Fully Momentum-Conserving Reduced Deformable Bodies with Collision, Contact, Articulation, and Skinning

Rahul Sheth*
Stanford University
Pixar Animation Studios

Wenlong Lu*
Stanford University

Yue Yu*
Stanford University

Ronald Fedkiw*
Stanford University
Industrial Light & Magic

Abstract

We propose a novel framework for simulating reduced deformable bodies that fully accounts for linear and angular momentum conservation even in the presence of collision, contact, articulation, and other desirable effects. This was motivated by the observation that the mere excitation of a single mode in a reduced degree of freedom model can adversely change the linear and angular momentum. Although unexpected changes in linear momentum can be avoided during basis construction, adverse changes in angular momentum appear unavoidable, and thus we propose a robust framework that includes the ability to compensate for them. Enabled by this ability to fully account for linear and angular momentum, we introduce an impulse-based formulation that allows us to precisely control the velocity of any node in spite of the fact that we only have access to a lower-dimensional set of degrees of freedom. This allows us to model collision, contact, and articulation in a robust and high visual fidelity manner, especially when compared to penalty-based forces that merely aim to coerce local velocities. In addition, we propose a new "deformable bones" framework wherein we leverage standard skinning technology for "bones," "bone" placement, blending operations, etc. even though each of our "deformable bones" is a fully simulated reduced deformable model.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: deformable bodies, elasticity, collisions, skinning, model reduction, subspace

1 Introduction

Simulations of deformable objects are known to be both visually interesting and computationally demanding. One way to lessen the computational expense of these simulations is to use a reduced-order (or subspace) model to represent the internal dynamics. The performance of such a model depends on the size of the subspace r as well as the basis vectors chosen to represent that space. If n is the size of the full-order model, optimal performance benefits are achieved when $r \ll n$ and the simulation takes $O(r^p)$ time for small values of p.

Basis construction is a difficult problem and many solutions have been proposed, including modal analysis [Pentland and Williams 1989], a method to include the derivatives of linear modes (i.e. modal derivatives) [Idelsohn and Cardona 1985], mass-weighted principal component analysis (PCA) of an existing simulation dataset [Barbič and James 2005], augmentation of the basis with localized displacement vectors [Harmon and Zorin 2013], etc. Regardless of the specific method used, most of the bases constructed by these methods are linear and can thus be written in a matrix form. Our method is compatible with all of the aforementioned approaches yet does not require mass normalization, but does require full column rank. This keeps the framework as general as possible and enables an example-based reduced-order scheme that we discuss in Section 8.

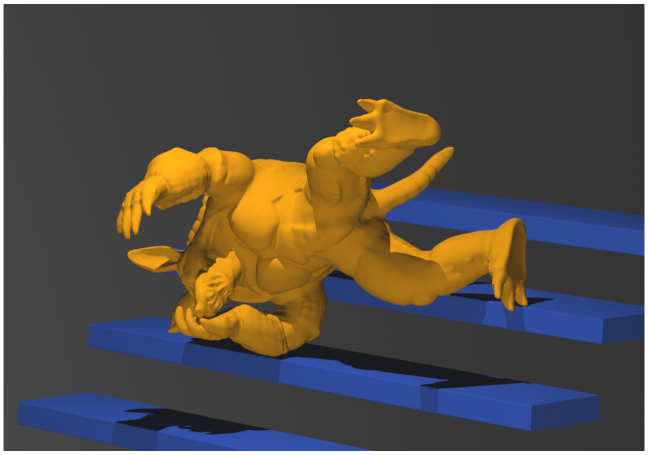

Figure 1: *An articulated reduced deformable body falls down a flight of stairs undergoing collision and self-collision. Even with simple linear finite elements, the results have high visual fidelity due to our new "deformable bones" skinning strategy. We stress that this is a pure simulation of a reduced deformable body, and no misleading techniques such as first simulating a rigid body and then attaching a deformable skin after the fact were employed.*

We use a reduced-order approximation of the displacement $\vec{u} = S\vec{q}$ where the columns of S represent particle displacements from the rest pose $\vec{x}_0$. Here $\vec{u}$ and $\vec{x}_0$ have dimensions n by 1, S has n rows and r columns, and the entries of $\vec{u}$, $\vec{x}_0$ and S are all 3-vectors. Then, particle positions are given by $S\vec{q} + \vec{x}_0$, where $\vec{q}$ is a reduced space vector which when set to a basis vector (i.e. $\vec{q} \in \{\vec{e}_1, \vec{e}_2, ..., \vec{e}_r\}$) results in $S\vec{e}_k + \vec{x}_0$ reconstructing the k^{th} basis shape. To enable fully unconstrained simulation, we embed this system in a rigid frame

$$\vec{x} = \mathbf{R}(S\vec{q} + \vec{x}_0) + \vec{\mathbf{T}} \tag{1}$$

where $\mathbf{R}$ is a diagonal matrix of rotations R and $\vec{\mathbf{T}}$ is a vector of translations $\vec{t}$. Note that this formulation reduces to a rigid body when there are no internal deformation modes, i.e. when $S = 0$. This is similar to the method of [Terzopoulos and Witkin 1988] where deformation occurs in a rigid frame.

After assigning each particle a mass (consistent with the requirement in Section 4), we adjust the rest state $\vec{x}_0$ and each basis shape $S\vec{e}_k + \vec{x}_0$ such that their centers of mass are at the origin. This

*e-mail: {rbsheth,wenlongl,yuey,fedkiw}@cs.stanford.edu

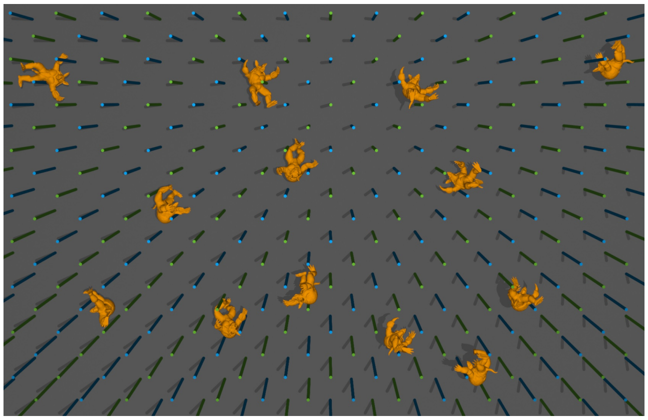

Figure 2: *Thirteen out of the thirty-one articulated reduced deformable armadillos we drop through a Pachinko machine.*

guarantees that the mass-weighted column sums of S are $\vec{0}$ and that $S\vec{q} + \vec{x}_0$ contains no translation of the center of mass. Thus, any translation of the center of mass in Equation 1 is represented entirely by $\vec{t}$. This is useful for formulating collisions and constraints in maximal coordinates using impulses, because the total change in linear momentum is due solely to changes in $\vec{t}$. Unfortunately, a similar approach for angular momentum is generally not feasible, as changes in $\vec{q}$ affect the particle configuration, the inertia tensor, and the angular velocity. Similarly, changes in $\dot{\vec{q}}$ affect the angular momentum in complex ways, and thus R and $\vec{q}$ are intrinsically coupled. The fact that $\vec{q}$ and $\dot{\vec{q}}$ may have no effect on the linear momentum is either serendipitous or explicitly exploited in many reduced model formulations, whereas the inability to decouple $\vec{q}$ and $\dot{\vec{q}}$ from R and $\dot{R}$ with regards to angular momentum has not received the cautionary attention that it should.

We derive in detail our method to calculate the angular momentum caused by changes in $\vec{q}$ and $\dot{\vec{q}}$ and subsequently compensate for it by adjusting the rigid frame's angular momentum. Linear momentum is not changed by $\dot{\vec{q}}$ due to the aforementioned basis processing step but could be handled in a similar fashion if necessary. Per-body conservation enables us to formulate conservative methods for collisions, contact, and articulation, allowing us to maintain the physically correct total momentum of the system (deformation plus rigid motion) using a simple integration scheme. We combine these capabilities with a decomposition and a skinning technique to enable stable, visually pleasing reduced simulations of complex deformable models.

