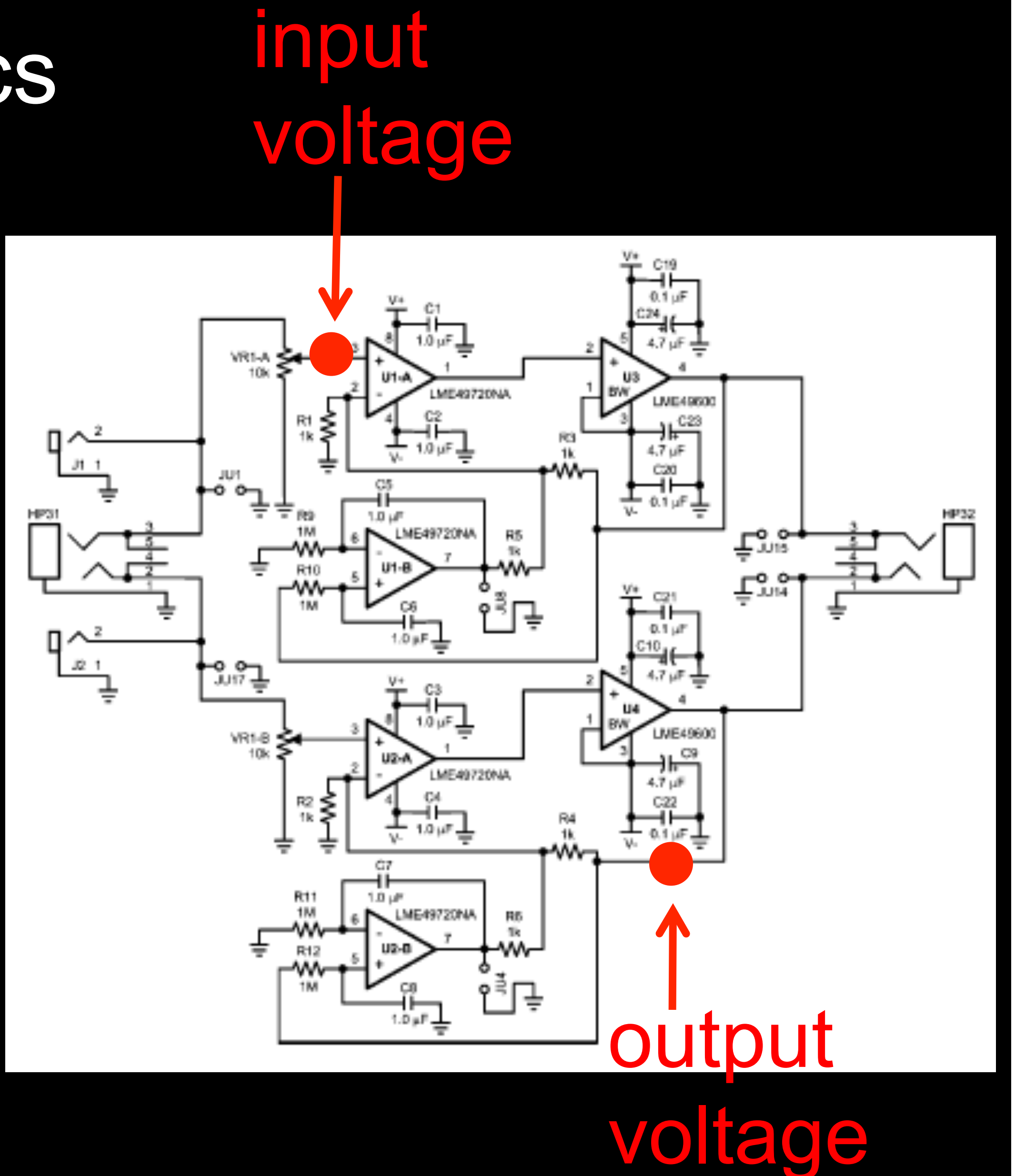


[Carlberg and Farhat 2010]

- Electric circuits
- Electromagnetics
- Microelectromechanical systems
- Aeronautics: Navier-Stokes equations, coupled fluid-structure problems

Model Reduction Outside of Computer Graphics

- Mostly linear systems
- Low-dimensional input, low-dimensional output



In Computer Graphics:

high-dimensional output (object shape)

need different reduction methods

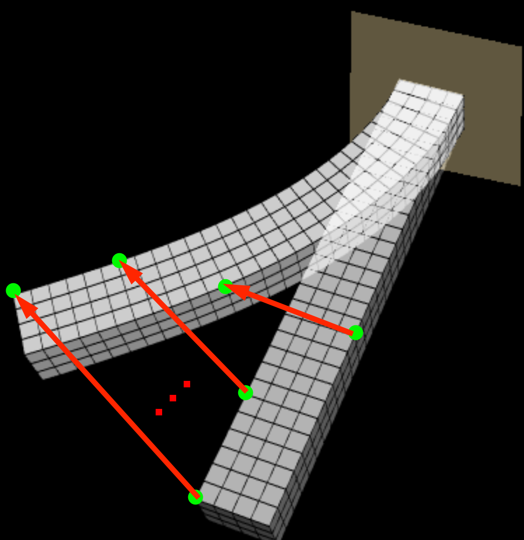
Outline

- Vega FEM
- Introduction to Model Reduction
- Linear Modal Analysis
- Model Reduction of Nonlinear Deformations
- Applications of Model Reduction

Model Reduction of Linear Systems

Notation:
Deformation Vector (3D meshes)

- Contains the 3D deformation vectors for all the mesh vertices



$$u = $$

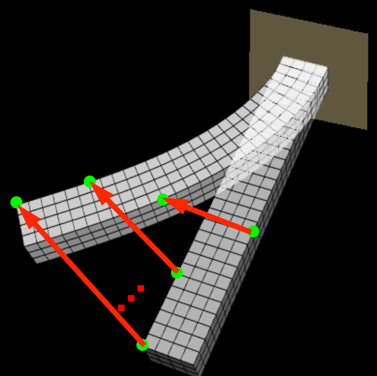
$3n$

1

Linear Equations of Motion of 3D Solid Deformable Object [Shabana 1990]

$$M\ddot{u} + D\dot{u} + Ku = f(t)$$

- u = deformation vector



- M = mass matrix
- D = damping matrix
- K = stiffness matrix
- f(t) = external forces

Linear Equations of Motion of 3D Solid Deformable Object [Shabana 1990]

$$M\ddot{u} + D\dot{u} + Ku = f(t)$$

- 3D linear continuum mechanics + FEM
- Widely used (e.g., earthquake simulation)
- Captures transient waves
- Supports small deformations only
- High-dimensional; no reduction
- Slow for very complex meshes (supercomputers)

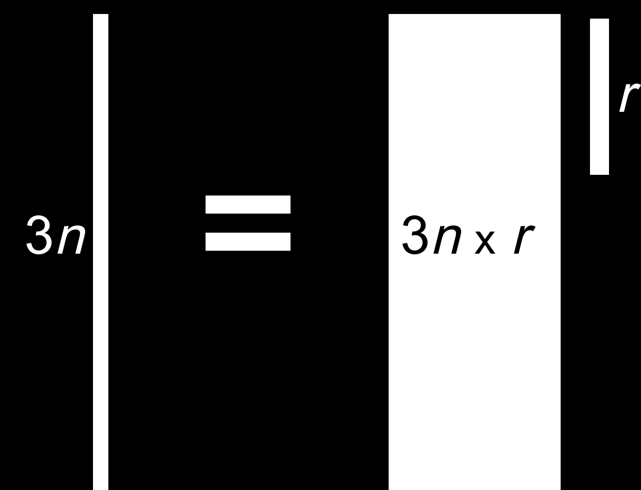
Applying Model Reduction to

$$M\ddot{u} + D\dot{u} + Ku = f(t)$$

Express deformation vector *u* as:

$$u = Uq$$

reduced coordinates

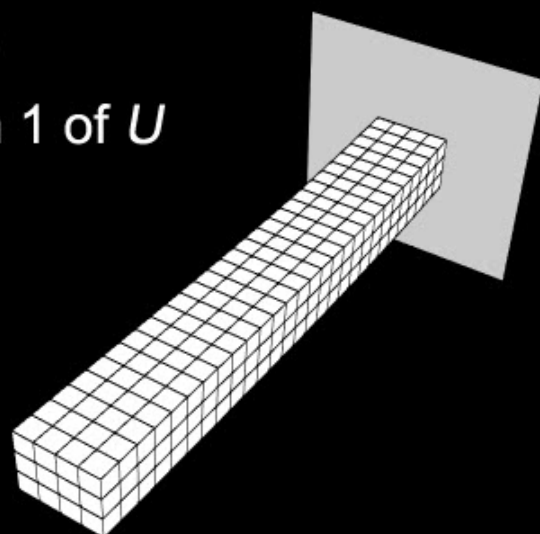


$r << 3n$

(for some appropriately chosen basis matrix *U*)

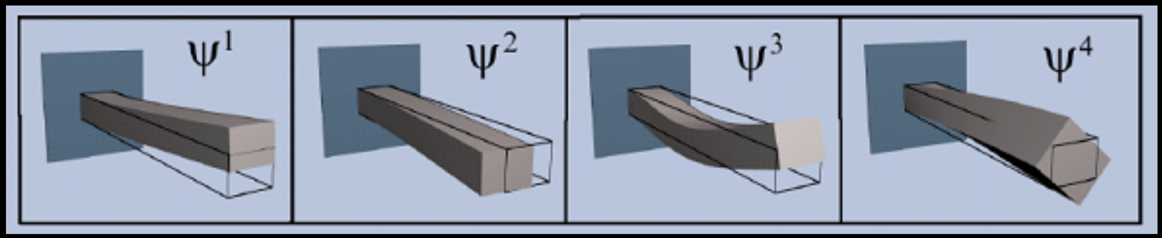
Columns of *U* are deformation basis vectors

Mode 1 =
= Column 1 of *U*



What is a good choice of basis?

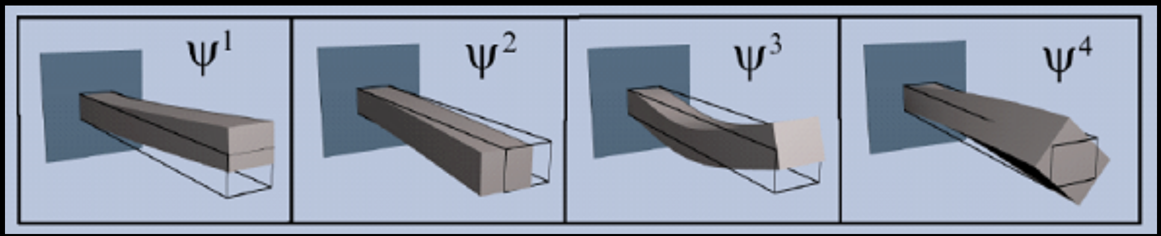
Linear Modes



Linear Modes
k = 4 shown

- Shapes with the ***least*** resistance to deformation
- “Natural” deformations of a structure
- Depend on boundary conditions (fixed vertices)

Linear Modes



Linear Modes

k = 4 shown

- Only good for small deformations
- In the k $\longrightarrow$ 3n limit, one obtains the full linear model

$$M\ddot{u} + D\dot{u} + Ku = f(t)$$

Linear Modes

Linear Modes are Shapes with the Least Resistance to Deformation

For a given amount of deformation,

subject to fixed vertices,

which shape increased the elastic

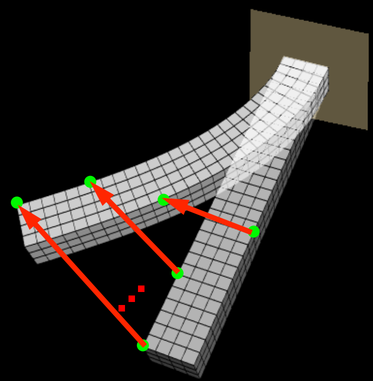
strain energy by the ***least*** amount?

Linear Modes are Shapes with the Least Resistance to Deformation

- Measure mesh displacement:

½**<M u, u>** = "total amount of displaced mass"

u = deformation vector



Note: **<u, u>** is **not** a good measure!

