Real-time Large-deformation Substructuring

Jernej Barbič, Yili Zhao

University of Southern California

Model Reduction + FEM + Domain Decomposition

1435 Domains

11,972 Total DOFs

5 FPS

Detail

1435 Domains

11,972 Total DOFs

5 FPS

Assumptions

3D volumetric mesh

Geometrically nonlinear FEM

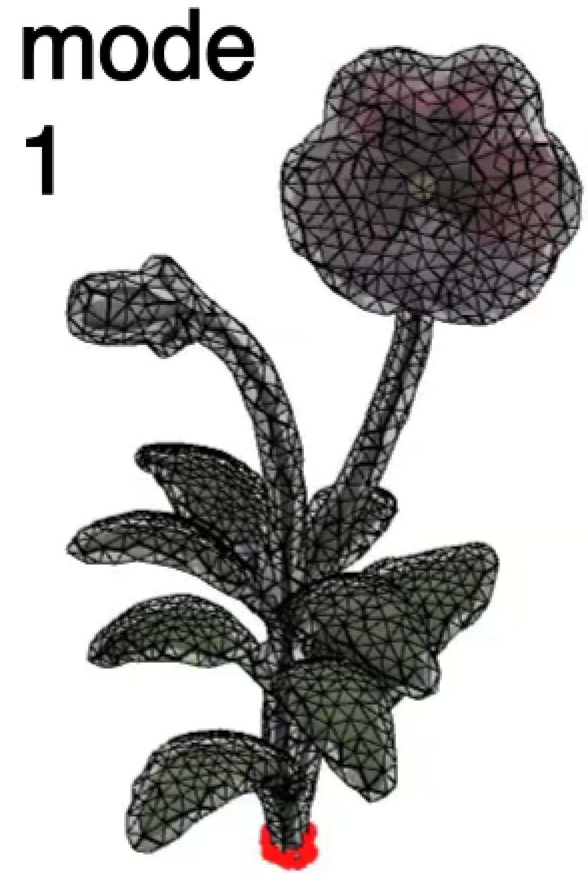
Boundary and initial conditions given

Want to simulate quickly, with rich deformations and local detail

Speed

7600 tetrahedra
3 fps

model
reduction



45 modes,
150 fps

Problem: large basis required for detail

Coupled equations

Complexity is $O(r^3)$

Slow for $r > 100$

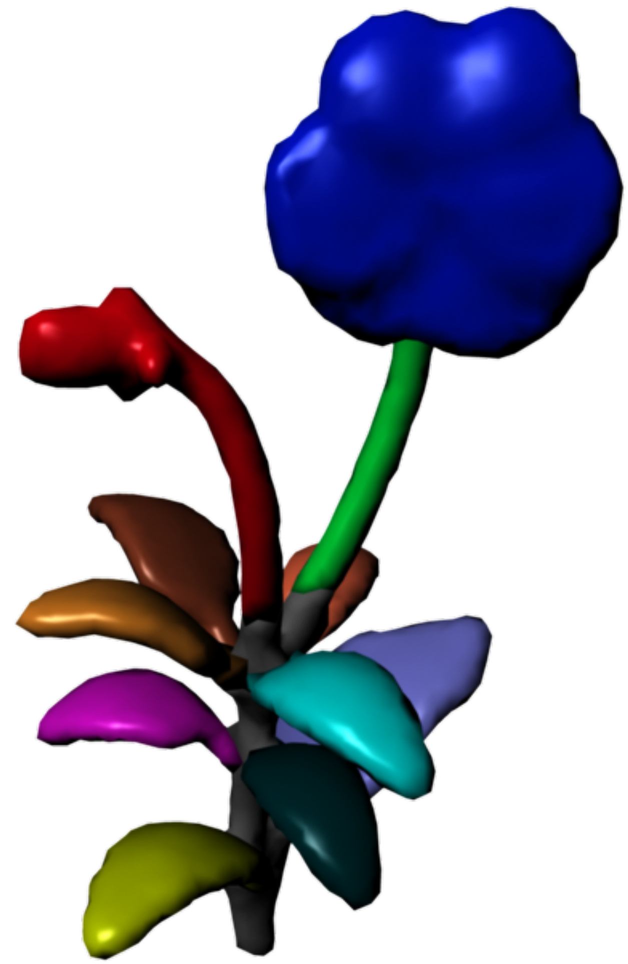
$r = 45$ modes

Domain Decomposition

1. divide the object into domains

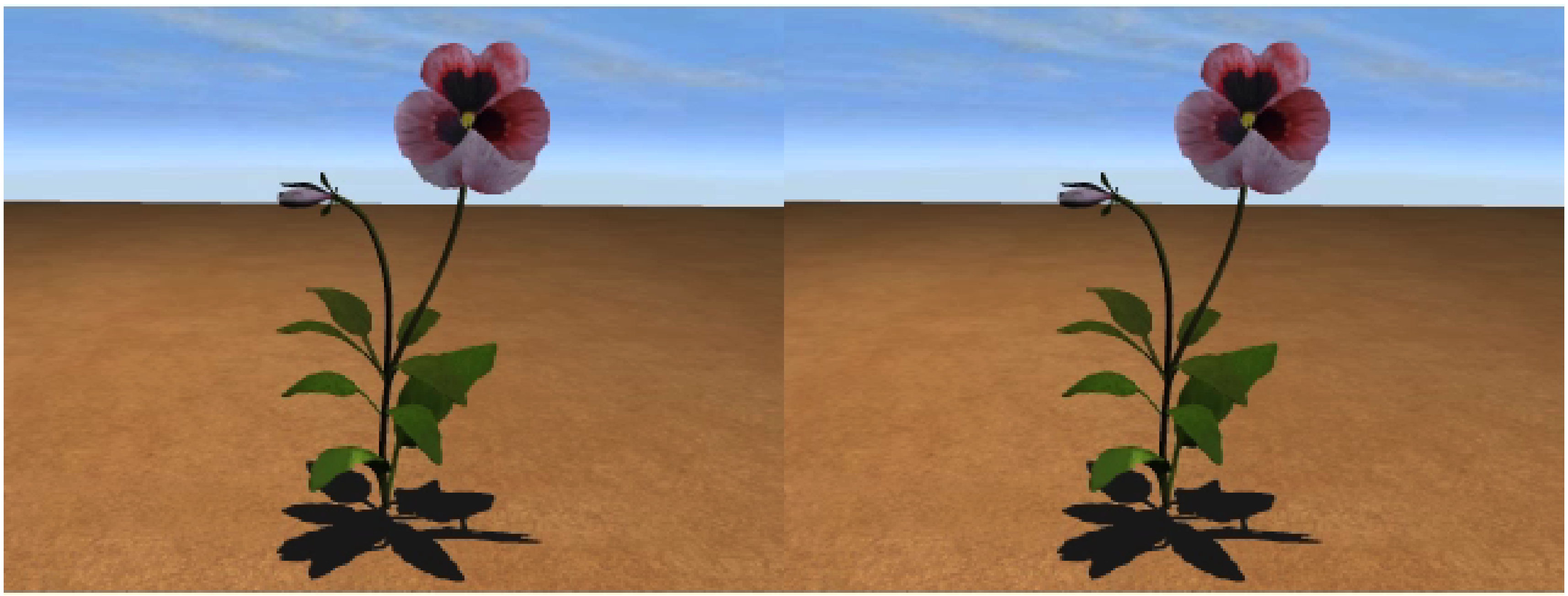
2. reduce each domain

3. couple domains with proper forces



the domains

Richer deformations for same computational effort



global basis
$r = 45$

our result
$r = 240$

Related Work: FEM with Reduction

Several methods

Single-domain simulations

[Metaxas and Terzopoulos 1992]

[Barbic and James 2005]

[An and colleagues 2008]

[Kim and James 2009]

[Kharevych and colleagues 2009]

[Nesme and colleagues 2009]

Related Work: Domain Decomposition

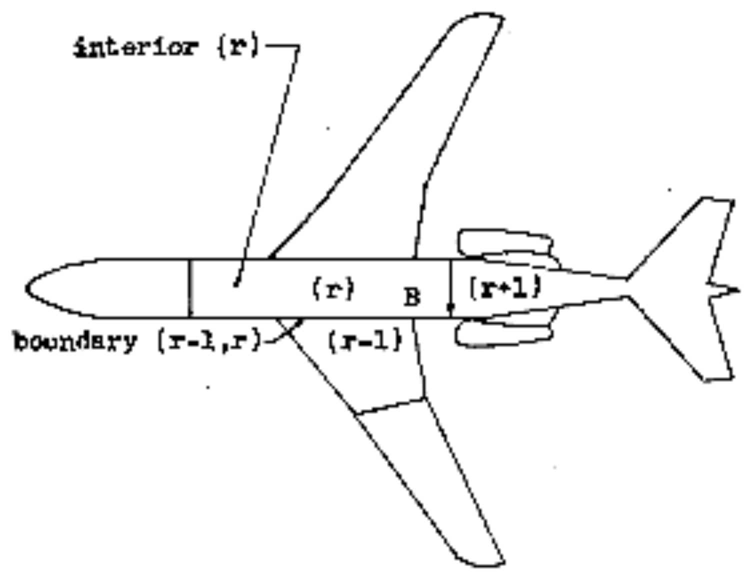
Old, established method

[Patnaik and colleagues 1994]

