



**HAL**  
open science

## New Techniques for Combined FEM-Multibody Anatomical Simulation

John E Lloyd, Antonio Sánchez, Erik Widing, Ian Stavness, Sidney Fels,  
Siamak Niroomandi, Antoine Perrier, Yohan Payan, Pascal Perrier

► **To cite this version:**

John E Lloyd, Antonio Sánchez, Erik Widing, Ian Stavness, Sidney Fels, et al.. New Techniques for Combined FEM-Multibody Anatomical Simulation. J.M.R.S. Tavares & P.R. Fernandes. New Developments on Computational Methods and Imaging in Biomechanics and Biomedical Engineering, pp.75-92, 2019, 10.1007/978-3-030-23073-9\_6 . hal-02358380

**HAL Id: hal-02358380**

**<https://hal.science/hal-02358380v1>**

Submitted on 11 Nov 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# New Techniques for Combined FEM-Multibody Anatomical Simulation

John E. Lloyd, Antonio Sánchez, Erik Widing, Ian Stavness, Sidney Fels, Siamak Niroomandi, Antoine Perrier, Yohan Payan, and Pascal Perrier

**Abstract** This article describes a number of new techniques useful for the construction of biomechanical and anatomical models, particularly those that employ combined FEM-multibody simulation. They are being introduced to the ArtiSynth mechanical modeling system, and include: reduced coordinate modeling, in which an FEM model is made more computationally efficient by reducing it to a low degree-of-freedom 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. All these techniques are based on the principle of virtual work, and we illustrate their application with a number of examples, including a reduced FEM tongue model, subject-specific skeletal registration, skinning applied to modeling the human airway, and a detailed model of the human masseter.

## 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

---

Lloyd, Sánchez and Fels:  
Electrical and Computer Engineering, University of British Columbia

Widing and Stavness:  
Computer Science, University of Saskatchewan

Niroomandi, A. Perrier and Payan:  
TIMC, Université Joseph Fourier, Grenoble, France

P. Perrier:  
Institut de la Communication Parlée, INGP & Université Grenoble Alpes, France

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 [12] ([www.artisynth.org](http://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

Reduced coordinate modeling is a technique in which a deformable body is modeled using a restricted deformation basis instead of a collection of deformable finite elements [10]. It spans the gap between FEM methods and rigid bodies (which are themselves reduced models condensed to purely rigid motions), and can be 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 model has  $n$  nodes (each with 3 degrees of freedom), 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

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{U}\mathbf{q}, \quad (1)$$

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 [14] (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 [3] or applying additional linear transformations to the basis vectors [22]. 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 [10].

## 2.1 Reduced Dynamics

Background material for reduced dynamics modeling can be found in [20]. Here we provide an overview of reduced dynamics within the ArtiSynth modeling framework. An FEM model advances in time according to the dynamics

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

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 a semi-implicit integrator, to solve for the motion of the FEM model. In order to handle reduced models, it is necessary to find the equivalent reduced dynamics,

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

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 would 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):

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

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 must be 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 [5]. This process, sometimes known as *matrix assembly*, has  $O(n)$  complexity, with a proportionality constant depending on the nodal connectivity.

If  $\tilde{\mathbf{K}}$  is evaluated using (4) directly, the resulting complexity will be  $O(r^2n)$  (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

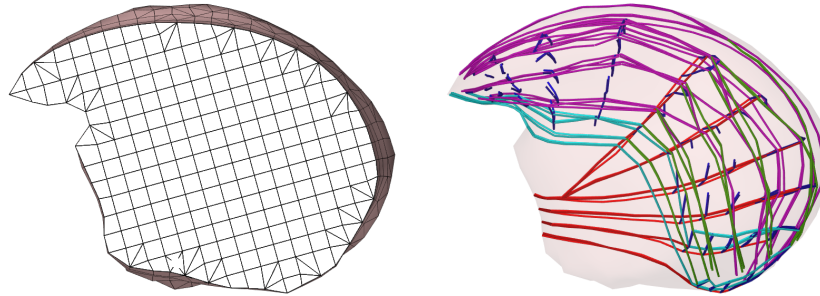
$O(r)$  (such that the number of points is proportional to  $r$  and not  $n$ ). For example, one can select  $O(r)$  elements, use a single integration point in the middle of each, and then rather than forming  $\mathbf{K}$ , instead accumulate the local stiffness matrix  $\mathbf{K}_j$  associated 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  $O(r^3)$  [1].