- Measure (linearized) strain energy: ½**<K u, u>**

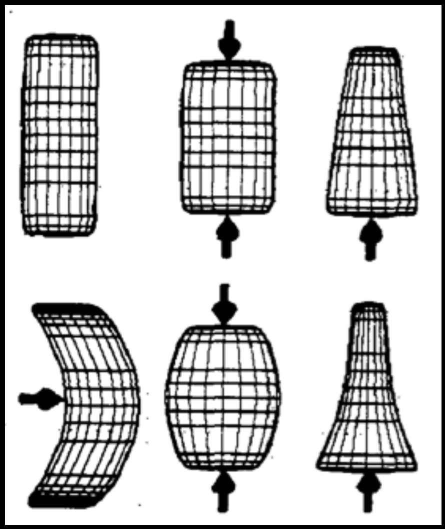
Linear Modes are Shapes with the Least Resistance to Deformation

$$\psi_1 = \arg\min_{u;<Mu,u>=1} < Ku, u >$$

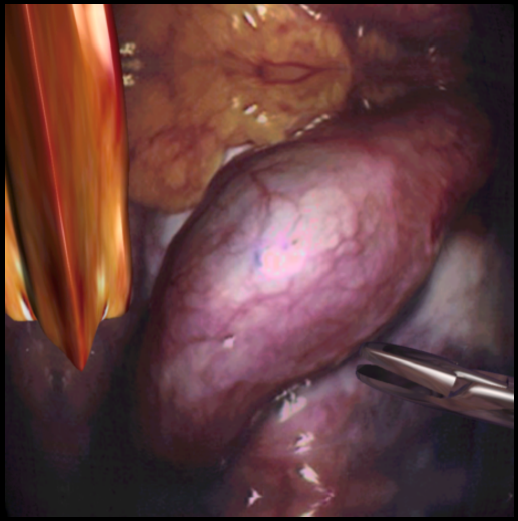
$$\psi_2 = \arg\min_{u;u\perp\psi_1,<Mu,u>=1} < Ku, u >$$

$$\psi_3 = \arg\min_{u;u\perp\psi_1,u\perp\psi_2,<Mu,u>=1} < Ku, u >$$

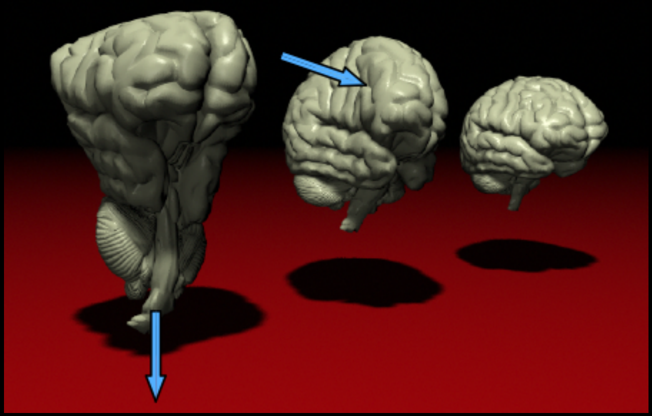
Linear Modes in Computer Graphics



[Pentland and Williams 1989]



[James and Pai 2002]



[Hauser, Shen, O'Brien 2003]

Applications of Linear Modes

- Fast deformable object simulation (games, virtual surgery, fast previewing)
- Modeling of deformed shapes (interactive design of animations)
- Force feedback rendering / haptics
- Sound simulation

Computing Linear Modes

- Remove rows and columns corresponding to fixed vertices from K and M

$\Rightarrow \bar{K}, \ \bar{M}$

- Solve generalized eigenvalue problem:

$$\bar{K}x = \lambda \bar{M}x$$

- Can use ARPACK (free eigensolver)
- $\lambda = \omega^2$, $\omega = 2\pi / T$, T = oscillation period

ARPACK

- Free eigensolver for large sparse matrices:

$$A\,x = \lambda\,B\,x$$

- Arnoldi iteration
- Danny C. Sorensen, Rice University, mid-1990s
- http://www.caam.rice.edu/software/ARPACK/

ARPACK

- Works very well
- Written in Fortran; compiles (today) without much difficulty
- Compilation instructions for Windows:

  http://www.jernejbarbic.com/arpack.html

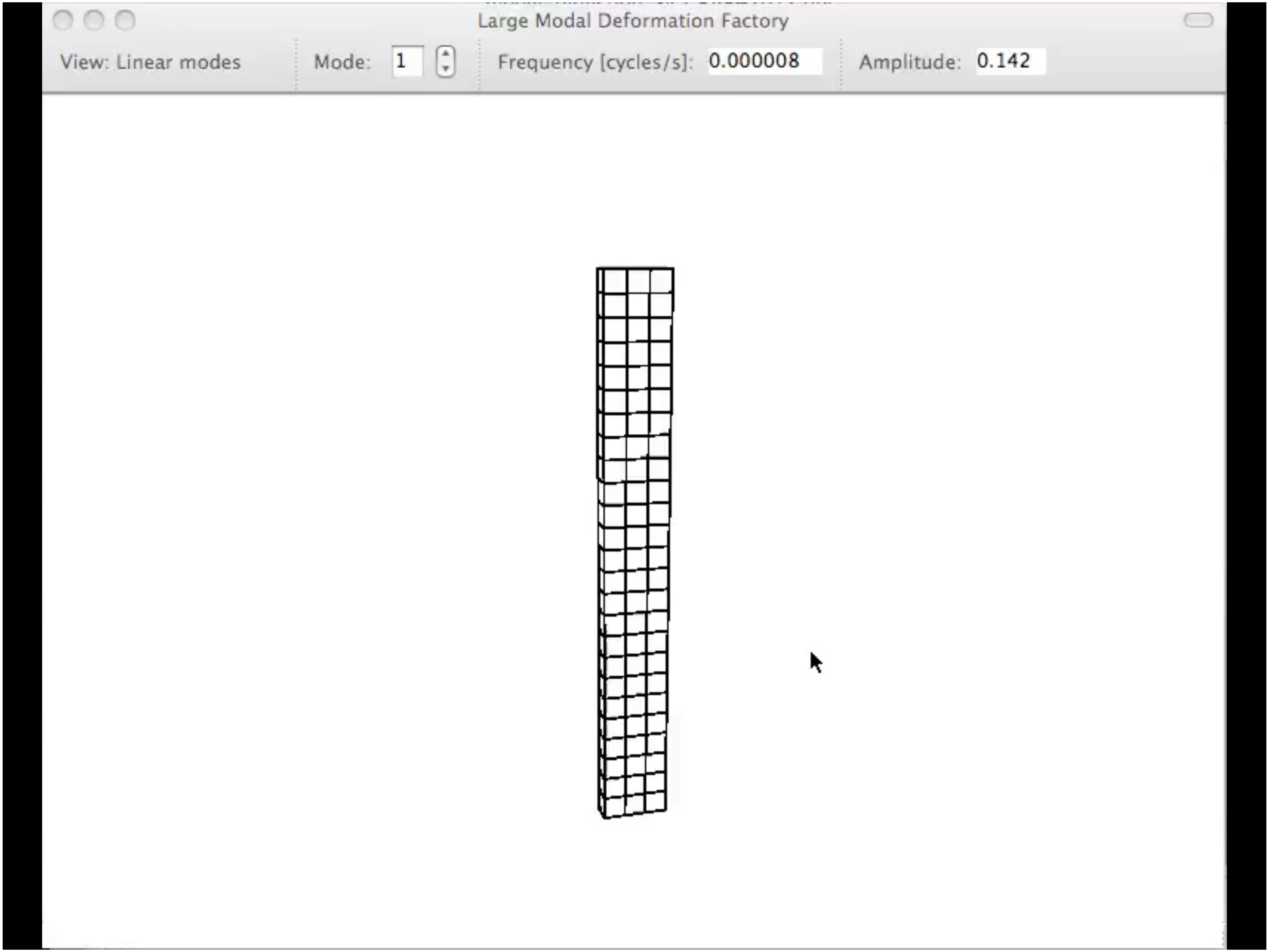
When no fixed vertices: “free-fly” modes

- Useful for free-flying objects
- First six modes correspond to:

  all rigid translations (3 modes), and

  all infinitesimal rotations (3 modes)
- Zero frequency
- These modes are often discarded

[Hauser, Shen, O’Brien 2003]


Large Modal Deformation Factory
View: Linear modes
Mode: 1
Frequency [cycles/s]: 0.000008
Amplitude: 0.142

Use Linear Modes for Reduction

$$M\ddot{u} + D\dot{u} + Ku = f(t)$$

Substitute $u = Uq$

Project U^T

$$\ddot{q} + U^T D U \dot{q} + \Lambda q = U^T f(t)$$

Reduced Equations of Motion

Independent modal oscillators

- If $D = \alpha M + \beta K$ (Rayleigh damping), then $U^T D U$ is ***diagonal***.

$$\ddot{q} + \zeta \dot{q} + \Lambda q = U^T f(t)$$

Decouples

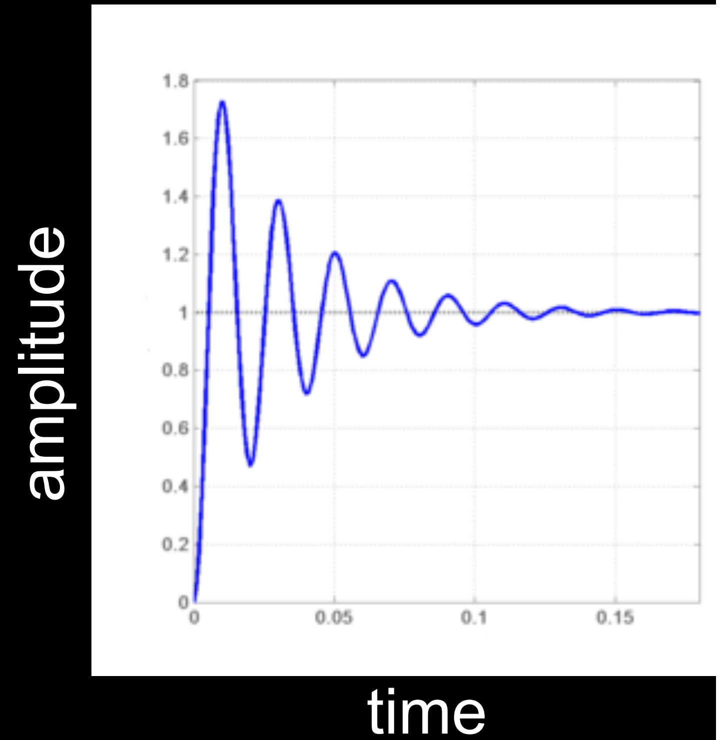
$$\ddot{q}_i + \zeta_i \dot{q}_i + \Lambda_i q_i = \tilde{f}_i(t)$$

Decoupled 1D modal oscillators

Integrating modal oscillators

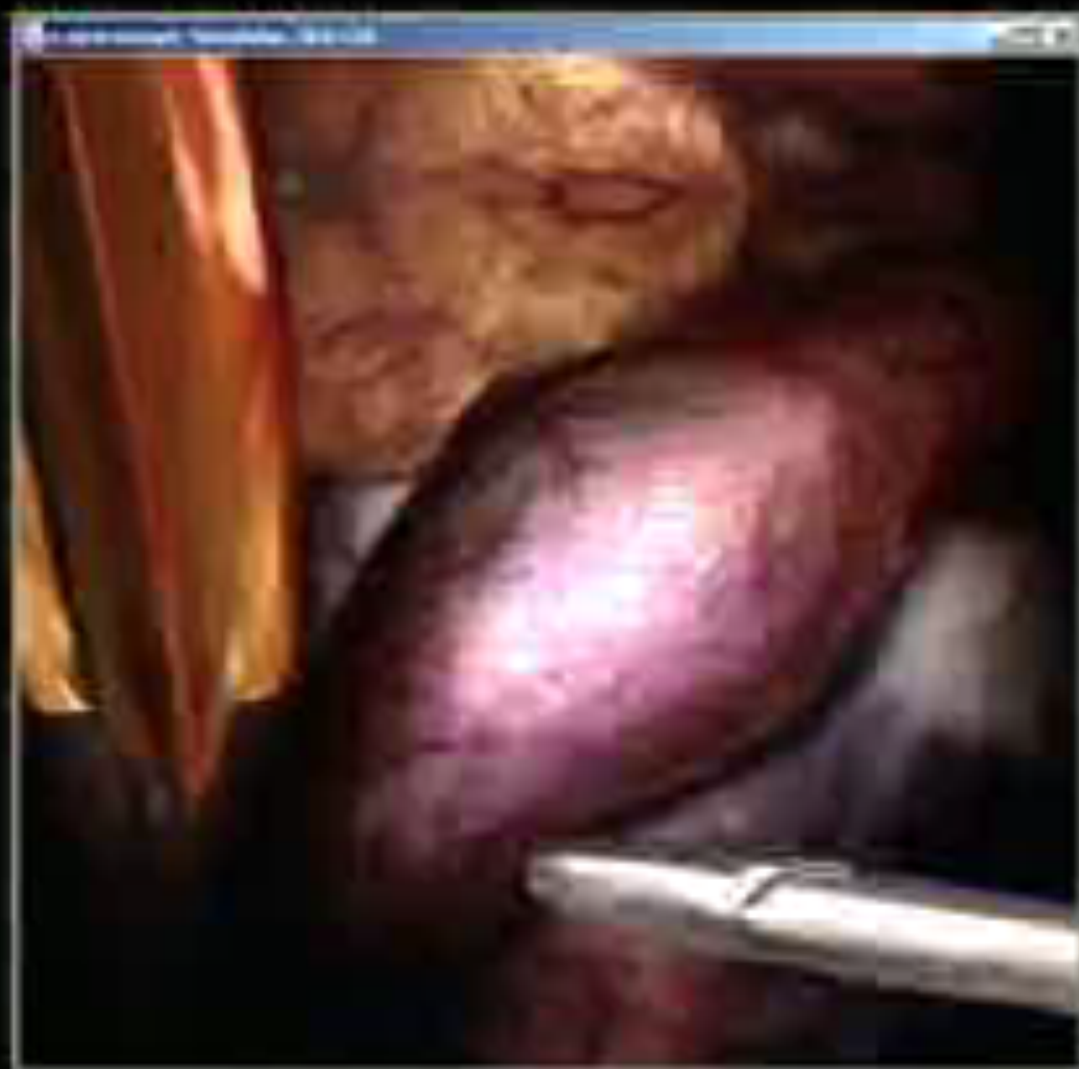
$$\ddot{q}_i + \zeta_i \dot{q}_i + \Lambda_i q_i = \tilde{f}_i(t)$$