2 Related Work

Equation 1 bears some resemblance to other approaches, such as [Metaxas and Terzopoulos 1992; Hauser et al. 2003; Lall et al. 2003; Gilles et al. 2011; Barbič and Zhao 2011; Fan et al. 2013]. In particular, [Shabana 2005] intrinsically couples R, $\vec{q}$, and $\vec{t}$ since they are solving for general rigid linkages with deformable components and do not necessarily have control over the coordinate systems. In contrast, we can separate the treatment of $\vec{t}$ from R and $\vec{q}$, making R and $\vec{q}$ only one-way dependent on $\vec{t}$ (i.e. $\vec{t}$ is independent of R and $\vec{q}$) by moving all the basis shapes' centers of mass to the origin, whereas in [Shabana 2005] the generalized Newton-Euler matrix is dense and $\vec{t}$ is two-way coupled to R and $\vec{q}$. As noted above, it is not generally possible to process the basis shapes in a way that separates R from $\vec{q}$; however, we do take a novel approach that quantifies the relationship between R and $\vec{q}$ allowing us to formulate the general impulse equations between and among reduced deformable models.

The articulation of reduced deformable bodies has been discussed previously in [Kry et al. 2002; Choi and Ko 2005]. However, these methods did not attempt to solve the rigid and the reduced deformable systems in a fully coupled manner. In particular, [Kim and James 2011] did not consider two-way coupling between R, $\vec{q}$, and $\vec{t}$ since they assumed R and $\vec{t}$ were prescribed by the underlying motion of the articulated figure, whereas in the general case that we address (similar to [Shabana 2005]) R and $\vec{t}$ vary based on the laws of physics. Generally speaking, [Shabana 2005] takes a force-based generalized coordinates approach to the equations, since their focus is on articulated linkages where arbitrary, unpredictable collisions, contact, closed-loops, etc. are less common, whereas in a graphical simulation system they are commonplace and therefore we take a maximal coordinates approach using impulses.

Skinning articulated rigid bodies is a widely-used technique in graphics, see e.g. [Lewis et al. 2000; Kavan et al. 2007; Baran and Popović 2007]. The use of skinning combined with two-way coupled full-space deformations for articulated bodies was explored in [Liu et al. 2013]. Our novel contribution is not the skinning directly but rather the application of skinning to articulated reduced deformable body parts of a character. Output-sensitive collision detection for reduced deformable bodies has been discussed in [James and Pai 2004], and we extend their BD-tree ideas to work on a skin made up of multiple underlying reduced deformable bodies. A similar method for standard skinning models was described in [Kavan and Žára 2005]. [Barbič and James 2010; Teng et al. 2014] explored self-collision pre-computation schemes, but not on skinned bodies. Their work could potentially be extended to speed up the detection of self-collisions on our skinned bodies as necessary.

3 Kinematic Framework

Each row of Equation 1

$$\vec{x}_i = R(S_i\vec{q} + \vec{x}_{0,i}) + \vec{t} \tag{2}$$

describes the position of a single particle. Differentiating Equation 2 with respect to time, the velocity is

$$\dot{\vec{x}}_i = \dot{R}(S_i\vec{q} + \vec{x}_{0,i}) + RS_i\dot{\vec{q}} + \dot{\vec{t}} = \vec{\omega}^*_R\vec{r}_i + RS_i\dot{\vec{q}} + \dot{\vec{t}} \tag{3}$$

where $\vec{r}_i = R(S_i\vec{q} + \vec{x}_{0,i})$ is the moment arm for particle i, $\vec{\omega}_R$ is the angular velocity associated with the change in R, * represents the skew-symmetric cross product matrix for a given vector, and $\dot{R} = \vec{\omega}^*_R R$.

The total linear velocity of the center of mass $\vec{v}$ is defined in world space as the linear velocity of the frame plus the net linear velocity of the deformation. As such, it may have contributions from two separate terms in Equation 3, namely $\dot{\vec{t}}$ and $RS_i\dot{\vec{q}}$. Likewise, the total angular velocity $\vec{\omega}$ may have contributions from $\vec{\omega}^*_R\vec{r}_i$ and $RS_i\dot{\vec{q}}$. Therefore, we write

$$\vec{v} = \dot{\vec{t}} + \vec{v}' \tag{4a}$$

$$\vec{\omega} = \vec{\omega}_R + \vec{\omega}' \tag{4b}$$

where $\vec{v}'$ and $\vec{\omega}'$ indicate the contributions from $RS_i\dot{\vec{q}}$. Equations 3 and 4 allow us to write

$$\dot{\vec{x}}_i = \vec{v} + \vec{\omega}^*\vec{r}_i + \left[RS_i\dot{\vec{q}} - \vec{v}' - \vec{\omega}'^*\vec{r}_i\right] \tag{5}$$

where the term in brackets should have no net contribution to the total linear or angular velocity.

Since the term in brackets in Equation 5 should have no net contribution to the total linear momentum, i.e. all the linear momentum is contained in $\vec{v}$, we may write

$$R\sum_{i=1}^{n} m_i S_i \dot{\vec{q}} - M\vec{v}' = \vec{0} \tag{6}$$

where $M = \sum_{i=1}^{n} m_i$ is the total mass of all n particles of the object. Since we adjusted our basis so that S would have mass weighted column sums of $\vec{0}$, the first term in Equation 6 is zero and thus $\vec{v}' = \vec{0}$. That is, $\vec{v} = \dot{\vec{t}}$ and the velocity of the internal modes $\dot{\vec{q}}$ makes no contribution to the total linear momentum of the object.

A similar discussion leads to

$$\sum_{i=1}^{n} m_i \vec{r}_i^{\,*} \left[RS_i\dot{\vec{q}} - \vec{\omega}'^{*}\vec{r}_i\right] = \vec{0} \tag{7}$$

which can be re-arranged and solved for $\vec{\omega}'$ to obtain

$$\vec{\omega}' = I^{-1}\sum_{i=1}^{n} m_i \vec{r}_i^{\,*} RS_i\dot{\vec{q}} \tag{8}$$

where $I = \sum_{i=1}^{n} m_i (\vec{r}_i^{\,*})^T \vec{r}_i^{\,*}$ is the inertia tensor. Ideally, one could remove all internal rotation from the basis so that $\vec{\omega}' = 0$ and $\vec{\omega} = \vec{\omega}_R$, which is done trivially for rigid bodies. However it is not clear that one can remove this rotation in general, and therefore one must account for both $\vec{\omega}'$ and $\vec{\omega}_R$.

To further elucidate Equation 8, consider the total global angular momentum of a body

$$\vec{L}_{global} = \vec{x}_M^{\,*} M\vec{v} + \sum_{i=1}^{n} m_i \vec{r}_i^{\,*}\vec{v}_i = \vec{x}_M^{\,*} M\vec{v} + \sum_{i=1}^{n} I_i\vec{\omega}_i \tag{9}$$

where $\vec{x}_M$ is the world space location of the center of mass, $\vec{v}_i$ is the relative velocity of particle i with respect to the center of mass velocity $\vec{v}$, $I_i = m_i \left(\vec{r}_i^{\,*}\right)^T \vec{r}_i^{\,*}$ is particle i's contribution to the inertia tensor, and $\vec{\omega}_i = \vec{r}_i^{\,*}\vec{v}_i / \vec{r}_i^{\,T}\vec{r}_i$ is the angular velocity of particle i due to $\vec{v}_i$ with respect to the center of mass. The second equality in Equation 9 is true because $\vec{r}_i^{\,*}\vec{v}_i = \vec{r}_i^{\,*}\vec{\omega}_i^{\,*}\vec{r}_i$ since the dilational component of $\vec{v}_i$ is annihilated by the cross product with $\vec{r}_i$ leaving only the rotational component. The summation term in Equation 9 is the angular momentum of the body with respect to its center of mass, $\vec{L}_{body}$. Thus, one can more rigorously define the total angular velocity $\vec{\omega}$ for the deforming body as

$$\vec{\omega} \overset{\text{def}}{\equiv} I^{-1}\sum_{i=1}^{n} I_i\vec{\omega}_i = I^{-1}\vec{L}_{body}. \tag{10}$$

Multiplying Equation 4b by I then results in

$$\vec{L}_{body} = I\vec{\omega} = I\left(\vec{\omega}_R + \vec{\omega}'\right). \tag{11}$$

The reduced degrees of freedom $\dot{\vec{q}}$ make a non-zero contribution to the global angular momentum, which can be accounted for by adding a global rotational velocity $\vec{\omega}'$ that offsets the contribution from $\dot{\vec{q}}$ keeping the total global angular momentum as represented by $\vec{\omega}$ constant. We emphasize that the ability to make a correction to the global angular velocity along the lines of $\vec{\omega}'$ is quite important for conserving angular momentum; the fact that our framework in Equation 1 contains the rotation matrix R allows us to readily make this modification.

