15th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering
and
3rd Conference on Imaging and Visualization
CMBBE 2018
P. R. Fernandes and J. M. Tavares (Editors)

New Techniques for Combined FEM-multibody Anatomical Simulation

John E. Lloyd*, Antonio Sánchez*, Erik Widing†, Ian Stavness†, Sidney Fels*

*Electrical and Computer Engineering, University of British Columbia
†Computer Science, University of Saskatchewan

Keywords: finite element methods, multibody dynamics, anatomical simulation, hybrid multibody/FEM modeling

Abstract: *This article describes new techniques being introduced to the ArtiSynth mechanical modeling system to improve its utility for combined FEM-multibody simulation. These include: reduced coordinate modeling, in which an FEM model is made more computationally efficient by reducing it to a low DOF subspace; new methods for connecting points and coordinate frames directly to deformable bodies; and the ability to create skin and embedded meshes that are connected to underlying FEM models and other dynamic components. The unifying concept behind all these techniques is the principal of virtual work.*

1. INTRODUCTION

Effective simulation of human anatomical structure and function can benefit from combining low-fidelity models with fast computation times and high-fidelity models that emulate detailed tissue dynamics but have slower computation times. Multibody methods are typically used for the former, modeling structures such as bones, joints and point-to-point muscles, while finite element methods (FEM) are typically used for the latter, modeling deformable tissues and capturing internal stress/strain dynamics. Combining the two can enable the creation of models with efficient, and possibly interactive, simulation times while also providing appropriate fidelity in an area of interest.

In this article, we describe new techniques that are being introduced into ArtiSynth [1] (www.artisynth.org), an open source simulation platform that permits researchers to combine multibody and FEM techniques and hence leverage the advantages of both. These new techniques include: reduced coordinate modeling, attaching points and frames to deformable bodies, and skinning and embedded meshes.

2. REDUCED COORDINATE MODELING

This is a technique in which a deformable body is modeled using a restricted deformation basis instead of a collection of deformable finite elements [2]. It spans the gap between FEM methods and rigid bodies (which are themselves reduced models condensed to purely rigid motions), and can be very effective in speeding up simulation times for models in which the range of typical deformations is constrained (such as tongue motions in speech production).

To create a reduced coordinate model, it is often convenient to begin with a standard FEM model. One can then construct a *basis* $\mathbf{U}$ of nodal deformations (with respect to the nodal rest positions) which spans the set of all possible deformations for the reduced model. Assume the FEM modal has n nodes, and let $\mathbf{x}$, $\mathbf{x}_0$ and $\mathbf{u}$ denote composite vectors of their positions, rest positions, and displacements, such that $\mathbf{x} = \mathbf{x}_0 + \mathbf{u}$. Then if $\mathbf{q}$ is a vector of the r reduced coordinates, we have

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{U}\mathbf{q}, \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{3n \times r}$. The basis $\mathbf{U}$ does not have to be constant but often is and will be assumed to be for the remainder of this article. Determining an appropriate basis is one of the principal challenges in constructing a reduced model. Automatic techniques include linear modal analysis [3] (when the deformation is small), along with various ways to extend a modal basis with additional vectors to handle large deformations, such as using modal derivatives [4] or applying additional linear transformations to the basis vectors [5]. In practice, better results are often obtained by creating the basis via a training method in which a non-reduced FEM model is used to recreate the deformations that are required for the modeling application [2].

2.1 Reduced dynamics

Background to some of the concepts in this section can be found in [6]. An FEM model advances in time according to the dynamics

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} + \mathbf{K}\delta\mathbf{x} = \mathbf{f}_{\text{int}}(\mathbf{x}) + \mathbf{f}_{\text{ext}}, \tag{2}$$

where $\mathbf{M}$ and $\mathbf{D}$ are mass and damping matrices, $\mathbf{K}$ is the local stiffness matrix, $\delta\mathbf{x}$ is the local change in $\mathbf{x}$, and $\mathbf{f}_{\text{int}}$ and $\mathbf{f}_{\text{ext}}$ are the internal and external forces. Note that the matrices in (2) are almost always sparse.