- Fast (1D simulation)
- Can use any numerical integrator
- Over-damped vs under-damped, depending on damping strength
- Exact integration possible using IIR filters [James and Pai 2002]



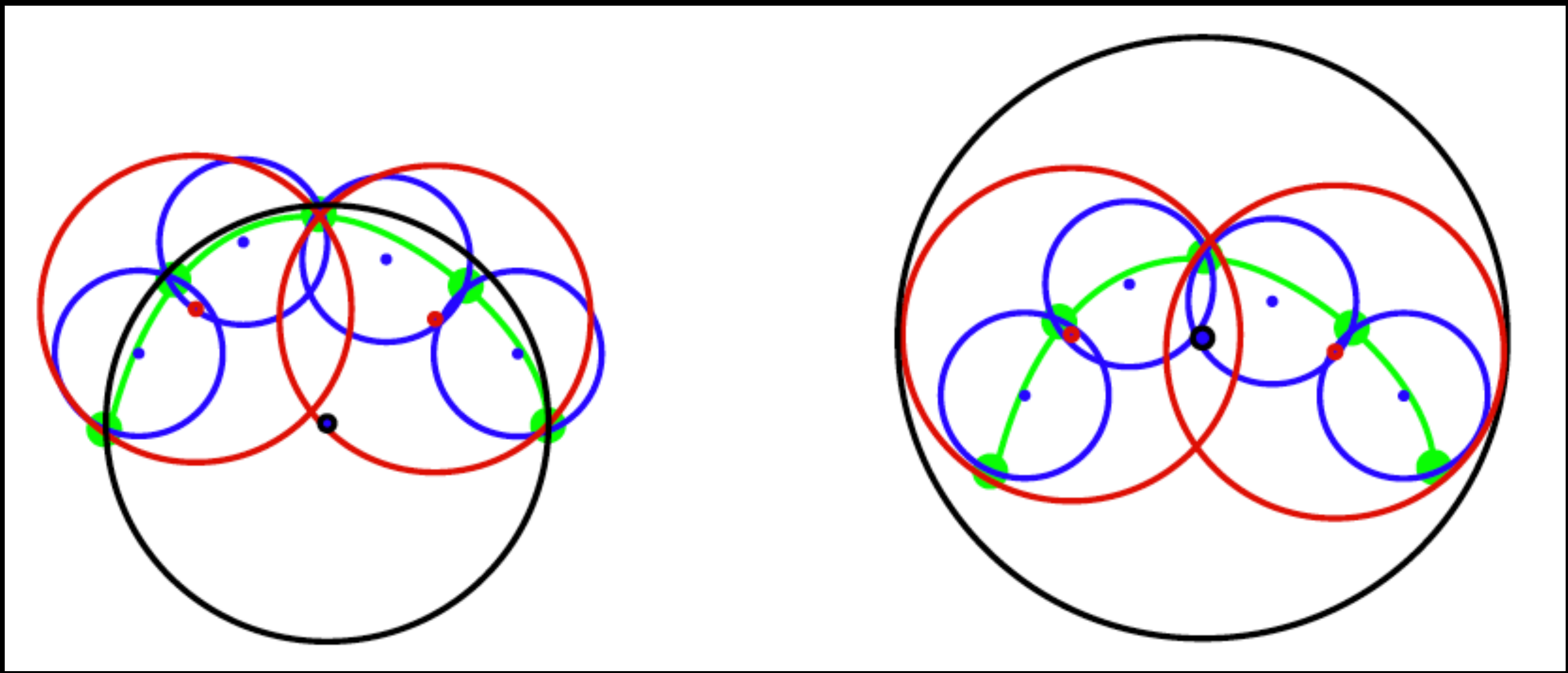
Linear modal simulation

[James and Pai 2002]



Collision Detection for Reduced Deformable Models

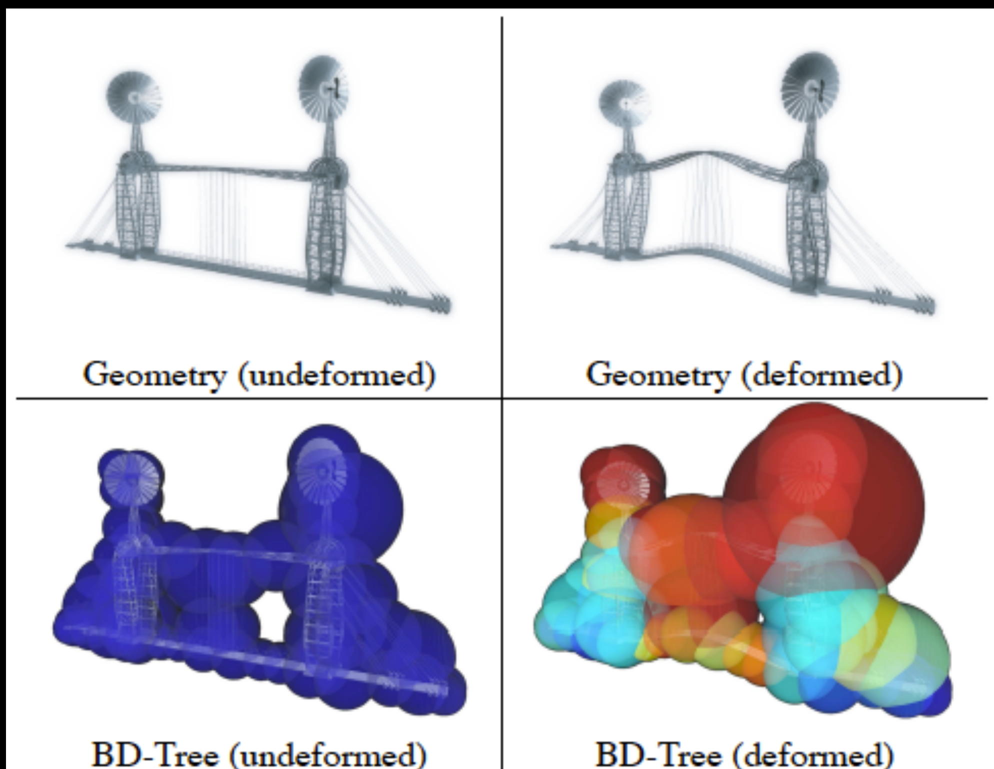
BD-Tree [James and Pai 2004]

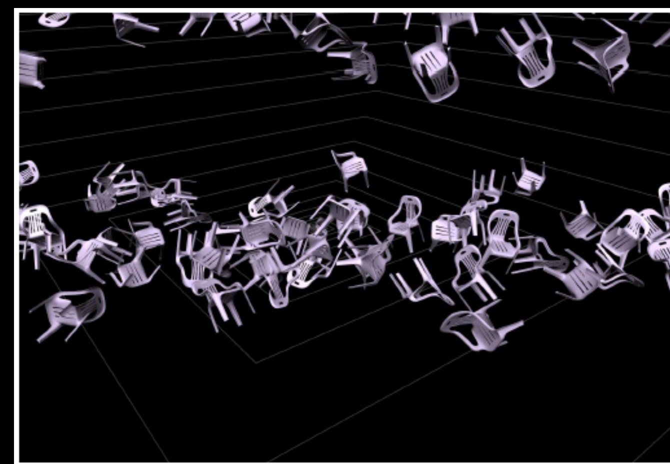


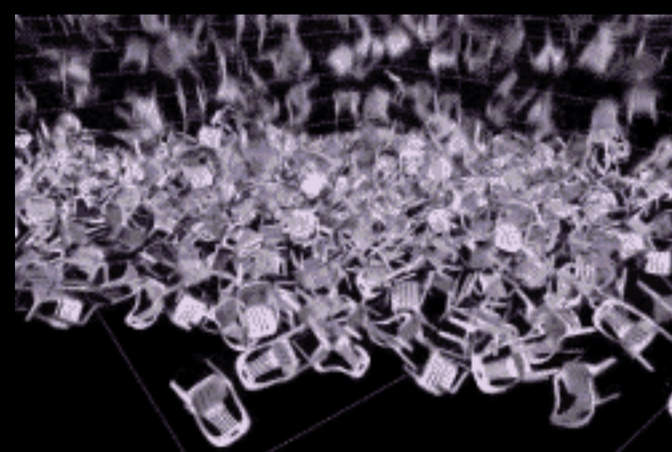
undeformed deformed

Collision Detection for Reduced Deformable Models

BD-Tree [James and Pai 2004]

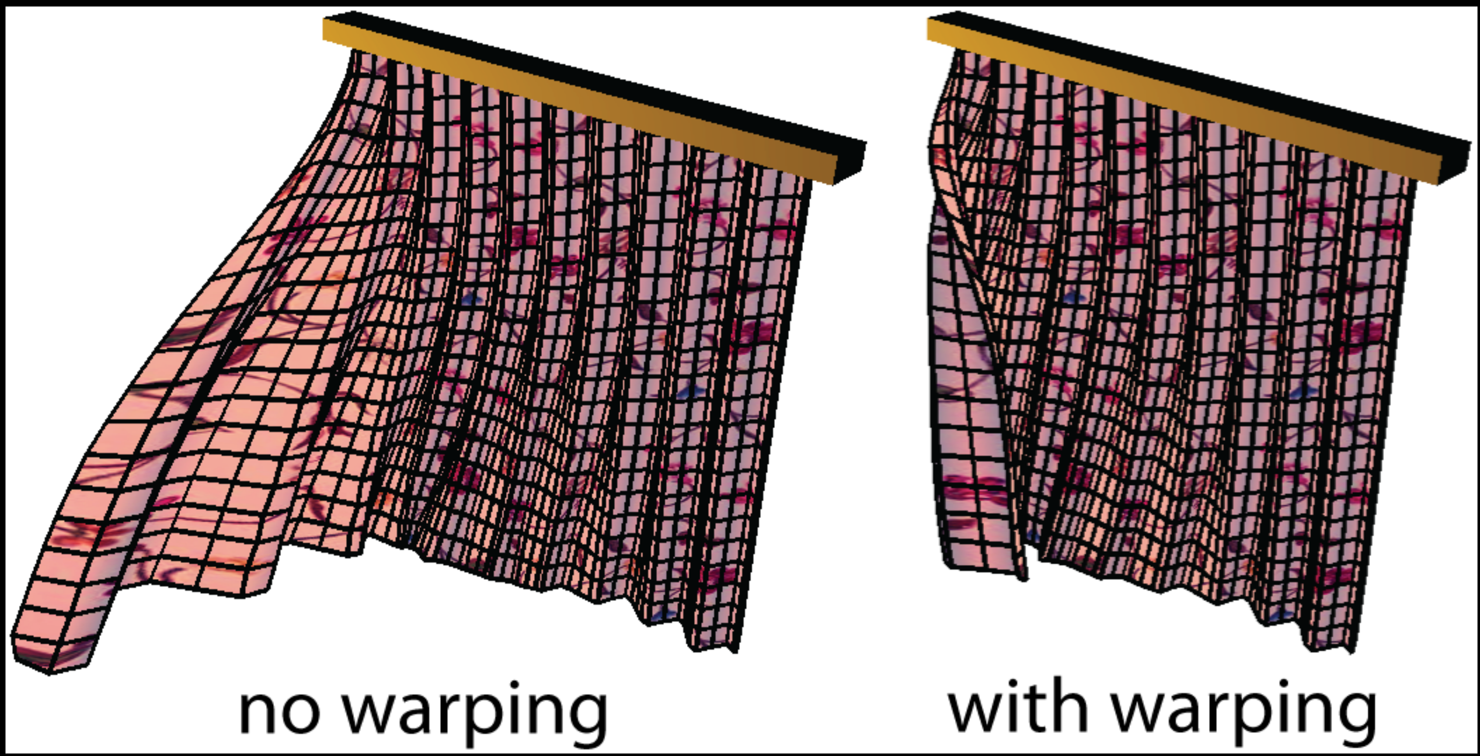




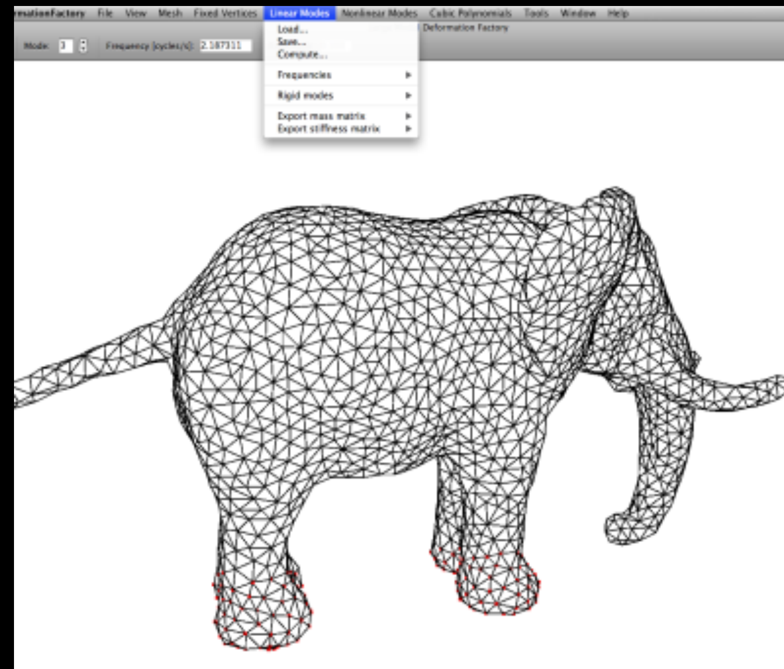


Correcting Artifacts of Large Deformations : Deformation Warping

- Two flavors: [Choi and Ko 2005], [Huang et al. 2011]