Figure 3 illustrates the difference our correction term can have. We have found that the conservation of angular momentum is crucial for plausible collision behavior. The strange and numerically unstable behavior we observed while simulating unconstrained reduced bodies did in fact disappear once we added enough anchor constraints. However, angular momentum conservation allows us to make the unconstrained simulations much more stable in addition to being more physically correct.

Whereas the summation in Equation 8 appears to be $O(n)$, its computation can be optimized by pre-computing terms and calculating the reduced angular momentum in object space. Similarly, the $O(n)$ inertia tensor calculation can also be optimized. Although updating $\vec{\omega}'$ and I can be relatively expensive, these costs are only fully incurred when the positional terms change.

4 Dynamic Framework

We differentiate Equation 3 with respect to time in order to obtain the acceleration

$$\ddot{\vec{x}}_i = (\dot{\vec{\omega}}_R^{\,*} + \vec{\omega}_R^{\,*}\vec{\omega}_R^{\,*})\vec{r}_i + 2\vec{\omega}_R^{\,*}RS_i\dot{\vec{q}} + RS_i\ddot{\vec{q}} + \ddot{\vec{t}}. \tag{12}$$

Three of these terms lead to the fictitious forces: $\dot{\vec{\omega}}_R^{\,*}\vec{r}_i$ leads to the Euler or transverse force, $\vec{\omega}_R^{\,*}\vec{\omega}_R^{\,*}\vec{r}_i$ leads to the centrifugal force, and $2\vec{\omega}_R^{\,*}RS_i\dot{\vec{q}}$ leads to the Coriolis force.

Newton's second law is

$$\mathbf{M}\mathbf{R}^{-1}\ddot{\vec{x}} = \vec{F}^{int} + \mathbf{R}^{-1}\vec{F}^{ext} \tag{13}$$

where the internal forces are defined in object space, the external forces are defined in world space, and $\mathbf{M}$ is the consistent finite element mass matrix defined in object space. Substituting Equation 12 into Equation 13, solving for $\mathbf{M}S\ddot{\vec{q}}$, and pre-multiplying both sides by S^T results in

$$\begin{aligned} S^T\mathbf{M}S\ddot{\vec{q}} = {} & S^T\vec{F}^{int} + S^T\mathbf{R}^{-1}\vec{F}^{ext} - 2S^T\mathbf{M}\mathbf{R}^{-1}\vec{\omega}_R^{\,*}\mathbf{R}S\dot{\vec{q}} \\ & - S^T\mathbf{M}\mathbf{R}^{-1}\left(\dot{\vec{\omega}}_R^{\,*} + \vec{\omega}_R^{\,*}\vec{\omega}_R^{\,*}\right)\mathbf{R}(S\vec{q} + \vec{x}_o) \end{aligned} \tag{14}$$

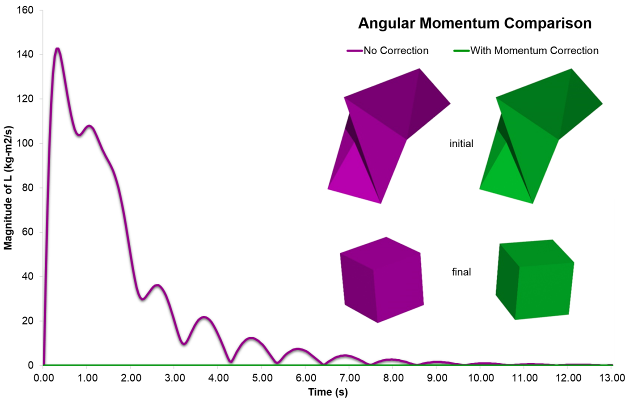

Figure 3: *A simple reduced deformable cube with six tetrahedral elements, eight particles, and a mass of 1000 kg is given a random initial reduced displacement with zero reduced velocity and embedded in a frame. We simulate the cube without and with our angular momentum correction scheme using the same initial conditions. Even though there is no angular momentum in the system at the start, the uncorrected simulation quickly accumulates significant erroneous angular momentum whereas our scheme preserves the momentum of the system exactly. Note the change in final orientations.*

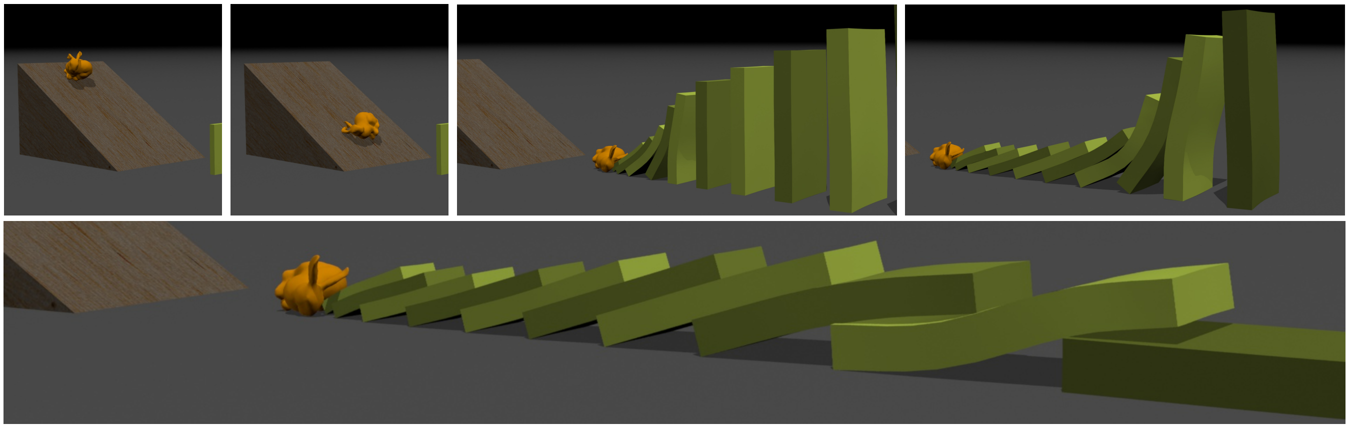

Figure 4: *A reduced deformable bunny falls and rolls down a slope, and then topples a domino set consisting of reduced deformable tiles.*

which can be solved for $\ddot{\vec{q}}$ because $S^T\mathbf{M}S$ is symmetric positive definite (SPD) as long as our examples are linearly independent. Since the mass-weighted column sums of S are $\vec{0}$, S^T projects $\mathbf{M}\mathbf{R}^{-1}\ddot{\vec{\mathbf{T}}}$ to zero as long as the row sums of $\mathbf{M}$ are equal to the individual masses of the particles. If this is not the case, one could redefine the individual particle masses as the row sums of $\mathbf{M}$ in order to satisfy this condition.

Fictitious forces that appear in Equation 14, which we collectively call $\vec{F}_r^{fict}$, can be computed efficiently by noting that $\mathbf{R}^{-1}\vec{\omega}_R^*\mathbf{R}$ and $\mathbf{R}^{-1}\left(\dot{\vec{\omega}}_R^* + \vec{\omega}_R^*\vec{\omega}_R^*\right)\mathbf{R}$ can be computed by rotating $\vec{\omega}_R^*$ and $\dot{\vec{\omega}}_R^*$ into object space.

4.1 External Forces

We sum all the external forces in order to get a net external force, which is used to update the rigid frame as

$$\dot{\vec{v}} = \frac{1}{M}\sum_{k=1}^{n} \vec{f}_k^{ext} \tag{15}$$

$$\dot{\vec{L}}_{body} = \sum_{k=1}^{n} \vec{r}_k^{\,*}\vec{f}_k^{ext} = \sum_{k=1}^{n} \vec{\tau}_k^{ext} \tag{16}$$

where $\vec{F}^{ext}$ is ideally a sparse vector and thus this summation is not $O(n)$. When putting the external forces (or impulses) into object space, one could project away the translational component beforehand

$$P\vec{F}^{ext} = \vec{F}^{ext} - \begin{pmatrix} m_1 \\ \vdots \\ m_n \end{pmatrix} \dot{\vec{v}} \tag{17}$$

where P is a projection matrix that performs this linear operation. However, this is not necessary since any net translation is already in the null space of our reduced basis. Similarly, aiming to keep as much of the rotation in R as possible, we could also project away the rotational component

$$P\vec{F}^{ext} = \vec{F}^{ext} - \begin{pmatrix} m_1(\vec{r}_1^{\,*})^T \\ \vdots \\ m_n(\vec{r}_n^{\,*})^T \end{pmatrix} I^{-1}\dot{\vec{L}}_{body}. \tag{18}$$

Generally speaking, Equation 18 is useful in the sense that we would like to minimize $\vec{\omega}'$, although it is not entirely clear that Equation 18 minimizes $\vec{\omega}'$ since one cannot readily predict the effect of $P\vec{F}^{ext}$ on the reduced basis (although it is a good start). Combining Equations 17 and 18 into a single projection P converts $\vec{F}^{ext}$ into a conservative force with no net translation or rotation.

