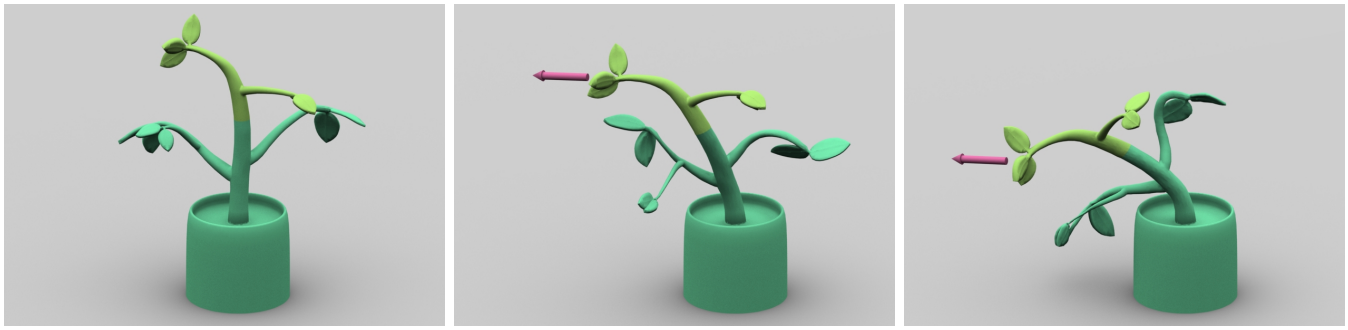
Pacific Graphics 2016
E. Grinspun, B. Bickel, and Y. Dobashi
(Guest Editors)

Volume 35 (2016), Number 7

Incremental Deformation Subspace Reconstruction

R. Mukherjee[1] X. Wu[1] H .Wang[†1]

[1]The Ohio State University



(a) Two domains (b) Stitched body simulated in the original subspace (c) Stitched body simulated in the new subspace

Figure 1: The plant example. Our system can quickly recalculate the subspace of a deformable body, after it gets modified by simple stitches, i.e., zero-length springs. The new subspace allows the two domains in (a) to be connected and simulated as shown in (c). In contrast, if the simulator uses the deformation subspace constructed for the original mesh before stitching happens, it will produce the locking issue due to the inconsistency of the subspace, as shown in (b).

Abstract

Recalculating the subspace basis of a deformable body is a mandatory procedure for subspace simulation, after the body gets modified by interactive applications. However, using linear modal analysis to calculate the basis from scratch is known to be computationally expensive. In the paper, we show that the subspace of a modified body can be efficiently obtained from the subspace of its original version, if mesh changes are small. Our basic idea is to approximate the stiffness matrix by its low-frequency component, so we can calculate new linear deformation modes by solving an incremental eigenvalue decomposition problem. To further handle nonlinear deformations in the subspace, we present a hybrid approach to calculate modal derivatives from both new and original linear modes. Finally, we demonstrate that the cubature samples trained for the original mesh can be reused in fast reduced force and stiffness matrix evaluation, and we explore the use of our techniques in various simulation problems. Our experiment shows that the updated subspace basis still allows a simulator to generate visual plausible deformation effects. The whole system is efficient and it is compatible with other subspace construction approaches.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation.

1. Introduction

Many real-time simulation applications demand interactive editing on a deformable body. In virtual surgeries, surgeons perform cutting and stitching operations to modify the connectivity of a soft tissue. During computer-aided design, designers and artists adjust the stiffness and the size of a 3D model to produce its optimal shape. These topological and geometric editing operations should be efficiently handled on the fly, without noticeable lags that compromise user's experience.

Subspace deformation, also known as *dimensional model reduction* or *reduced-order deformation*, has been an important graphics research topic in recent years, thanks to its ability to significantly

† Corresponding author: whmin@cse.ohio-state.edu

accelerate deformable body simulation. The basic strategy of this technique is to restrict the deformation into a subspace, so that the dynamical system is reduced from the full space to the subspace, in which the system solution can be found in real time. Although subspace simulation is fast, the computational cost spent on calculating the subspace is often non-trivial, which may take several seconds or even minutes for a high-resolution mesh. For applications that do not involve interactive editing on the rest shape of a deformable body, this is not an issue since the subspace can be calculated ahead of time as a pre-computation step. However, if an application requires to interactively update the rest shape, the subspace recalculation cost cannot be ignored. Without subspace recalculation, the simulation result suffers from the locking issue as shown in Figure 1b, because the original subspace does not offer enough freedom for the model to deform.

For faster subspace recalculation, one natural idea is to divide the mesh into multiple domains and build the subspace of each domain separately. If the interface between two domains is small, each domain can be simulated in its own subspace with the interface implemented as a rigid constraint [BZ11]. If the interface is large, its deformation cannot be ignored and it can be incorporated into the subspace simulation of each domain as hard constraints by Lagrangian multipliers [HLB*06]. Since the subspaces may not be consistent, hard constraints often suppress subspace deformations. To address this issue, Kim and James [KJ11] replaced hard constraints by soft spring constraints, which allow two domains not to be strictly aligned at the interface. Yang and colleagues [YXG*13] developed a boundary-aware method to make the subspaces more consistent at the interface. To do so, their method incorporates more deformation modes into the subspace and it considers linear modes only. [HZ13] handles local deformation by supplementing subspace basis with additional basis functions. Multidomain subspace based simulations were also used in [ZB13] to demonstrate the tuning of stiffness in an arbitrary domain in a real-time setting. Recently, Wu and collaborators [WMW15] proposed to simulate rigid motions and subspace simulations of all of the domains under a unified framework, to better eliminate artifacts near the interface. In general, existing domain decomposition techniques still suffer from a variety of limitations, and there exists no effective way to fully address the interface issue yet.

Different from recent research [vTSSH13, YLX*15] on expediting the subspace construction process itself, our work tries to study the problem in an incremental fashion. *Specifically, after a mesh gets modified by some small changes, can we quickly obtain its new subspace from the original subspace?* Here the mesh can be a connected mesh, or a mesh with disjointed domains whose subspaces can be combined into a single subspace. By incorporating mesh changes into the recalculated subspace, we do not need to apply additional constraints in runtime simulation. Our work is based on the assumption that the subspace can be efficiently and incrementally calculated, after a small number of changes each time. To achieve this goal, we made the following technical contributions.

- **Recalculation of linear modes.** Given the existing linear subspace of a deformable body, we show how its linear deformation modes can be quickly updated through incremental eigenvalue decomposition, when the body is modified by a single stitch, i.e., a zero-length spring.
- **Recalculation of modal derivatives.** Barbič and James [BJ05] introduced the use of modal derivatives to handle nonlinear and large deformations. By avoiding the expensive recalculation of Hessian stiffness tensor and reducing the number of tensor-vector-vector evaluations, we present a fast approach for generating modal derivatives of the stitched body.
- **Cubature-based subspace simulation.** We demonstrate the use of our new deformation modes in subspace simulation. Since mesh changes are small, reduced forces and matrices can still be evaluated by original cubature approximation.

Figure 1c illustrates that the new subspace, generated by our techniques in two seconds, can realistically handle large and nonlinear deformations in simulation. Our techniques are especially fast when the number of stitches is small. In case the deformable body undergoes too many changes, we can still use other approaches [BJ05, YLX*15] and build the subspace from scratch.