Software for model reduction
(by Jernej Barbic)



- Compute linear modes for any tet mesh or triangle mesh
- Compute modal derivatives
- Compute the basis
- Compute cubic polynomials
- Timestep reduced models at runtime
- Available at:

www.jernejbarbic.com/code

Live Demo:

Computing Linear Modes

Live Demo:

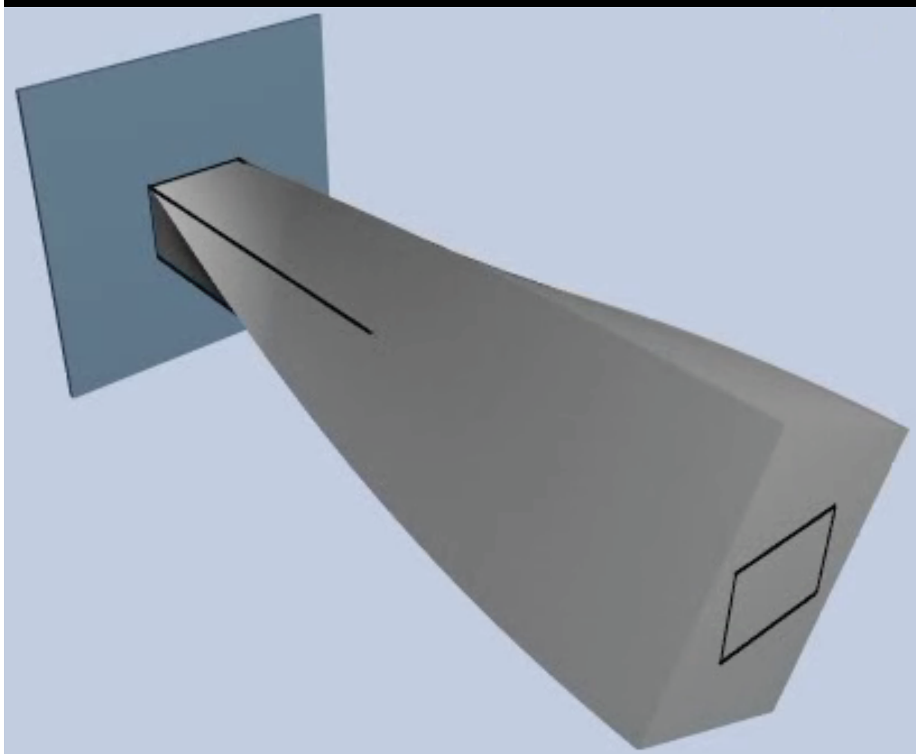
Building a Reduced Nonlinear Simulation

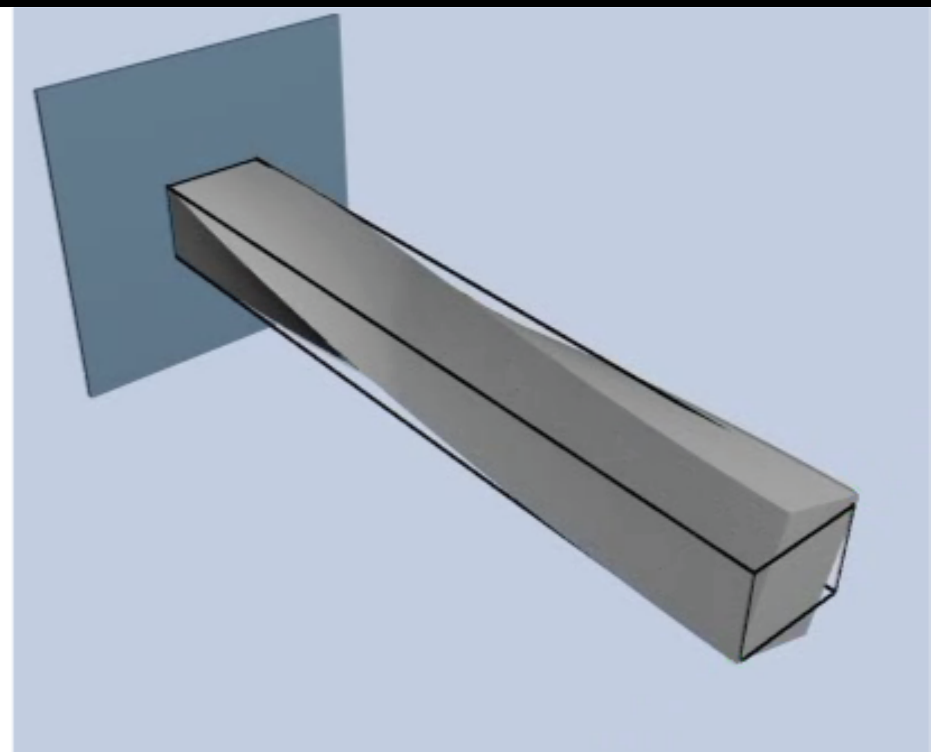
Outline

- Vega FEM
- Introduction to Model Reduction
- Linear Modal Analysis
- Model Reduction of Nonlinear Deformations
- Applications of Model Reduction

Model Reduction of Nonlinear Deformations

Motivation



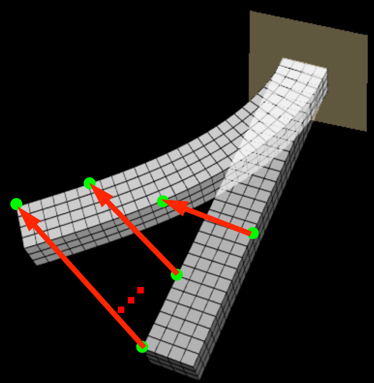


3D Continuum Mechanics + FEM:

Equations of Motion **[Euler, Lagrange]**

$$M\ddot{u} + D\dot{u} + \boxed{f_{\text{int}}(u)} = f_{\text{ext}}(t)$$

- u = deformation vector



- Supports large deformations
- Nonlinear

$$M\ddot{u} + D\dot{u} + f_{\mathrm{int}}(u) = f_{\mathrm{ext}}(t)$$

High-dimensional system of ODEs

Not real-time for large models

How to approximate
it for interactive
applications ?

Reduced equations of motion

$$M\ddot{u} + D\dot{u} + f_{\text{int}}(u) = f_{\text{ext}}(t)$$

Substitute $u = Uq$

Project U^T

$$\ddot{q} + U^T D U \dot{q} + \boxed{U^T f_{\text{int}}(Uq)} = U^T f_{\text{ext}}(t)$$

Reduced Equations of Motion

Unreduced

$$M\ddot{u} + D\dot{u} + f_{\text{int}}(u) = f_{\text{ext}}(t)$$

- High-dimensional (e.g. 3*n* = 75,000)

Reduced

$$\ddot{q} + U^T D U \dot{q} + U^T f_{\text{int}}(Uq) = U^T f_{\text{ext}}(t)$$

- Low-dimensional (e.g. *r* = 30)

Reduced internal forces:

$$\tilde{f}_{\text{int}}(q) = U^T f_{\text{int}}(Uq)$$

slow to evaluate

Assumption:

Large strain + linear material ("geometrically nonlinear FEM")

$$\tilde{f}_{\text{int}}(q) = U^T f_{\text{int}}(Uq) =$$

$$= \begin{pmatrix} p_1(q) \\ p_2(q) \\ \vdots \\ \\ \\ p_r(q) \end{pmatrix}$$

[Barbic and James 2005]

r cubic polynomials in components of *q*

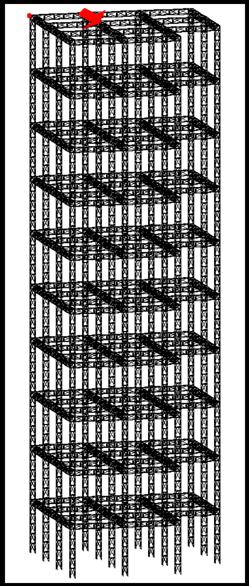
Coefficients depend on:
- object geometric shape
- material properties

How to select the basis U ?

$$u = Uq$$

$\boxed{\text{Basis must capture typical nonlinear deformations}}$

Motion basis selection

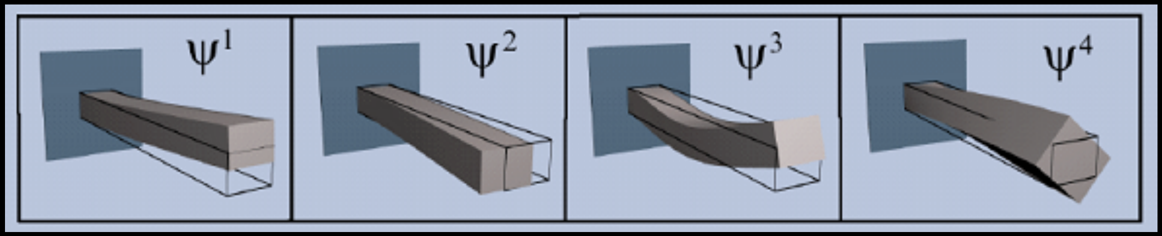


0. Example motion [Krysl et al. 2001]

1. Recorded user interaction
   [Barbic and James 2005]

2. Modal Derivatives (automatic)
   [Barbic and James 2005]

Motion basis from modal derivatives



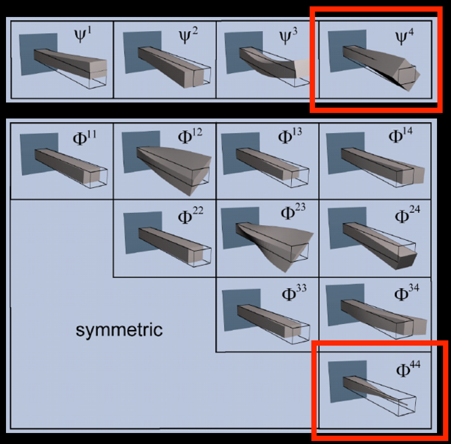
Linear Modes

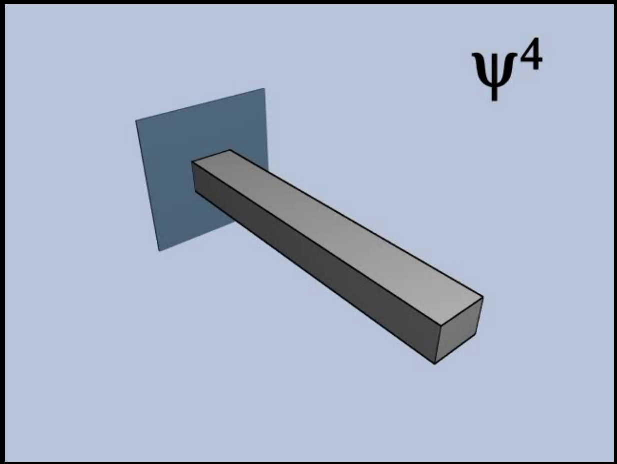
k=4 shown

- Linear modes only good for very small deformations

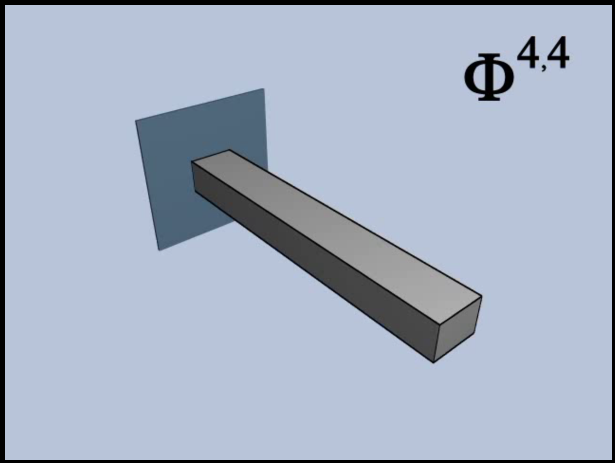
Modal derivatives are nonlinear corrections to linear modes