Note that computing $S^T\mathbf{R}^{-1}P\vec{F}^{ext}$ is $O(n)$ since $P\vec{F}^{ext}$ is no longer sparse, even if $\vec{F}^{ext}$ was. However, we can first rotate the external forces into object space and then apply the equivalent appropriate projection along the lines of Equations 17 and 18 in object space, essentially switching the order of $\mathbf{R}^{-1}$ and P by modifying P. This allows us to precompute the various $O(n)$ cost terms in S^TP so that computing the projected external force in the reduced space only requires consideration of the sparse entries in $\vec{F}^{ext}$, i.e.

$$\vec{F}_r^{ext} = \sum_k \left[S^TP\right]_k R^{-1}\vec{f}_k^{ext} \tag{19}$$

where $\left[S^TP\right]_k$ is the k^{th} column of S^TP and the summation is only over all non-zero $\vec{f}_k^{ext}$.

4.2 Internal Forces

Although one could choose any valid reduced internal force model (such as [Barbič and James 2005; An et al. 2008]), for simplicity we chose a linearized finite element model

$$\vec{F}^{int} = -\mathbf{K}\vec{u} - \mathbf{C}\dot{\vec{u}} = -\mathbf{K}S\vec{q} - \mathbf{C}S\dot{\vec{q}} \tag{20}$$

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}. \tag{21}$$

To find the reduced internal force, we project these forces onto the basis by pre-multiplying by S^T

$$\vec{F}_r^{int} = -S^T\mathbf{K}S\vec{q} - S^T\mathbf{C}S\dot{\vec{q}} = -\mathbf{K}_r\vec{q} - \mathbf{C}_r\dot{\vec{q}}. \tag{22}$$

5 Collisions and Contact

Starting from the conservation of momentum laws for two bodies, we can describe the post-collision momenta in terms of the pre-collision ones

$$M_a\vec{v}_a^{\,+} + M_b\vec{v}_b^{\,+} = M_a\vec{v}_a^{\,-} + M_b\vec{v}_b^{\,-} \tag{23}$$

$$\begin{aligned}\vec{x}^*_{M_a}M_a\vec{v}_a^{\,+} + \vec{L}^+_{body,a} + \vec{x}^*_{M_b}M_b\vec{v}_b^{\,+} + \vec{L}^+_{body,b} \\ = \vec{x}^*_{M_a}M_a\vec{v}_a^{\,-} + \vec{L}^-_{body,a} + \vec{x}^*_{M_b}M_b\vec{v}_b^{\,-} + \vec{L}^-_{body,b}\,.\end{aligned} \tag{24}$$

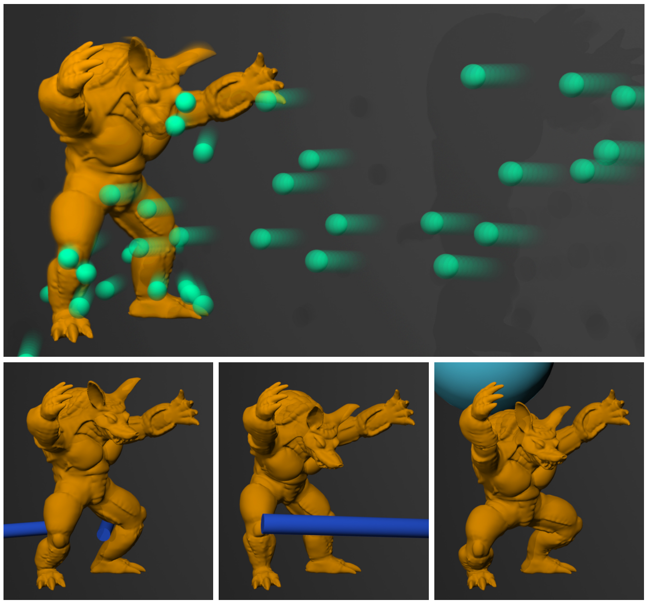

Figure 5: *We make the armadillo's hands and feet static and allow the rest of the body to deform with articulation constraints. We then hit the armadillo with various rigid bodies causing deformations in the armadillo while keeping its hands and feet in place. The hands and feet could also have been moving according to artist-specified kinematic paths.*

An impulse $\vec{j}_k$ applied to particle k changes both linear and angular momentum as follows

$$M_l \vec{v}_l^+ = M_l \vec{v}_l^- \pm \vec{j}_k \tag{25}$$

$$\begin{aligned}\vec{x}^*_{M_l} M_l \vec{v}_l^+ &+ \vec{L}^+_{body,l} \\ &= \vec{x}^*_{M_l} M_l \vec{v}_l^- + \vec{L}^-_{body,l} \pm (\vec{x}_{M_l} + \vec{r}_{k,l})^* \vec{j}_k\end{aligned} \tag{26}$$

for $l = a, b$. By substituting Equation 25 into Equation 26, we can simplify Equation 26 to

$$\vec{L}^+_{body,l} = \vec{L}^-_{body,l} \pm \vec{r}^*_{k,l} \vec{j}_k. \tag{27}$$

Using the definition of $\vec{\omega}$ in Equation 10, we can rewrite Equation 27 as

$$\vec{\omega}_l^+ = \vec{\omega}_l^- \pm I_l^{-1} \vec{r}^*_{k,l} \vec{j}_k. \tag{28}$$

Applying only a single impulse via Equation 14 allows us to write $\left(S^T \mathbf{M} S\right) \Delta\dot{\vec{q}} = \left[S^T P\right]_k R^{-1} \vec{j}_k$ with the aid of Equation 19, which we solve for $\Delta\dot{\vec{q}} = \dot{\vec{q}}^+ - \dot{\vec{q}}^-$ to obtain

$$\Delta\dot{\vec{q}} = \left[\left(S^T \mathbf{M} S\right)^{-1} S^T P\right]_k R^{-1} \vec{j}_k = A_k R^{-1} \vec{j}_k \tag{29}$$

where A_k can be precomputed. Starting from Equation 5 for $\dot{\vec{x}}_k^+$, we substitute Equation 25 for $\vec{v}^+$, Equation 28 for $\vec{\omega}^+$, and Equation 29 for $\dot{\vec{q}}^+$ to obtain

$$\begin{aligned}\dot{\vec{x}}_k^+ &= \vec{v}^- + \vec{j}_k/M + \left(\vec{\omega}^- + I^{-1} \vec{r}_k^* \vec{j}_k\right)^* \vec{r}_k \\ &+ \left[R S_k \left(\dot{\vec{q}}^- + A_k R^{-1} \vec{j}_k\right) - \left(\vec{\omega}'^- + \Delta\vec{\omega}'\right)^* \vec{r}_k\right]\end{aligned} \tag{30}$$

where the unknown change $\Delta\vec{\omega}'$ equals $\vec{\omega}'^+ - \vec{\omega}'^-$. Rearranging terms and using Equation 5 to substitute in for $\dot{\vec{x}}_k^-$, we obtain

$$\begin{aligned}\dot{\vec{x}}_k^+ &= \dot{\vec{x}}_k^- + \left(1/M - \vec{r}_k^* I^{-1} \vec{r}_k^*\right) \vec{j}_k \\ &\quad + R S_k A_k R^{-1} \vec{j}_k + \vec{r}_k^* \Delta\vec{\omega}'.\end{aligned} \tag{31}$$

Using the definition of $\vec{\omega}'$ in Equation 8 and the definition of $\Delta\dot{\vec{q}}$ in Equation 29 we can write

$$\Delta\vec{\omega}' = I^{-1} \sum_{i=1}^{n} m_i \vec{r}_i^* R S_i A_k R^{-1} \vec{j}_k. \tag{32}$$

Then, we may write

$$\dot{\vec{x}}_k^+ = \dot{\vec{x}}_k^- + (K_1 + K_2)\vec{j}_k = \dot{\vec{x}}_k^- + K\vec{j}_k \tag{33a}$$

$$K_1 = 1/M - \vec{r}_k^* I^{-1} \vec{r}_k^* \tag{33b}$$

$$K_2 = R S_k A_k R^{-1} + \vec{r}_k^* I^{-1} \sum_{i=1}^{n} m_i \vec{r}_i^* R S_i A_k R^{-1} \tag{33c}$$

where K_1 is the standard rigid body impulse factor and K_2 is the new impulse factor that contains terms for object space deformations. Note that the combined operator is still linear, so we can solve for impulses quickly just as in the rigid body case. [Hauser et al. 2003] presents an equation similar to Equation 33, although their equations are incorrect unless $\vec{\omega}' = \vec{0}$. In our examples, we have seen that the contribution to angular momentum from $\vec{\omega}'$ is quite significant and cannot be ignored.