We can envision a variety of usages for our methods as we have illustrated through examples : Changing Material Stiffness such as the Oball Example (Figure 6) ; stitching applications in virtual surgery simulations such as the Stomach Example (Figure 11); Multidomain simulations such as the Plant Example (Figure 1, 10) and complex deformation scenarios such as the Dinosaur Example (Figure 4c, 9).

2. Other Related Work

Subspace simulation. The history of subspace simulation in engineering and mathematics can be traced back to [Lum67]. In computer graphics, Pentland and Williams [PW89] used linear modal analysis to build a subspace and developed the subspace simulation of a dynamical system for the first time. Compared with linear deformation, nonlinear deformation is much more difficult to handle in a subspace. A straightforward approach is to compute the deformation modes around several states of the physical system and use them to span the subspace. Specifically, this approach requires to solve several large eigenvalue problems, which are too expensive in practice. Researchers in engineering and graphics [IC85, BJ05, HSTP11, Tis11] have advocated the use of modal derivatives to enrich the subspace basis. The idea behind modal derivatives is to calculate the natural second-order system response to large deformation around the rest shape, so that they can approximate nonlinear deformation better when incorporated into the basis. Recently, von Tycowicz and collaborators [vTSSH13] proposed to widen the subspace by applying affine transformations to each linear mode. Their method is fast and easy to implement, but it creates a larger subspace that cannot be easily reduced afterwards. Subspace simulation of nonlinear deformation is relatively easier to handle, if the subspace is built from a set of deformed shape examples [KLM01]. Kim and James [KJ09] suggested to build the subspace on the fly, using previously simulated results as example data. In this work, we prefer not to recalculate the subspace in an example-based style, since we cannot afford running full-space simulation and we do not know user interaction ahead of time.

Subspace simulation can be several orders of magnitude faster

than full-space simulation. But if the mesh resolution is high, the evaluation of reduced forces and Jacobian matrices is still costly and can be the bottleneck of the system. Barbič and James [BJ05] found that the internal forces of a geometrically nonlinear material can be expressed as cubic polynomial functions of reduced space coordinates. So they proposed to precompute the polynomial coefficients and quickly evaluate reduced forces and matrices. Instead of performing the evaluation exactly, An and colleagues [AKJ08] used cubature approximation to estimate reduced forces and matrices. Their method effectively reduced the evaluation cost to $O(r^3)$, when the number of deformation modes is r and the number of cubature points is $O(r)$.

Other problems. When simulating a deformable body in the subspace, an interesting question is whether we can handle its self collisions in the subspace as well. Assuming that an object cannot be in self collision if it does not deform much, Barbič and James [BJ10] developed "certificates" to quickly avoid unnecessary collision tests using reduced subspace coordinates. Teng and colleagues [TOK14] proposed a pose-based cubature scheme, so they can detect self contacts of an articulated body, without explicitly checking the collision of two primitives. Recently, Sheth and colleagues [SLYF15] presented a skinning-based framework to ensure momentum conservation in subspace simulation, even in the presence of collisions and contacts.

Graphics researchers have also explored the use of subspace simulation in other fields. Hahn and colleagues [HTC*14] used a pose-varying subspace basis to simulate detailed clothes dressed on a human body. Xu and collaborators [XLCB15] developed an interactive tool to design heterogeneous material properties of a 3D model, based on subspace simulation techniques. Researchers [TLP06, WST09, KD13, ATW15] have also studied subspace fluid simulation.

3. Background

Given a base mesh with N vertices, we can formulate the equation describing its motion as,

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{f}^{\text{int}}(\mathbf{u}) = \mathbf{f}^{\text{ext}}, \tag{1}$$

in which $\mathbf{u} \in \mathbb{R}^{3N}$ is the stacked vertex displacement vector, $\mathbf{M} \in \mathbb{R}^{3N\times 3N}$ and $\mathbf{D} \in \mathbb{R}^{3N\times 3N}$ are the mass and damping matrices, and $\mathbf{f}^{\text{int}} \in \mathbb{R}^{3N}$ and $\mathbf{f}^{\text{ext}} \in \mathbb{R}^{3N}$ are the stacked internal and external force vectors. Using implicit time integration, we can use Equation 1 to update $\mathbf{u}$ by solving a $3N \times 3N$ sparse linear system. For a high-resolution mesh with a large N, the linear solve becomes the bottleneck of the whole simulator.

The basic idea behind subspace simulation is to constrain the displacement vector $\mathbf{u}$ into a subspace spanned by r deformation modes: $\{\phi_1, \phi_2, ..., \phi_r\}$. These modes can be assembled together into a $3N \times r$ matrix $\mathbf{U}$, known as the subspace basis. Since $\mathbf{u}$ is in the subspace defined by $\mathbf{U}$, we can define it as: $\mathbf{u} = \mathbf{U}\mathbf{q}$, in which $\mathbf{q} \in \mathbb{R}^r$ contains the reduced coordinates of $\mathbf{u}$ in the subspace. Assuming that $\mathbf{U}$ is mass-orthogonal: $\mathbf{U}^\mathsf{T}\mathbf{M}\mathbf{U} = \mathbf{I}$, we now obtain the governing equation in the subspace:

$$\ddot{\mathbf{q}} + \mathbf{U}^\mathsf{T}\mathbf{D}\mathbf{U}\dot{\mathbf{q}} + \mathbf{U}^\mathsf{T}\mathbf{f}^{\text{int}}(\mathbf{U}\mathbf{q}) = \mathbf{U}^\mathsf{T}\mathbf{f}^{\text{ext}}. \tag{2}$$

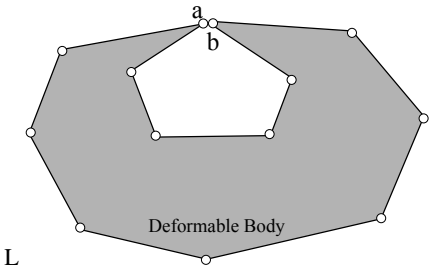


Figure 2: One stitch case. After a zero-length spring connects two co-located vertices a and b, our goal is to quickly calculate the subspace of the stitched body from the old one.

Implicit time integration of Equation 2 requires to solve a dense $r \times r$ linear system. Fortunately, if r is significantly smaller than $3N$, subspace simulation becomes orders of magnitude faster than full-space simulation.

4. One Stitch Case

Our research is focused on the reconstruction of the subspace basis, after simple editing operations. To begin with, let us consider stitching two co-located vertices together by a zero-length spring, as Figure 2 shows. In Subsection 4.1, we will present our incremental approach for computing the linear deformation modes of the stitched body. In Subsection 4.2, we will show how to recompute modal derivatives for nonlinear deformation. Finally in Subsection 4.3, we will discuss the use of the new basis in simulation.

4.1. Incremental Linear Modal Analysis

Let $\mathbf{K} = \partial\mathbf{f}^{\text{int}}/\partial\mathbf{u}$ be the $3N \times 3N$ stiffness matrix of the deformable body evaluated at $\mathbf{u} = \mathbf{0}$, where $\mathbf{f}^{\text{int}}$ is the internal force and $\mathbf{u}$ is the vertex displacement vector. Linear modal analysis solves a sparse, generalized eigenvalue decomposition problem:

$$\mathbf{K}\phi_i = \omega_i^2\mathbf{M}\phi_i, \tag{3}$$

where $\mathbf{M}$ is the invertible mass matrix and ϕ_i is the i-th vibration mode with frequency ω_i. Since high-frequency deformations are hardly noticeable in simulation, we ignore the modes with high frequencies and the six rigid modes with zero frequencies. The remaining modes can then be organized into a $3N \times r$ linear modal matrix for subspace simulation: $\mathbf{U} = \{\phi_1, \phi_2, ..., \phi_r\}$.

