Subspace Dynamic Simulation Using Rotation-Strain Coordinates

Zherong Pan Hujun Bao* Jin Huang*
State Key Lab of CAD&CG, Zhejiang University

Abstract

In this paper, we propose a full featured and efficient subspace simulation method in the rotation-strain (RS) space for elastic objects. Sharply different from previous methods using the rotation-strain space, except for the ability to handle non-linear elastic materials and external forces, our method correctly formulates the kinetic energy, centrifugal and Coriolis forces which significantly reduces the dynamic artifacts. We show many techniques used in the Euclidean space methods, such as modal derivatives, polynomial and cubature approximation, can be adapted to our RS simulator. Carefully designed experiments show that the equation of motion in RS space has less non-linearity than its Euclidean counterpart, and as a consequence, our method has great advantages of lower dimension and computational complexity than state-of-the-art methods in the Euclidean space.

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

Keywords: elastic animation, model reduction

1 Introduction

In computer graphics, modeling elastic bodies undergoing large deformation is of great interest. Denoting $\mathbf{q}:\Omega\times\mathbb{R}\to\mathbb{R}^3$ as the deformation offset that takes a point p in the rest shape Ω to its offset $\mathbf{q}(p,t)$ at time t, it is well-known that the dynamics of such system is mainly characterized by the pose dependent potential energy $V(\mathbf{q})$ and velocity dependent kinetic energy $T(\dot{\mathbf{q}})$. Considering the complexity of the shape and its motion, $\mathbf{q}$ is usually discretized on a tetrahedral mesh with large number of nodes N and elements T . In other words, the configuration $\mathbf{q}$ is discretized as $|N|$ node offsets $\mathbf{u}_i(t)\in\mathbb{R}^3$. Conventional FEM method uses the high dimensional vector $\mathbf{u}(t)\in\mathbb{R}^{3|N|}$ composed of $\mathbf{u}_i(t)$ as discrete configuration space. However, two problems leading to the hurdle of efficiency are still challenging.

First, solving a high dimensional equation of motion is computationally too demanding. This motivates a lot of works about dimension reduction, which restricts the deformation in a linear subspace as $\mathbf{u}(t)=\sum_{i=1}^{|\mathbf{B}|}\mathbf{B}_i\mathbf{x}_i(t)$ with $|\mathbf{B}|\ll|N|$ basis vectors $\mathbf{B}_i\in\mathbb{R}^{3|N|}$ and the corresponding coordinates $\mathbf{x}_i(t)\in\mathbb{R}$ and turns the equation of motion into a low dimensional one with respect to $\mathbf{x}(t)\in\mathbb{R}^{|\mathbf{B}|}$. Among the early works following this idea is the neat theory of linear modal analysis [Pentland and Williams 1989]. Being simple and fast, these methods cannot represent large displacements and thus [Barbič and James 2005] proposed to use more basis vectors from modal derivatives or even non-physical ones via mass-PCA to cover the prominent deformations. Although one can always expand the subspace by adding more basis vectors for lower reduction error, the benefits drop down quickly.

Figure 1: *We propose a novel deformable model capable of large deformation in a very small configuration space. The state variables can be efficiently related to Euclidean positions allowing us to handle all kinds of constraints and external forces. In this example, we simulate 8 deformable letters undergoing large deformation caused by collision and contact with only* 10 *RS space basis vectors and* 65 *Euclidean space basis vectors for each letter.*

This way becomes even less economic if the algorithm has high complexity in terms of $|\mathbf{B}|$. Unfortunately, this is usually the case because the potential energy V is non-quadratic in $\mathbf{q}$ and its linearly discretized counterpart $\mathbf{u}$ and $\mathbf{x}$. For example, the overhead of frame scales like $O(|\mathbf{B}|^4)$ using [Barbič and James 2005] or $O(|\mathbf{B}|^2|\mathcal{C}|)$ using [An et al. 2008], where $|\mathcal{C}|$ is the number of cubature points used to approximate the non-linear reduced forces.