It is also possible to show that for  $O(r)$  integration points,  $\tilde{\mathbf{f}}_{\text{int}}$  can be determined in  $O(r^2)$  time [1]. Other computations involving the reduction of external forces, and updating of the original FEM nodal positions (such as for graphic display), have a complexity of  $O(nr)$ .

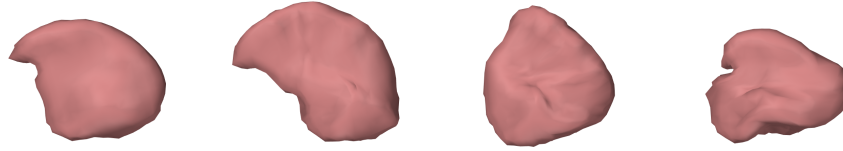
## 2.2 Application to an FEM Tongue Model



**Fig. 1** Left: Cutaway view of the FEM tongue model, showing the mesh structure; Right: fiber directions for the five muscles (HG, IL, SL, GGP, TRANS) used in this paper.

As a test case, we applied the above reduction method to a finite element model of the human tongue [8] for modeling the tongue motions associated with speech production. The mesh is hex dominant and contains 4255 elements and 2961 nodes. The constitutive material is the same as that used for an earlier model [6]: a nearly-incompressible Mooney Rivlin material with  $C_{10} = 1037$ ,  $C_{20} = 486$ , and bulk modulus  $\kappa = 10370$ .

Tongue deformation is effected by embedding within the model a number of fiber fields corresponding to the different tongue muscles. Each field is associated with an additional anisotropic constitutive law that results in directed stresses along the fiber directions when the muscle is activated. The work described in this paper uses five tongue muscles: the hyoglossus (HG), inferior longitudinal (IL), superior



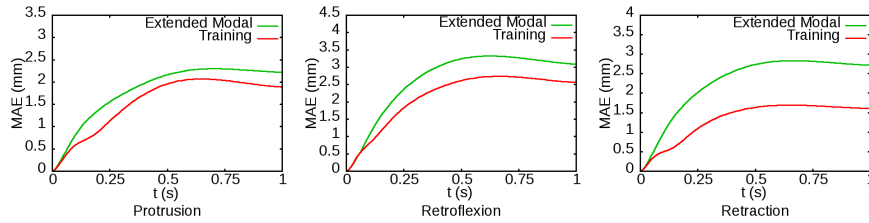
**Fig. 2** Different tongue motions used in this study. From left to right: rest position, protrusion, retroflexion, and retraction.

motion	GGP	TRANS	SL	HG	IL
protrusion	0.4	0.4	0.1	-	-
retroflexion	-	0.4	0.6	-	-
retraction	-	-	-	0.6	0.4

**Table 1** Maximum muscle excitation levels associated with the different motions.

longitudinal (SL), posterior genioglossus (GGP), and the transversalis (TRANS). Fig. 1 shows a cutaway view of the tongue mesh, along with a representation of the fiber directions for these muscles.

Our study consisted of creating two different reduced models for this tongue, using two different basis generation techniques, and then examining how well these were able to recreate three different tongue motions (protrusion, retroflexion and retraction) as produced by the original model. Fig. 2 shows the different motions, while Table 1 shows the muscle excitations associated with each.

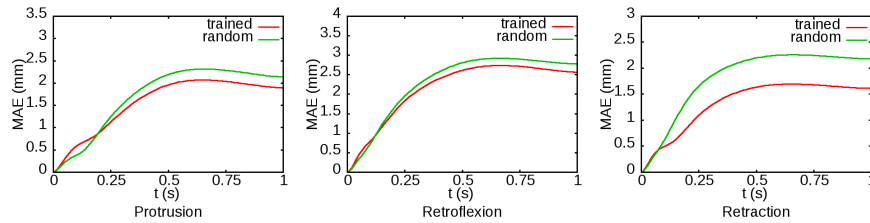


**Fig. 3** Mean average errors (mm) of the tongue motions for the reduced model (with 600 integration points) as compared to the unreduced model. From left to right: protrusion, retroflexion, and retraction. Each graph shows the error for both the extended modal and trained basis.

The first basis was generated using modal analysis combined with a linear extension technique, as described in [22], which provides the extra degrees of freedom needed to accommodate large deformations. Specifically, 6 linear modes were extended by applying 9 linearly independent affine transformations, resulting in a basis with  $r = 54$  vectors. The second basis was determined by a training technique as described in [10], during which the tongue was exercised through activation of all muscles, one at a time, and the displacements of all nodes were recorded. The final reduced basis, consisting of 20 modes, was then determined through a principal

orthogonal decomposition of the recorded data and selection of the most significant modes.