ArtiSynth uses (2), in conjunction with one of its implicit integrators, to solve for the motion of the FEM model. To handle reduced models, it is necessary to find the equivalent reduced dynamics,

$$\tilde{\mathbf{M}}\ddot{\mathbf{q}} + \tilde{\mathbf{D}}\dot{\mathbf{q}} + \tilde{\mathbf{K}}\delta\mathbf{q} = \tilde{\mathbf{f}}_{\text{int}}(\mathbf{q}) + \tilde{\mathbf{f}}_{\text{ext}}, \tag{3}$$

where $\tilde{\mathbf{M}}$, $\tilde{\mathbf{D}}$, and $\tilde{\mathbf{K}}$ are the reduced mass, damping and stiffness matrices, and $\tilde{\mathbf{f}}_{\text{int}}$ and $\tilde{\mathbf{f}}_{\text{ext}}$ are the reduced internal and external forces. Note that all of the matrices in (3) are dense.

Model reduction implies a linear relationship between the nodal velocities $\dot{\mathbf{x}}$ of the original FEM model and the velocities $\dot{\mathbf{q}}$ of the reduced model:

$$\dot{\mathbf{x}} = \mathbf{U}\dot{\mathbf{q}}.$$

(Note that even if $\mathbf{U}$ were not constant, this will still be true locally). Then from the principle of virtual work, we know that the work done in nodal coordinates $\mathbf{f}^T\dot{\mathbf{x}}$ must equal the work done in reduced coordinates $\tilde{\mathbf{f}}^T\dot{\mathbf{q}}$, and therefore

$$\tilde{\mathbf{f}} = \mathbf{U}^T\mathbf{f}.$$

This allows us to determine the reduced quantities in (3):

$$\tilde{\mathbf{M}} = \mathbf{U}^T\mathbf{M}\mathbf{U}, \quad \tilde{\mathbf{D}} = \mathbf{U}^T\mathbf{D}\mathbf{U}, \quad \tilde{\mathbf{K}} = \mathbf{U}^T\mathbf{K}\mathbf{U}, \quad \tilde{\mathbf{f}}_{\text{int}} = \mathbf{U}^T\mathbf{f}_{\text{int}}, \quad \tilde{\mathbf{f}}_{\text{ext}} = \mathbf{U}^T\mathbf{f}_{\text{ext}}. \tag{4}$$

ArtiSynth normally employs a lumped mass model in which $\mathbf{M}$ is constant, and so $\tilde{\mathbf{M}}$ is also constant and can be precomputed. The damping matrix $\mathbf{D}$ is also often constant and so $\tilde{\mathbf{D}}$ is also typically easy to determine. However, in any model involving large deformations, $\mathbf{K}$ is almost always non-constant, and is re-evaluated at each simulation time step by integrating the stress/strain relationships of the model's constitutive materials over a set of integration points within each FEM model element [7]. This process, sometimes known as *matrix assembly*, has $O(n)$ complexity, with the constant depending on the nodal connectivity.

If $\tilde{\mathbf{K}}$ is evaluated using (4) directly, the resulting complexity will be $O(r^2 n)$ (since $\mathbf{K}$ is sparse with $O(n)$ entries). For larger r, however, this can be burdensome. A more efficient approach is to use a smaller number of integration points, generally $O(r)$. For example, one can select $O(r)$ elements, use a single integration point in the middle of each, and then rather than forming $\mathbf{K}$, instead accumulate the local stiffness matrix $\mathbf{K}_j$ associated with each integration point directly into $\tilde{\mathbf{K}}$:

$$\tilde{\mathbf{K}} = \sum_j \mathbf{U}^T \mathbf{K}_j \mathbf{U}.$$

Because each $\mathbf{K}_j$ has $O(1)$ size, the resulting $\tilde{\mathbf{K}}$ can be formed in $O(r^3)$ [8].