[Storaasli and Bergan 1987]

Component mode synthesis

Only small domain deformations



[Craig and Bampton 1968]

Related Work: Domain Decomposition in Graphics

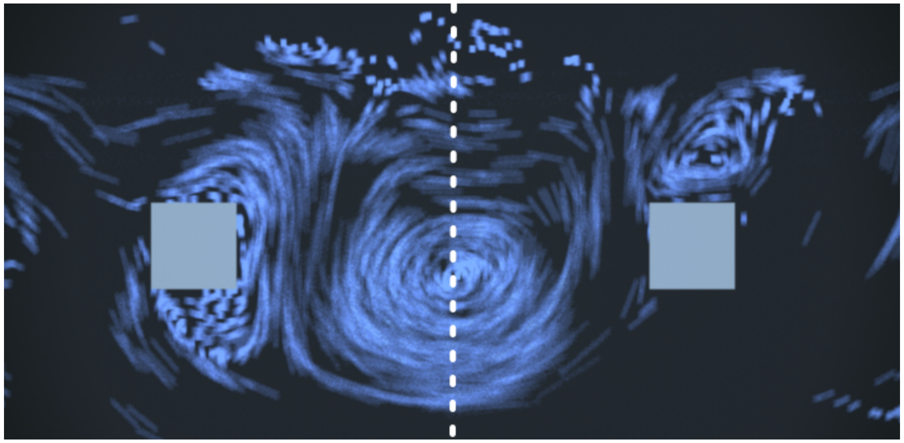
Linear domain deformations

[James and Pai 2002]

[Huang and colleagues 2006]

Spatially-adaptive model reduction

[Wicke and colleagues 2009]

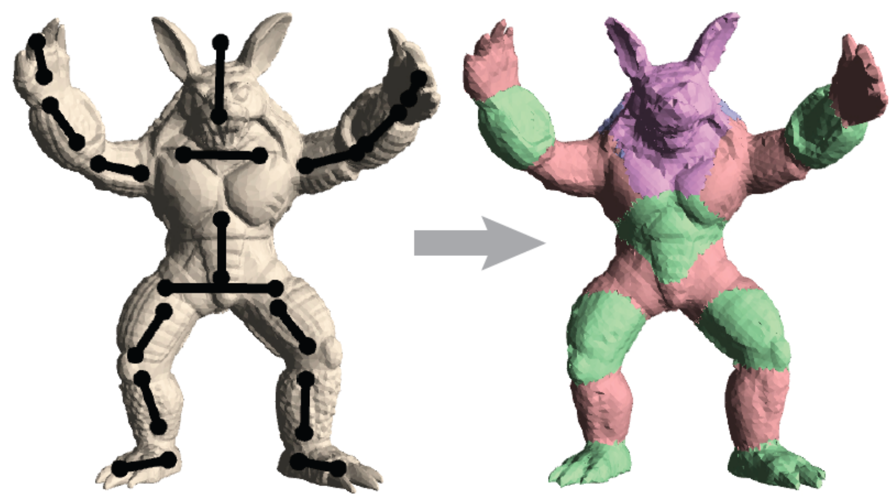


Related Work: Model Reduction + Domain Decomposition

Embedded skeleton

Connect domains with penalty forces

Concurrent work

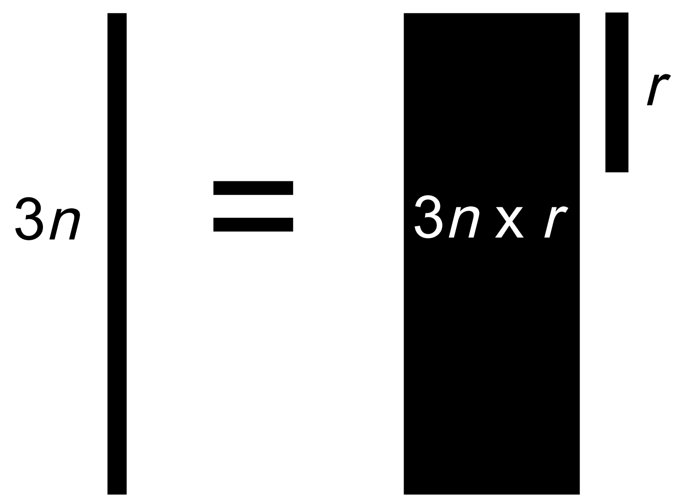


[Kim and James 2011]

Model Reduction

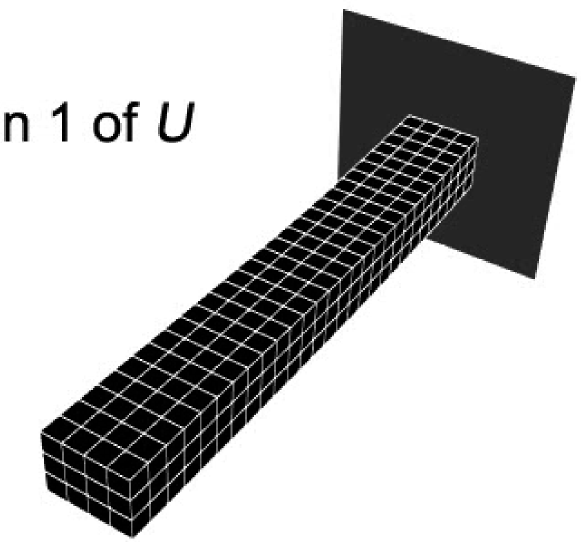
Reduction-Order Deformations

$$\mathbf{u} = \mathbf{U}\mathbf{q}$$



$r << 3n$

Mode 1
= Column 1 of *U*



Examples:
Linear & nonlinear modal analysis,
PCA, polynomial deformers,
FFD/embedded, splines, subdivision, etc.

Model Reduction

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

$$u = Uq$$

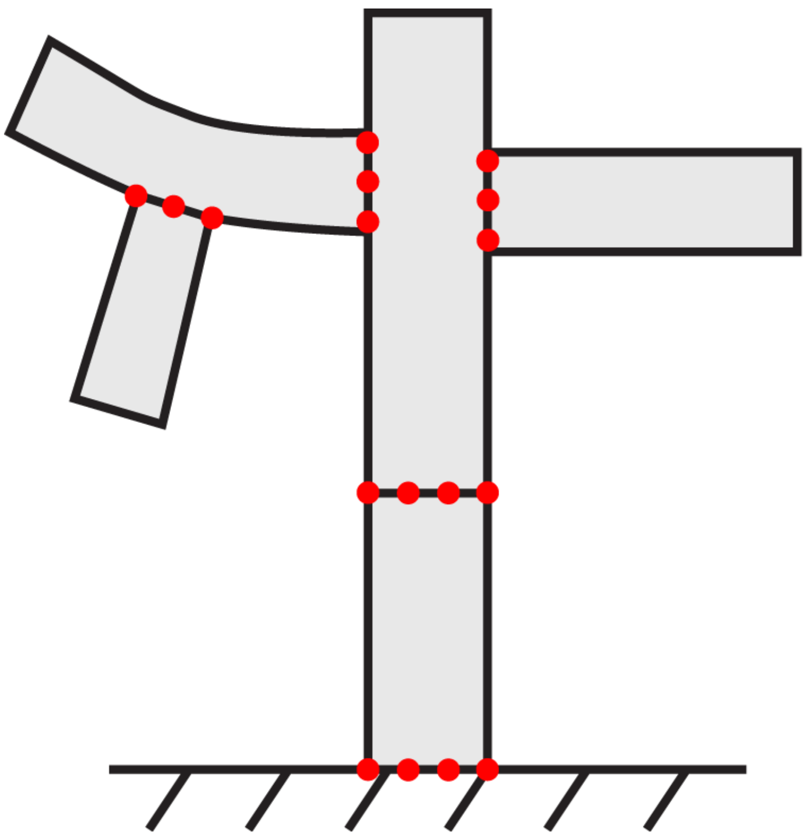
Pre-multiply with U^T

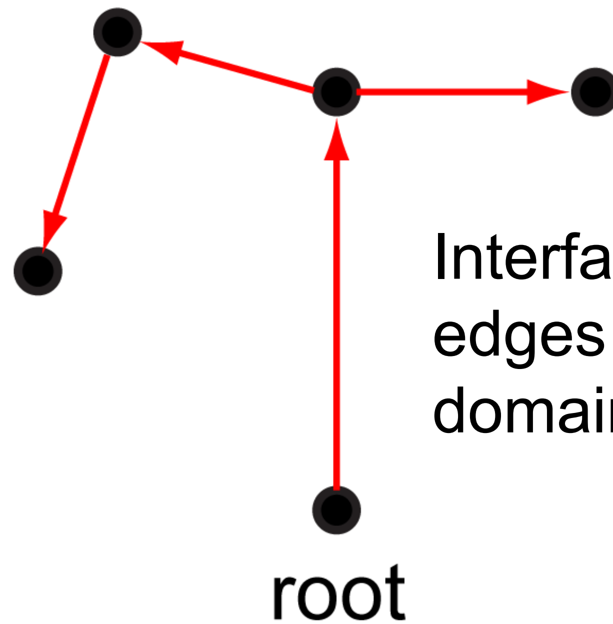
Low-dimensional approximation: $\ddot{q} = U^T F(Uq, U\dot{q}, t)$

Elasticity, fluids, voltages, etc.