5.1 Reduction to Rigid and Deformable Models

We stress that Equation 33 is the most general form of the impulse equation within our framework and is the superset of both the rigid and deformable cases. In the rigid case, $S = 0$, $\vec{\omega}' = \vec{0}$, and $K_2 = 0$, leaving only K_1 which is the standard impulse factor. In the case of deformable bodies, R is the identity, $\vec{t} = \vec{0}$, and S functions similar to an identity except that it packs all the scalar components of $\vec{q}$ into a vector of 3-vectors in $\vec{u}$. In addition, P is the identity and the net effect of Equation 29 is to divide $\vec{j}_k$ by m_k. Then K_2 reduces to $1/m_k + \vec{r}_k^* I^{-1} \vec{r}_k^*$. Thus the inertia tensor terms from K_1 and K_2 cancel, leaving only $1/M + 1/m_k$. At first glance, the $1/M$ seems misplaced, but this is because we have set $\vec{t}$ identically zero and have not likewise set $\vec{v}' = \vec{v}$ according to Equation 4a. Doing so introduces another term in K_2 that cancels out $1/M$ leading to the expected result.

5.2 Friction

We model friction after the method in [Guendelman et al. 2003], which first calculates the impulse required to apply static friction, then checks if it is in the friction cone via $\left\|\vec{j} - \left(\vec{j} \cdot \vec{n}\right) \vec{n}\right\|_2 \leq \mu \left(\vec{j} \cdot \vec{n}\right)$ and applies it when within this cone—otherwise applying kinetic friction. However, we have observed that the global nature of deformation in reduced deformable models means that $\vec{j} \cdot \vec{n}$ can be negative even when $\Delta\vec{v} \cdot \vec{n} = K\vec{j} \cdot \vec{n}$ is positive. This is due to the fact that our angular momentum correction can directly affect the angular velocity of a body in a nonintuitive way and is reminiscent of the Painlevé Paradox (see [Kaufman et al. 2008] for a related discussion on rigid bodies). Therefore, we modified the check from [Guendelman et al. 2003] to be $\left\|K\vec{j} - \left(K\vec{j} \cdot \vec{n}\right) \vec{n}\right\|_2 \leq \mu \left(K\vec{j} \cdot \vec{n}\right)$. That is, we check the magnitude of the resulting velocity, not the impulse. This situation illustrates a further problem with penalty forces, where one could apply a penalty force to a node in order to make it go in a certain direction but the resulting object deformation could instead push that node in the opposite direction. Of course, continually applying the force for a long enough time may eventually coax the entire object to go in the desired direction, meaning that the penalty force can work in an unconstrained scenario. However, if the object's motion

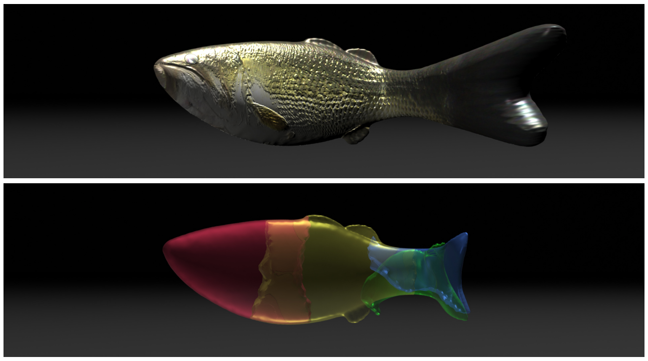

Figure 6: *The top figure shows a reduced deformable fish with 40 modes, which exhibits well-known linearized rotation artifacts from linearized finite elements. The bottom figure shows the fish segmented into four reduced deformable "bones" (shown in red, yellow, green and blue) with 10 modes each and articulated. After skinning the segmented fish along the lines of Figure 7 we obtain the high visual fidelity shown in Figures 8 and 9.*

is constrained in various ways, the penalty force may not properly influence the object to move or deform in any intuitive way whatsoever and instead just adds energy to the system.

6 Time Integration

The first step in time integration is to solve Equations 15 and 16 in order to update the velocity and total angular momentum of the rigid frame. Then, we discretize Equation 14 as

$$S^T \mathbf{M} S \dot{\vec{q}}^{\,n+1} = S^T \mathbf{M} S \dot{\vec{q}}^{\,n} + \Delta t \left(\vec{F}_r^{int} + \vec{F}_r^{ext} + \vec{F}_r^{fict} \right) \quad (34)$$

where $\vec{F}_r^{int}$ and $\vec{F}_r^{ext}$ are defined in Equations 22 and 19 respectively. We solve for $\vec{F}_r^{int}$ implicitly and use $\vec{q}^{\,n+1} = \vec{q}^{\,n} + \Delta t \dot{\vec{q}}^{\,n+1}$ to obtain

$$\begin{aligned} &\left(S^T \mathbf{M} S + \Delta t \mathbf{C}_r + (\Delta t)^2 \mathbf{K}_r \right) \dot{\vec{q}}^{\,n+1} \\ &\qquad = S^T \mathbf{M} S \dot{\vec{q}}^{\,n} - \Delta t \mathbf{K}_r \vec{q}^{\,n} + \Delta t \left(\vec{F}_r^{ext} + \vec{F}_r^{fict} \right). \quad (35) \end{aligned}$$

Once $\vec{v}^{n+1}$, $\vec{L}_{body}^{n+1}$, and $\dot{\vec{q}}^{\,n+1}$ have been updated, we calculate $\vec{\omega}'^{\,n+1}$ using Equation 8 and $\vec{\omega}_R^{n+1}$ using Equation 11. We can then proceed to update the position terms as normal

$$\vec{t}^{\,n+1} = \vec{t}^{\,n} + \Delta t \vec{v}^{n+1} \quad (36)$$

$$Q^{n+1} = \hat{Q}\left(\Delta t \vec{\omega}_R^{n+1} \right) Q^n \quad (37)$$

$$\vec{q}^{\,n+1} = \vec{q}^{\,n} + \Delta t \dot{\vec{q}}^{\,n+1} \quad (38)$$

where Q is a quaternion, Equation 37 represents the standard quaternion update rule, and R^{n+1} is retrieved from Q^{n+1}. Finally, we update the inertia tensor to I^{n+1} and re-compute both $\vec{\omega}'^{\,n+1}$ and $\vec{\omega}_R^{n+1}$. To handle collisions and contact, we use the scheme from [Guendelman et al. 2003] in conjunction with our new equations.

7 Articulation

We follow [Weinstein et al. 2006] for articulating reduced deformable bodies using both pre-stabilization and post-stabilization. For simplicity, we will only consider point joints, but note that the same ideas can be extended to other types of joints. For the sake of exposition, assume that we are merely constraining particle i on body a to particle k on body b, noting that we could constrain arbitrary points using barycentric weights (similar consideration applies to collisions in Section 5). Post-stabilization is rather straightforward. We simply apply the impulse that results from equating $\dot{\vec{x}}_i^+$ and $\dot{\vec{x}}_k^+$ using Equation 33a.