It is also possible to show [8] that for $O(r)$ integration points, $\tilde{\mathbf{f}}_{\text{int}}$ can be determined in $O(r^2)$ time. Other computations involving the reduction of external forces, and updating of the orginal FEM nodal positions (such as for graphic display), have a complexity of $O(nr)$.

2.2 Application to an FEM tongue model

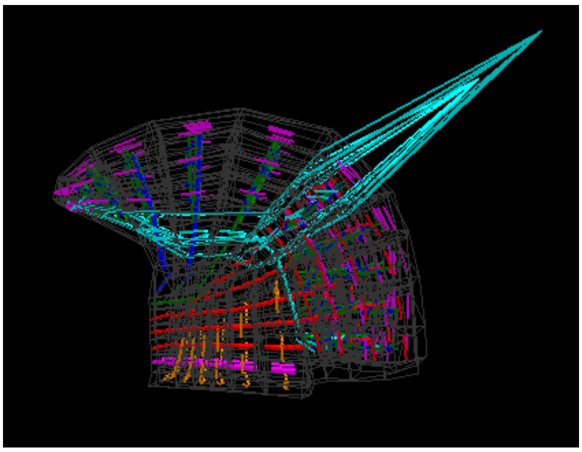

Figure 1. Cutaway view of the FEM tongue model, with colored lines showing the fiber directions for different muscles.

As a test case, we applied the above reduction method to an FEM tongue model that we had previously created using ArtiSynth [9], based on earlier work by [10]. The original FEM model contains 740 elements and 948 nodes and has a simulation time of about 60 msec per time step on an Intel Core i7 laptop.

We determined $\mathbf{U}$ using a linear extension to the modal basis as described in [5], and the integration points using a training process also described in [5]. The use of a training process to determine integration points (sometimes known as cubature points) is often advisable, particularly in the case of biomechanical models, which may contain a variety of materials and structures that vary throughout the model. Integration point selection must account for this, or otherwise simulation fidelity may be lost. In particular, our tongue model contains a variety of muscle fiber fields to simulate the motion of different tongue muscles (Figure 1), and the integration points need to sufficiently cover the different fiber directions to correctly reproduce the different muscle motions.

The resulting reduced model contains $r = 54$ basis vectors, with 66 integration points, and has a computation time of about 10 msec per time step, which is interactive for a 10 msec step size. A flexion motion performed using this reduced model is shown in Figure 2.

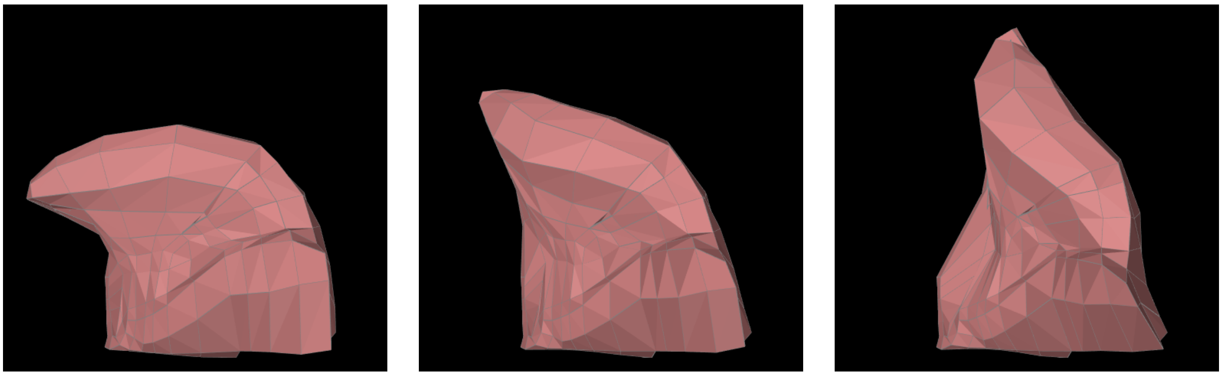

Figure 2. Flexion motion for a reduced tongue model with 54 DOF and 66 integration points. The images shown are formed from the original FEM surface mesh, updated to correspond to the underlying reduced coordinates.