[Idelsohn and Cardona 1985]





The “twist” linear mode and artifacts for large deformations

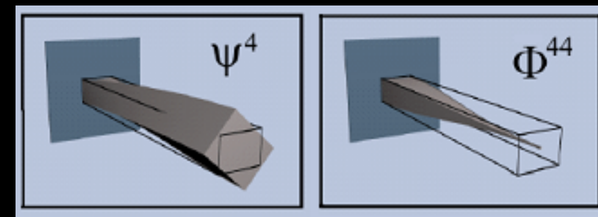


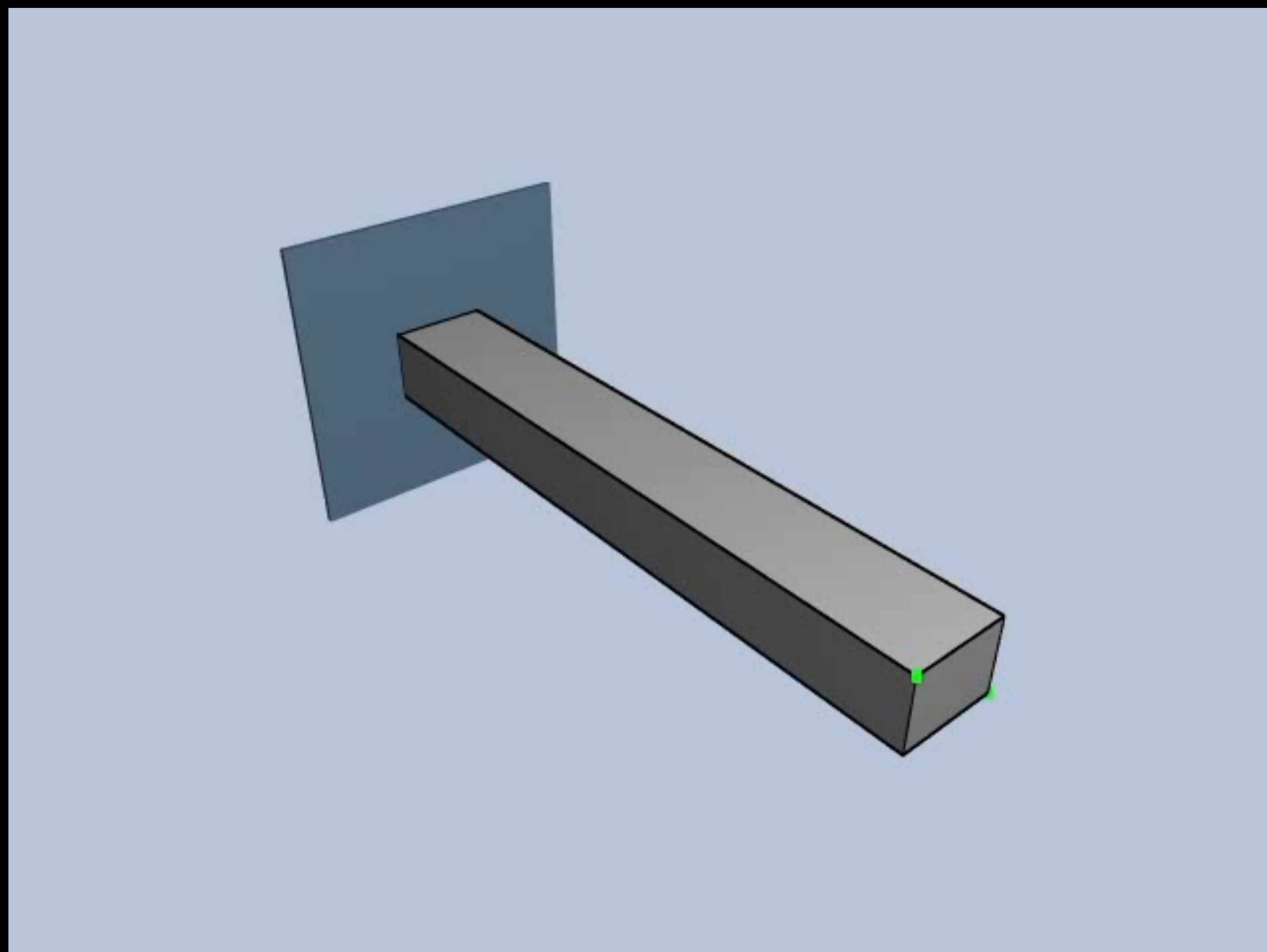
Modal derivative cancels volume growth

Runtime simulation:

Modal Derivatives, r = 2

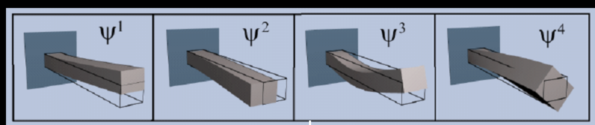
Basis

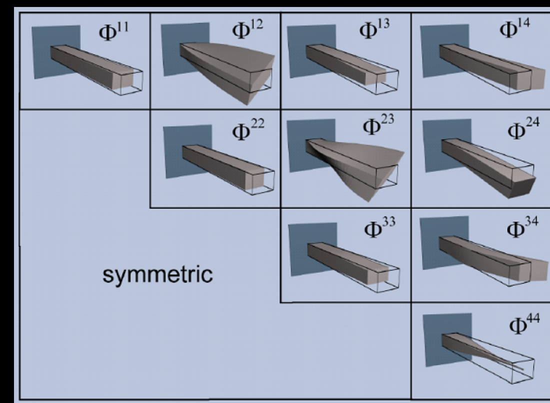




[Barbic and James 2005]

Motion basis from modal derivatives





Scale by frequency

Scale by frequency pair

Mass-scaled PCA

Basis of motion U

Comparison:

Full simulation vs reduced simulation

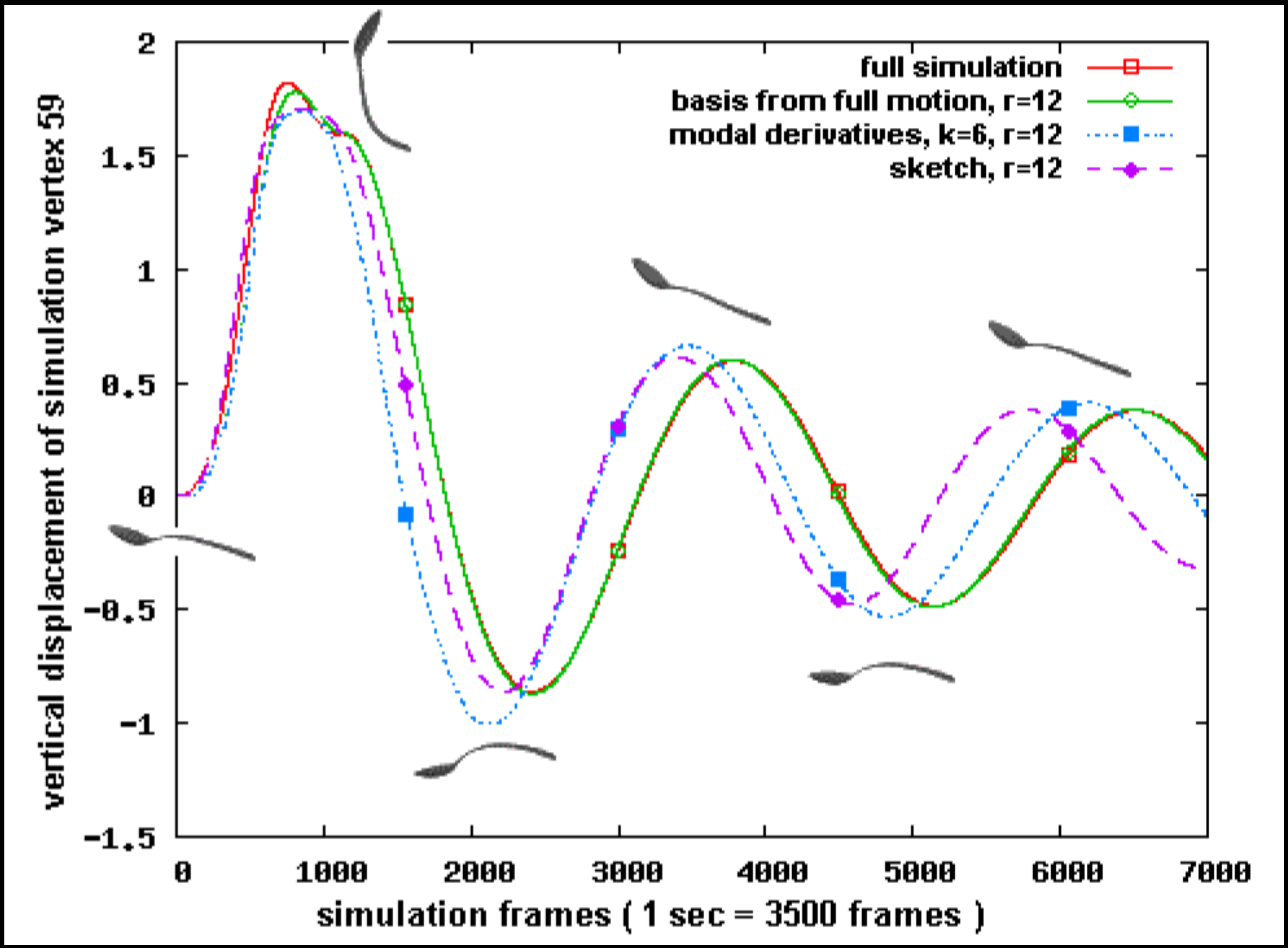
Unreduced
3n=11094

Modal derivatives
k=6, r=12

Computation time: 10 hours

Computation time: 0.71 sec
50,000x faster

Spoon experiment accuracy plot



Modal Derivatives

$$f_{\text{int}}(u) = f$$

What load causes a displacement aligned with mode ψ_i ?

Answer:
$f = \lambda_i M \psi_i$

Proof:
$f_{int}(\psi_i) \approx K \psi_i = \lambda_i M \psi_i$

Modal Derivatives

For any $p \in \mathbb{R}^k$:

$$f_{\text{int}}(u(p)) = M U_{\text{lin}} \Lambda p$$

This defines a function u = u(p).

Taylor series:

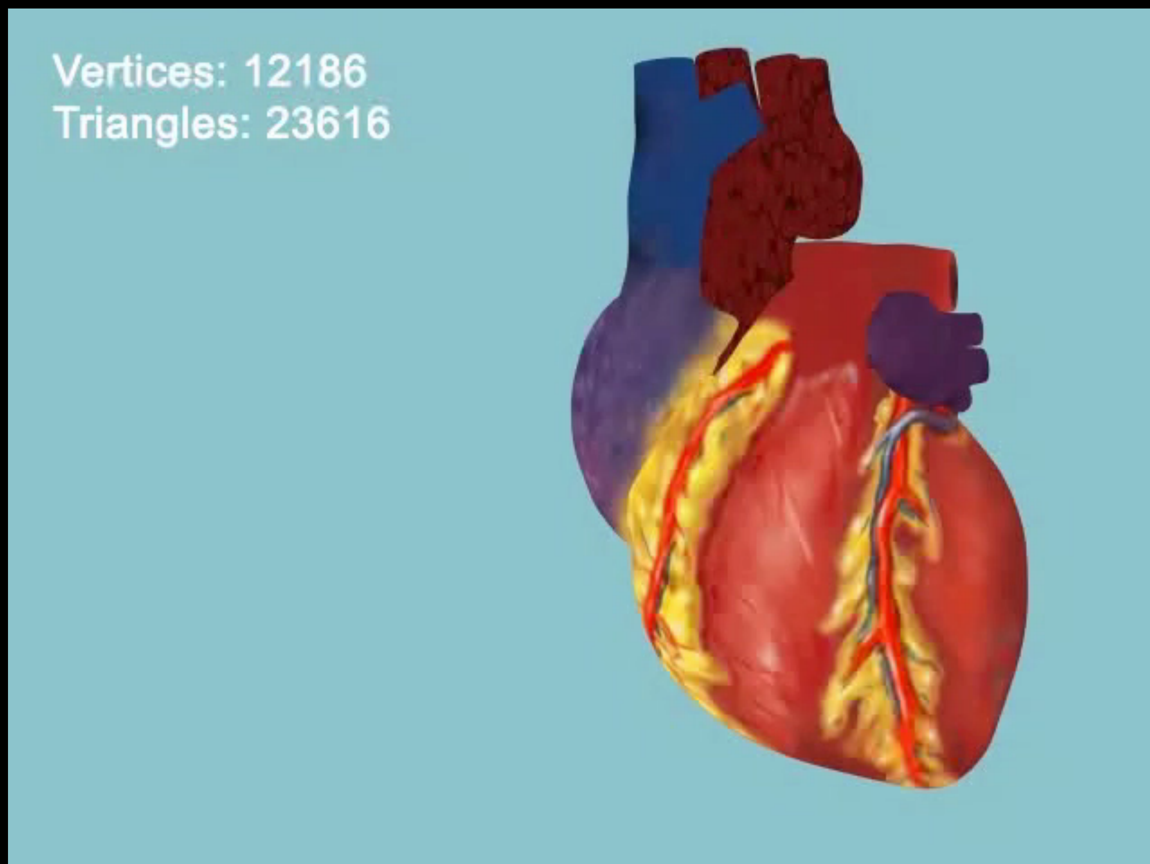
$$u(p) = \sum_{i=1}^{k} \Psi^i p_i + \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} \Phi^{ij} p_i p_j + O(p^3)$$