In this paper we argue that presenting the configuration in the Rotation-Strain space (RS space) [Huang et al. 2011] has great advantages. Specifically, a deformation $\mathbf{q}$ in the Euclidean space can be non-linearly mapped into RS space as $\tilde{\mathbf{q}}$, which can be reversely mapped back into Euclidean space. We reformulate the potential and kinetic energy with respect to $\tilde{\mathbf{q}}$, leading to a new equation of motion in the RS space, which can be similarly discretized and reduced as the coordinates in the RS space, i.e. RS coordinates. As a first advantage, the prominent deformations can be much more efficiently reduced into a low dimensional space. In typical cases, to achieve comparable reduction error under $|\mathbf{B}|$ Euclidean space basis vectors, approximately $\sqrt{2|\mathbf{B}|}$ RS space basis vectors are enough. Besides, the non-linearity in the equation of motion can be largely replaced by the non-linear map between $\mathbf{q}$ and $\tilde{\mathbf{q}}$. For instance, the

*Corresponding authors: hj@cad.zju.edu.cn, bao@cad.zju.edu.cn

potential energy can be simply quadratic even for large deformation. Although our method pays the cost of having a non-quadratic kinetic energy in RS space, we still achieve lower computational complexity, which is the second advantage.

It is worth mentioning that previous methods using RS coordinates either simply view it as a geometric post-warping or wrongly keep the kinetic energy quadratic with respect to the RS space velocity. As a consequence, these methods are built on an incomplete equation of motion in RS space and are restricted to linear elasticity or applications where dynamic error can be tolerated and external forces are absent.

1.1 Contribution

This paper proposes the first complete equation of motion in the RS space and derives a full featured elastic simulation method with the correct kinetic energy, the ability of handling external forces, collisions and non-linear materials. We show that applying existing techniques, such as modal derivatives for basis construction, rational and cubature approximation for acceleration, our method can achieve comparable accuracy with lower dimension and complexity than its Euclidean counterpart.

2 Related work

Physically based modelling of elastic deformation in computer graphics can be dated back to [Terzopoulos et al. 1987]. Supported by solid mechanics and finite element analysis, the method has been well established and understood. Vast improvements have been made over time by [Hirota et al. 2000; Capell et al. 2002; Müller and Gross 2004; Irving et al. 2006] etc., tailoring it to various graphics applications. All these methods use large number of nodes making them largely unavailable to interactive applications.

On the other hand, subspace methods have been recognized as an efficient tool for modeling salient global deformation. A simple but efficient method of this kind is linear modal analysis [Pentland and Williams 1989; James and Pai 2002; Hauser et al. 2003]. It runs on a small set of global basis vectors which are linearly related to Euclidean space. Our key observation here is that it is exactly this linear relationship that prevented LMA from large deformation. If such relationship is to be kept, [Barbič and James 2005] showed that it is still possible to model non-linear deformations if one uses a larger subspace and precomputes the coefficients of high order polynomials defining the stiffness matrix. But since the online evaluation scales like $\mathcal{O}(|\mathbf{B}|^4)$, the method can only handle a subspace with $|\mathbf{B}| \leq 30$ in real time. Like [Barbič and James 2005], one implementation of our method depends on high order polynomials but the evaluation scales like $\mathcal{O}(|\mathbf{B}|^3)$ which is more efficient. Recently, subspace methods have been greatly enhanced to handle local deformations [Harmon and Zorin 2013] and extended to incorporate other physical models such as fluid [Kim and Delaney 2013; Ando et al. 2015].