3. ATTACHING POINTS AND FRAMES TO DEFORMABLE BODIES

Biomechanical models are often built using a variety of components and sub-models, which must then be connected together. ArtiSynth provides a number of means for doing this, including general constraints, joints, and the ability to directly attach points and coordinate frames directly to other components. This section focuses on the latter capability, and describes the recently enhanced mechanism for connecting either points or frames to deformable bodies. This allows (for example) a point-to-point muscle to be connected directly to an FEM tissue model.

The ArtiSynth attachment mechanism works by defining the coordinates $\mathbf{x}_a$ of the *attached* component to be a function of the coordinates $\mathbf{x}_m$ of one or more *master* components to which it is attached:

$$\mathbf{x}_a = f(\mathbf{x}_m). \tag{5}$$

This then implies that the velocities are related by a linear relationship of the form

$$\dot{\mathbf{x}}_a = \mathbf{G}_{am}\dot{\mathbf{x}}_m, \quad \mathbf{G}_{am} \equiv \nabla f(\mathbf{x}_m). \tag{6}$$

From the principle of virtual work, discussed above, forces $\mathbf{f}_a$ on the attached components then propagate back to forces $\mathbf{f}_m$ on the master components via

$$\mathbf{f}_m = \mathbf{G}_{am}^T \mathbf{f}_a. \tag{7}$$

For the case of a point attached to an FEM model, its position (and velocity) is given by a weighted sum of nearby nodal positions $\mathbf{x}_j$:

$$\mathbf{x}_a = \sum_j w_j \mathbf{x}_j, \quad \dot{\mathbf{x}}_a = \sum_j w_j \dot{\mathbf{x}}_j.$$

Forces $\mathbf{f}_a$ applied to the point then propagate back to each node according to

$$\mathbf{f}_j = w_j \mathbf{f}_a.$$

Often the local nodes are chosen to be the ones associated with the element containing the node, but in some cases it may be desirable to spread the attachment across a larger set of nodes, in order to better distribute forces imparted by the attached point across the FEM model (Figure 3).

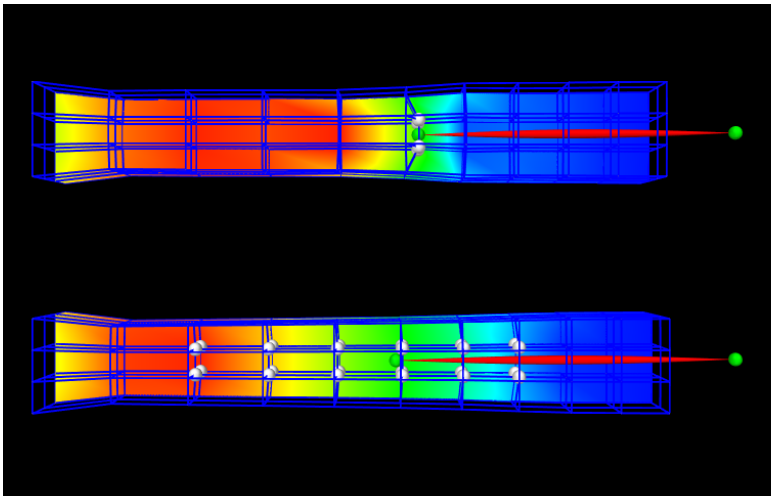

Figure 3. Two examples of a point-to-point muscle attached to an FEM model, using 4 support nodes (top) and 24 support nodes (bottom). The resulting stress/strain pattern is smoother and more diffuse with the larger number of support nodes.