Modal derivative

$$K\Phi^{ij} = -(H : \psi_j)\psi_i$$

Simplified Heart (modal derivatives)

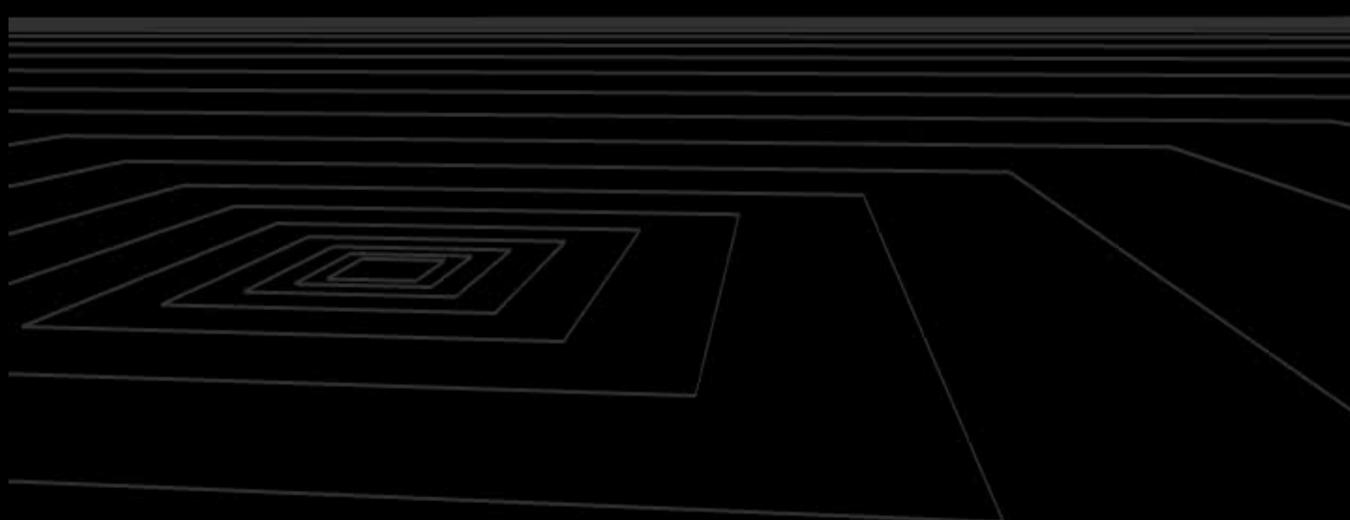
r =30; 1.4 msec; speedup = 21,000x



Multibody dynamics (modal derivatives)

512 baskets (r = 35), 1.2 sec / time-step

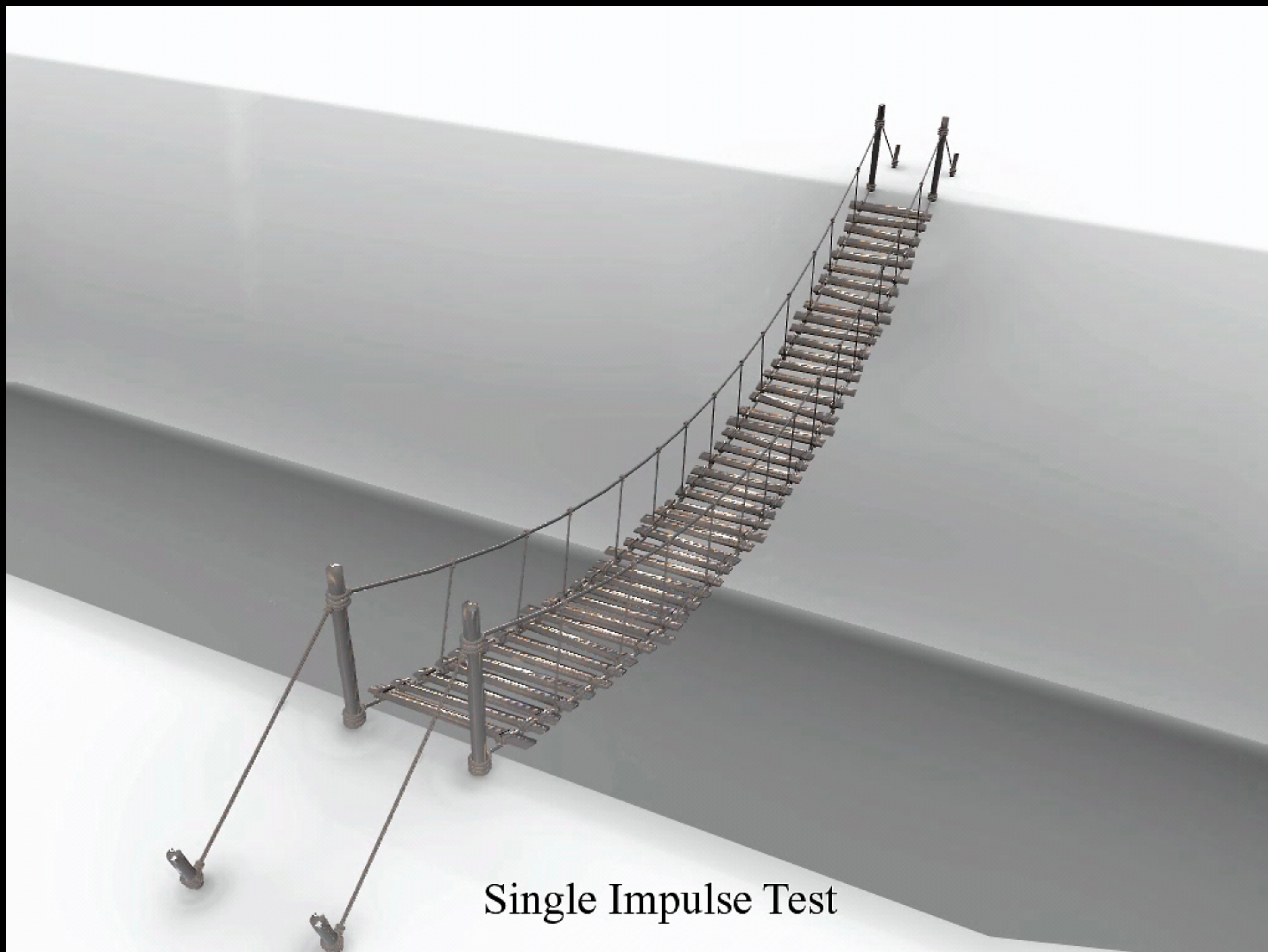
BD-Tree for collision detection



Nonlinear materials + reduction

- StVK cubic polynomial scales as O(r^4) (r = #modes)
- Can approximate reduced forces and reduced stiffness matrix in O(r^3) time, using numerical ***cubature*** [An, Kim and James 2008]
- Supports arbitrary nonlinear materials

Nonlinear materials + reduction



[An, Kim and James 2008]

Combining full simulation with model reduction [Kim and James 2009]

- Adaptively decides whether to take a full step or reduced step, at runtime ⟹ **online model reduction**
- Makes it possible to *throttle* the simulation

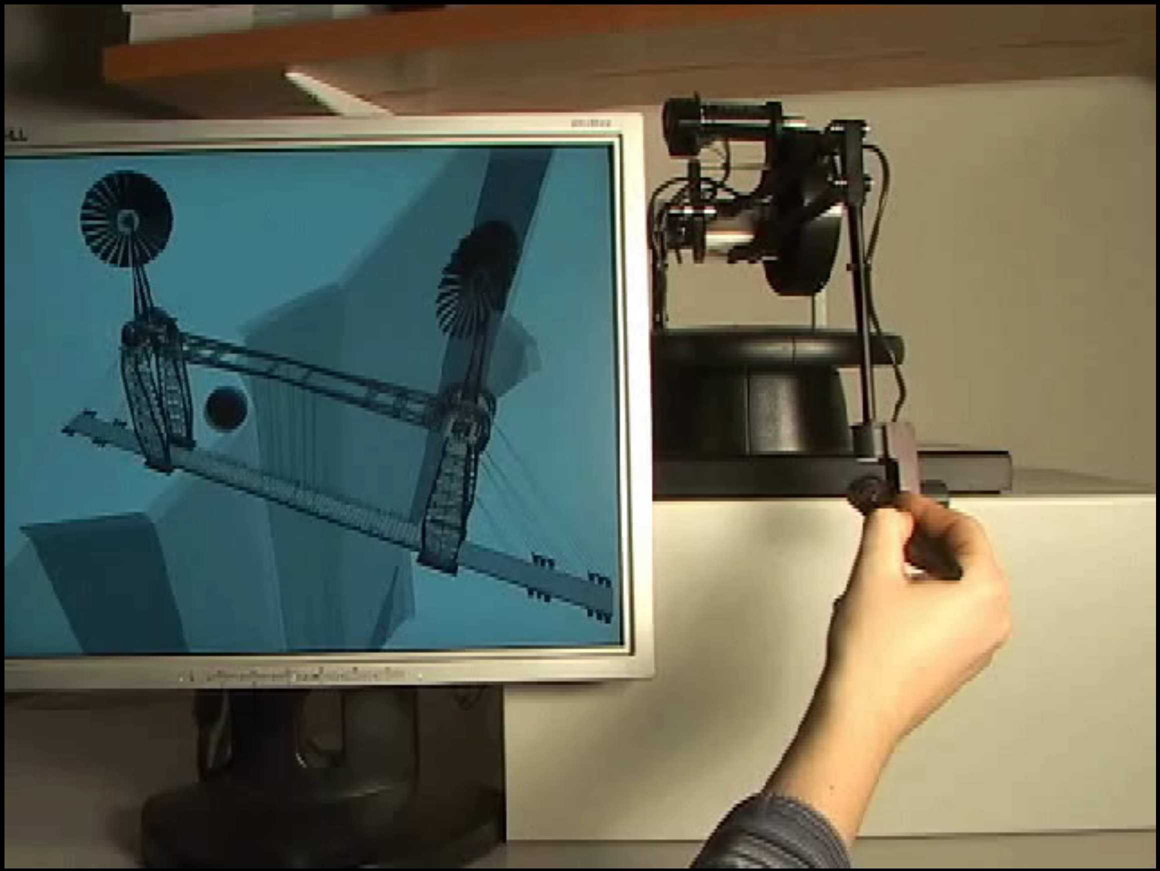
Outline

- Vega FEM
- Introduction to Model Reduction
- Linear Modal Analysis
- Model Reduction of Nonlinear Deformations
- Applications of Model Reduction

Haptics (*Greek;* pertaining to the sense of touch)

- User feels forces generated by the haptic device.