Kinematics

The Decomposition and Domain Graph

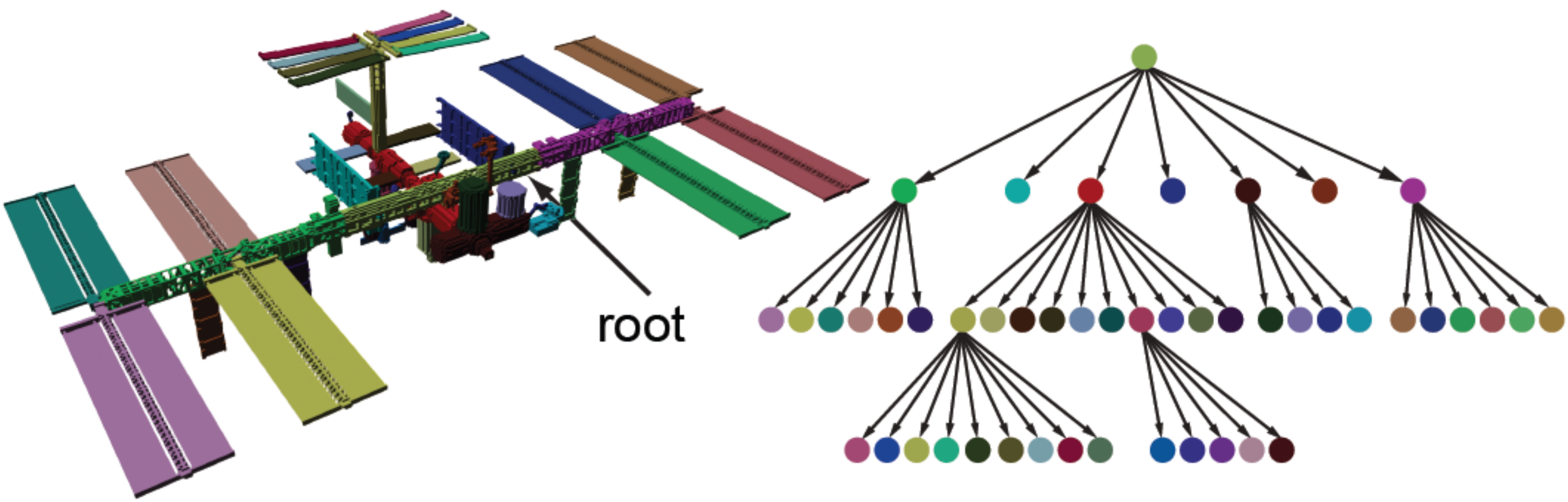




Interfaces are edges in the domain graph

Assumption: no cycles

The Decomposition and Domain Graph

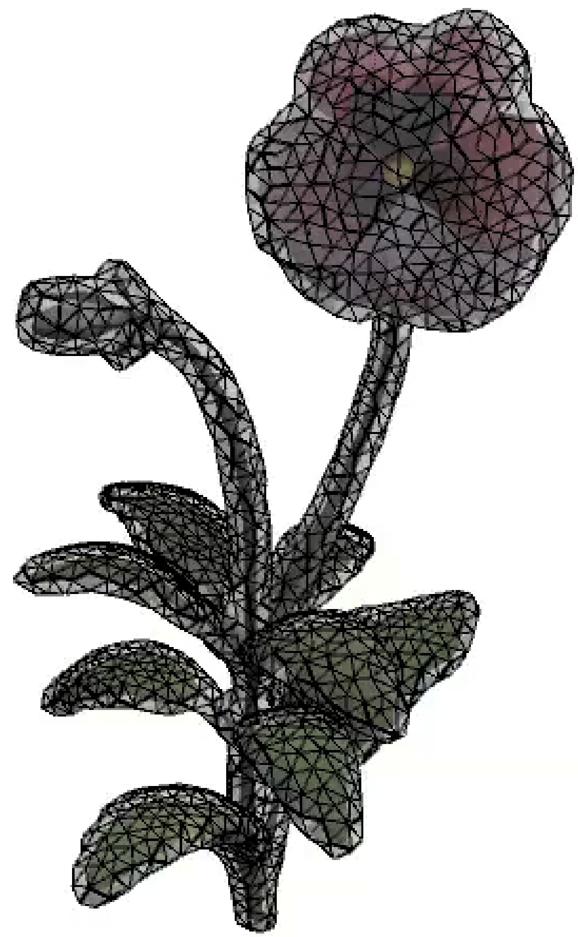


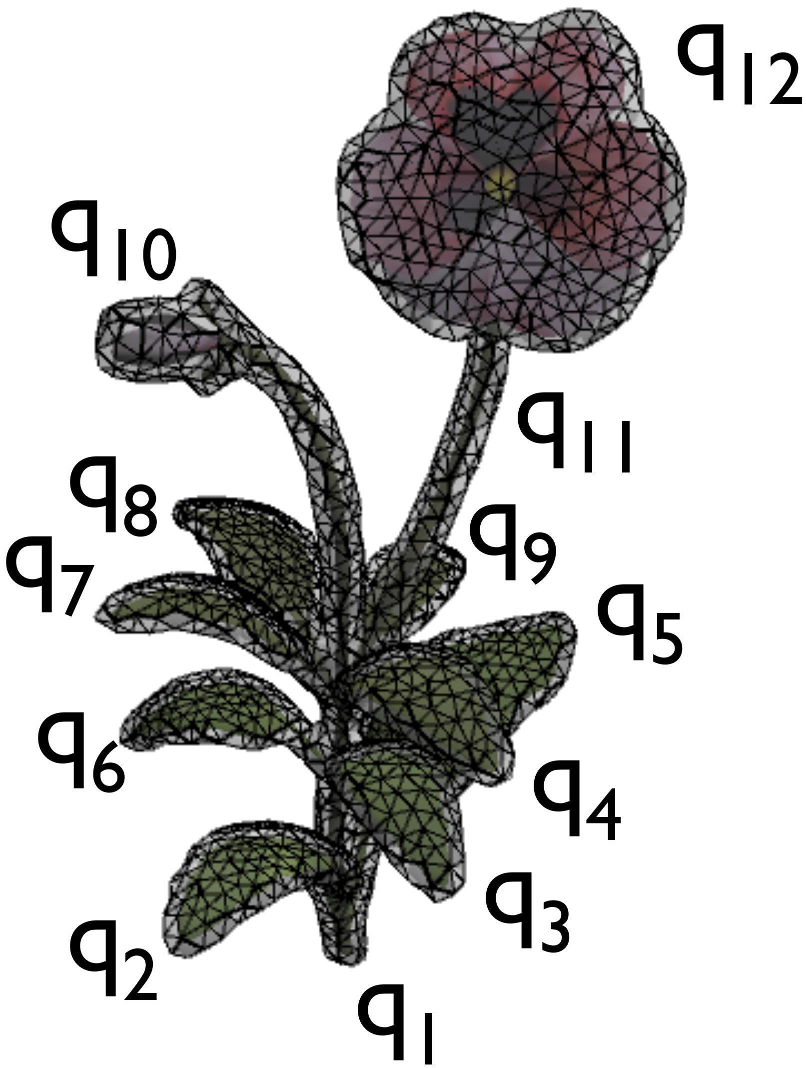
Compute Modes for Domains

Boundary condition:

fix vertices at interface to parent (red)

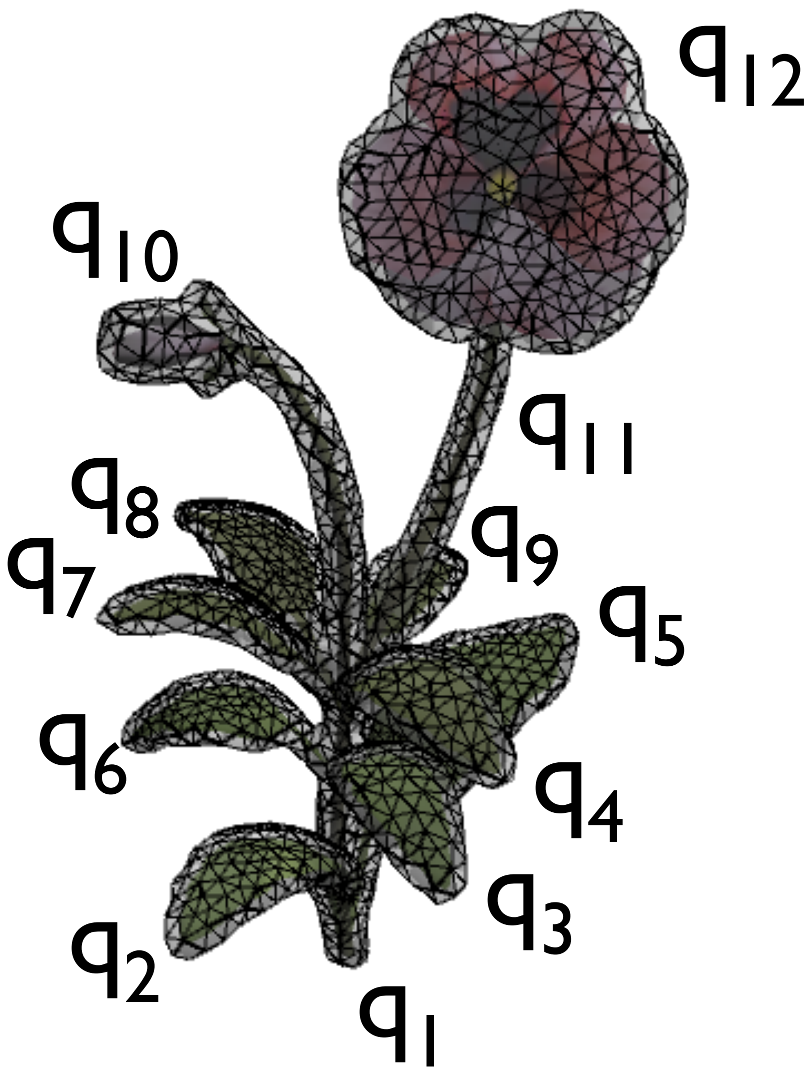
Note: Domains can be rigid





How to define a consistent, connected model ?