Points can be attached to a reduced model in essentially the same way, only now the support nodes are themselves controlled by the underlying reduced coordinates:

$$\mathbf{x}_a = \sum_j w_j (\mathbf{x}_{j0} + \mathbf{U}_j \mathbf{q}), \quad \dot{\mathbf{x}}_a = \sum_j w_j \mathbf{U}_j \dot{\mathbf{q}},$$

where $\mathbf{x}_{j0}$ and $\mathbf{U}_j$ are the rest position and the submatrix of $\mathbf{U}$ corresponding to node j. With respect to (6), $\mathbf{G}_{am}$ takes the form

$$\mathbf{G}_{am} = \sum_j w_j \mathbf{U}_j.$$

One difference between reduced and FEM model attachments is that for the former it is often less necessary to be concerned about distributing the stress/strain across multiple nodes, since the model reduction process tends to do this automatically.

One feature that is *not* likely to be supported soon in ArtiSynth is the ability to directly connect two reduced models together. This is because there is no easy way to guarantee that the reduced motion of each body would be mutually compatible with the attachment, particularly when the attachment has spatial extent. Any future implementation of such an attachment would presumably have to "relax" its constraints to accommodate the motion range of the reduced models.

Coordinate frames can also be connected to deformable bodies in much the same way as for points. The frame origin is attached as a point, and the orientation $\mathbf{R}$ can then be determined in one of two ways:

- *Element shape functions:* If the nodes are associated with an element, then the local deformation $\mathbf{F}$ gradient can be determined using element shape functions in the standard FEM manner, with $\mathbf{R}$ then determined from $\mathbf{F}$ using a polar decomposition $\mathbf{F} = \mathbf{RP}$.

- *Procrustean method:* If the nodes are arbitrary, then $\mathbf{R}$ can be estimated based on a Procrustean approach. First we compute a matrix $\mathbf{F}$ according to

$$\mathbf{F} = \sum_j w_j (\mathbf{r}_j \mathbf{r}_{0j}^T),$$

  where w_j are the nodal weights and $\mathbf{r}_j$ and $\mathbf{r}_{0j}$ are the current and rest positions of the nodes described with respect to the coordinate frame origin. $\mathbf{R}$ can then again be determined from $\mathbf{F}$ using a polar decomposition $\mathbf{F} = \mathbf{RP}$.

The ability to connect frames means in particular that rigid bodies can be connected directly to deformable models, as shown in Figure 4, right.

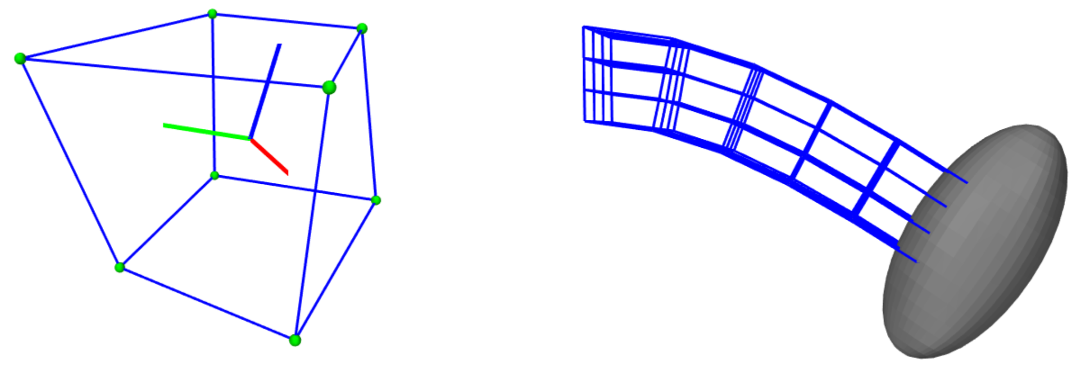

Figure 4. Attaching frames to deformable bodies. Left: a frame connected directly to the nodes of a single FEM element. Right: an ellipsoidal rigid body connected to the end of an FEM beam.