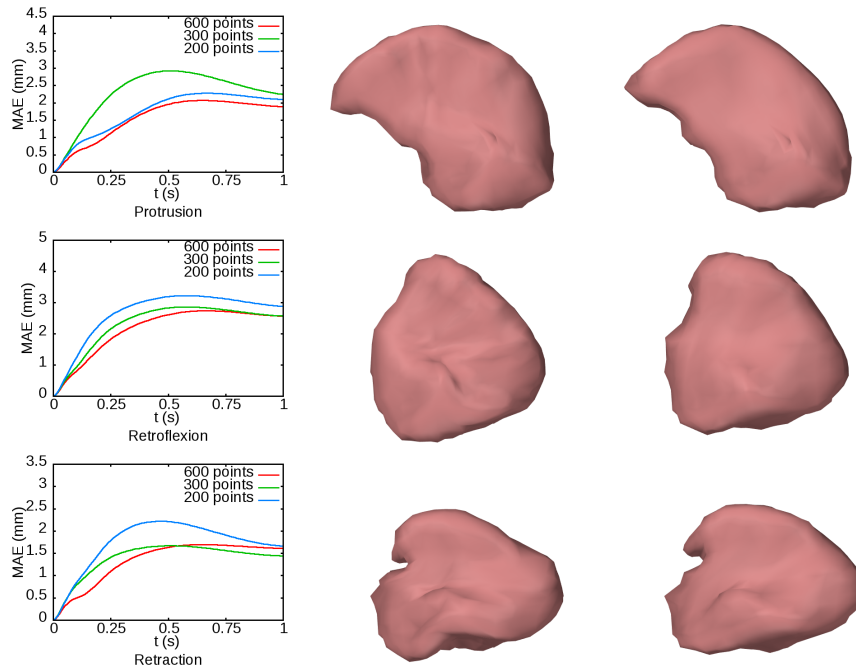
As mentioned in Sect. 2.1, the solution complexity for the reduced model is  $O(r^3)$  only if the number of integration points is  $O(r)$ . In particular, this means that the number of integration points must be subsampled in comparison to the original model. This can be achieved in a couple of ways: (a) by choosing a certain number of points randomly within the model, and (b) by using a training method to select the integration points, based on [22]. The training approach can be expected to give better results since it has more ability to ensure sufficient coverage of the muscle fiber fields, without which simulation fidelity may be lost. Results for both approaches are presented below. Also, to get a better idea of how error varied with the number of sampled integration points, we computed three different distributions of 600, 300 and 200 points, respectively.



**Fig. 4** Mean average errors (mm) of the tongue motions (using the trained basis) with 600 integration points that are (a) determined using a training method, and (b) selected at random.

To compare the effectiveness of the reduced model under each basis, each of the three tongue motions was executed, using 600 integration points, over a one second time interval, with the excitations ramped from zero to full strength over the interval  $t \in [0, 0.5]$  sec. and the displacement then being allowed to settle for an additional 0.5 sec. These motions were then compared with those of the original (unreduced) model, with the error associated with each node’s position being computed and averaged to determined a *mean average error* over time. The results for each motion are shown in Fig. 3, with mean average errors given in mm. For a sense of scale, the approximate tongue diameter is around 60 mm. These results show that the trained basis exhibited somewhat less error for a significantly smaller value of  $r$  (20 vs 54). We also observe that the reduced model has some dynamic lag with respect to the unreduced model, as evidenced by the small error overshoot near  $t = 0.5$ .

To study how the error was affected by the choice of integration points, the same tests were performed again (using the trained basis), with different point distributions. First, we tested the difference between choosing points randomly vs. identifying them with a training method. The results, for 600 points, are shown in Fig. 4. The training method produces slightly lower errors for two cases and significantly lower errors for another. This suggests the while training is likely to be preferred, a random method may work instead if the number of points is sufficiently large. Second, we tested the effect of using fewer points (200 and 300); results for this are



**Fig. 5** Mean average errors (mm) of the tongue motions, using the trained basis, with the number of integration points varying between 600, 300, and 200. From top to bottom: protrusion, retroflexion, and retraction. To get a sense of the absolute error, static images are provided of the unreduced model (middle) and 300-point reduced model (right) at the point of maximum error.

shown in Fig. 5. To provide a better sense of the worst-case error, static views are shown of the unreduced and 300 point reduced model at the point of maximum error. In general, fewer integration points results in a larger error, but not always: for the protrusion task, 300 integration points gives a larger error than 200. The dynamic lag also tends to increase as the number of points decreases.