There are two other closely related works, [Choi and Ko 2005] and [Huang et al. 2011]. They are both built on linear modal analysis but can approximate non-linear deformations. Their efficacy has been exploited in [Barbič et al. 2012] for interactive animation editing. More recently, [Li et al. 2014; Schulz et al. 2014] proposed methods to accurately handle position constraints under this framework. However, these methods result in severe dynamic errors in case of large deformations or high velocity as shown in this paper. Another pitfall is that they cannot capture non-linear mode coupling, although the Poisson reconstruction of [Huang et al. 2011] is a global procedure. If simply viewing RS method as a geometric post warping applied to the Euclidean space linear modal analysis, the dynamics cannot be framed into an equation of motion which is the true cause of above problems. We show that by viewing the RS coordinates as the generalized coordinates, these problems can be easily addressed. We further enhance the RS method by coupling it with floating frame [Terzopoulos et al. 1987]. The dynamics of the whole system including centrifugal and Coriolis forces is simulated via the variational formulation [Martin et al. 2011; Hahn et al. 2012; Bouaziz et al. 2014].

More recently, subspace methods have witnessed greater flexibility with the introduction of cubature optimization [An et al. 2008]. Contrary to [Barbič and James 2005], the method enables efficient evaluation of approximate reduced forces for arbitrary potential energy including many hyperelastic materials such as Mooney-Rivlin [Rivlin 1948] and Arruda-Boyce [Arruda and Boyce 1993]. This flexibility can actually be exploited in our method by accelerating the evaluation of non-linear mass matrix using a cubature approximation. Furthermore, we will show that, to achieve comparable accuracy with [An et al. 2008], fewer cubature points would be needed on the kinetic side. The idea of cubature optimization [An et al. 2008] can be further exploited here to replace linear elasticity with other hyperelastic materials. Moreover, we show that the induced non-linear potential term in our case can actually be partially linearized, making online cubature evaluation more efficient. We test this idea on Fung's hyperelastic material [Fung 1981] as an example.

In principle, any cubature optimizer can be used with our method, of which the state-of-the-art method is the NN-HTP solver [von Tycowicz et al. 2013]. Being much faster and more accurate than [An et al. 2008], their method requires user to estimate the number of cubature points $|\mathcal{C}|$ as cardinality constraint. This can often result in either too many or too few points than is actually needed. In view of this, we propose an alternative cubature solver based on the Alternating Direction Method of Multiplier (ADMM) [Yang and Zhang 2011] in Appendix A. Unlike [von Tycowicz et al. 2013], our method tries to find a set $\mathcal{C}$ as small as possible under a user specified error bound.

3 Rationale of RS Space

In the RS space [Huang et al. 2011], the configuration $\tilde{\mathbf{q}}$ is represented as a field composed of a 3×3 antisymmetric part ω and a symmetric part ϵ for rotation and strain respectively (or 3 rotations + 3 shears + 3 compressions/extensions). We use $\tilde{\nabla}$ to denote the (offset) deformation gradient evaluated from ω, ϵ, i.e.:

$$\tilde{\nabla}(\omega(p), \epsilon(p)) = \exp(\omega(p))(\epsilon(p) + \mathrm{Id}) - \mathrm{Id}\,. \tag{1}$$

Solving ω, ϵ from the following equation gives a simple mapping from the Euclidean space to the RS space:

$$\nabla\mathbf{q}(p) = \tilde{\nabla}(\omega(p), \epsilon(p)). \tag{2}$$

To reconstruct $\mathbf{q}$ from $\tilde{\mathbf{q}}$, they solve the following Poisson problem with position constraints to eliminate global translation:

$$\min_{\mathbf{q}} \int_{\Omega} \|\nabla\mathbf{q}(p) - \tilde{\nabla}(\omega(p), \epsilon(p))\|^2 dp. \tag{3}$$