Now let us consider stitching two co-located vertices a and b together, as Figure 2 shows. We model this stitch as a zero-length spring with stiffness k, which applies a spring force on vertex a as: $\mathbf{f}_{ab} = k(\mathbf{u}_a - \mathbf{u}_b)$. The stiffness matrix resulted from this spring is a sparse block matrix:

$$\mathbf{K}_{ab} = \begin{bmatrix} \frac{\partial \mathbf{f}_{ab}}{\partial \mathbf{u}_a} & \frac{\partial \mathbf{f}_{ab}}{\partial \mathbf{u}_b} \\ \frac{\partial \mathbf{f}_{ba}}{\partial \mathbf{u}_a} & \frac{\partial \mathbf{f}_{ba}}{\partial \mathbf{u}_b} \end{bmatrix} = \begin{bmatrix} k\mathbf{I} & -k\mathbf{I} \\ -k\mathbf{I} & k\mathbf{I} \end{bmatrix}, \tag{4}$$

where the blocks are corresponding to vertex a and b respectively.

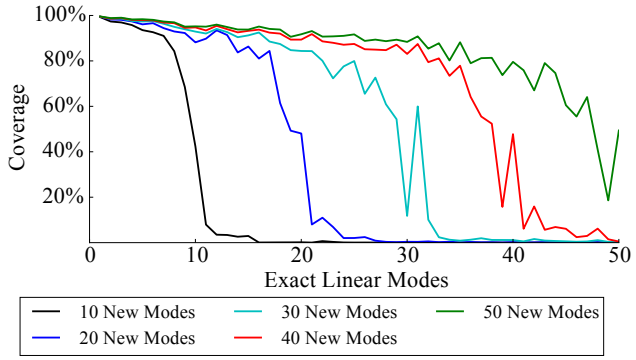


Figure 3: Subspace coverage. This plot illustrates how well each exact linear mode can be represented by the recalculated linear basis shown as each curve. The recalculated basis can represent the exact modes more closely, if it contains more modes.

It is straightforward to see that $\mathbf{K}_{ab}$ is a rank-3 matrix and it can be decomposed into: $\mathbf{K}_{ab} = \mathbf{A}\mathbf{A}^\mathsf{T}$, in which $\mathbf{A}$ is a $3N \times 3$ matrix:

$$\mathbf{A} = \begin{bmatrix} \quad & \sqrt{k}\mathbf{I} & \quad & -\sqrt{k}\mathbf{I} & \quad \end{bmatrix}^\mathsf{T}. \tag{5}$$

Here the two blocks are the a-th and the b-th blocks.

Performing linear modal analysis on the newly stitched body is equivalent to solving a new generalized eigenvalue problem: $\hat{\mathbf{K}}\hat{\Phi}_i = \hat{\omega}_i^2\mathbf{M}\hat{\Phi}_i$, in which $\hat{\mathbf{K}} = \mathbf{K} + \mathbf{K}_{ab}$ is the stiffness matrix of the new body. This generalized eigenvalue problem is computationally expensive to solve. Fortunately, we can approximate $\mathbf{K}$ by its low-frequency component and solve a low-rank eigenvalue problem afterwards. Let $\mathbf{U}$ and $\mathbf{U}_\mathrm{H}$ be low-frequency and high-frequency modes of $\mathbf{K}$, and Λ and Λ_H be their diagonal eigenvalue matrices. By definition, the joint modal matrix must be mass-diagonal:

$$\begin{bmatrix} \mathbf{U} & \mathbf{U}_\mathrm{H} \end{bmatrix}^\mathsf{T}\mathbf{M}\begin{bmatrix} \mathbf{U} & \mathbf{U}_\mathrm{H} \end{bmatrix} = \mathbf{I}. \tag{6}$$

So we can split $\mathbf{K}$ into two parts:

$$\begin{aligned} \mathbf{K} &= \mathbf{M}\begin{bmatrix} \mathbf{U} & \mathbf{U}_\mathrm{H} \end{bmatrix}\begin{bmatrix} \Lambda & \\ & \Lambda_\mathrm{H} \end{bmatrix}\begin{bmatrix} \mathbf{U} & \mathbf{U}_\mathrm{H} \end{bmatrix}^{-1} \\ &= \mathbf{M}\begin{bmatrix} \mathbf{U} & \mathbf{U}_\mathrm{H} \end{bmatrix}\begin{bmatrix} \Lambda & \\ & \Lambda_\mathrm{H} \end{bmatrix}\begin{bmatrix} \mathbf{U} & \mathbf{U}_\mathrm{H} \end{bmatrix}^\mathsf{T}\mathbf{M} \\ &= \mathbf{M}\mathbf{U}\Lambda\mathbf{U}^\mathsf{T}\mathbf{M} + \mathbf{M}\mathbf{U}_\mathrm{H}\Lambda_\mathrm{H}\mathbf{U}_\mathrm{H}^\mathsf{T}\mathbf{M}, \end{aligned} \tag{7}$$

in which $\mathbf{M}\mathbf{U}\Lambda\mathbf{U}^\mathsf{T}\mathbf{M}$ contains the low-frequency component and $\mathbf{M}\mathbf{U}_\mathrm{H}\Lambda_\mathrm{H}\mathbf{U}_\mathrm{H}^\mathsf{T}\mathbf{M}$ contains the high-frequency component. We then solve the generalized eigenvalue problem on $\mathbf{M}\mathbf{U}\Lambda\mathbf{U}^\mathsf{T}\mathbf{M} + \mathbf{K}_{ab}$.

Our idea is based on the assumption that the newly added spring is stiff and it cannot cause high-frequency deformations to become low-frequency. Therefore, the low-frequency subspace of the exact matrix can be well covered by the low-frequency subspace of the approximation matrix. To evaluate the plausibility of this idea, we apply generalized eigenvalue decomposition on $\mathbf{K} + \mathbf{K}_{ab}$, and then measure the distance from each exact low-frequency mode to its projection in the subspace spanned by our recalculated modes. Using this distance, we can calculate the percentage of the exact mode being covered by the recalculated basis. Figure 3 shows the first few exact modes can be well represented by the first few recalculated modes. After that, more exact modes can be represented by more recalculated modes, as expected.

4.1.1. Numerical Implementation

Since $\mathbf{M}\mathbf{U}\Lambda\mathbf{U}^\mathsf{T}\mathbf{M}$ is a rank-r matrix and $\mathbf{K}_{ab}$ is a rank-3 matrix, the rank of their sum $\bar{\mathbf{K}}$ cannot be higher than $r+3$. Inspired by the fast low-rank modification framework [Bra06], we propose to formulate the approximation of $\bar{\mathbf{K}}$ as:

$$\mathbf{L}\begin{bmatrix} \mathbf{L}^\mathsf{T}\mathbf{U} & \mathbf{L}^{-1}\mathbf{A} \end{bmatrix}\begin{bmatrix} \Lambda & \\ & \mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{L}^\mathsf{T}\mathbf{U} & \mathbf{L}^{-1}\mathbf{A} \end{bmatrix}^\mathsf{T}\mathbf{L}^\mathsf{T}, \tag{8}$$