Reduced model effectiveness is also illustrated by an online video, showing a side-by-side comparison of task execution between an unreduced model and a reduced model with 300 integration points:

[www.artisynth.org/videos/CMBBE2018ModelReduction.mp4](http://www.artisynth.org/videos/CMBBE2018ModelReduction.mp4)

Basis	600	300	200
Extended modal	53.2	33.0	23.6
Trained	27.7	15.2	12.1

**Table 2** Average per-step computation times (msec) for each basis with different numbers of integration points.



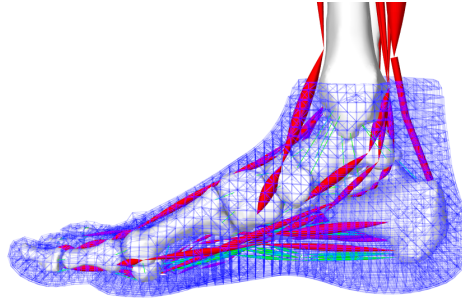
Finally, average per-step computation times were compared for each of the bases, using 200, 300 and 600 integration points, with the results shown in Table 2. Computations were performed on an Intel quad-core i7-7700 desktop with 16 GB of RAM. The average per-step computation time for the unreduced model was 188.9 msec, using the Pardiso [19] multicore direct solver utilizing a hybrid direct/iterative scheme. The reduced computation times were roughly linear in the number of integration points, as we would expect from the discussion of Sect. 2.1, while appearing to be roughly linear in  $r$ . However, that is because of the  $O(nr)$  overheads associated with system assembly and the updating of nodal positions from the reduced coordinates.

Typical maximum absolute errors in the above results are around 3 mm, which for an approximate tongue diameter of 60 mm corresponds to a relative error of around 5%. Meanwhile, for 300 or 200 integration points, the speed-up with the trained basis is more than 10 times, and reduces the average per-step computation time to interactive rates.

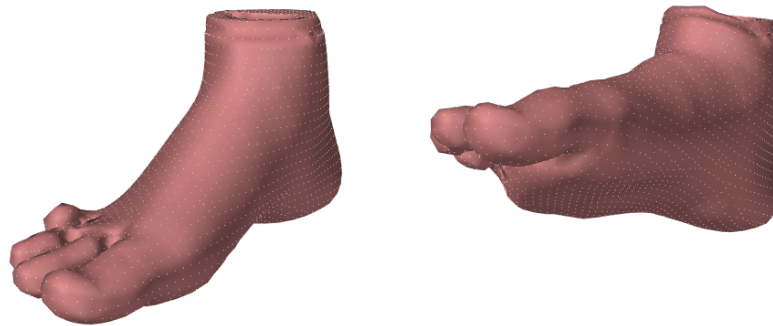
### 2.3 Application to an FEM Foot Model

As an example involving a larger FEM model containing internal rigid structures, we describe preliminary work on model reduction for the detailed biomechanical foot model described in [15]. Developed in ArtiSynth, this contains all the bones of the foot, together with tendons and ligaments, embedded within an FEM model that emulates the skin and other soft tissues (Fig. 6). The bones are modeled as rigid bodies, to which nearby FEM nodes are connected using point-to-frame attachments. The FEM model itself uses a neo-Hookean material and contains 23,298 elements and 13,087 nodes. Joints between the bones are modeled using unilateral contact, which provides more realistic joint motions, but, when combined with the FEM embedding, increases per-step computation times to around 2600 msec on a Intel quad-core i7-7700 desktop with 16 GB of RAM. When run by itself without any embedded structures, the FEM model has an execution speed of around 650 msec.

To help make this model suitable for clinical applications, we are investigating model reduction to reduce its computation time to the point where it can be run interactively. We developed a reduced basis for the model by creating a series of nodal displacements corresponding to various excitation levels of the tibialis anterior muscle and then applying a principal orthogonal decomposition in a manner similar to that used for the tongue model (Sect. 2.2). The basis contains 6 vectors, and when combined with 100 randomly-chosen integration points allows the model to be run at a real-time rate of around 10 msec/step (Fig. 7). Further investigations will consider other muscle excitations and external loadings such as floor contact.