Degrees of Freedom

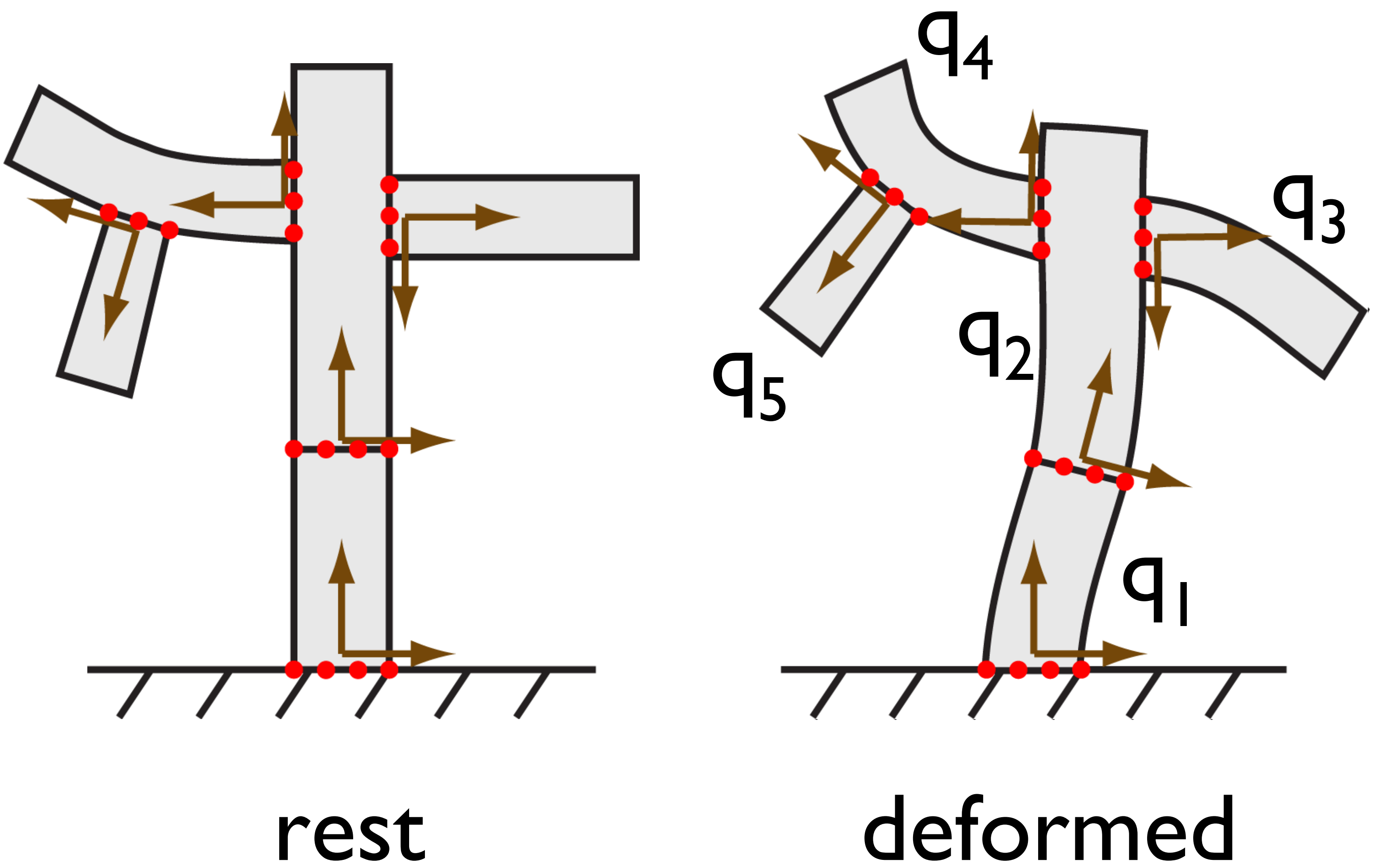


$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \\ q_{10} \\ q_{11} \\ q_{12} \end{bmatrix}$$

240 DOFs
(20 each
domain)

Frames

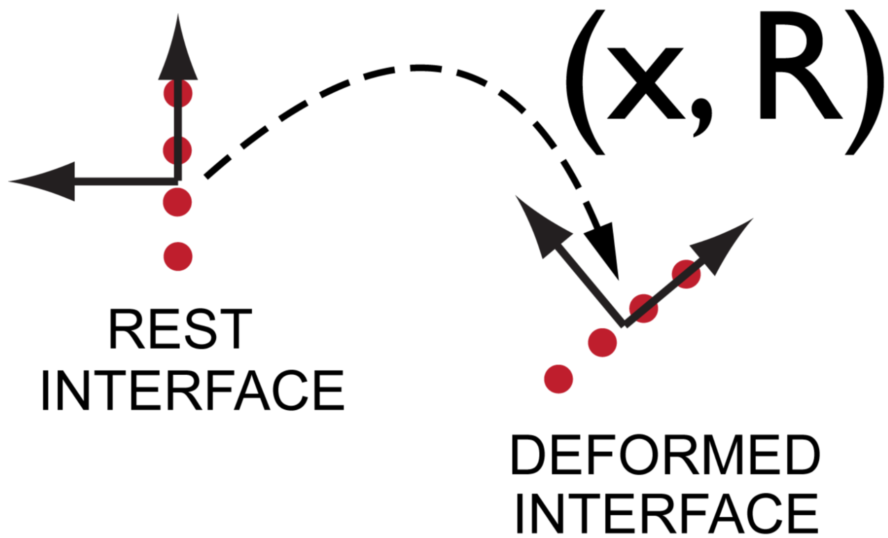
Assumption: interfaces undergo mostly **rigid** motion



Fit Best Rigid Transformation to Interface Vertices

Use polar decomposition

[Mueller and colleagues 2005]



A = R • S

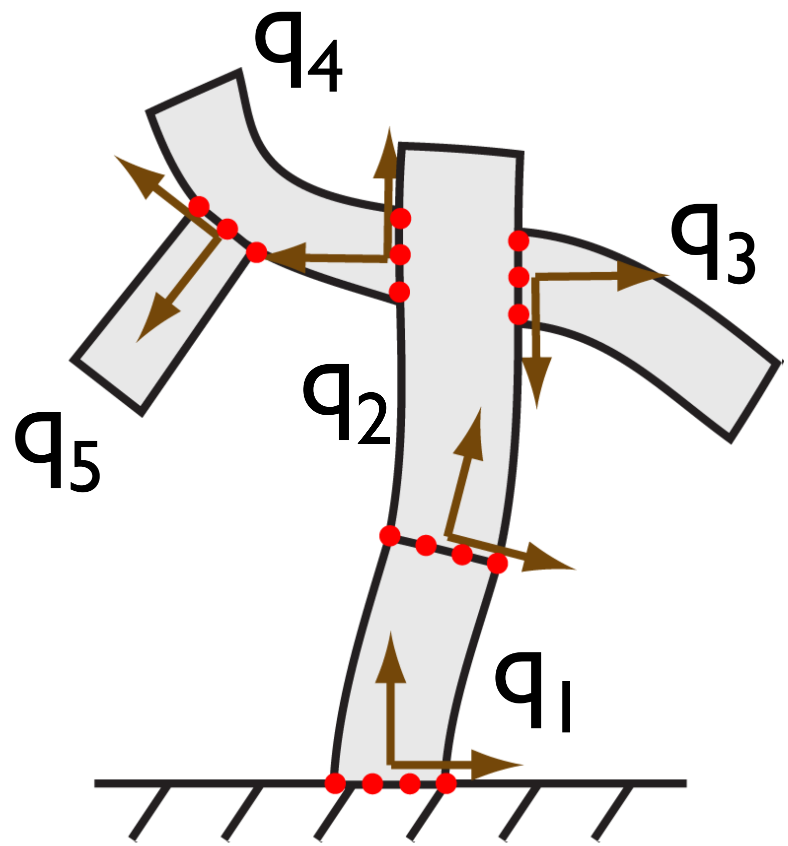
rotation   symmetric

Can be done entirely in the reduced space!