- Requires high simulation update rates (1000 Hz)



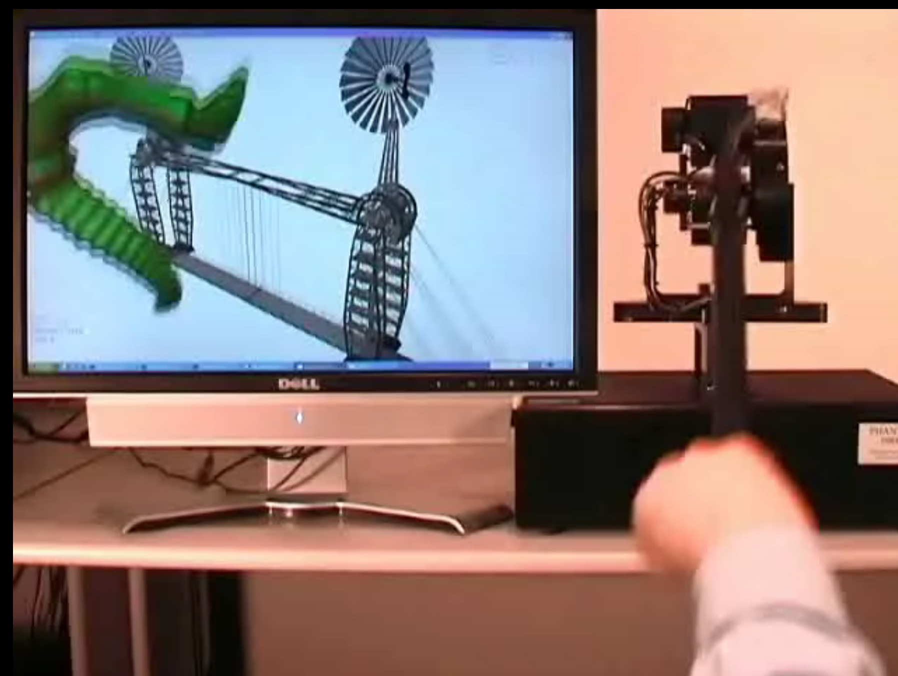
[Barbic and James 2005]

Haptic rendering of distributed contact

- Runs at 1000 Hz:

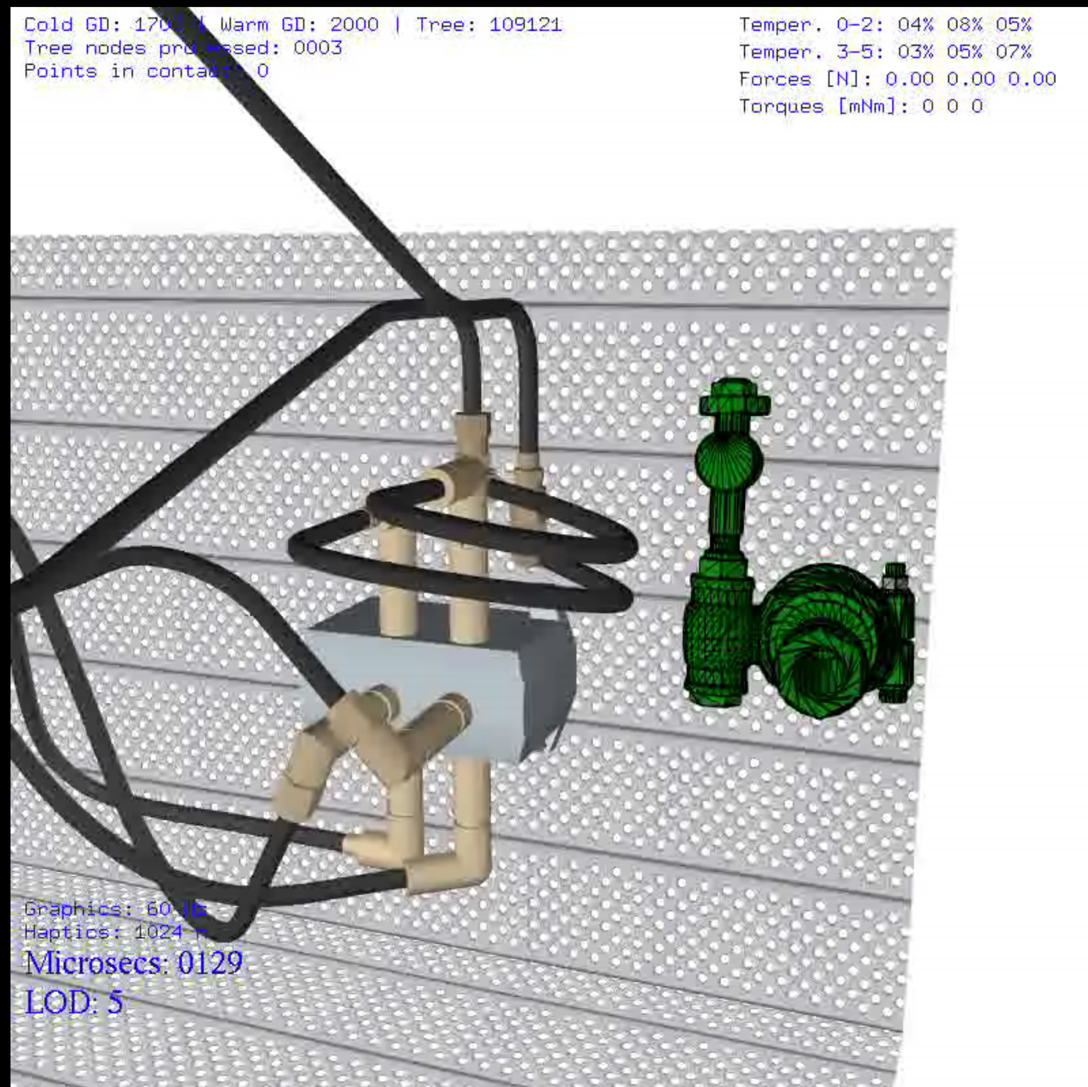
  deformable dynamics +
  collision detection +
  contact force computation.

- Adapts contact force accuracy to computer speed



[Barbic and James 2008]

Virtual assembly (aircraft geometry)



Both forces and torques rendered.

Deformable vs deformable contact

Deformable dragon and deformable dinosaur

Five-level hierarchical pointshell

256,000 points

15-dim deformation basis

Deformable distance field

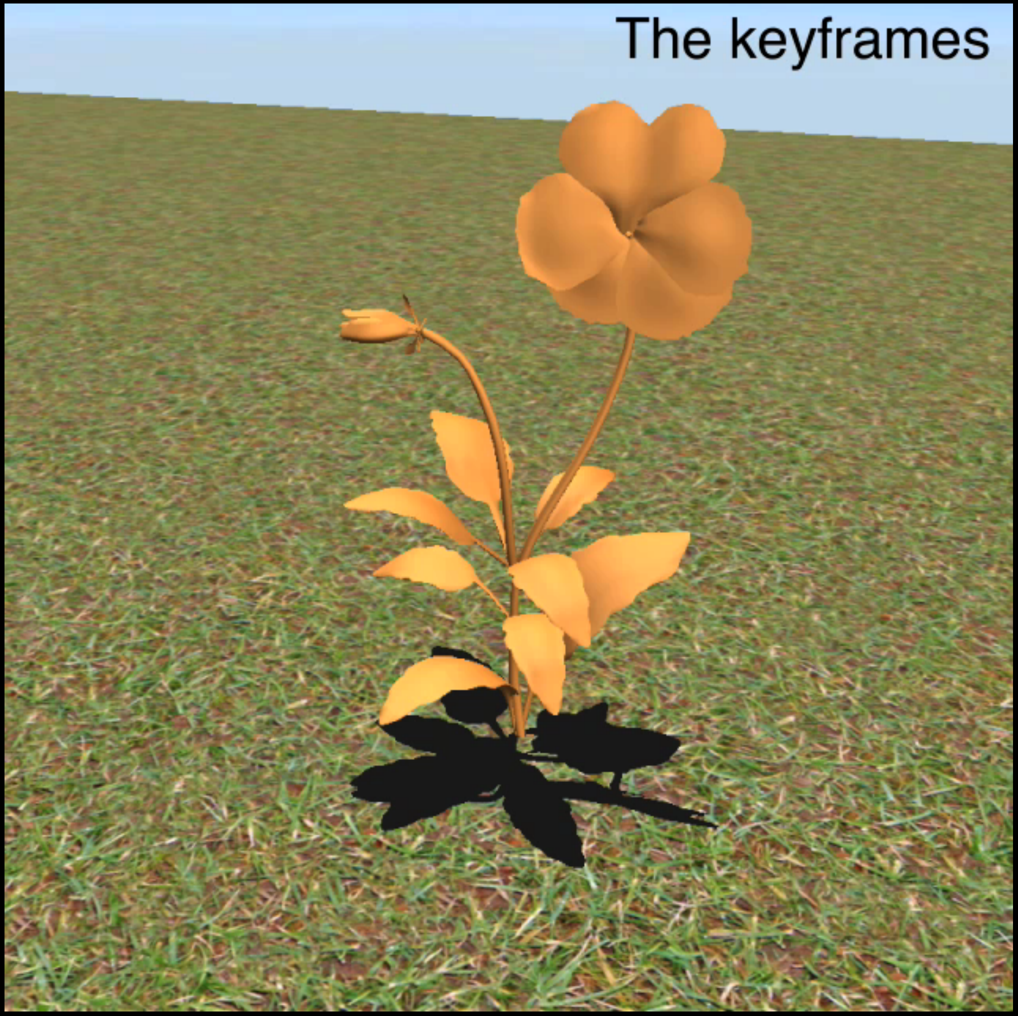
256x256x256

5 domains, 40 proxies total

15-dim deformation basis

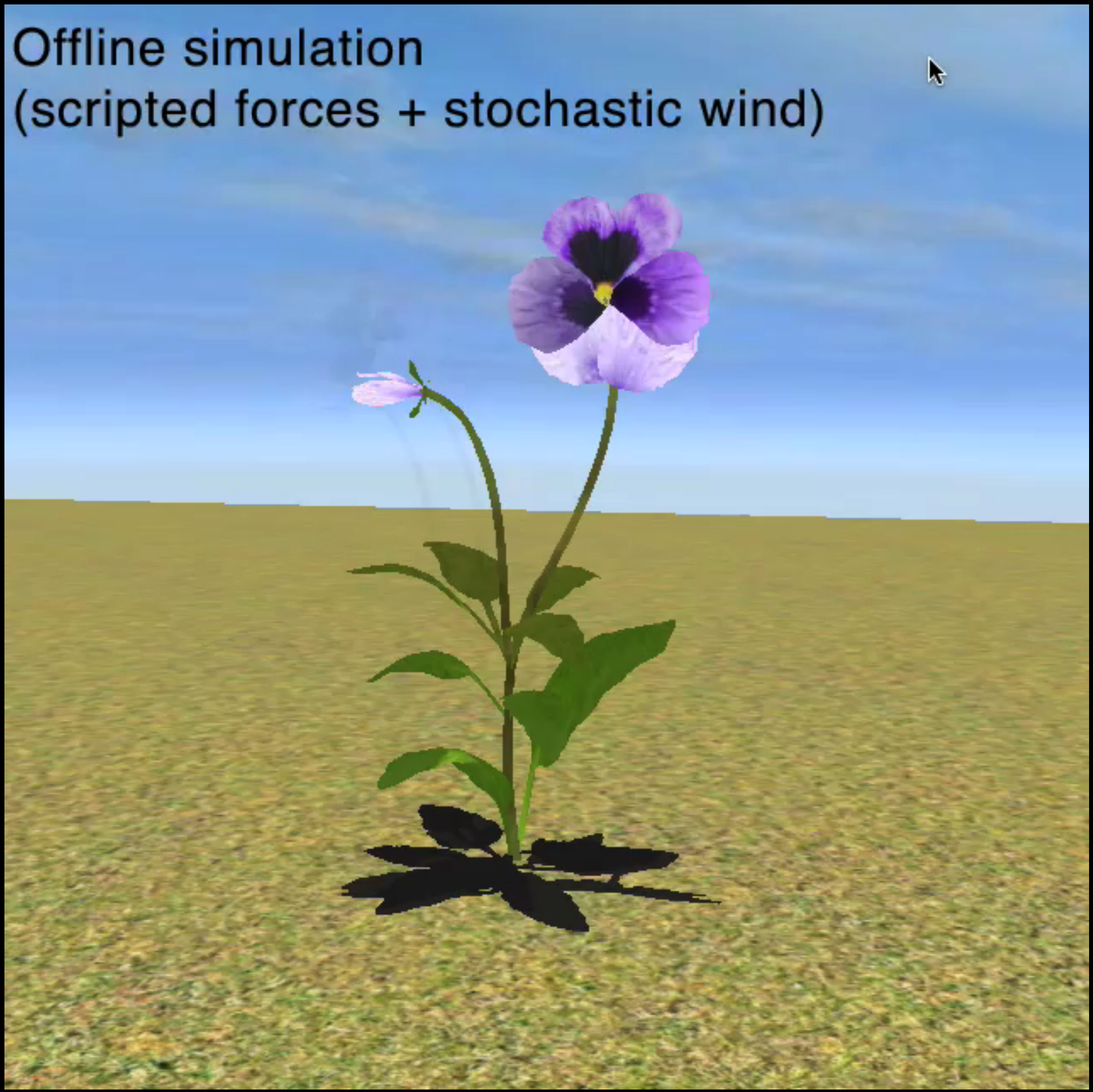
the domains

Optimal Control Using the Adjoint Method



[Barbic, da Silva and Popovic 2009]

Real-time tracking controller (using Linear Quadratic Regulators)



[Barbic and Popovic 2008]

Real-time tracking controller (using Linear Quadratic Regulators)