**Fig. 6** The foot model, showing bones, tendons, ligaments, and the embedding FEM mesh.



**Fig. 7** The reduced foot model at rest (left) and after muscle excitation (right).

### 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. An illustrative example of this is the FRANK model [2] ( Fig. 8), which provides a reference model of human head and neck anatomy. Implemented in ArtiSynth [12], FRANK is designed to simulate anatomical functions related to swallowing, chewing and speech, and consists of various components modeled using finite elements, rigid bodies, point-to-point springs and muscles structures. ArtiSynth provides a number of means for connecting such components together, 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*



**Fig. 8** The FRANK model of human head and neck anatomy, with some structures (such as the jaw) not shown to reveal internal detail.

components to which it is attached:

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

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

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

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

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

### 3.1 Point Attachments

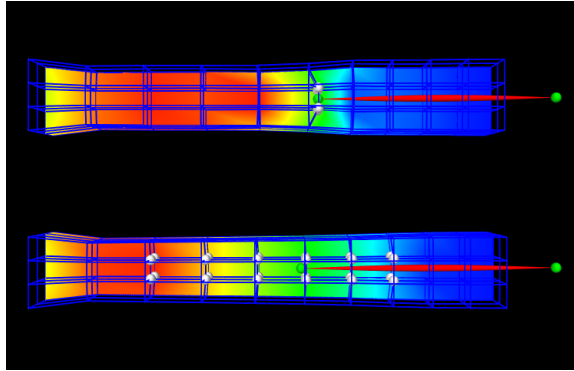
For the case of a point attached to an FEM model, its position (and velocity) is given by a weighted sum of nearby nodal positions  $\mathbf{x}_j$ :

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

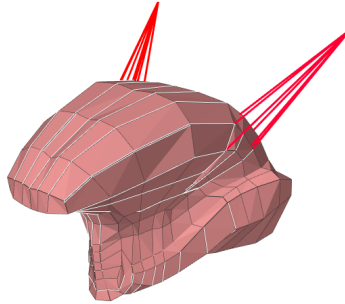
Forces  $\mathbf{f}_a$  applied to the point then propagate back to each node according to

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

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 (Fig. 9). A good case example of this is shown in Fig. 10, where the styloglossus muscles of a tongue are modeled as external point-to-point muscles connected to the main FEM tongue model.



**Fig. 9** 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.



**Fig. 10** A finite element model of the tongue, with the styloglossus modeled as externally connected point-to-point muscles (red). It may be desirable to distribute the styloglossus/tongue attachment across the nodes of multiple elements.

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:

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

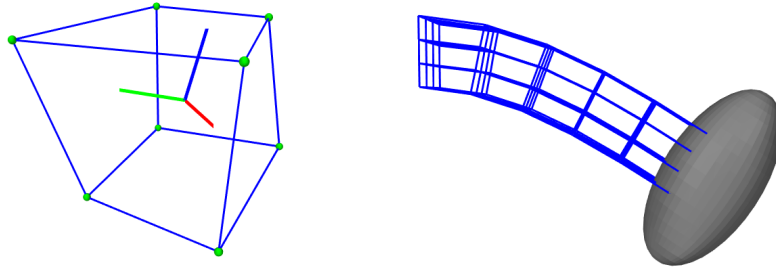
where  $\mathbf{x}_{j0}$  and  $\mathbf{U}_j$  are the rest position and the submatrix of  $\mathbf{U}$  corresponding to node  $j$ . With respect to (6),  $\mathbf{G}_{am}$  takes the form

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

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.

A current open problem 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 need to “relax” its constraints to accommodate the motion range of the reduced models.

### 3.2 Frame Attachments



**Fig. 11** 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.

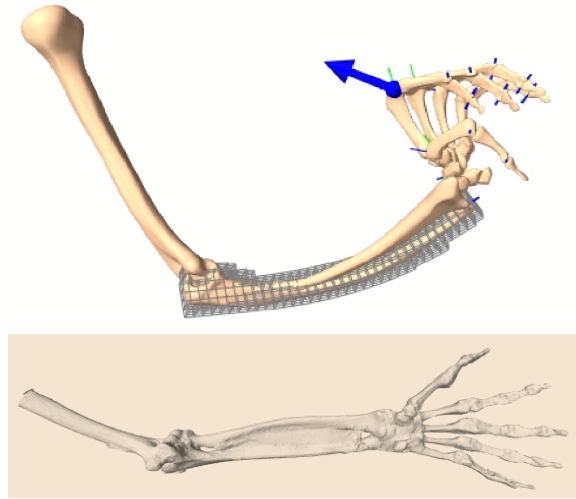
Coordinate frames can also be connected to deformable bodies in much the same way as for points. The frame origin is attached as a point, and the orientation  $\mathbf{R}$  can then be determined in one of two ways:

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

The ability to connect frames means in particular that rigid bodies can be connected directly to deformable models, as shown in Fig. 11, right.