where $\mathbf{M} = \mathbf{L}\mathbf{L}^\mathsf{T}$ is the Cholesky decomposition of $\mathbf{M}$. Here $\mathbf{L}^\mathsf{T}\mathbf{U}$ is an orthogonal basis, since $\mathbf{U}$ is mass-diagonal and $\mathbf{U}^\mathsf{T}\mathbf{L}\mathbf{L}^\mathsf{T}\mathbf{U} = \mathbf{I}$. The component of $\mathbf{L}^{-1}\mathbf{A}$ that is orthogonal to $\mathbf{L}^\mathsf{T}\mathbf{U}$ can be defined as: $(\mathbf{I} - \mathbf{L}^\mathsf{T}\mathbf{U}\mathbf{U}^\mathsf{T}\mathbf{L})\mathbf{L}^{-1}\mathbf{A}$. Let $\mathbf{P}$ be its orthogonal basis calculated using the Gram-Schmidt algorithm. We must have $\mathbf{L}^\mathsf{T}\mathbf{U}$ and $\mathbf{P}$ orthogonal to each other and they span the space of $\mathbf{L}^{-1}\mathbf{A}$. Therefore, we have:

$$\begin{bmatrix} \mathbf{L}^\mathsf{T}\mathbf{U} & \mathbf{L}^{-1}\mathbf{A} \end{bmatrix} = \begin{bmatrix} \mathbf{L}^\mathsf{T}\mathbf{U} & \mathbf{P} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{U}^\mathsf{T}\mathbf{A} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}, \tag{9}$$

where $\mathbf{R} = \mathbf{P}^\mathsf{T}(\mathbf{I} - \mathbf{L}^\mathsf{T}\mathbf{U}\mathbf{U}^\mathsf{T}\mathbf{L})\mathbf{L}^{-1}\mathbf{A}$. We can then reformulate the approximation into:

$$\bar{\mathbf{K}} = \mathbf{L}\begin{bmatrix} \mathbf{L}^\mathsf{T}\mathbf{U} & \mathbf{P} \end{bmatrix}\mathbf{C}\begin{bmatrix} \mathbf{L}^\mathsf{T}\mathbf{U} & \mathbf{P} \end{bmatrix}^\mathsf{T}\mathbf{L}^\mathsf{T}, \tag{10}$$

in which $\mathbf{C}$ is a $(r+3)\times(r+3)$ matrix:

$$\mathbf{C} = \begin{bmatrix} \Lambda & \\ & \end{bmatrix} + \begin{bmatrix} \mathbf{U}^\mathsf{T}\mathbf{A} \\ \mathbf{R} \end{bmatrix}\begin{bmatrix} \mathbf{U}^\mathsf{T}\mathbf{A} \\ \mathbf{R} \end{bmatrix}^\mathsf{T}. \tag{11}$$

Let $\mathbf{C} = \mathbf{u}\bar{\Lambda}\mathbf{u}^\mathsf{T}$ be the eigenvalue decomposition of $\mathbf{C}$, we have:

$$\bar{\mathbf{K}} = \mathbf{L}\begin{bmatrix} \mathbf{L}^\mathsf{T}\mathbf{U} & \mathbf{P} \end{bmatrix}\mathbf{u}\bar{\Lambda}\mathbf{u}^\mathsf{T}\begin{bmatrix} \mathbf{L}^\mathsf{T}\mathbf{U} & \mathbf{P} \end{bmatrix}^\mathsf{T}\mathbf{L}^\mathsf{T}. \tag{12}$$

The novel linear modal matrix can then be calculated as: $\bar{\mathbf{U}} = \mathbf{L}^{-\mathsf{T}}\begin{bmatrix} \mathbf{L}^\mathsf{T}\mathbf{U} & \mathbf{P} \end{bmatrix}\mathbf{u}$ and $\bar{\Lambda}$ contains its generalized eigenvalues. This is because $(\mathbf{L}^{-1}\bar{\mathbf{K}}\mathbf{L}^{-\mathsf{T}})(\mathbf{L}^\mathsf{T}\bar{\mathbf{U}}) = (\mathbf{L}^\mathsf{T}\bar{\mathbf{U}})\bar{\Lambda}$.

If $\mathbf{M}$ is a diagonal matrix, both $\mathbf{L}$ and $\mathbf{L}^{-1}$ are easy to calculate. However, if $\mathbf{M}$ is not a diagonal matrix, $\mathbf{L}^{-1}$ can become dense and we cannot calculate it directly. Fortunately, since $\mathbf{L}$ is a lower triangular matrix and $\mathbf{A}$ has three columns only, we can solve $\mathbf{L}^{-1}\mathbf{A}$ by three forward substitutions. Similarly, since the rank of $(\mathbf{I} - \mathbf{L}^\mathsf{T}\mathbf{U}\mathbf{U}^\mathsf{T}\mathbf{L})\mathbf{L}^{-1}\mathbf{A}$ cannot be greater than 3, its orthogonal basis $\mathbf{P}$ cannot have more than three columns. We can solve $\mathbf{L}^{-\mathsf{T}}\mathbf{P}$ by at most three backward substitutions. To reduce the computational cost, we precompute $\mathbf{L}$ as a sparse matrix, and we maintain both $\mathbf{U}$ and $\mathbf{L}^\mathsf{T}\mathbf{U}$ during runtime.

If the body is unconstrained, we found that it is not a good practice to ignore the six rigid modes, even though their generalized eigenvalues are zeros and they are not useful in subspace simulation. Instead, we keep them through the whole incremental linear modal analysis process, and remove them only when we use the basis for simulation or additional processes.

4.2. Basis Extension for Nonlinear Deformation

Large and nonlinear deformation cannot be handled by linear vibration modes. One solution to this problem is modal warping [CK05, HTZ*11], which tries to remove the artifacts caused by linear modes directly. Unfortunately, it cannot correctly handle nonlinear material behaviors and it can cause drifting artifacts in free falling cases. Alternatively, von Tycowicz and colleagues [vTSSH13] proposed to expand the basis, by decoupling each linear mode into nine new modes. Although their method can more accurately represent nonlinear deformation, it significantly increases the basis size to $9r+12$, which cannot be easily reduced for more efficient subspace simulation.

4.2.1. Modal Derivatives

To address the nonlinear deformation issue, we propose to use the modal derivative approach proposed by Barbič and James [BJ05]. Given two new linear modes $\bar{\phi}_i$ and $\bar{\phi}_j$ in the basis matrix $\bar{\mathbf{U}}$, we calculate their derivative $\bar{\phi}_{ij}$ by solving the linear system:

$$(\mathbf{K}+\mathbf{K}_{ab})\bar{\phi}_{ij} = -(\bar{\mathbf{H}} : \bar{\phi}_i)\bar{\phi}_j, \tag{13}$$

in which $\bar{\mathbf{H}} = \partial\bar{\mathbf{K}}/\partial\mathbf{u}$ at $\mathbf{u}=0$, an order-3 tensor representing the first-order derivative of $\bar{\mathbf{K}}$, also known as the *Hessian stiffness tensor*. According to Equation 4, $\mathbf{K}_{ab}$ is constant and its Hessian stiffness tensor is trivial. So we have $\bar{\mathbf{H}} = \mathbf{H}$, which can be precomputed using the base mesh.

Let r be the total number of non-rigid linear modes. Their combinations can provide us $r(r+1)/2$ modal derivatives, each of which requires to solve the linear system in Equation 13 once. Although this seems to be computationally expensive, Barbič and James [BJ05] pointed out that $\mathbf{K}+\mathbf{K}_{ab}$ stays the same for all of the systems, so it can be pre-factorized for fast direct solve. According to our experiment, the real bottleneck is the evaluation of the tensor-vector-vector product on the right-hand side of Equation 13.