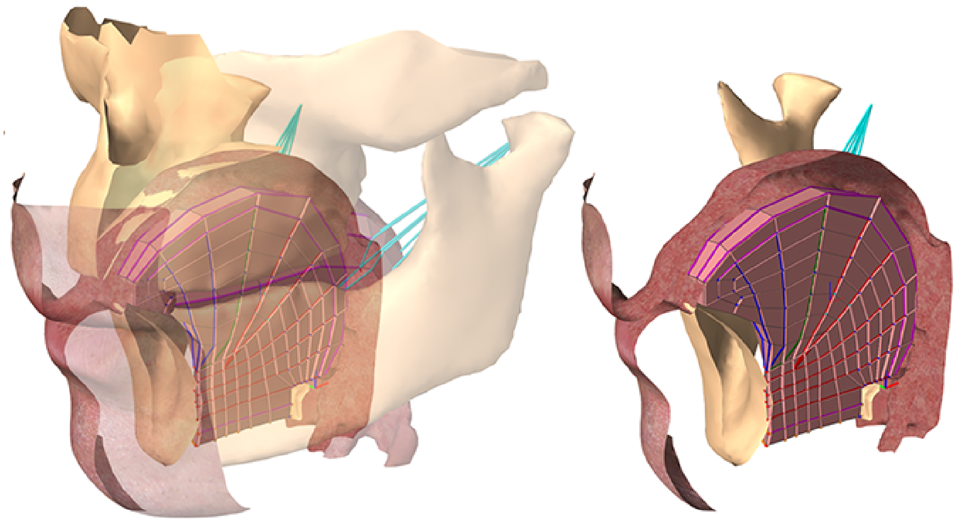

Figure 5. A skin mesh used to delimit the boundary of the human upper airway, connected to various surrounding structures including the palate, tongue, and jaw [11].

4. SKINNING AND EMBEDDED MESHES

Another useful technique for creating anatomical and biomechanical models is to attach a passive mesh to an underlying set of dynamically active bodies so that it deforms in accordance with the motion of those bodies. ArtiSynth allows meshes to be attached to collections of both rigid bodies and FEM models, facilitating the creation of structures that are either embedded in, or connect or envelope a set of underlying components.

The underlying method uses the attachment mechanism (equations 5-7), with the mesh vertices being the "attached components". The mesh deforms in response to the attachment configuration, while external forces applied to the mesh can be propagated back to the dynamic components via (7).

There are a variety of uses for such constructions. One is to create a continuous skin surrounding an underlying set of components. For example, when modeling the vocal tract, a disparate set of models describing the tongue, jaw, palate and pharynx can be connected together with a surface skin to get an airtight mesh [11] (Figure 5). Another use is to embed fine structures within an existing FEM mesh, for purposes of visualization and/or affecting material behavior. Figure 6 (left) shows this employed to create a detailed model of the human masseter, as described in [12]. Finally, embedded meshes can be used as a means to create a reduced model (Figure 6, right), where now the detailed model is the embedded mesh and the "reduced" model is the embedding FEM (which presumably has many fewer nodes tham the mesh). The resulting behavior can be made more realistic by weighting the mass and the stiffness values of the embedding FEM to account for regions where the embedded mesh is absent [13].