**Fig. 12** Subject-specific registration of skeletal structures. Each bone mesh of a reference model (top) is made deformable by embedding it within a regular FEM grid (shown here for the ulna). The deformable bones are then connected using FEM joints, allowing the reference to be more easily registered to subject specific data (bottom) using ICP or similar techniques.

Since frames can be attached to deformable bodies, this also means that joints (which implement constraints between frames) can also be attached to deformable bodies. A useful application of this is the subject-specific registration of skeletal anatomy, as presented in [17] and illustrated in Fig. 12. This involves taking a reference model of a skeletal structure, including bones and joints, and registering (i.e., deforming) it to a specific subject based on medical imaging data (e.g., a CT scan). To do this, the bones in the reference model must be made deformable (which can be done by placing them within an embedding mesh, as described in Sect. 4) and then “attracted” to the subject data using a technique such as iterative closest point (ICP) [4]. In order to preserve the structural integrity of the reference model, the (deformable) bones are connected with joints appropriate to the anatomy, allowing the model to bend freely at each joint (up to joint limits) and to simultaneously register the shape and pose of each bone. If instead the reference was modeled as a single deformable structure without such joints, it would be necessary to greatly

reduce the stiffness near each joint location, in an anisotropic way that emulated the constraints of each joint.

## 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-within, connected-to, or enveloped-by a set of underlying components. Such mesh embedding approaches are well known in the computer graphics community and have more recently been applied to biomechanics [13], and also figure prominently in the SOFA system [7].

It should be noted that mesh embedding provides an alternate way to perform model reduction, in the sense that the number of dynamic DOFs for the resulting system is determined by the number of nodes in the embedding mesh. By using a course embedding mesh, the number DOFs can be significantly reduced.

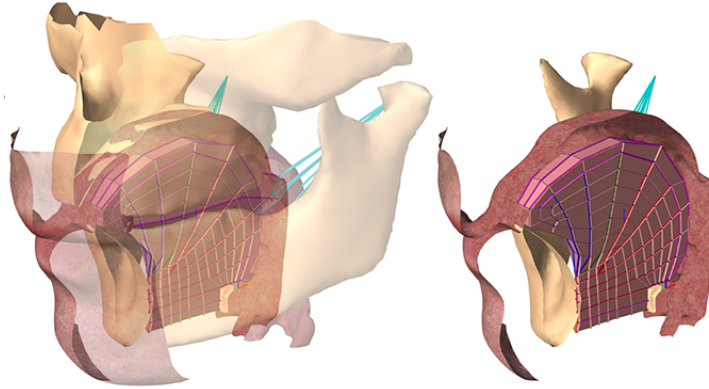
The underlying method uses the attachment mechanism (equations 5-7), with the mesh vertices being the “attached components”. The mesh deforms in response to the attachment configuration, while external forces applied to the mesh can be propagated back to the dynamic components via (7). This technique is useful in a variety of applications, as presented below.

### 4.1 Skinning for Modeling the Human Airway

One application is to create a continuous skin surrounding an underlying set of anatomical components. For example, for modeling the human airway, a disparate set of models describing the tongue, jaw, palate and pharynx can be connected together with a surface skin to form a seamless airtight mesh (Fig. 13), as described in [21]. This then provides a uniform boundary for handling air or fluid interactions associated with tasks such as speech or swallowing. In [21], each skin mesh vertex is attached to one or more *master* components, which can be either an FEM model or a rigid body. The position  $\mathbf{q}_v$  of each vertex is given by a weighted sum of contributions from the master components, according to

$$\mathbf{q}_v = \mathbf{q}_{v,0} + \sum_i^M w_i f_i(\mathbf{q}_m, \mathbf{q}_{m0}, \mathbf{q}_{v,0}) \quad (8)$$

where  $\mathbf{q}_{v,0}$  is the initial position of the skinned vertex,  $\mathbf{q}_m$  and  $\mathbf{q}_{m0}$  give the positions and rest positions of the  $M$  master components, and  $w_i$  and  $f_i$  are the skinning weight and blend function associated with the  $i$ -th master component. Further details are



**Fig. 13** A skin mesh used to delimit the boundary of the human upper airway, connected to various surrounding structures including the palate, tongue, and jaw [21].