During pre-stabilization, the goal is to combat positional drift and make the positions of the two particles identical using

$$R_a \left(S_i^a \vec{q}_a + \vec{x}_{0,i}^{\,a} \right) + \vec{t}_a = R_b \left(S_k^b \vec{q}_b + \vec{x}_{0,k}^{\,b} \right) + \vec{t}_b \quad (39)$$

at time $n+1$. Since all the terms in Equation 39 are evaluated at time $n+1$, we can substitute the right hand sides of Equations 36, 37, and 38 to obtain

$$\begin{aligned} &\hat{Q}\left(\Delta t \vec{\omega}_{R_a}^{n+1} \right) Q_a^n \Big[S_i^a \Big(\vec{q}_a^{\,n} + \Delta t \dot{\vec{q}}_a^{\,n+1} \Big) + \vec{x}_{0,i}^{\,a} \Big] + \vec{t}_a^{\,n} + \Delta t \vec{v}_a^{\,n+1} = \\ &\hat{Q}\left(\Delta t \vec{\omega}_{R_b}^{n+1} \right) Q_b^n \Big[S_k^b \Big(\vec{q}_b^{\,n} + \Delta t \dot{\vec{q}}_b^{\,n+1} \Big) + \vec{x}_{0,k}^{\,b} \Big] + \vec{t}_b^{\,n} + \Delta t \vec{v}_b^{\,n+1}. \quad (40) \end{aligned}$$

We can then define the objective function that we want to make zero via impulses applied to the time $n+1$ velocities as

$$\begin{aligned} \vec{f}\left(\vec{j} \right) &= \vec{t}_a^{\,n} + \Delta t \left(\vec{v}_a^{\,n+1} + \vec{j}/M_a \right) \\ &+ \hat{Q}\left(\Delta t \left(\vec{\omega}_{R_a}^{n+1} + \left(I_a^n \right)^{-1} \left(\hat{\vec{r}}^{\,*} \vec{j} - \Delta \vec{\omega}_a'^{\,n} \right) \right) \right) Q_a^n \\ &\quad \left[S_i^a \left(\vec{q}_a^{\,n} + \Delta t \left(\dot{\vec{q}}_a^{\,n+1} + A_{a,i}^n \left(R_a^n \right)^{-1} \vec{j} \right) \right) + \vec{x}_{0,i}^{\,a} \right] \\ &- \vec{t}_b^{\,n} - \Delta t \left(\vec{v}_b^{\,n+1} - \vec{j}/M_b \right) \\ &- \hat{Q}\left(\Delta t \left(\vec{\omega}_{R_b}^{n+1} + \left(I_b^n \right)^{-1} \left(\hat{\vec{r}}^{\,*} \vec{j} - \Delta \vec{\omega}_b'^{\,n} \right) \right) \right) Q_b^n \\ &\quad \left[S_k^b \left(\vec{q}_b^{\,n} + \Delta t \left(\dot{\vec{q}}_b^{\,n+1} - A_{b,k}^n \left(R_b^n \right)^{-1} \vec{j} \right) \right) + \vec{x}_{0,k}^{\,b} \right] \quad (41) \end{aligned}$$

where $\hat{\vec{r}}$ is defined as the world space midpoint between the two time n positions and $\vec{\omega}'$ is defined using the time n quantities in Equation 32. To solve this equation we use Newton iteration, and the derivation of $\partial \vec{f} / \partial \vec{j}$ needed for the solve follows along the lines of that provided in [Weinstein et al. 2006].

The ability to articulate reduced deformable bodies directly allows us to segment (or decompose) a simulation mesh into different pieces and connect them using joints. However, since the bodies can deform but cannot deform arbitrarily, there is no guarantee

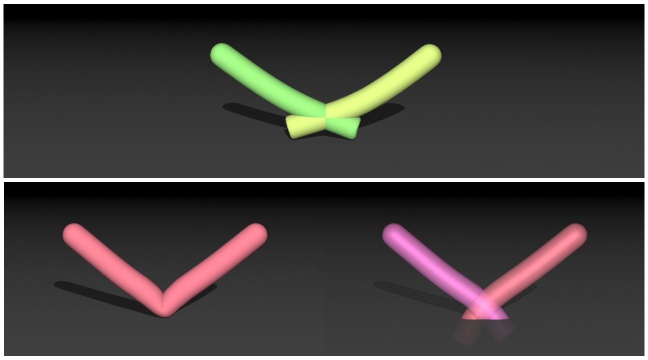

Figure 7: *Naïvely articulating together two reduced deformable rods (top) produces erroneous collision with the ground. Treating the two bodies as "bones" of a single deformable body and skinning them produces improved results (bottom left). Note that only the skin collides, and the simulated bodies do not interact with the environment (bottom right).*

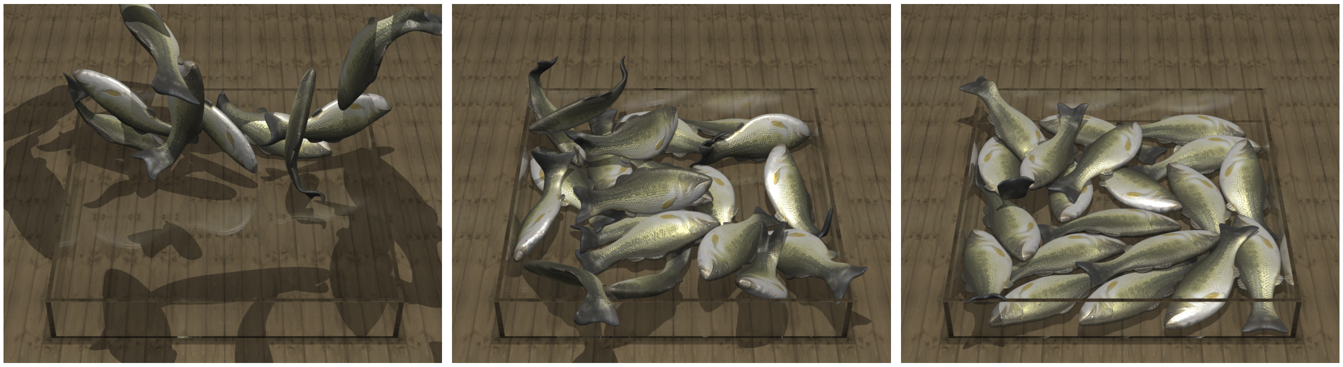

Figure 8: *Twenty reduced deformable fish are dropped into an empty glass box. Each fish is actuated by three balanced forces applied on the surface that are scaled by a unique function of time for each fish so that the fish actively flop (as seen in the video).*

that all joint constraints can be satisfied. Similar to [Barbič and Zhao 2011; Kim and James 2011], there may be discontinuities at the boundaries between articulated meshes. Our novel approach of reinterpreting each reduced deformable body as an abstract "deformable bone" with various dynamically changing shapes allows us to remedy this with a straightforward skinning-based approach.

8 Skinning

In standard skinning, one has a bone with a shape that may be a function of various parameters, and we extend this to the notion of a "deformable bone" where a traditional kinematic bone is replaced by a reduced deformable model given by Equation 1. Essentially, our "deformable bone" has a time-dependent shape based on simulation, yet is similar to existing skinning bones in every other way. Skinning already solves the problem of how to tie multiple parts of a character together, so it is very natural to leverage skinning for reduced deformable models. In addition, work on skinning has already investigated issues such as the number and placement of bones, blending operations, linear artifacts, etc., and all this be exploited in our "deformable bones" framework.

Using standard rigging software, we assign standard skinning weights to a surface mesh based on standard bones. Then, in a non-standard fashion we propagate these weights into the interior elements. Using a threshold, we decompose the mesh into different subdomains of elements associated with each bone. These thresholds allow for subdomains to overlap, which means that multiple subdomains can each replicate and incorporate elements from the original simulation mesh (i.e. there is no guarantee of a unique mapping from an element on the original mesh to a subdomain). During simulation, each subdomain is simulated independently, with the addition of about three point constraints inside overlapping regions to articulate one subdomain with another. See Figure 6 for an example decomposition.

For simplicity, we use linear blend skinning, though more complex methods could be used. The position of particle i on the skin $\vec{x}_i^s$ is given by

$$\vec{x}_i^s = \sum_k w_i^k \left(R_k(S_i^k \vec{q}_k + \vec{x}_{0,i}^k) + \vec{t}_k \right) \tag{42}$$

where k iterates through the reduced deformable bodies that contribute to the skin position of $\vec{x}_i^s$, and w_i^k is the skinning weight for particle i on body k. The velocity of a skinned particle, $\dot{\vec{x}}_i^s$, can be calculated via Equation 3. We assume that the skinning weights for particle i sum to one—standard practice for skinning.

Our proposed skinning framework readily mixes reduced deformable bodies, rigid bodies, and standard kinematic bones. This allows an artist, for example, to replace the midsection of a traditionally animated/skinned character with one of our "deformable bones" and proceed to add elastic and vibrational modes such as "belly jiggle" or even collisions—see Figure 5. In fact, motivated by the example-based work of [Martin et al. 2011; Schumacher et al. 2012; Coros et al. 2012] as well as [Koyama et al. 2012; Koyama and Igarashi 2013], one could augment the basis with example poses sculpted by the artist (such as an indentation caused by being hit in the stomach). Furthermore, this mode could be emphasized by lowering its corresponding eigenvalue. Finally, the results are readily blended back in with the artist's performance for the rest of the character. Similar results were achieved using a different framework in [Hahn et al. 2012]. Although we have only considered standard linear blend skinning, it would be quite interesting to extend our methodology to a variety of other skinning models so that a large number of artist tools (and thus artists) could be impacted.