No constraints

Frames are uniquely determined by **q**
(**q** = “joints”)

“**Minimal**” formulation
[Featherstone 1987]



Fix interfaces with blending

no blending
(domain gaps visible)

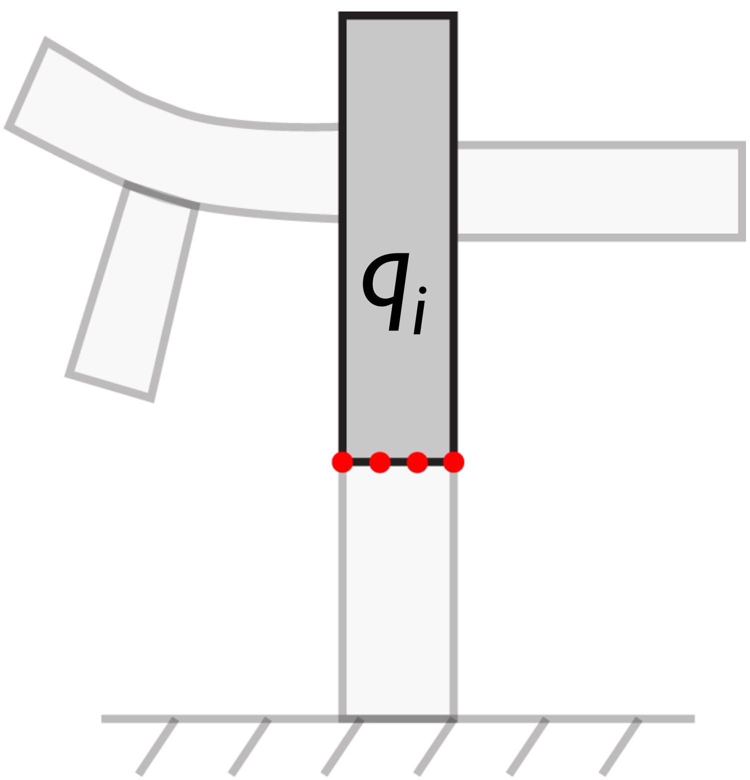
C0 blending

Dynamics

How to evolve **q** in time ?

Equations of motion

$$M\ddot{q}_i + D_i\dot{q}_i + f_i^{\text{int}}(q_i) = f_i^{\text{ext}} + f_i^{\text{system}} + f_i^{\text{interface}}$$



Simulate each domain in its local coordinate system

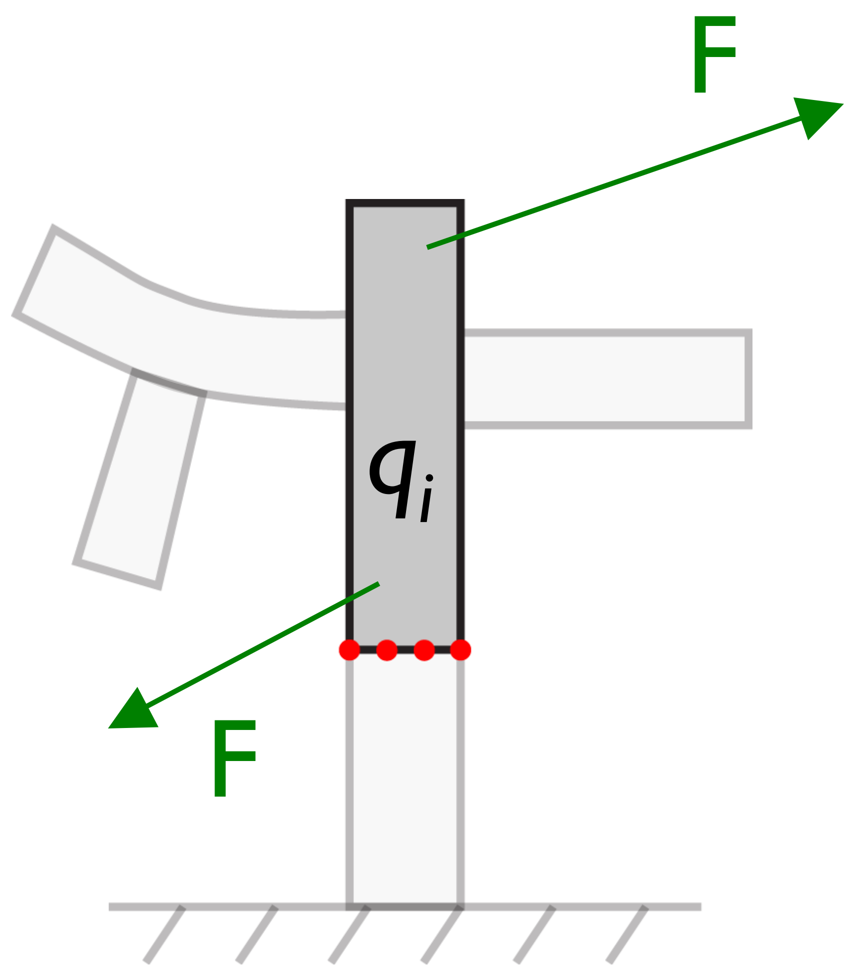
External Forces

$$M\ddot{q}_i + D_i\dot{q}_i + f_i^{\text{int}}(q_i) = \boxed{f_i^{\text{ext}}} + f_i^{\text{system}} + f_i^{\text{interface}}$$

User Forces

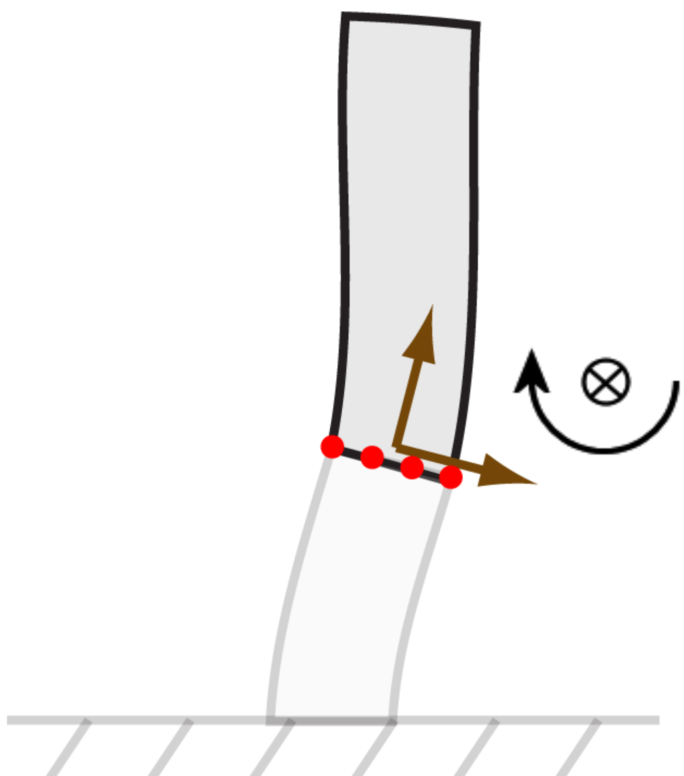
Wind

Gravity



System Forces

$$M\ddot{q}_i + D_i\dot{q}_i + f_i^{\text{int}}(q_i) = f_i^{\text{ext}} + \boxed{f_i^{\text{system}}} + f_i^{\text{interface}}$$



v

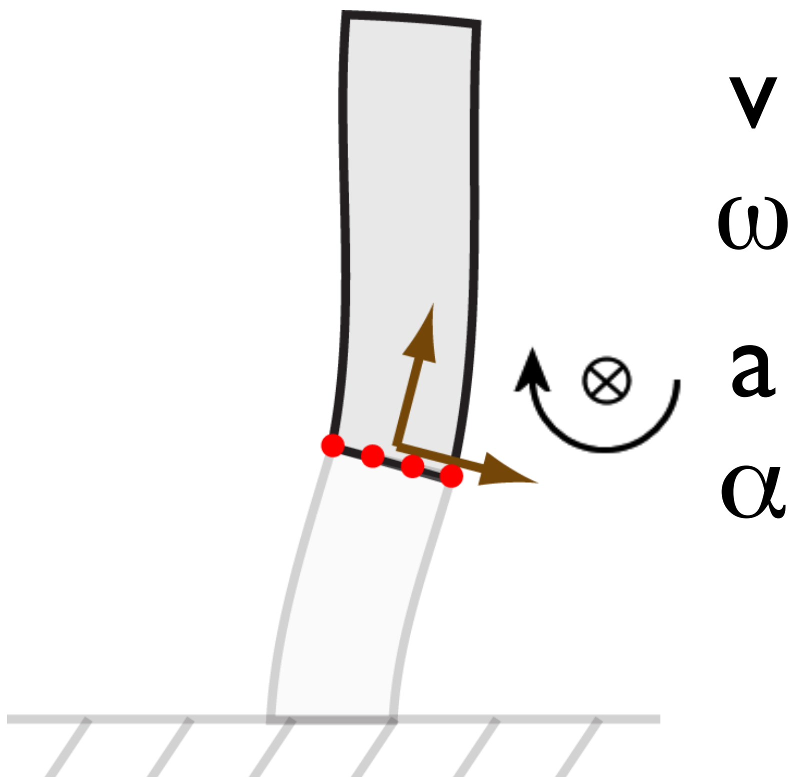
ω

a

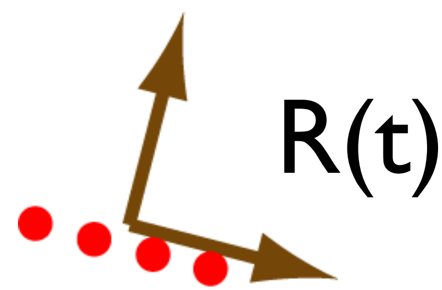
α

forces
due to frame
acceleration

System Forces



How to compute frame acceleration?