Bee lands on the flower

Controlled deformable object

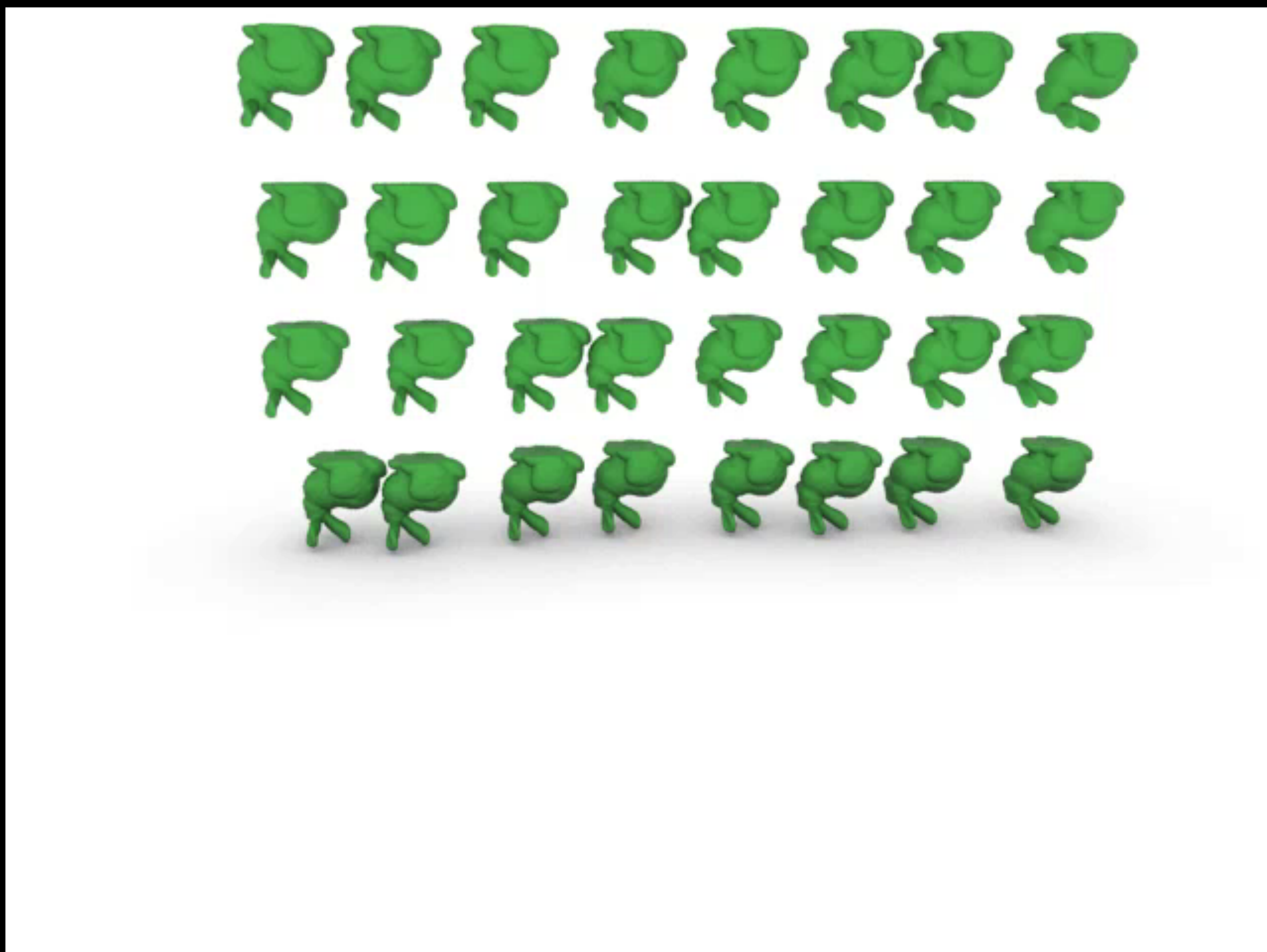
Fixed trajectory

[Barbic and Popovic 2008]

Uncontrolled

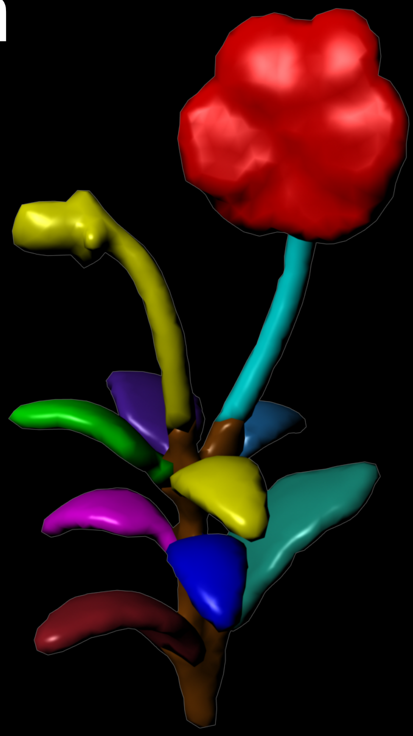
[Barbic and Popovic 2008]

Multibody dynamics with self-collision detection



[Barbic and James 2010]

Model Reduction + FEM + Domain Decomposition



- Decompose the object
- Simulate each domain using reduction
- Couple the domains
- Two approaches:
  - Via polar decomposition gradients [Barbic and Zhao 2011]
  - Via inter-domain spring forces [Kim and James 2011]

Model Reduction + FEM + Domain Decomposition

Via polar decomposition gradients

[Barbic and Zhao 2011]

Detail [Barbic and Zhao 2011]

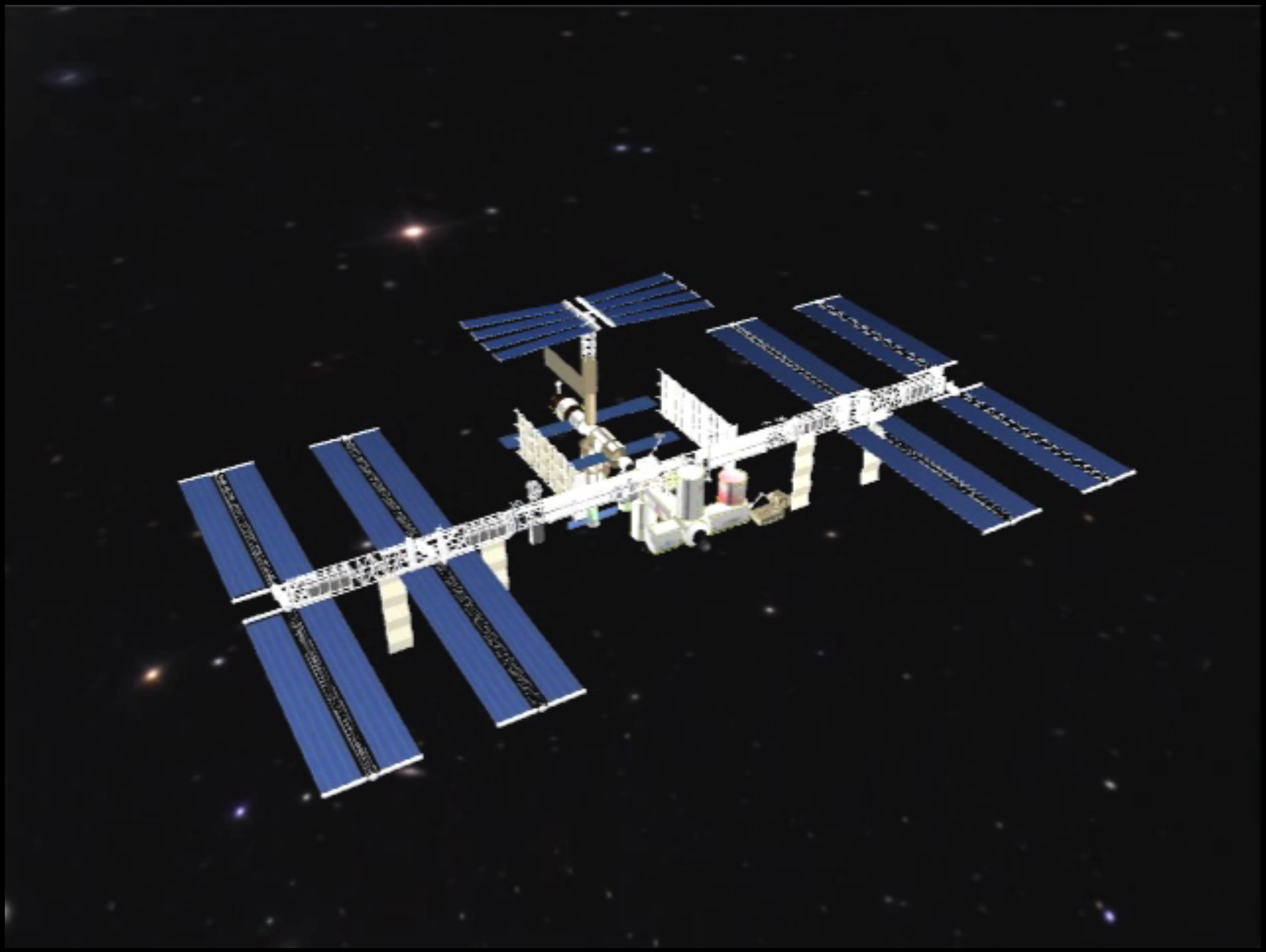
1435 Domains

11,972 Total DOFs

5 FPS

Space Station [Barbic and Zhao 2011]

dynamics: 75 fps, **2500x** speedup



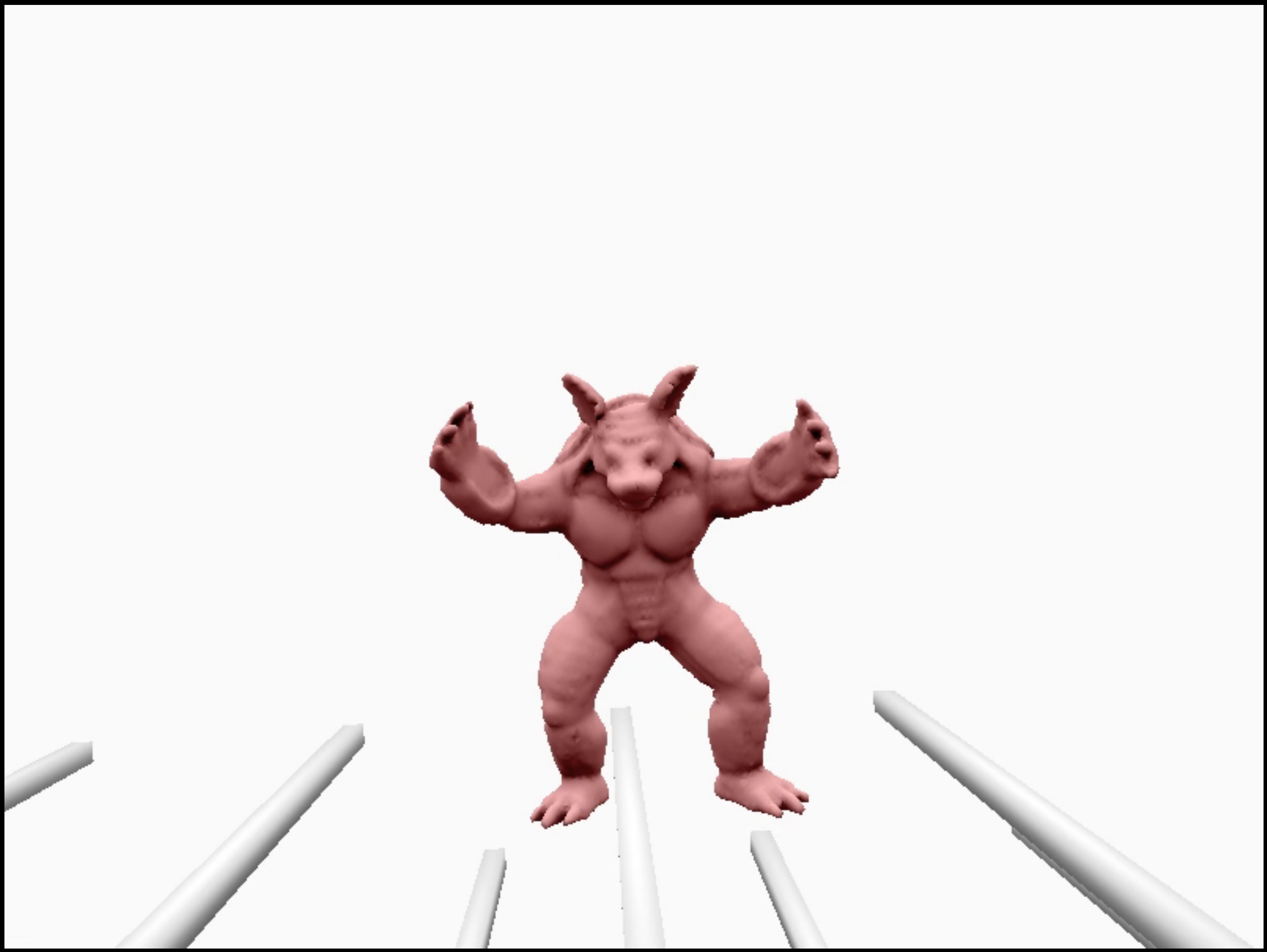
107,556 voxels

48 domains

921 DOFs

Model Reduction + FEM + Domain Decomposition

Via inter-domain spring forces



[Kim and James 2011]

Acknowledgments

- National Science Foundation (CAREER-53-4509-6600)

- Eftychios Sifakis

- Yili Zhao