For collisions, the skin particles follow the derivation in [Sifakis et al. 2007]. For simplicity, consider a skin particle i which depends on two bodies a and b. Then an impulse $\vec{j}$ is conservatively distributed to the bodies using $w_i^a \vec{j}$ and $w_i^b \vec{j}$ to obtain

$$\dot{\vec{x}}_i^{a+} = \dot{\vec{x}}_i^{a-} + w_i^a K_a \vec{j} \tag{43}$$

$$\dot{\vec{x}}_i^{b+} = \dot{\vec{x}}_i^{b-} + w_i^b K_b \vec{j}. \tag{44}$$

The skin particle i's new velocity is then

$$\begin{aligned} \dot{\vec{x}}_i^{s+} &= \dot{\vec{x}}_i^{s-} + w_i^a \left(w_i^a K_a \vec{j} \right) + w_i^b \left(w_i^b K_b \vec{j} \right) \\ &= \dot{\vec{x}}_i^{s-} + K_s \vec{j}. \end{aligned} \tag{45}$$

Using the new skinned particle impulse factor K_s, we can solve directly for $\vec{j}$ for use in both collisions and self-collisions. See Figure 7.

Using a standard mesh simplification algorithm that can preserve original particle positions (such as [Garland and Heckbert 1997]), we build a coarser skin proxy for use in collisions. This new mesh contains a subset of the original skinned particles with a new topology. In this way, we use geometry to simplify the collision point set instead of finding points using existing simulation data ([Teng et al. 2014]). Using a coarser collision proxy is especially useful when simulating reduced deformable models, where individual particles are not free to move on their own since the degrees of freedom are

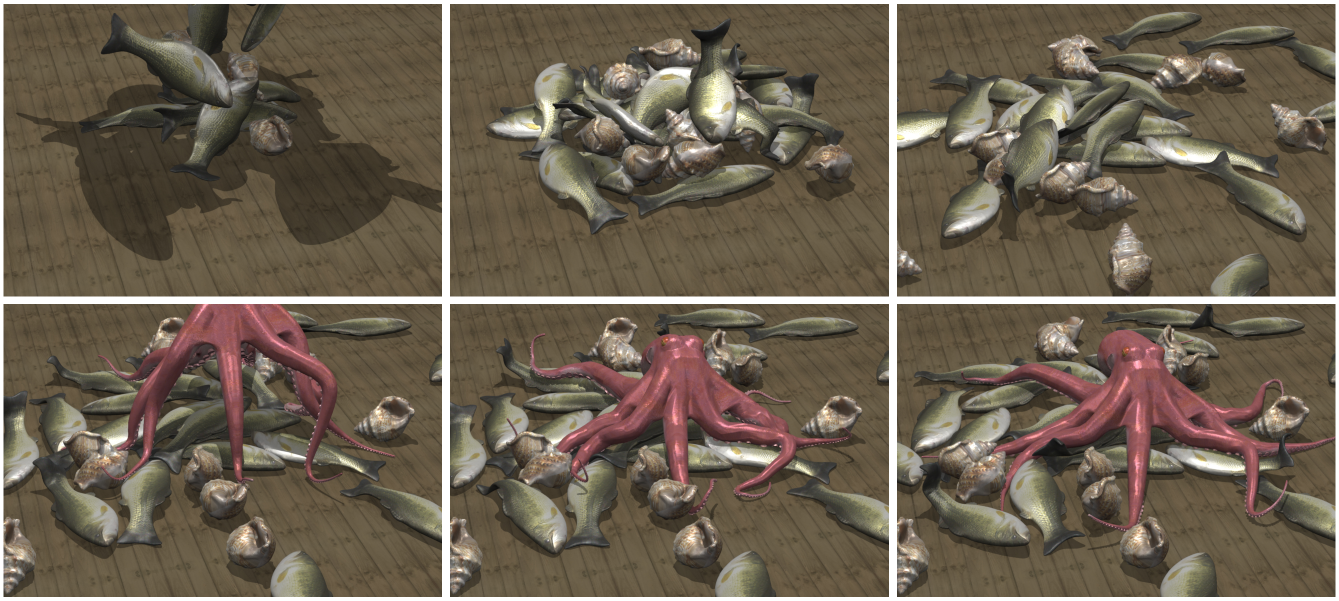

Figure 9: *Twenty-four reduced deformable fish and twelve rigid seashells are dropped onto a slightly slippery floor. After the pile settles, a reduced deformable octopus (skinned from 50 "bones") drops on top of the fish and seashells, eventually settling with its tentacles being nudged by the flopping fish tails.*

basis shapes. In addition to our collision proxies, we build a BD-tree (see [James and Pai 2004]) on the skin for further accelerations. Although skinned bodies have no intrinsic reduced degrees of freedom, we would like to express sphere radii in terms of $\vec{q}_k$ for each "bone" k for efficiency. Thus our radius bound of each skin sphere is defined as the maximum of its (maximum contained skinning weight-scaled) "bone" radii over all "bones." This bound is potentially looser than that in [James and Pai 2004]. The center updates can be losslessly accelerated by precomputing for each body k a skinning weight average of rows of S_k, $\vec{x}_{0,k}$, and $\vec{1}_k$ corresponding to each sphere's contained particles.

9 Results and Discussion

We provide several examples to illustrate the power of our method, noting that each example uses linear finite elements (i.e. linear elasticity and linear shape functions) with $r \leq 10$ unconstrained eigenmodes per reduced body unless otherwise specified. Modal analysis was performed on each reduced body individually; precomputation times and other mesh-specific details can be found in Table 1. Skeletons were created by first manually placing standard bones in the rest structures of each mesh and obtaining standard linear skinning weights using Pinocchio ([Baran and Popović 2007]). Segmentation into "deformable bones" was then done by mapping the skinning weights from the surface onto the tetrahedral volume, thresholding the weights to create a new tetrahedral volume for each segment, and subsequently re-normalizing the weights over the entire mesh. Rigid joints were created by automatically laying out three point joints in a cross section of the overlapping joint region, varying the distance among the three point joints to control rigidity of the connected "deformable bones."

	Linear FEM	Basis Gen.	BD-Tree	Other	# of Bones	# of Particles
Bunny	0.73	1.93	0.54	0.22	13	7k
Domino	0.0081	0.0151	0.0016	0.0035	1	2.7k
Armadillo	7.60	45.26	6.20	2.02	21	74k
Armadillo (Coarse)	1.14	2.63	1.76	0.39	21	19k
Fish	0.66	1.91	0.47	0.18	4	10k
Octopus	21.01	320.99	25.76	6.00	50	244k

Table 1: *Pre-computation times (in seconds) and complexity details for all of our example meshes.*

We focused on the algorithmic portions of our implementation and as such there is much room for optimization to achieve better performance, including parallelization. The timing information in Table 2 hints at the main bottleneck in our simulations, which is the reduced deformable-reduced deformable collision (and self-collision) step. This aspect of the system could be optimized by implementing or designing alternative acceleration structures for our skinned reduced deformable bodies, perhaps even using a standard kd-tree instead of a BD-tree. The costs of our inertia tensor update and angular momentum correction are negligible compared to the cost of collisions, but they could also be optimized similar to how [An et al. 2008] reduced the cost of [Barbič and James 2005]. One could also attempt less frequent updates of the inertia tensor, ignoring fictitious forces, and so on to increase speed at the expense of accuracy, but doing so may introduce instabilities into the system. Our code currently allows us to run a skinned armadillo with 30k triangles, 19k particles, 21 member "bones," and 87 point joints with a collision proxy in the Pachinko scenario at the interactive rate of about 7fps on a single core of an Intel Xeon X5680 CPU.

Although we have used linear finite elements to simplify our discussion, our method can handle non-linear materials as well via [Barbič and James 2005]. Our derivations assume a linear basis (i.e. $\vec{u} = S\vec{q}$), so we would require modal derivatives in the basis to reasonably handle non-linear materials. One could extend the ideas in this paper to even more complex subspaces, including those with multiple linear bases. We note that it is straightforward to apply our skinning framework to standard finite element models as well—for example one could use a very low degree of freedom linear full space finite element model attached to a rigid frame via Equation 1 for each "deformable bone" (or even a very coarse nonlinear finite

	# of Frames	Average Sec/Frame	# of Bones	# of Points of Articulation	# of Particles	# of Surface Particles	# of Collision Proxy Surface Particles	Self-Collision	Reduced-Reduced Collision
Bunny and 10 Dominoes	600	35.4	23	64	10k	6k	-	on	on
Armadillo and Stairs	400	32.4	21	83	74k	36k	-	on	on
Armadillo with Projectiles	500	4.2	21	83	74k	36k	-	off	off
Armadillos in a Pachinko Machine (Each Armadillo)	650	0.15	21	87	19k	15k	1536	off	off
20 Fish in a Box	480	122.4	80	300	200k	116k	5600	off	on
20 Fish in a Box	480	13.38	80	300	200k	116k	5600	off	off
24 Fish and 12 Shells	480	81.6	96	360	240k	139k	6720	off	on
Fish, Shells, and an Octopus	204	344.4	146	591	484k	240k	11762	on	on

Table 2: *Timing and mesh sizes for all of our examples, which were run using a single core of an Intel Xeon X5680 CPU. For the Pachinko example, we set collision, contact, and poststabilization iterations to 1 while setting the iterative articulation tolerance to 0.1. We set collision, contact, and poststabilization iterations to 5 and our iterative articulation tolerance to 1e-7 for all other examples.*

element model). A better friction model than the one we chose to use could also be investigated in future work.