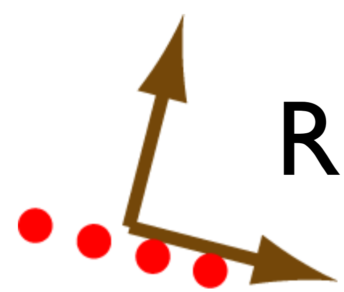


Polar Decomposition Gradient and Hessian

Let A(t) be 3x3 matrix that depends on a scalar parameter.

For any t, perform polar decomposition:

$$A(t) = \underbrace{R(t)}_{\text{rotation}}\ \underbrace{S(t)}_{\text{symmetric}}$$



R(t)

Polar Decomposition Gradient and Hessian

$$A(t) = R(t)\, S(t)$$

If $A(t)$, $A'(t)$, $A''(t)$ are known,

$$R'(t) = ?$$

$$S'(t) = ?$$

$$R''(t) = ?$$

$$S''(t) = ?$$

We derive formulas in the paper.
Code is on our website.

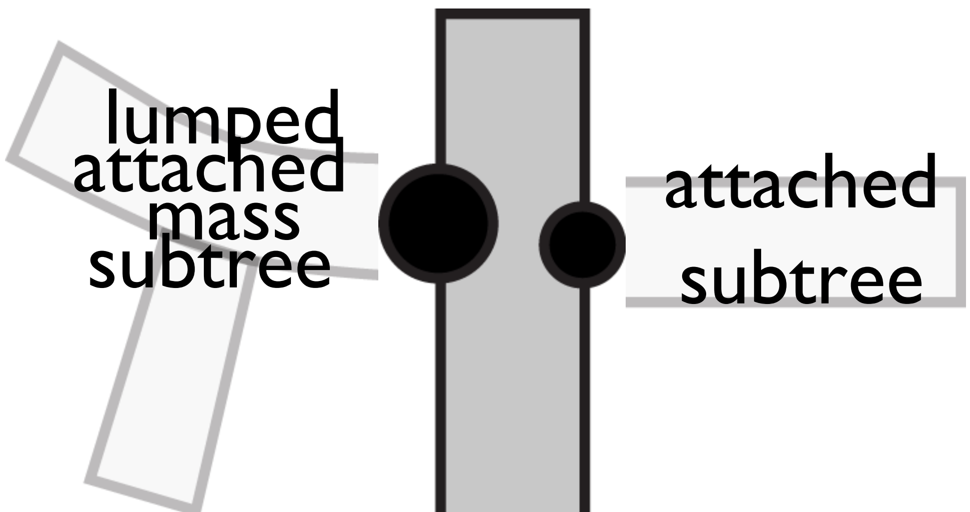
Interface Forces

$$M\ddot{q}_i + D_i\dot{q}_i + f_i^{\text{int}}(q_i) = f_i^{\text{ext}} + f_i^{\text{system}} + \boxed{f_i^{\text{interface}}}$$

Due to the
mass inertia of the
attached subtrees.

Approximation:
interface lumping

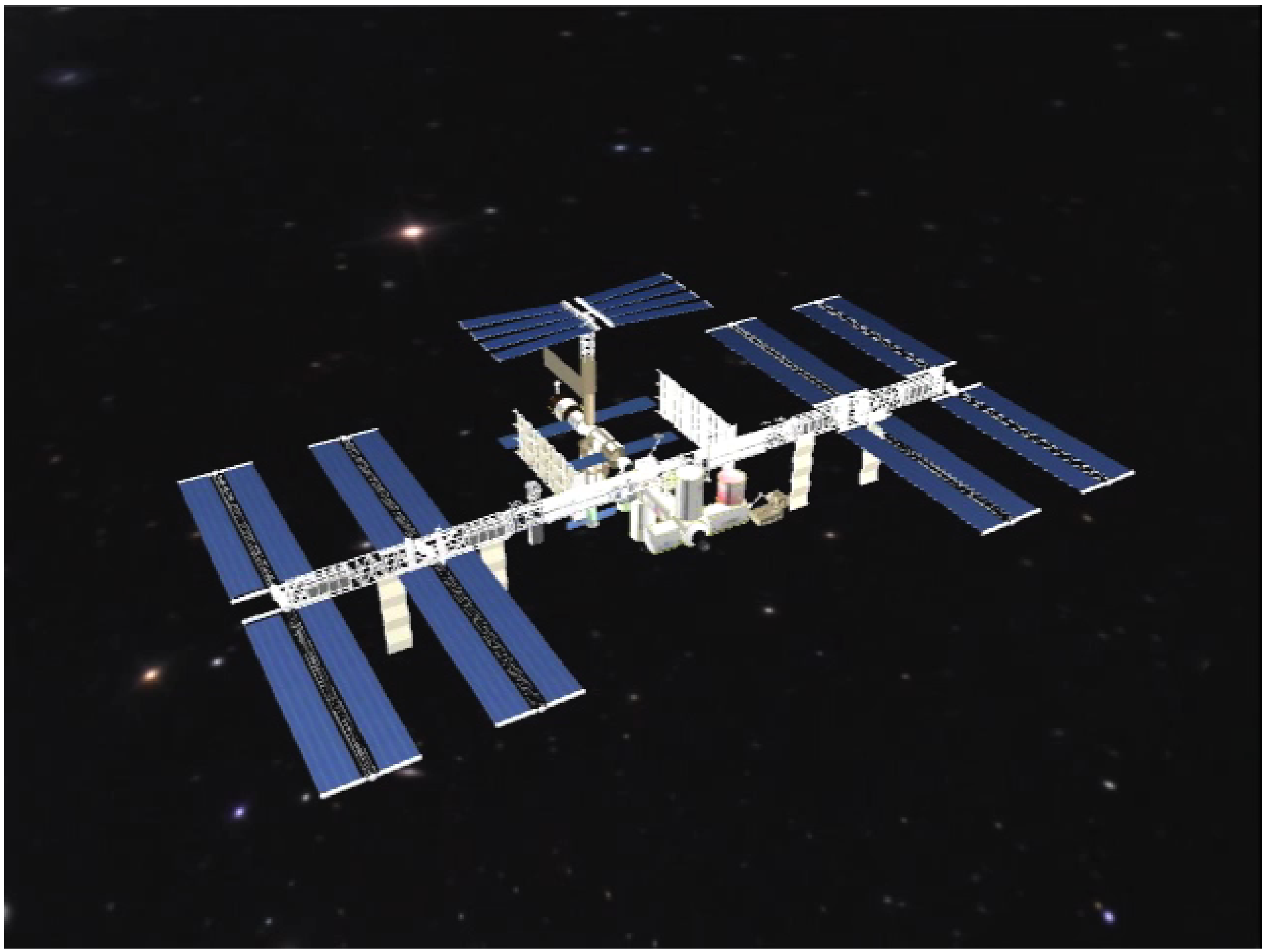
Exact: two-pass recursion
[Featherstone 1987]



Results

Space Station

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



107,556 voxels

48 domains

921 DOFs

Dragon with self-collision detection

dynamics: 75 fps, **500x** speedup

160,553 tetrahedra

40 domains

454 DOFs

Live Demo

Conclusion

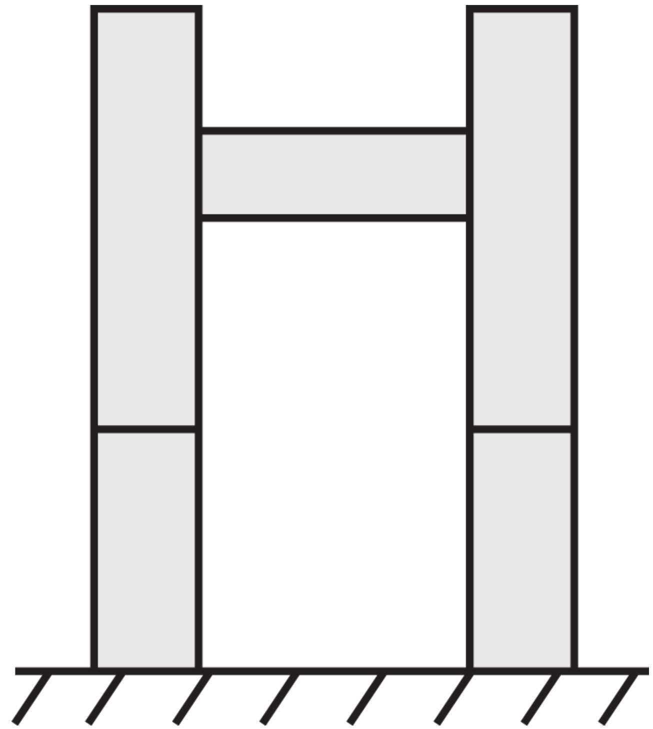
Domain Decomposition + Model Reduction

Local Detail at Interactive Rates

Gradient and Hessian of Polar Decomposition

Future work

- Loops
- Arbitrary-depth hierarchies
- Deformable interfaces
- Joints (articulation)
- Implicit integration of system forces
- Multicore implementations