As shown in [Huang et al. 2011], $\mathbf{q}$ and $\tilde{\mathbf{q}}$ can be simply discretized as full space coordinates $\mathbf{u} \in \mathbb{R}^{3|N|}$ and $\tilde{\mathbf{u}} \in \mathbb{R}^{9|T|}$ respectively on a tetrahedral mesh with $|N|$ nodes and $|T|$ elements. For efficiency, the bases $\mathbf{B}$ and $\tilde{\mathbf{U}}$ reduce the full space coordinates as $\mathbf{u} = \mathbf{B}\mathbf{x}$ and $\tilde{\mathbf{u}} = \tilde{\mathbf{U}}\tilde{\mathbf{x}}$ respectively [Li et al. 2014], and result in the following reduced version of Equation 3:

$$\mathbf{min}_{\mathbf{x}} \sum_{i=1}^{|T|} \|\nabla_i(\mathbf{B}\mathbf{x}) - \tilde{\nabla}_i(\tilde{\mathbf{U}}\tilde{\mathbf{x}})\|^2 |T_i|, \tag{4}$$

where ∇_i and $\tilde{\nabla}_i$ compute the deformation gradient in the i-th element from the Euclidean and RS coordinates respectively. Since ∇_i is a linear operator, the solution to this minimization can be abbreviated in matrix notation:

$$\mathbf{x}(\tilde{\mathbf{x}}) = (\mathbf{B}^T \nabla^T D \nabla \mathbf{B})^{\dagger} \mathbf{B}^T \nabla^T D \tilde{\nabla}(\tilde{\mathbf{x}}) \triangleq \Phi \tilde{\nabla}(\tilde{\mathbf{x}}), \qquad (5)$$

with ∇ assembled from ∇_i and $\tilde{\nabla}$ from $\tilde{\nabla}_i$. D is a diagonal matrix assembled from volume of elements $|T_i|$. The superscript † denotes matrix inverse restricted to a set of user provided position constraints. This complication can be removed if $\mathbf{B}$ satisfies these position constraints by itself. We take this assumption and use $^{-1}$ in place of † hereafter. More details about this reduced RS method can be found in [Li et al. 2014].

4 The Dynamics in RS Space

In this section, we first present our continuous and discrete version of RS-space kinetic and potential energies. They are then plugged into a timestepping equation in its variational form (see Section 4.2). The whole procedure is illustrated in Appendix C.

All physically based models of deformable body are governed by Lagrangian mechanics. Given their respective kinetic part T and potential part V:

$$T(\dot{\mathbf{q}}) = \int_{\Omega} \frac{1}{2} \rho(p) \|\dot{\mathbf{q}}(p)\|^2 dp, \quad V(\mathbf{q}) = \int_{\Omega} W(\nabla \mathbf{q}(p)) dp,$$

the locus of motion can be found from the Euler-Lagrange equation, where $\rho(p), \dot{\mathbf{q}}(p)$ are the density and velocity at point p respectively. $W(\nabla \mathbf{q}(p))$ is the potential energy density at point p with a specific elastic material W, which is usually non-linear.

After discretization and reduction,these two energies can be simply represented in terms of $\mathbf{x}, \dot{\mathbf{x}}$:

$$T(\dot{\mathbf{x}}) = \frac{1}{2} \dot{\mathbf{x}}^T \mathbf{M} \dot{\mathbf{x}}, \quad V(\mathbf{x}) = \sum_{i=1}^{|T|} W(\nabla_i(\mathbf{B}\mathbf{x})) |T_i|.$$

Since the potential energy is usually only related to the rotation invariant strain part ϵ as is the case with corotated linear elasticity, one remarkable feature of RS space dynamics is that the potential energy can be largely linearized since ϵ is encoded separately:

$$V(\tilde{\mathbf{x}}) = \sum_{i=1}^{|T|} \tilde{W}(\epsilon_i) |T_i|, \qquad (6)$$

where ϵ_i is the strain part in $(\tilde{\mathbf{U}}\tilde{\mathbf{x}})_i$, and $\tilde{W}$ has much less non-linearity than W. For corotated elastic material, $\tilde{W}$ is actually quadratic.