10 Conclusion

We have presented a general framework for simulating reduced deformable bodies (each may specialize to be rigid or fully deformable) that conserves linear and angular momentum by compensating for the often overlooked change in angular momentum due to changes in the reduced degrees of freedom. Enforcing momentum conservation allows us to achieve plausible and stable simulations of unconstrained bodies. We have developed a scheme for impulse-based control of individual particle velocities to achieve physically accurate collisions, contact, and point position articulation. Our skinning of reduced deformable "bones," when combined with articulation, achieves domain decomposition and produces detailed and directable local deformations.

11 Acknowledgements

R.S. thanks Tony DeRose, Mark Meyer, and Forrester Cole at Pixar Animation Studios for relevant and insightful discussions. The authors also thank Like Gobeawan and Zahid Hossain for their help. R.S. was supported in part by the Department of Energy Office of Science Graduate Fellowship Program (DOE SCGF), made possible in part by the American Recovery and Reinvestment Act of 2009 and administered by ORISE-ORAU under contract no. DE-AC05-06OR23100. Y.Y. was supported by a Stanford Graduate Fellowship. This research was supported in part by ONR N00014-13-1-0346, ONR N00014-11-1-0707, ARL AHPCRC W911NF-07-0027, and the Intel Science and Technology Center for Visual Computing. Computing resources were provided in part by ONR N00014-05-1-0479.

References

An, S. S., Kim, T., and James, D. L. 2008. Optimizing cubature for efficient integration of subspace deformations. In *ACM SIGGRAPH Asia 2008 Papers*, ACM, New York, NY, USA, SIGGRAPH Asia '08, 165:1–165:10.

Baran, I., and Popović, J. 2007. Automatic rigging and animation of 3d characters. *ACM Trans. Graph. 26*, 3.

Barbič, J., and James, D. 2005. Real-time subspace integration of St. Venant-Kirchhoff deformable models. *ACM Trans. Graph. (SIGGRAPH Proc.) 24*, 3, 982–990.

Barbič, J., and James, D. 2010. Subspace self-collision culling. In *Proc. of ACM SIGGRAPH 2010*, 81:1–81:9.

Barbič, J., and Zhao, Y. 2011. Real-time large-deformation substructuring. *ACM Trans. on Graphics (SIGGRAPH 2011) 30*, 4, 91:1–91:7.

Choi, M. G., and Ko, H.-S. 2005. Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Trans. Viz. Comput. Graph. 11*, 91–101.

Coros, S., Martin, S., Thomaszewski, B., Schumacher, C., Sumner, R., and Gross, M. 2012. Deformable objects alive! *ACM Trans. Graph. 31*, 4 (July), 69:1–69:9.

Fan, Y., Litven, J., Levin, D. I. W., and Pai, D. K. 2013. Eulerian-on-lagrangian simulation. *ACM Trans. Graph. 32*, 3 (July), 22:1–22:9.

Garland, M., and Heckbert, P. S. 1997. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '97, 209–216.

Gilles, B., Bousquet, G., Faure, F., and Pai, D. K. 2011. Frame-based elastic models. *ACM Trans. Graph. 30*, 2 (Apr.), 15:1–15:12.

Guendelman, E., Bridson, R., and Fedkiw, R. 2003. Nonconvex rigid bodies with stacking. *ACM Trans. Graph. 22*, 3, 871–878.

Hahn, F., Martin, S., Thomaszewski, B., Sumner, R., Coros, S., and Gross, M. 2012. Rig-space physics. *ACM Trans. Graph. 31*, 4 (July), 72:1–72:8.

Harmon, D., and Zorin, D. 2013. Subspace integration with local deformations. *ACM Trans. Graph. 32*, 4 (July), 107:1–107:10.

Hauser, K. K., Shen, C., and O'Brien, J. F. 2003. Interactive deformation using modal analysis with constraints. In *Graphics Interface*, A K Peters, CIPS, Canadian Human-Computer Commnication Society, 247–256. ISBN 1-56881-207-8, ISSN 0713-5424.

Idelsohn, S. R., and Cardona, A. 1985. A reduction method for nonlinear structural dynamic analysis. *Computer Methods in Applied Mechanics and Engineering 49*, 3, 253 – 279.

James, D., and Pai, D. 2004. Bd-tree: Output sensitive collision detection for reduced deformable models. *ACM Trans. Graph. (SIGGRAPH Proc.) 23*, 393–398.

Kaufman, D., Sueda, S., James, D., and Pai, D. 2008. Staggered projections for frictional contact in multibody systems. *ACM Transactions on Graphics (SIGGRAPH Asia 2008) 27*, 5, 164:1–164:11.

Kavan, L., and Žára, J. 2005. Fast collision detection for skeletally deformable models. *Computer Graphics Forum 24*, 3, 363–372.

Kavan, L., Collins, S., Žára, J., and O'Sullivan, C. 2007. Skinning with dual quaternions. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '07, 39–46.

Kim, T., and James, D. L. 2011. Physics-based character skinning using multi-domain subspace deformations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '11, 63–72.

Koyama, Y., and Igarashi, T. 2013. View-dependent control of elastic rod simulation for 3d character animation. In *Symposium on Computer Animation*, ACM, J. Chai, Y. Yu, T. Kim, and R. W. Sumner, Eds., 73–78.

Koyama, Y., Takayama, K., Umetani, N., and Igarashi, T. 2012. Real-time example-based elastic deformation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '12, 19–24.

Kry, P., James, D., and Pai, D. 2002. Eigenskin: real time large deformation character skinning in hardware. In *Proc. of the ACM SIGGRAPH Symp. on Comput. Anim.*, ACM Press, 153–159.

Lall, S., Krysl, P., and Marsden, J. E. 2003. Structure-preserving model reduction for mechanical systems. *Physica D: Nonlinear Phenomena 184*, 1, 304–318.

Lewis, J., Cordner, M., and Fong, N. 2000. Pose space deformations: A unified approach to shape interpolation a nd skeleton-driven deformation. *Comput. Graph. (SIGGRAPH Proc.)*, 165–172.

Liu, L., Yin, K., Wang, B., and Guo, B. 2013. Simulation and control of skeleton-driven soft body characters. *ACM Trans. Graph. 32*, 6, 215:1–215:8.

Martin, S., Thomaszewski, B., Grinspun, E., and Gross, M. 2011. Example-based elastic materials. In *ACM SIGGRAPH 2011 papers*, ACM, New York, NY, USA, SIGGRAPH '11, 72:1–72:8.

Metaxas, D., and Terzopoulos, D. 1992. Dynamic deformation of solid primitives with constraints. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '92, 309–312.

Pentland, A., and Williams, J. 1989. Good vibrations: modal dynamics for graphics and animation. *Comput. Graph. (Proc. SIGGRAPH 89) 23*, 3, 215–222.

Schumacher, C., Thomaszewski, B., Coros, S., Martin, S., Sumner, R., and Gross, M. 2012. Efficient simulation of example-based materials. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '12, 1–8.

Shabana, A. 2005. *Dynamics of Multibody Systems*. Cambridge University Press.

Sifakis, E., Shinar, T., Irving, G., and Fedkiw, R. 2007. Hybrid simulation of deformable solids. In *Proc. of ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 81–90.

Teng, Y., Otaduy, M. A., and Kim, T. 2014. Simulating articulated subspace self-contact. *ACM Trans. Graph. 33*, 4 (July), 106:1–106:9.

Terzopoulos, D., and Witkin, A. 1988. Physically based models with rigid and deformable components. In *Graph. Interface*, 146–154.

Weinstein, R., Teran, J., and Fedkiw, R. 2006. Dynamic simulation of articulated rigid bodies with contact and collision. *IEEE TVCG 12*, 3, 365–374.