Deformable Structure with Loops

Thank you

- Eitan Grinspun,
  Doug L. James,
  James O'Brien
- Anonymous Reviewers
- National Science Foundation
- Zumberge Innovation Fund at USC

with self-collision detection

Extra slides

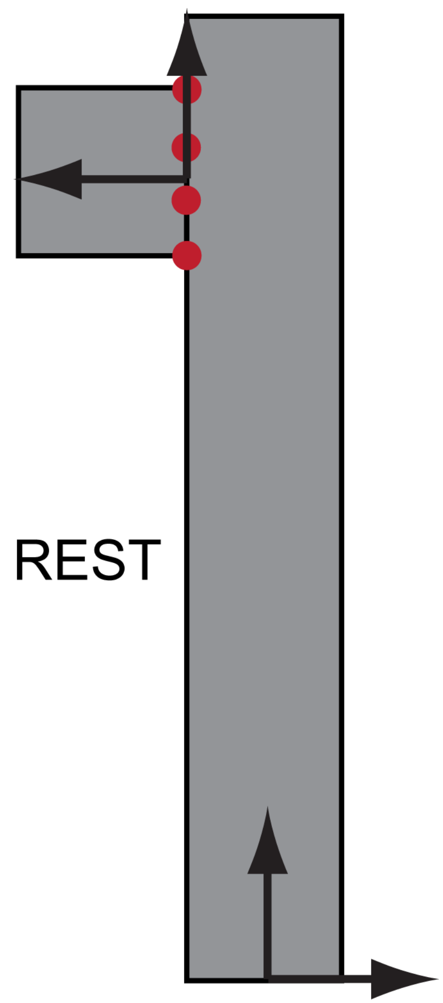
Model Reduction + FEM + Domain Decomposition

41 Branches
+ 1394 Leaves

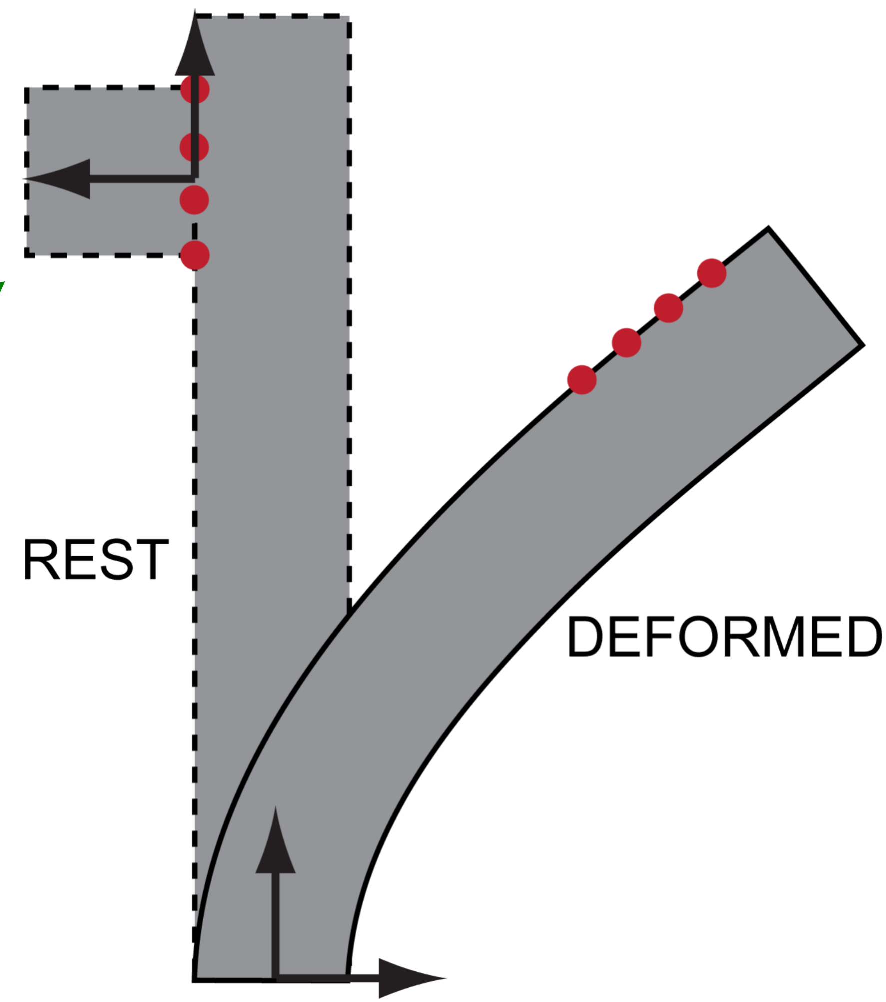
1435 Domains

triangle mesh   volumetric mesh

Computing Frames

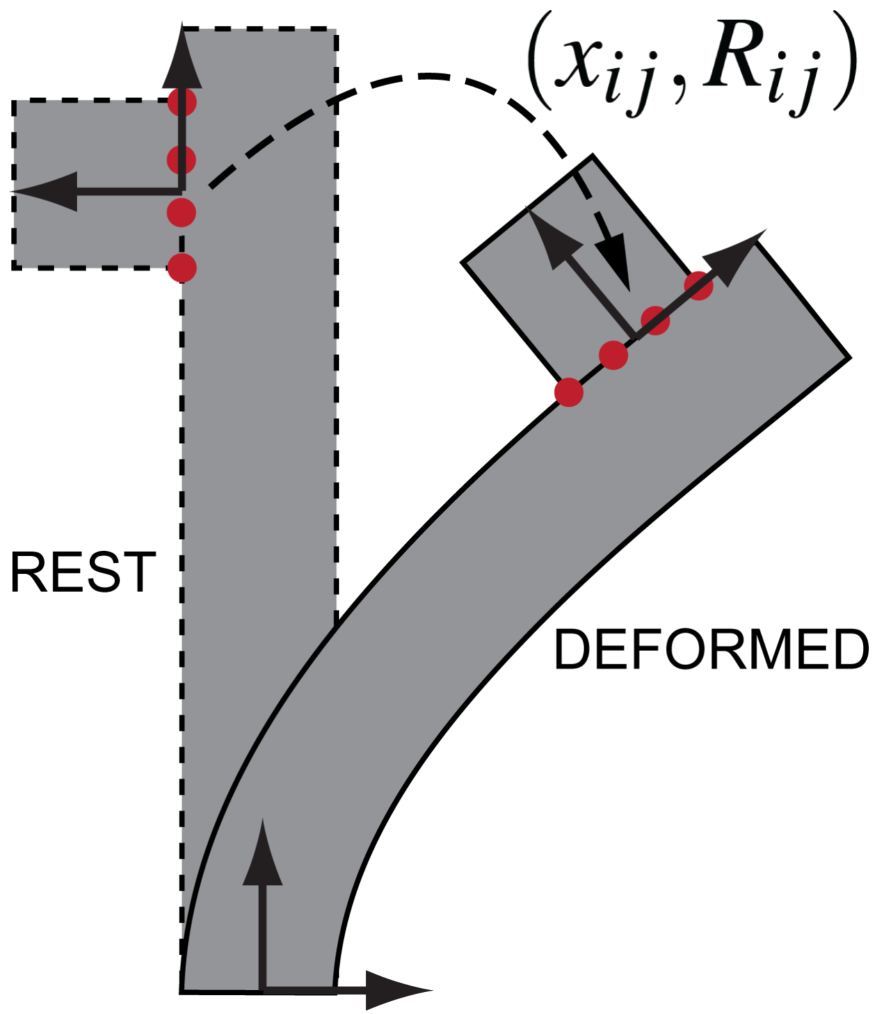


Computing Frames



Where should this domain go?