So instead of using all of the r modes to calculate modal derivatives, we choose to use the first six non-rigid linear modes only. Since the resulting 21 modal derivatives are typically insufficient to cover enough nonlinear deformations, we reuse 40 percent of the right-hand sides evaluated from the base mesh to generate additional modal derivatives, assuming that the nonlinear relationships among the modes after stitching is similar to those before stitching. The plant example in Figure 1 shows that this hybrid strategy still allows the resulting basis to capture large and nonlinear deformation. Meanwhile, it reduces the computational time spent on modal derivatives from 6.82s to 0.89s, thanks to fewer Tensor-vector-vector products. Note that our strategy can also introduce some high-frequency vibrations back into the basis, which were lost due to low-rank approximation discussed in Subsection 4.1.

4.2.2. Basis reduction

So far we have collected many linear modes and their modal derivatives. For faster subspace simulation, Barbič and James [BJ05] recommended the use of principal component analysis (PCA) to reduce the subspace basis again. We adopt this idea and we use the randomized PCA method proposed by Halko and colleagues [HMT11]. Previous research showed that randomized PCA can run orders-of-magnitude faster than standard PCA, yet it has similar accuracy and robustness. Before performing randomized PCA, we scale each linear mode $\bar{\phi}_i$ by $\bar{\omega}_i/\bar{\omega}_1$ and each modal derivative $\bar{\phi}_{ij}$ by $\bar{\omega}_i\bar{\omega}_j/\bar{\omega}_1^2$. Here $\bar{\omega}_i$ is the frequency of $\bar{\phi}_i$ given by $\bar{\Lambda}$ in Equation 12, and $\bar{\omega}_1$ is the frequency of the first mode, i.e., the lowest frequency. These scalings are used to prevent low-frequency modes from being dominated by high-frequency modes, as suggested in [BJ05].

4.3. Cubature Approximation

Given the new basis $\bar{\mathbf{U}}$, we are now ready to formulate the dynamical system for subspace simulation described in Equation 2. Suppose that the evaluation of the reduced internal force was accelerated by using cubature samples. An important question is: *how can we still quickly evaluate the reduced internal force, when the subspace basis gets changed?*

Let $\mathbf{f}^{\text{int}}$ be the internal force of the original mesh and $\mathbf{f}_{ab}$ be the new stitching force. The total reduced force is: $\bar{\mathbf{U}}^{\mathsf{T}}\mathbf{f}^{\text{int}}(\bar{\mathbf{U}}\bar{\mathbf{q}}) + \bar{\mathbf{U}}^{\mathsf{T}}\mathbf{f}_{ab}(\bar{\mathbf{U}}\bar{\mathbf{q}})$, where $\bar{\mathbf{q}}$ contains the reduced coordinates of the current shape in the new subspace. We assume that the cubature samples trained for the original mesh can still be used to evaluate the reduced internal force in the new subspace. So we have:

$$\bar{\mathbf{U}}^{\mathsf{T}}\mathbf{f}^{\text{int}}(\bar{\mathbf{U}}\bar{\mathbf{q}}) \approx \sum_k w_k \bar{\mathbf{U}}^{\mathsf{T}}\mathbf{f}_k^{\text{int}}(\bar{\mathbf{U}}\bar{\mathbf{q}}), \tag{14}$$

where k and w_k are the cubature sample and weight trained for the original mesh. To evaluate $\bar{\mathbf{U}}^{\mathsf{T}}\mathbf{f}_{ab}$, we simply calculate $\mathbf{f}_{ab}$ as a sparse vector and perform the matrix-vector product. In total, the cost of reduced force evaluation is $O(r^2)$, where r is the number of the deformation modes. We calculate the reduced stiffness matrix in the subspace using a similar approach.

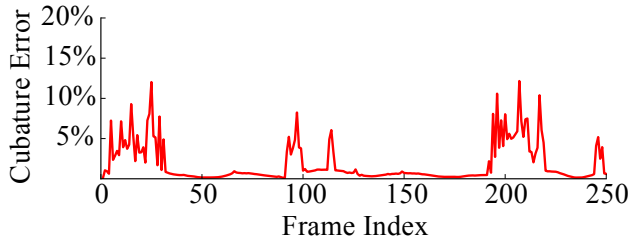
Figure 4a illustrates the approximation error produced by our cubature scheme in each frame, using the dinosaur example. These errors are typically small, except when user interaction happens. To further understand how the errors can be accumulated to affect the simulation, we run the same simulation twice: once without cubature approximation and once with cubature approximation. Although the two simulation results are not identical, they are visually similar as shown in Figure 4b and 4c. So we think our cubature scheme is still effective.

5. Complex Editing Cases

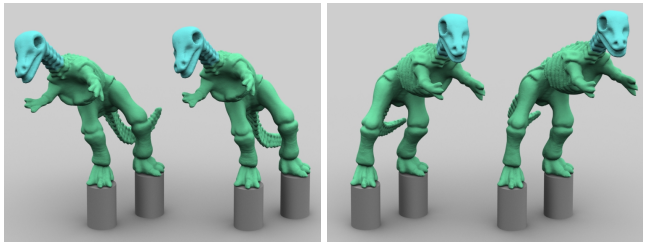
In this section, we will discuss how our techniques can be adopted to handle more complex editing operations.

5.1. Multiple Stitches and Domains

It is straightforward to extend the method described in Section 4 for multiple stitches, i.e., zero-length springs. Every spring adds three new rows and columns to the matrix $\mathbf{C}$ in Equation 11, so its size becomes $(r+3s)\times(r+3s)$. Here s is the number of stitches. Our method can efficiently handle a small number of stitches, as Figure 5 shows. When s increases, our method needs more computational cost and it becomes less attractive. In that case, we may



(a) Per-frame cubature approximation error



(b) Frame 112 (c) Frame 210

Figure 4: Quantitative and qualitative results without and with cubature optimization. In (b) and (c), the results without cubature approximation are shown on the left side, and the results with cubature approximation are shown on the right side.

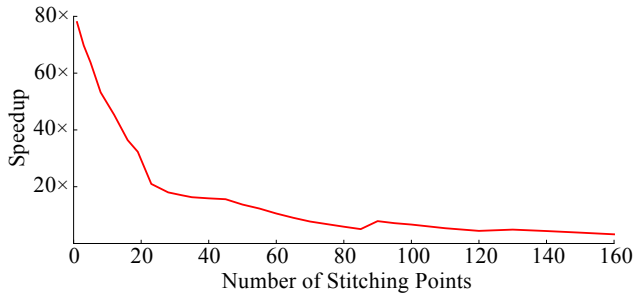
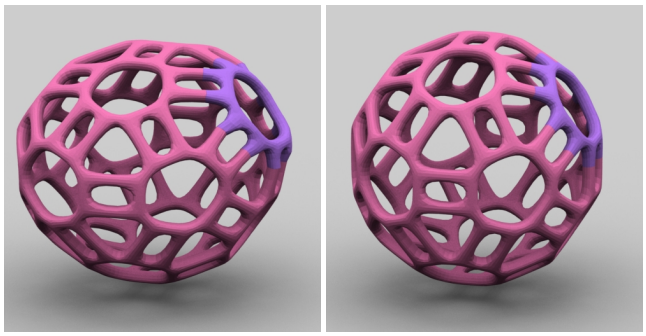


Figure 5: The relationship between the number of stitches and the speedup gained by using our methods. In general, using more stitches makes our methods less effective.

need to recalculate the subspace basis from scratch again, using the expedited approach [YLX*15] for example.