given in [21]. The position equation (8) can be differentiated to yield a velocity relationship

$$\mathbf{u}_v = \mathbf{G}\mathbf{u}_m,$$

where  $\mathbf{u}_v$  and  $\mathbf{u}_m$  are the velocities of the skin vertex and the master components and  $\mathbf{G}$  is a (local) matrix. Then from the principle of virtual work, vertex forces  $\mathbf{f}_v$  can be propagated back onto the master component forces  $\mathbf{f}_m$  via

$$\mathbf{f}_m = \mathbf{G}^T \mathbf{f}_v.$$

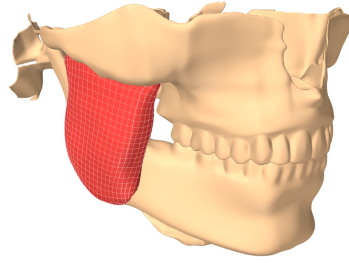
In general, this means that forces, pressures or contact interactions applied to the skin can be reflected back onto the underlying master components. In the case of the airway model, these external loads could involve pressures from air, fluid or food bolus interactions.

## 4.2 Mesh Embedding Applied to Modeling the Masseter

Embedding and attachment techniques have recently been applied to the creation of a finite element model of the human masseter [16], which is the principal muscle used in chewing (Fig. 14). This model contains detailed information about the internal muscle fascicles and aponeuroses, with a primary purpose being to study the importance of aponeuroses stiffness in the transmission of force within the masseter.

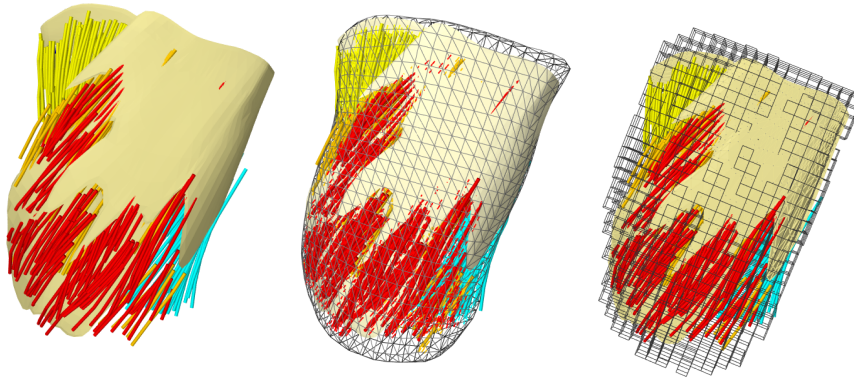
Fascicle and aponeuroses data were obtained using the dissection and digitization procedure of [9]. Fascicle data was discretized into a set of line segment meshes, while the aponeuroses (tendon sheets) were discretized into triangular surface meshes. A muscle volume was then constructed by constructing a wrapping surface around this fiber and aponeuroses data, in a manner similar to [11]. Orig-





**Fig. 14** Model of the masseter connected to the jaw and mandible.

inally, this was used as the surface mesh from which a conforming hex-dominant volumetric mesh was constructed [18] (Fig. 15, middle). However, creating such a conforming mesh was both time consuming and also yielded a number of poorly conditioned elements, and so a mesh embedding approach was adopted instead in which the muscle volume surface was embedded inside a coarse but highly regular and well-conditioned grid (Fig. 15, right). Model fidelity was improved using the technique of [13], in which the mass and stiffness values of the embedding FEM are weighted to account for regions where the embedded mesh is absent.



**Fig. 15** Left: raw digitized data for masseter fascicles and aponeuroses. Middle: FEM masseter model based on a conforming mesh generated from a wrapping surface enclosing the data. Right: FEM model based on a regular embedding mesh. For both models, fascicles and aponeuroses are added as embedded structures.

Fascicles and aponeuroses were also incorporated within this primary FEM mesh. Fascicle data was embedded within the primary mesh, hence allowing it to deform accordingly. The fiber directions were used to determine the direction of muscle contraction used by the muscle constitutive law at nearby integration points of the primary mesh. Aponeuroses were modeled as thin membrane-like FEM

models, created by extruding their triangular surface data using wedge elements. These membrane FEMs were then attached to the primary FEM by connecting each membrane node to its containing element within the primary mesh (as described in Sect. 3.1), allowing the membrane stiffness to be transferred onto the entire structure.