Computing Frames

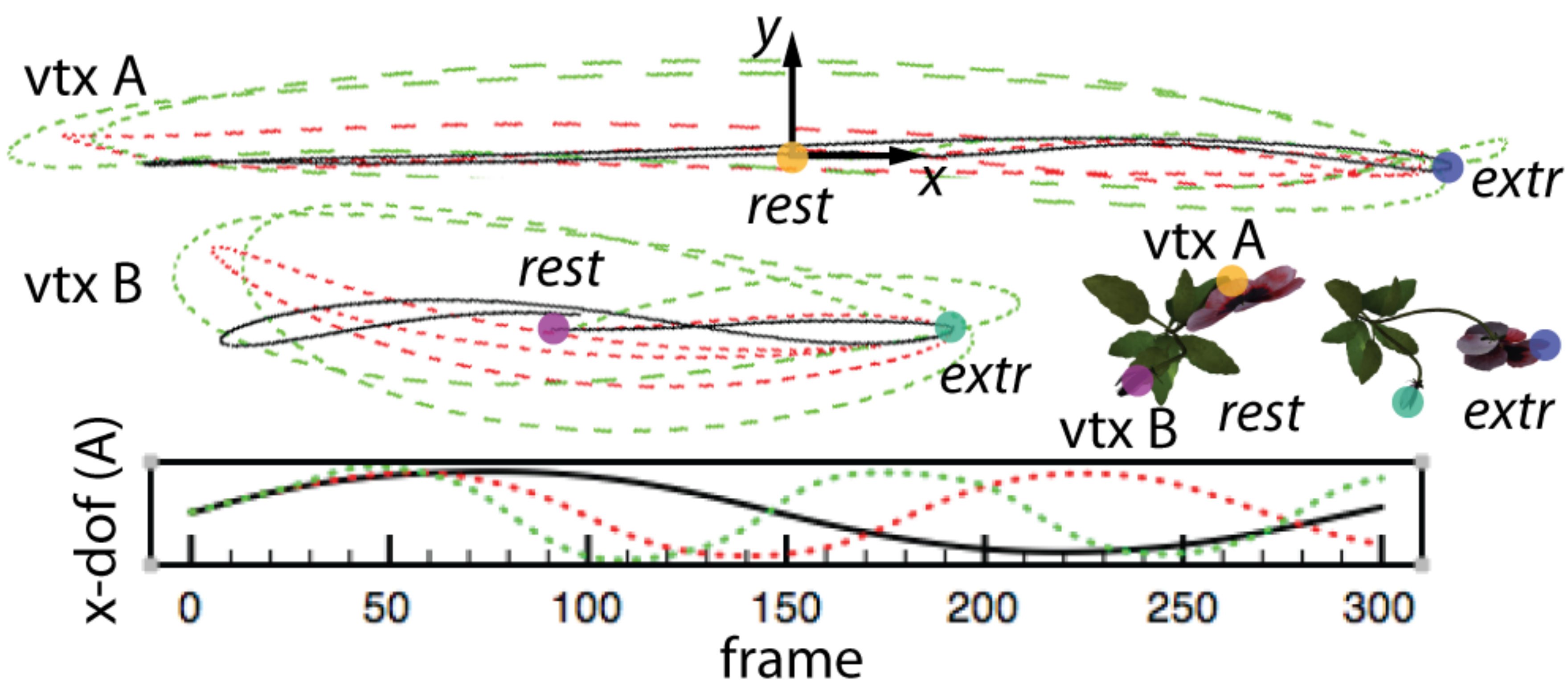


With and without blending

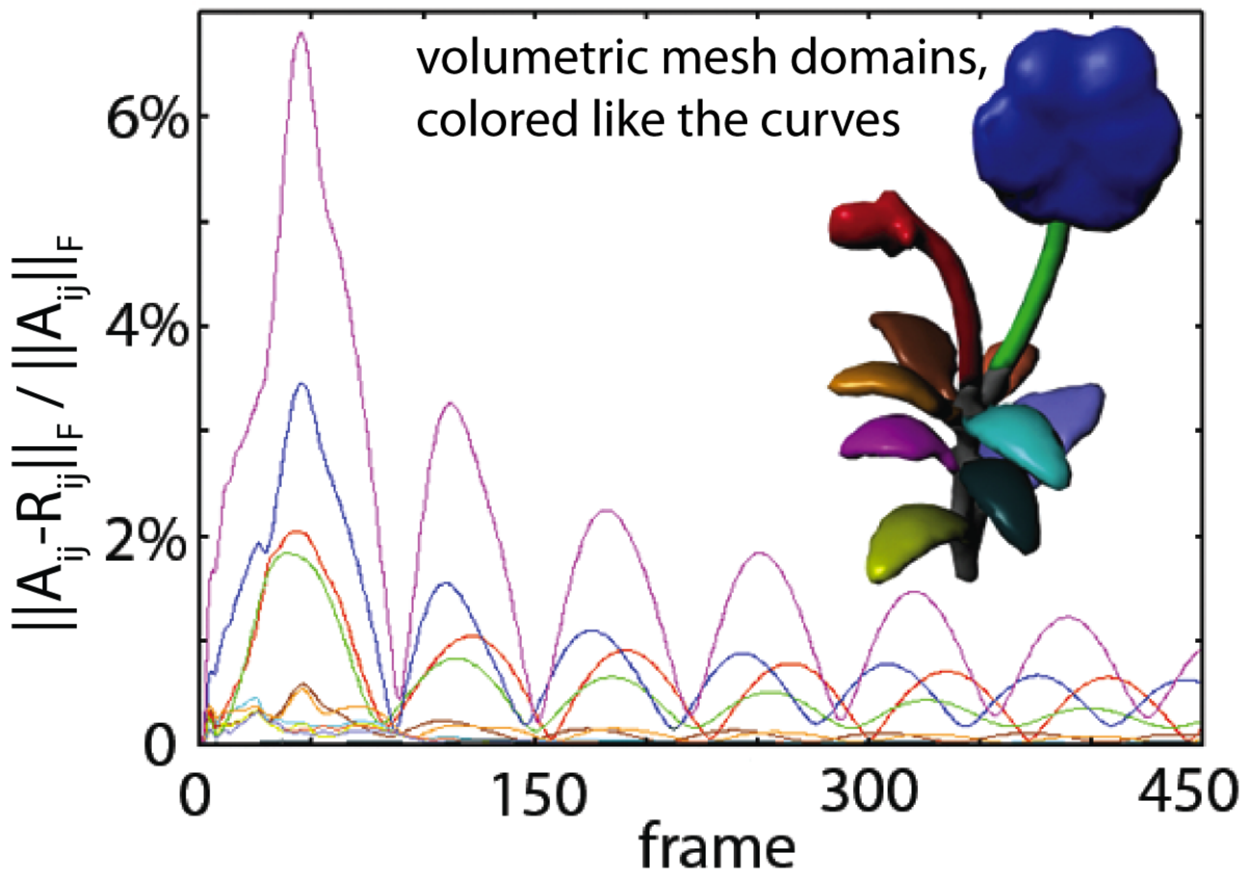
no blending
(domain gaps visible)

C0 blending

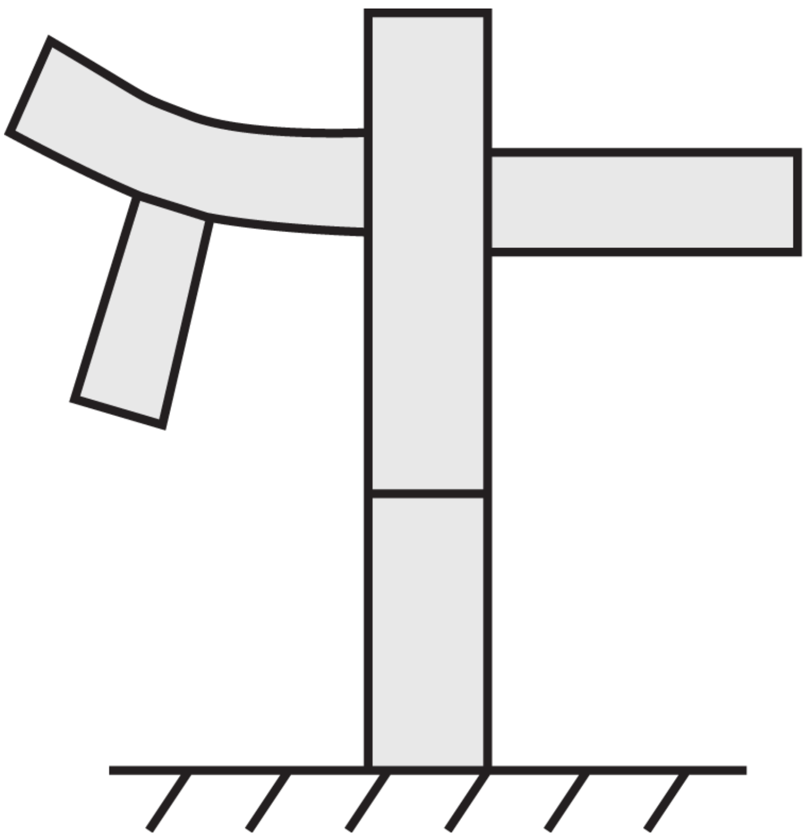
Accuracy of interface mass lumping

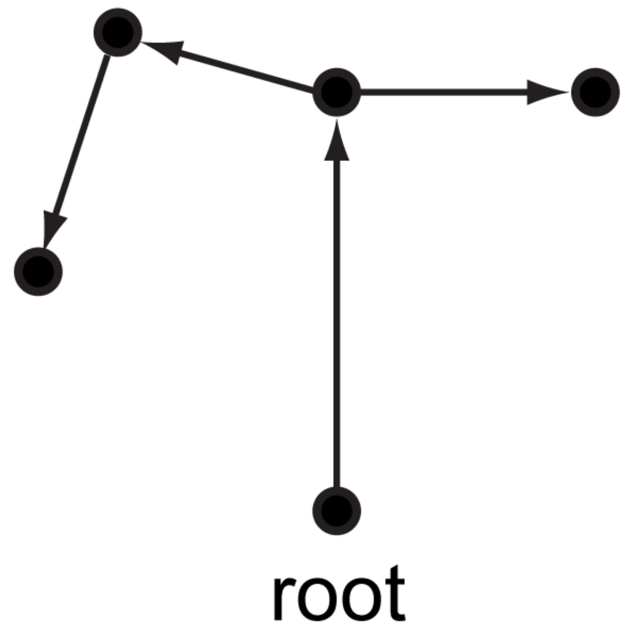


Deviation of interfaces from rotation



The Decomposition and Domain Graph





Assumption: no loops

Interfaces

Each domain has an interface to its parent domain.