If we consider the union of separate meshes as a single original mesh, we can use the method to reconstruct the subspace of a mesh with multiple domains as well. In that case, we can simply build the subspace of each domain separately. Let ϕ_i be a linear mode of domain k, we expand it into the linear mode of the joint original mesh as: $[\mathbf{0}^T,...,\mathbf{0}^T,\phi_i^T,\mathbf{0}^T,...,\mathbf{0}^T]^T$, where zeros represent no deformations in other domains. By expanding all of these linear modes, we obtain a set of linear modes for the joint original mesh. We then add stitches to connect the separate parts and apply our method to build the subspace of the newly stitched mesh as a whole.

Incorporating domain decomposition into our subspace reconstruction process has an obvious advantage: we can modify each domain separately, without affect the subspaces of other domains.



(a) Compliant base (b) Stiff base

Figure 6: The Oball example. Our system allows users to modify the stiffness of different domains, with a low subspace recalculation cost. In (a), the pink domain and the purple domain share the same stiffness, while in (b), the pink domain becomes 100 times stiffer.

For instance, we can increase or decrease the material stiffness of one domain. If the stiffness of the whole domain is scaled by a constant factor K, its linear modes remain unchanged and the frequencies are scaled by $\sqrt{K}$, according to Equation 3. Figure 6 shows the simulation results of an Oball example. When the pink domain of the ball becomes stiffer, our method can efficiently recalculate the subspace within 4.5s for simulation afterwards.

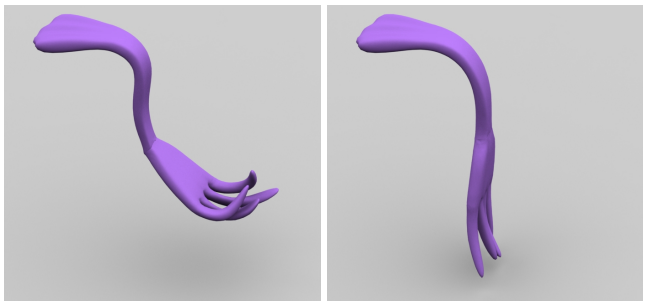
5.2. Unconstrained Bodies and Domains

When a deformable body is unconstrained, it has six rigid linear vibration modes with zero frequencies. Similar to [BZ11], we do not incorporate rigid modes into subspace simulation, to prevent other modes from becoming time-dependent. Instead, we separate rigid motion from subspace simulation and animate it by rigid body dynamics. Since subspace deformation is now in a non-inertial local frame, we must add inertial forces to account for Coriolis, inertial, Euler and centrifugal effects. We use fast sandwich transform [KJ11] to calculate these forces in the subspace.

When we stitch an unconstrained domain to constrained ones, such as the fork example in Figure 7a, we must consider its six rigid modes to avoid rotational artifacts. Specifically, we add the rigid modes into the subspace basis and then calculate the subspace for the whole stitched body, as discussed previously in Subsection 5.1. Doing this allows the fork head to be properly rotated, as shown in Figure 7b. Unfortunately, there is still one remaining problem: the linear modes of the unconstrained domain, i.e. the fork head, does not get rotated to reflect the deformation of the constrained domain. One possible solution is to couple the linear modes of different domains together through interface alignment, rather than padding them with zeros. However, we may need more computational cost to reduce the increased subspace size accordingly.

6. Results and Discussions

(Please watch the supplementary video for animation examples.) We implement our methods and tested our examples on an Intel



(a) Without six rigid modes (b) With six rigid modes

Figure 7: The fork example. Without considering the six rigid modes of the fork head, the simulator cannot properly handle large rotational motions as shown in (a).

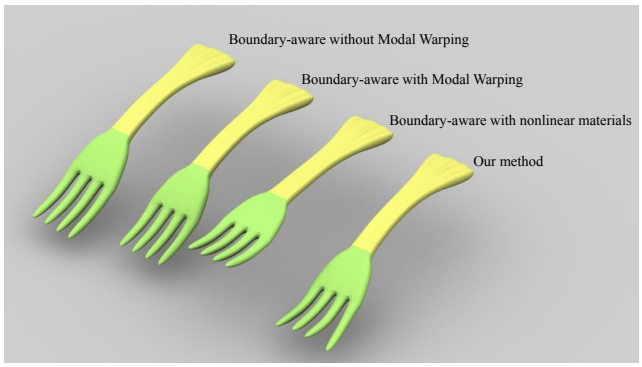
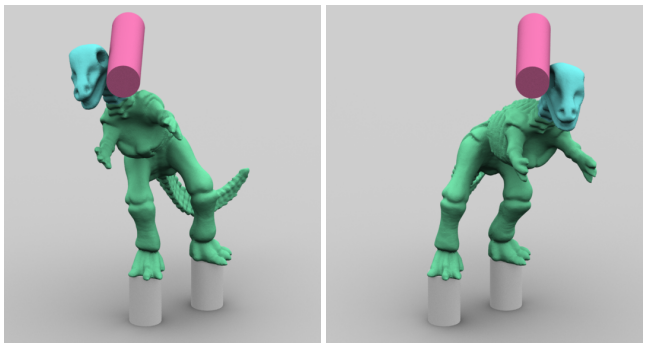


Figure 8: Comparison with boundary-aware subspace method.



(a) (b)

Figure 9: An extension of the Dinosaur example which shows how our system can handle deformations scenarios such as those involving collisions.

Core i7-3770S 3.10GHz processor. We use the Intel MKL library and the OpenMP API for parallelization. In addition, we apply Eigen and Armadillo linear algebra libraries for linear solves. We perform generalized eigenvalue decomposition on large sparse matrices by ARPACK.

Most of our examples have a Youngs's Modulus of 5.0×10^5 and the connection spring stiffness of 1.0×10^5. The value is same for all the springs in the same example. Our experiments show that as long as the connecting spring stiffness is within a reasonable order of magnitude of the connection springs, no additional parameter tuning is needed. However if springs of extremely large magnitude (say 10^{12} or above) is used for relatively less stiff materials, locking issues may occur. All of our examples set the time step to 0.01s. We render one frame every three time steps. We typically use 60 to 225 cubature samples for evaluating reduced forces and matrices. We cap the number of total modes in the subpace basis to 90, to ensure the simulation performance. The frame rates of our examples vary from 8 to 30 FPS, depending on the number of modes and the number of cubature samples. Table 1 summarizes the cost of each subspace recalculation step. It also provides the speedup compared with the standard approach that recalculates the subspace from scratch. Note that the speedup we gain in the PCA reduction step is mostly due to the use of randomized PCA, not our new methods.

We evaluate the "goodness" of our basis as compared to a regular subspace simulation using two methods. The first one quantitatively compares how well our linear modes approximates the actual linear modes as shows in Figure 3 . The second one is a qualitative comparison of the simulation generated by our method and a similar mesh undergoing subspace simulation as per [BJ05]. (See Video) We note that in this example our method has a visual performance similair to original subspace at a fraction of the computation cost. We shows two rounds of incremental stitching in this example - one which partially stitches the mesh and the other which completely stitches the mesh. Even though there is some loss of the high frequency modes which result in our method not able to replicate some of the more extreme deformations, the results are visually plausible.