By using embedding and attachment techniques, it is possible to create a masseter model with far fewer degrees of freedom (and hence a much faster simulation time) than would otherwise be possible. The embedding technique allows the primary mesh to be set at a resolution appropriate to the overall deformability of the muscle, rather than a need to accommodate the surface structure. More importantly, the use of attachments to connect the aponeuroses allow the primary and aponeuroses meshes to be non-conforming. Otherwise, it would be necessary to employ meshes with far higher resolutions, which would be both harder to construct and would result in much higher simulation times.

## 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. Most of the effort associated with model reduction involves determining both the basis and integration point distribution. Our results suggest that training techniques tend to yield the best basis results, yielding both a smaller basis and less error. Training is also useful for selecting integration points, although random point selection may sometimes work sufficiently well. We have also seen that model reduction has the potential to improve computational speeds to interactive rates.

The other methods include ways to attach points and frames to finite element models, along with skinning and mesh embedding. All of these are currently available in ArtiSynth and facilitate the dynamic interconnection of model components and the introduction of auxiliary mesh structures for both visualization and simulation purposes. They have already been applied to a number of applications in biomechanics; those described here include a large scale reference model of the head and neck, subject-specific skeletal registration, skinning applied to modeling the human airway, and a detailed model of the human masseter.

The unifying concept underlying all these techniques is the principle of virtual work, which explains the force relationship between coordinate systems when the velocity relationship is known.

## References

1. 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.
2. Peter Anderson, Sidney Fels, Negar M Harandi, Andrew Ho, Scott Moisik, C Antonio Sánchez, Ian Stavness, and Keyi Tang. Frank: A hybrid 3d biomechanical model of the head and neck. In *Biomechanics of Living Organs*, pages 413–447. Elsevier, 2017.
3. 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.
4. Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.
5. Javier Bonet and Richard D Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2008.
6. 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.
7. Francoois Faure, Christian Duriez, Hervé Delingette, Jérémie Allard, Benjamin Gilles, Stéphanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlik, et al. Sofa: A multi-model framework for interactive physical simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, pages 283–321. Springer, 2012.
8. Nicolas Hermant, Pascal Perrier, and Yohan Payan. Human tongue biomechanical modeling. In *Biomechanics of Living Organs*, pages 395–411. Elsevier, 2017.
9. Soo Y Kim, Erin L Boynton, Kajeandra Ravichandiran, Lillia Y Fung, Robert Bleakney, and Anne M Agur. Three-dimensional study of the musculotendinous architecture of supraspinatus and its functional correlations. *Clinical Anatomy: The Official Journal of the American Association of Clinical Anatomists and the British Association of Clinical Anatomists*, 20(6):648–655, 2007.
10. 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.
11. Dongwoon Lee, Kajeandra Ravichandiran, Ken Jackson, Eugene Fiume, and Anne Agur. Robust estimation of physiological cross-sectional area and geometric reconstruction for human skeletal muscle. *Journal of Biomechanics*, 45(8):1507–1513, 2012.
12. John E Lloyd, Ian Stavness, and Sidney Fels. Artistry: 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.
13. Matthieu Nesme, Paul G Kry, Lenka Jevrábková, and Francoois Faure. Preserving topology and elasticity for embedded deformable models. In *ACM Transactions on Graphics (TOG)*, volume 28, page 52. ACM, 2009.
14. Alex Pentland and John Williams. Good vibrations: \_ modal dynamics for graphics and animation. *Computer Graphics*, 23(3), 1989.
15. Antoine Perrier, Vincent Luboz, Marek Bucki, Francis Cannard, Nicolas Vuillerme, and Yohan Payan. Biomechanical modeling of the foot. In *Biomechanics of Living Organs: Hyperelastic Constitutive Laws for Finite Element Modeling*. Academic Press, 2017.
16. 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.
17. C. Antonio Sanchez, John E. Lloyd, Zhi Li, and Sidney Fels. Subject-specific modelling of articulated anatomy using finite element models. 13th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering (CMBBE), 2015.
18. C. Antonio Sanchez, Ian Stavness, John E. Lloyd, Alan G. Hannam, and Sidney Fels. Modelling mastication: the important role of tendon in force transmission. 12th International

- Symposium on Computer Methods in Biomechanics and Biomedical Engineering (CMBBE), 2014.
19. Olaf Schenk and Klaus Gärtner. Solving unsymmetric sparse systems of linear equations with pardiso. *Future Generation Computer Systems*, 20(3):475–487, 2004.
  20. 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.
  21. Ian Stavness, C Antonio Sánchez, John Lloyd, Andrew Ho, Johny 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.
  22. 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.