Viewing the $\tilde{\mathbf{x}}$ as a generalized coordinates, one can easily find that in all the previous RS related methods, $\tilde{\mathbf{x}}$ are indeed governed by an equation of motion with constant mass and stiffness matrices. In other words, they take a simple quadratic kinetic energy in terms of $\dot{\tilde{\mathbf{x}}}$. Similar to the linear elastic simulation in the Euclidean space that uses a quadratic potential energy, i.e. linear elastic forces, such a brute linearization in RS space results in the same well-known defects: mode coupling is left out and large error. But this time, the problem comes from the improper linearization of kinetic energy, instead of the potential one. In essence, the function $\mathbf{x}(\tilde{\mathbf{x}})$ is viewed in these works as a geometric post-warping instead of a transformation between two configuration spaces. Indeed, it is easy to correct this problem. Mathematically, the transformation $\mathbf{x}(\tilde{\mathbf{x}})$ can be directly formulated into the kinetic energy:

$$T(\tilde{\mathbf{x}}, \dot{\tilde{\mathbf{x}}}) = \frac{1}{2} \dot{\mathbf{x}}(\tilde{\mathbf{x}})^T \mathbf{M} \dot{\mathbf{x}}(\tilde{\mathbf{x}}), \qquad (7)$$

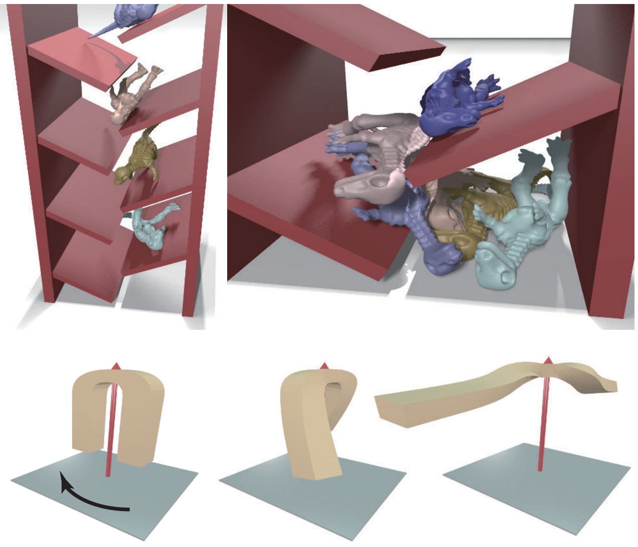


Figure 2: *RS coordinates augmented by rigid DOFs. Top: multiple bodies simulated with contact and collision. Bottom: A U-Shaped beam with an initial angular velocity, driven by centrifugal forces. The beam won't even deform when these forces are ignored.*

which combined with Equation 6 leads to a Lagrangian in RS space that solves the above problem. It is significantly different from the Lagrangian in the Euclidean space in that the non-linearity in the potential energy is largely shifted into the kinetic part. For linear elastic material, this formulation actually exchanges the linear and non-linear properties of these two energies.

4.1 RS Coordinates in Floating Frame

For modeling free-flying objects, additional degrees of freedom $\mathbf{t}, \Theta$ (floating frame) have to be introduced, where $\mathbf{t}$ is the 3×1 global translation left out by the Poisson reconstruction in Equation 3 and Θ is the 3×1 global rotation. With this extended configuration space $(\tilde{\mathbf{x}}, \mathbf{t}, \Theta)$, $\mathbf{u}_i$ now takes a more involved form:

$$\mathbf{u}_i(t) = \exp(\Theta(t))(\mathbf{B}\mathbf{x}(\tilde{\mathbf{x}}(t)) + \mathbf{r})_i + \mathbf{t}(t) - \mathbf{r}_i, \qquad (8)$$

where $\mathbf{r}$ is the rest shape vertex positions. With these components combined, we can model multiple bodies attached to floating frame under contact and collision, see Figure 2. Note that if $\mathbf{B} = \mathrm{Id}$, Θ is not needed as in [Huang et al. 2011]. But it is generally necessary for a subspace simulation.