Comparison with [YXG*13] : We also directly compared our method with the boundary-aware multi-domain subspace method proposed by Yang and colleagues. The fork example shown in Fig. 8 is used for the comparison. Without modal warping, boundary-aware subspace method suffers from excessive elongation artifacts as demonstrated by the fork in the extreme left. The artifact is corrected using modal warping as shown in the second (from left) fork. Our method of basis generation (both linear and non-linear) is approximately twice as fast compared to that of boundary-aware subspace method, which only computes linear basis. From the figure, we can see that the fork with boundary-aware subspace can bend slightly more than ours because it is modeled with linear elasticity. However the advantage of our method is its ability to support nonlinear materials. Compared to boundary-aware multi-domain methods where non-linear material completely fail to move the fork (as shown in the third fork) , our method can gracefully handle any arbitrary materials.

Each of our examples demonstrate one possible use of our method. The ball example demonstrates how our method can be used for controlling stiffness of different materials. The stomach example shows us the potential of this method in virtual surgery applications. The plant example (both using two domains as well

Model	Model Statistics #Vert, #Tet, #Spr	Time Original	Time Ours	Speedup
Stomach	37K, 118K, 30	16.89s	1.10s	**15.3**×
Plant	13K, 43K, 30	4.35s	0.42s	**10.3**×
Fork	13K, 55K, 40	6.03s	0.67s	**9.0**×
Dinosaur	57K, 192K, 38	14.12s	2.65s	**5.3**×
Oball	16K, 60K, 160	7.72s	2.30s	**3.4**×

(a) The cost of recalculating linear Modes.

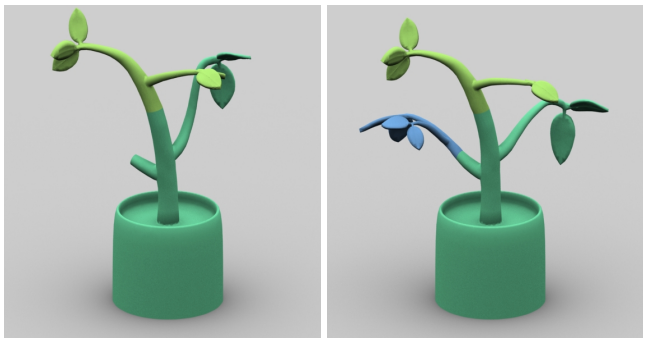
Model	Original #Deri	Original Time	Ours #Deri	Ours Time	Speedup
Stomach	201	26.08s	201	5.14s	**5.0**×
Plant	91	6.82s	91	0.89s	**7.6**×
Fork	143	6.50s	143	1.20s	**5.4**×
Dinosaur	91	34.09s	91	4.21s	**8.1**×
Oball	143	6.54s	143	1.25s	**5.2**×

(b) The cost of recalculating modal derivatives.

Model	Standard SVD #in	Standard SVD Time	Randomized SVD #in	Randomized SVD Time	Speedup
Stomach	231	15.76s	231	2.29s	**6.9**×
Plant	121	3.00s	121	0.80s	**3.8**×
Fork	173	3.75s	173	0.91s	**4.1**×
Dinosaur	121	13.74s	121	3.04s	**4.5**×
Oball	173	4.22s	173	0.98s	**4.3**×

(c) The cost of final PCA reduction.

Table 1: Model and timing statistics.



(a) Two Domains (b) Three domains

Figure 10: Multiple branches. Our system incrementally recalculates the deformation subspace of the whole plant model, when the branches are stitched to the model in multiple steps.

as three domains) shows its usefulness for domain decomposition methods. Finally the Fork and the Dinosaur example shows the application of our method in terms of real time basis recalculation speed and the ability of the method to handle complex collision scenarios.

Incremental editing. Since our system incrementally recalculates the deformation subspace, it is naturally suitable for handling multiple stitching steps. For example, when we stitch the second branch to the plant model as shown in Figure 10b, we can recalculate the subspace using the new basis calculated after stitching the first branch. Similarly, when sewing the cut on a stomach model as shown in Figure 11, we can incrementally recalculate the subspace twice: once to form a partially stitched model and once to form a fully stitched model. Incremental subspace reconstruction can effectively reduce the computational cost, when user needs to add multiple stitches over time. That being said, recalculating the subspace too many times can cause large error accumulation. In the future, we would like to know how to quickly evaluate the subspace quality and when to completely recalculate the subspace instead.

6.1. Limitations.

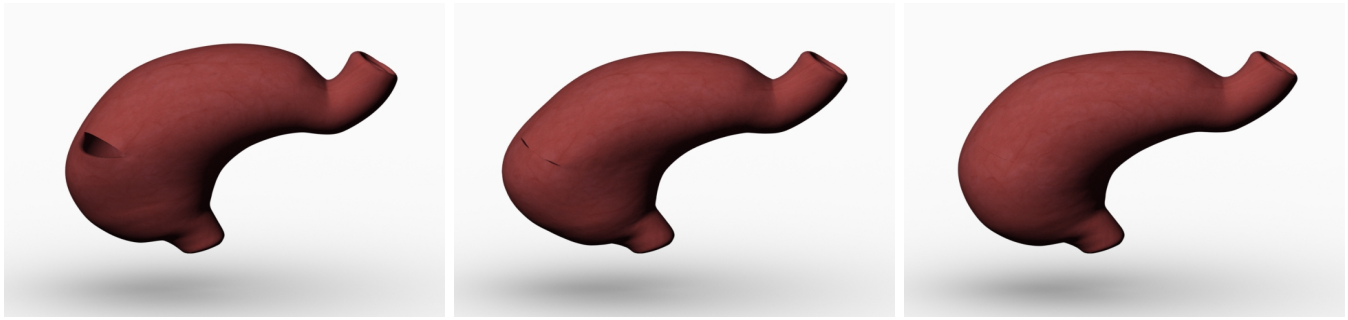
Our method provides a fast way to construct nonlinear deformation modes for subspace simulation. Although the method is based on [BJ05], the result is not meant to be the same as the modes constructed directly from the new mesh using their technique. If the deformation modes are incrementally updated when the mesh changes constantly, the error can be accumulated and the result can become unpredictable. The performance of our method relies on the assumption that the mesh is slightly changed each time. If the mesh is too significantly changed, our method needs more computational time (see Figure 5 for the relationship between the number of stitches and the corresponding speedup) and it may be even slower than just calculating the subspace from scratch. When adding new stitches, we assume that vertices are co-located and the springs have zero lengths. In other words, we cannot stitch two arbitrary vertices without aligning them ahead of time. Finally, when stitching unconstrained domains to constrained ones, we cannot allow constrained domains to have large deformations, or we cannot deform unconstrained domains correctly.

Our subspace simulation reuses the cubature samples calculated for the original subspace. If the mesh is changed significantly, cubature approximation will be poor and we must recalculate the samples as well, which is computationally expensive. Another point to note is that the success of the cubature reuse is greatly dependent on the accuracy of the original cubature training. In general we should ensure that the cubature error during training is as low as possible. This can be ensured by adjusting the parameters during cubature optimization. While such a setting may increase the overall training time, we must note that it is a one-time cost and such a well-trained cubature (with low error) will allow our algorithm to reuse the cubatures effectively even after many small incremental changes.

7. Conclusion and Future Work

In this paper, we demonstrate how to perform linear modal analysis in an incremental fashion, based on the assumption that the stiffness matrix can be approximated by its low-rank low-frequency part. Compared with linear motions, nonlinear and locally rigid motions are more complex to handle. So it is difficult or computationally expensive to cover them well in the recalculated subspace.