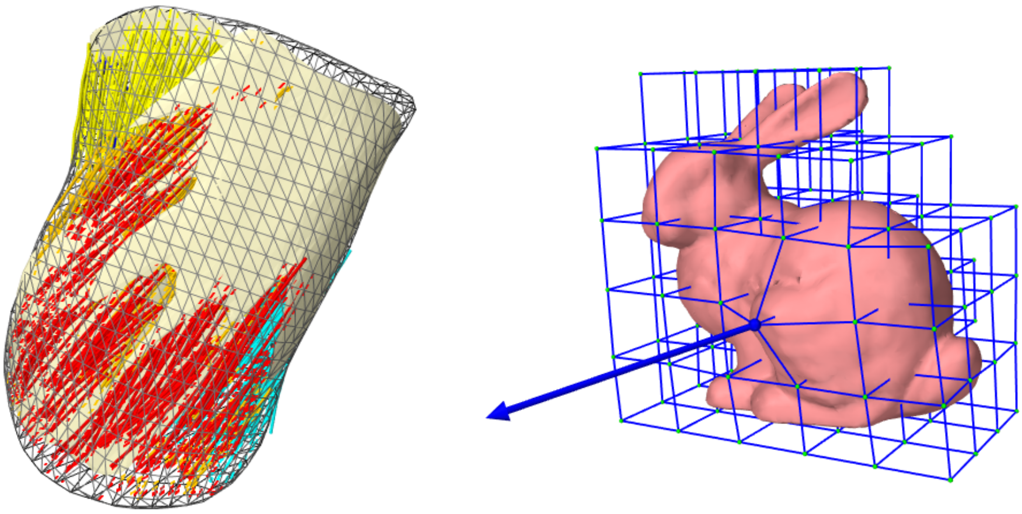

Figure 6. Left: an FEM model of the masseter (the principal muscle associated with chewing), showing embedded tendon sheets and muscle fiber fields. Right: the Stanford bunny made deformable by embedding it in a simple FEM hex mesh.

5. CONCLUSION

We have described a number of useful methods for enhancing the construction of biomechanical models. The first, model reduction, allows applications to implement complex deformable models at reasonable computational cost, and we are currently in the process of introducing this into the ArtiSynth modeling system. The other methods are currently available in ArtiSynth and facilitate the interconnection of model components and the introduction of auxiliary mesh structures for both visualization and simulation purposes. The unifying concept underlying them all is the principle of virtual work, which explains the force relationship between coordinate systems when the velocity relationship is known.

References

[1] John E Lloyd, Ian Stavness, and Sidney Fels. Artisynth: A fast interactive biomechanical modeling toolkit combining multibody and finite element simulation. In *Soft tissue biomechanical modeling for computer assisted surgery*, pages 355–394. Springer, 2012.

[2] P Krysl, S Lall, and JE Marsden. Dimensional model reduction in non-linear finite element dynamics of solids and structures. *International Journal for numerical methods in engineering*, 51(4):479–504, 2001.

[3] Alex Pentland and John Williams. Good vibrations: _ modal dynamics for graphics and animation. *Computer Graphics*, 23(3), 1989.

[4] Jernej Barbivc and Doug L James. Real-time subspace integration for st. venant-kirchhoff deformable models. *ACM transactions on graphics (TOG)*, 24(3):982–990, 2005.

[5] Christoph von Tycowicz, Christian Schulz, Hans-Peter Seidel, and Klaus Hildebrandt. An efficient construction of reduced deformable objects. *ACM Trans. Graph.*, 32(6):213:1–213:10, November 2013.

[6] Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, page 20. ACM, 2012.

[7] Javier Bonet and Richard D Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2008.

[8] Steven S An, Theodore Kim, and Doug L James. Optimizing cubature for efficient integration of subspace deformations. *ACM transactions on graphics (TOG)*, 27(5):165, 2008.

[9] Ian Stavness, John E Lloyd, and Sidney Fels. Automatic prediction of tongue muscle activations using a finite element model. *Journal of biomechanics*, 45(16):2841–2848, 2012.

[10] Stéphanie Buchaillard, Pascal Perrier, and Yohan Payan. A biomechanical model of cardinal vowel production: muscle activations and the impact of gravity on tongue positioning. *The Journal of the Acoustical Society of America*, 126(4):2033–2051, 2009.

[11] Ian Stavness, C Antonio Sánchez, John Lloyd, Andrew Ho, Johnty Wang, Sidney Fels, and Danny Huang. Unified skinning of rigid and deformable models for anatomical simulation s. In *SIGGRAPH Asia 2014 Technical Briefs*, page 9. ACM, 2014.

[12] C Antonio Sánchez, Zhi Li, Alan G Hannam, Purang Abolmaesumi, Anne Agur, and Sidney Fels. Constructing detailed subject-specific models of the human masseter. In *Imaging for Patient-Customized Simulations and Systems for Point-of-Care Ultrasound*, pages 52–60. Springer, 2017.

[13] Matthieu Nesme, Paul G Kry, Lenka Jevrábková, and Franccois Faure. Preserving topology and elasticity for embedded deformable models. In *ACM Transactions on Graphics (TOG)*, volume 28, page 52. ACM, 2009.