4.2 Timestepping Scheme

Given the continuous relationship $\dot{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}} \dot{\tilde{\mathbf{x}}}$, our novel kinetic energy can be discretized as $T = \frac{1}{2} \dot{\tilde{\mathbf{x}}}^T \tilde{\mathbf{M}} \dot{\tilde{\mathbf{x}}}$, where $\tilde{\mathbf{M}} = \frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}}^T \mathbf{M} \frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}}$ and $\mathbf{M} = \frac{\partial^2 T}{\partial \dot{\mathbf{x}}^2}$ are the mass matrices in RS and Euclidean space respectively. A naive timestepping equation can be derived from the Euler-Lagrange equation:

$$\frac{d}{dt}(\tilde{\mathbf{M}} \dot{\tilde{\mathbf{x}}}) + \frac{\partial V}{\partial \tilde{\mathbf{x}}} - \frac{\partial T}{\partial \tilde{\mathbf{x}}} = \mathbf{0}. \qquad (9)$$

However, the terms of $\frac{d}{dt}(\tilde{\mathbf{M}} \dot{\tilde{\mathbf{x}}})$ and $\frac{\partial T}{\partial \tilde{\mathbf{x}}}$ would involve high order derivatives of $\mathbf{x}(\tilde{\mathbf{x}})$, which are costly to evaluate. On the other hand, large dynamic error would be introduced, if one simply assume $\tilde{\mathbf{M}}$ constant in each time step (see Figure 3). To avoid their costly evaluation, we first look into the timestepping equation in the Euclidean space:

$$\frac{\mathbf{M}}{\Delta t^2}(\mathbf{x}_{n+1}-\mathbf{x}_n-\dot{\mathbf{x}}_n\Delta t)+\frac{\partial V}{\partial \mathbf{x}_{n+1}}=\mathbf{0}.$$

As noted in [Martin et al. 2011], this has an equivalent variational form:

$$\mathbf{argmin}_{\mathbf{x}_{n+1}} V(\mathbf{x}_{n+1})+\frac{1}{2}\|\mathbf{x}_{n+1}-\mathbf{x}_n-\dot{\mathbf{x}}_n\Delta t\|^2_{\mathbf{M}/\Delta t^2}.$$

We follow Equation 6 to replace $V(\mathbf{x}_{n+1})$ directly with $V(\tilde{\mathbf{x}}_{n+1})$, and use $\mathbf{x}(\tilde{\mathbf{x}}_{n+1})$ in place of $\mathbf{x}_{n+1}$ via Equation 5 for kinetic term only. After differentiation, we reach our final timestepping equation:

$$\frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}_{n+1}}^T \frac{\mathbf{M}}{\Delta t^2}(\mathbf{x}(\tilde{\mathbf{x}}_{n+1})-\mathbf{x}_n-\dot{\mathbf{x}}_n\Delta t)+\frac{\partial V}{\partial\tilde{\mathbf{x}}_{n+1}}=\mathbf{0}. \qquad (10)$$

Note that although we present our method using Backward Euler integrator for simplicity, other schemes are also available by first writing them in Euclidean space and then mapping the variational formulation into RS space. We solve this non-linear equation using Newton's method with the approximated Hessian evaluated as:

$$\frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}_n}^T\frac{\mathbf{M}}{\Delta t^2}\frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}_n}+\frac{\partial^2 V}{\partial \tilde{\mathbf{x}}_n^2}.$$

Just like Quasi-Newton or inexact Newton method, it slows down the convergence, but will reduce the cost of each iteration significantly without affecting the final solution. In all our experiments, we limit the number of iterations to 3 for fixed bodies and 10 for free-flying bodies.