We are actively looking for better ways to handle nonlinear and locally rigid motions by our recalculated subspace. We are also interested in studying fast ways to evaluate the quality of our sub-



(a) Before stitching (b) Partially stitched (c) Fully stitched

Figure 11: The stomach example. A cut on the original mesh can be partially or fully stitched in simulation, depending on the constraints.

space, so we can know when it should be replaced by the subspace calculated from scratch. How to stitch two arbitrary vertices without an expensive initial alignment step is another problem we plan to study. Finally, we would like to investigate the development of our techniques on the GPU for even faster performance.

Acknowledgment

The authors would like to thank Nvidia for their funding and equipment support.

References

[AKJ08] AN S. S., KIM T., JAMES D. L.: Optimizing cubature for efficient integration of subspace deformations. *ACM Trans. Graph. (SIGGRAPH Asia) 27*, 5 (Dec. 2008), 165:1–165:10. 3

[ATW15] ANDO R., THÜREY N., WOJTAN C.: A dimension-reduced pressure solver for liquid simulations. *Computer Graphics Forum (Eurographics)* (2015). 3

[BJ05] BARBIČ J., JAMES D. L.: Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Trans. Graph. (SIGGRAPH) 24*, 3 (July 2005), 982–990. 2, 3, 5, 7, 8

[BJ10] BARBIČ J., JAMES D. L.: Subspace self-collision culling. *ACM Trans. Graph. (SIGGRAPH) 29*, 4 (July 2010), 81:1–81:9. 3

[Bra06] BRAND M.: Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and Its Applications 415*, 1 (2006), 20–30. 4

[BZ11] BARBIČ J., ZHAO Y.: Real-time large-deformation substructuring. *ACM Trans. Graph. (SIGGRAPH) 30*, 4 (July 2011), 91:1–91:8. 2, 6

[CK05] CHOI M. G., KO H.-S.: Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Trans. Vis. Comp. Graph. 11*, 1 (Jan. 2005), 91–101. 5

[HLB*06] HUANG J., LIU X., BAO H., GUO B., SHUM H.-Y.: An efficient large deformation method using domain decomposition. *Comput. Graph. 30*, 6 (Dec. 2006), 927–935. 2

[HMT11] HALKO N., MARTINSSON P. G., TROPP J. A.: Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev. 53*, 2 (May 2011), 217–288. 5

[HSTP11] HILDEBRANDT K., SCHULZ C., TYCOWICZ C. V., POLTHIER K.: Interactive surface modeling using modal analysis. *ACM Trans. Graph. (SIGGRAPH) 30*, 5 (Oct. 2011), 119:1–119:11. 2

[HTC*14] HAHN F., THOMASZEWSKI B., COROS S., SUMNER R. W., COLE F., MEYER M., DEROSE T., GROSS M.: Subspace clothing simulation using adaptive bases. *ACM Trans. Graph. (SIGGRAPH) 33*, 4 (July 2014), 105:1–105:9. 3

[HTZ*11] HUANG J., TONG Y., ZHOU K., BAO H., DESBRUN M.: Interactive shape interpolation through controllable dynamic deformation. *IEEE Trans. Vis. Comp. Graph. 17*, 7 (2011), 983–992. 5

[HZ13] HARMON D., ZORIN D.: Subspace integration with local deformations. *ACM Trans. Graph. 32*, 4 (July 2013), 107:1–107:10. URL: http://doi.acm.org/10.1145/2461912.2461922, doi:10.1145/2461912.2461922. 2

[IC85] IDELSOHN S. R., CARDONA A.: A reduction method for nonlinear structural dynamic analysis. *Computer Methods in Applied Mechanics and Engineering 49*, 3 (1985), 253 – 279. 2

[KD13] KIM T., DELANEY J.: Subspace fluid re-simulation. *ACM Trans. Graph. 32*, 4 (July 2013), 62:1–62:9. 3

[KJ09] KIM T., JAMES D. L.: Skipping steps in deformable simulation with online model reduction. *ACM Trans. Graph. (SIGGRAPH Asia) 28*, 5 (Dec. 2009), 123:1–123:9. 2

[KJ11] KIM T., JAMES D. L.: Physics-based character skinning using multi-domain subspace deformations. In *Proceedings of SCA* (2011), pp. 63–72. 2, 6

[KLM01] KRYSL P., LALL S., MARSDEN J. E.: Dimensional model reduction in non-linear finite element dynamics of solids and structures. *International Journal for Numerical Methods in Engineering 51*, 4 (2001), 479–504. 2

[Lum67] LUMLEY J. L.: The structure of inhomogeneous turbulent flows. In *Atmospheric turbulence and radio propagation* (Moscow, 1967), Yaglom A. M., Tatarski V. I., (Eds.), Nauka, pp. 166–178. 2

[PW89] PENTLAND A., WILLIAMS J.: Good vibrations: Modal dynamics for graphics and animation. *SIGGRAPH Comput. Graph. 23*, 3 (July 1989), 207–214. 2

[SLYF15] SHETH R., LU W., YU Y., FEDKIW R.: Fully momentum-conserving reduced deformable bodies with collision, contact, articulation, and skinning. In *Proceedings of SCA* (2015), pp. 45–54. 3

[Tis11] TISO P.: Optimal second order reduction basis selection for nonlinear transient analysis. In *Modal Analysis Topics, Volume 3* (2011), pp. 27–39. 2

[TLP06] TREUILLE A., LEWIS A., POPOVIĆ Z.: Model reduction for real-time fluids. *ACM Trans. Graph. (SIGGRAPH) 25*, 3 (July 2006), 826–834. 3

[TOK14] TENG Y., OTADUY M. A., KIM T.: Simulating articulated subspace self-contact. *ACM Trans. Graph. (SIGGRAPH) 33*, 4 (July 2014), 106:1–106:9. 3

[vTSSH13] VON TYCOWICZ C., SCHULZ C., SEIDEL H.-P., HILDEBRANDT K.: An efficient construction of reduced deformable objects. *ACM Trans. Graph. (SIGGRAPH) 32*, 6 (Nov. 2013), 213:1–213:10. 2, 5

[WMW15] WU X., MUKHERJEE R., WANG H.: A unified approach for subspace simulation of deformable bodies in multiple domains. *ACM Trans. Graph. (SIGGRAPH Asia) 34*, 6 (Oct. 2015), 241:1–241:9. 2

[WST09] WICKE M., STANTON M., TREUILLE A.: Modular bases for fluid dynamics. *ACM Trans. Graph. (SIGGRAPH) 28*, 3 (July 2009), 39:1–39:8. 3

[XLCB15] XU H., LI Y., CHEN Y., BARBIVČ J.: Interactive material design using model reduction. *ACM Trans. Graph. (SIGGRAPH) 34*, 2 (Mar. 2015), 18:1–18:14. 3

[YLX*15] YANG Y., LI D., XU W., TIAN Y., ZHENG C.: Expediting precomputation for reduced deformable simulation. *ACM Trans. Graph. (SIGGRAPH Asia) 34*, 6 (Oct. 2015), 243:1–243:13. 2, 6

[YXG*13] YANG Y., XU W., GUO X., ZHOU K., GUO B.: Boundary-aware multidomain subspace deformation. *IEEE Tran. Vis. Comp. Graph. 19*, 10 (Oct 2013), 1633–1645. 2, 7

[ZB13] ZHAO Y., BARBIČ J.: Interactive authoring of simulation-ready plants. *ACM Trans. Graph. 32*, 4 (July 2013), 84:1–84:12. URL: http://doi.acm.org/10.1145/2461912.2461961, doi:10.1145/2461912.2461961. 2