This scheme has a notable advantage when applied to free-flying objects. As shown in Appendix B, there are non-diagonal entries in the mass matrix respect to the extended configuration space $(\tilde{\mathbf{x}},\mathbf{t},\Theta)$, which implicitly takes charge of centrifugal and Coriolis terms. Ignoring such entries, the floating frame will has large drifting. It is worth mentioning that our method properly captures non-linearity of the inertial force term in Equation 10. If one approximates this term to be linear to the "tangent mass matrix", i.e. $\left(\frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}_n}^T\frac{\mathbf{M}}{\Delta t^2}\frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}_n}\right)$, the accuracy in both deformation and the floating frame would be lost (see the inset).

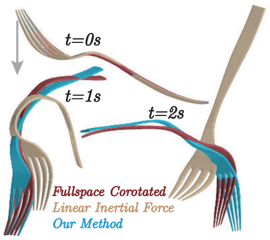


Figure 3: *Fork in a floating frame with strong forces applied on one end. We observed large deviation from groundtruth, if one uses linear inertial force approximation in each time step.*

5 Modal Reduction and Optimization

Our RS space method is especially suitable for reduced simulation. We first present a suitable choice of the basis set $\mathbf{B}$ and $\tilde{\mathbf{U}}$ in Section 5.1. For efficient simulation, two approximations to Equation 5 are proposed in Section 5.2.

5.1 Choice of Basis vectors

Theoretically, any set of basis vectors can fit into the above timestepping equation. The simplest strategy is to set $\tilde{\mathbf{U}},\mathbf{B}$ to identity, which results in common high dimensional "full space" discretization and low performance simulation. In this case, $\mathbf{x}$ can indeed be viewed as the "full space" coordinates of size $3|N|$. For better efficiency, one can use much fewer basis vectors. Since it has been shown in [Li et al. 2014] that $\tilde{\mathbf{x}}$ can be embedded in a much smaller subspace than $\mathbf{x}$, we propose to start with a few number of bases $\mathbf{U}$ coming from linear modal analysis or mass-PCA, and convert them into the RS space as $\tilde{\mathbf{U}}$, then expand $\mathbf{U}$ into $\mathbf{B}$. In order to derive $\mathbf{B}$, a classical method is load dependent basis [Idelsohn and Cardona 1985], which is introduced into graphics community in [Barbič and James 2005] as modal derivative.

The idea of [Idelsohn and Cardona 1985] is interesting in that it allows the basis set $\mathbf{B}$ to reflect the changing nature of the stiffness matrix. However, since the non-linearity has been shifted to the kinetic side, the stiffness matrix is constant in our case so that applying their idea would trivially lead to $\mathbf{B}=\mathbf{U}$. Instead, we want $\mathbf{B}$ to reflect the changing nature of the mass matrix $\tilde{\mathbf{M}}$. Based on the observation that the non-linearity comes from the transformation function $\mathbf{u}(\tilde{\mathbf{x}})$, an idea similar to [Idelsohn and Cardona 1985] can be readily applied. Specifically, we derive basis vectors in $\mathbf{B}$ from first and second order derivatives of $\mathbf{u}$ against $\tilde{\mathbf{x}}$ at origin:

$$\mathbf{B}=\left[(\nabla^T D\nabla)^{-1}\nabla^T D\frac{\partial\tilde{\nabla}}{\partial\tilde{\mathbf{x}}_i}(\mathbf{0}),(\nabla^T D\nabla)^{-1}\nabla^T D\frac{\partial\tilde{\nabla}^2}{\partial\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_j}(\mathbf{0})\right],$$

for $i\leq j$. The basis set constructed from our method is illustrated in Figure 4.

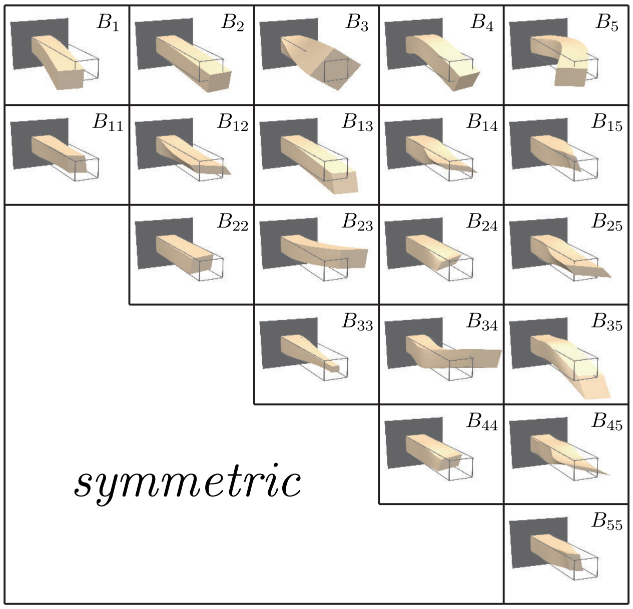


Figure 4: *Extended basis set* $\mathbf{B}$ *constructed for a beam from 5 linear modes, using our method. The first row shows first order basis vectors and the rest are second order ones.*

In our experiments, large displacements can be captured by the above choice of basis set pair. In order to further justify our choice of basis set size $|\tilde{\mathbf{U}}|$ and $|\mathbf{B}|$, we ran a large set of simulations on the fork and dinosaur model with $|\tilde{\mathbf{U}}|=3,5,10$ and $\mathbf{B}=\text{Id}$ (using subspace only in RS-space, leaving Euclidean space representation exact). A mass-PCA is then performed on the sequences of frames in Euclidean space to extract the dominant components forming a reasonably small subspace $\mathbf{B}$. Several criteria can be applied to determine how many components should be retained. Here, we simply retain basis vectors whose eigenvalues are bigger than 0.01% of the largest eigenvalue. This would result in fewer eigenvectors than the famous Kaiser criterion (retaining eigenvectors with eigenvalues bigger than 1) which is known to be an overestimation. In our experiments, $|\mathbf{B}|$ found following this criterion always gives plausible approximation to all the groundtruth shapes (case with $\mathbf{B} = \mathrm{Id}$) as illustrated in top Figure 5. Using our criterion, $|\mathbf{B}|$ is plotted against $|\tilde{\mathbf{U}}|$ in bottom Figure 5. This plot roughly verifies that the choice of $|\mathbf{B}| = \mathcal{O}(|\tilde{\mathbf{U}}|^2)$ is reasonable.

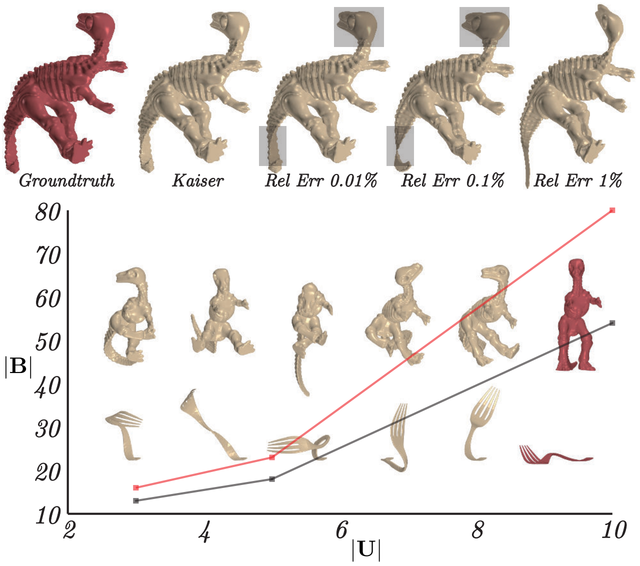


Figure 5: *Top: Our truncation criterion generates plausible approximation as Kaiser criterion. Allowing larger relative error would lead to visual artifact. Bottom: We ran simulation on the fork model using* $|\tilde{\mathbf{U}}| = 3, 5$ *and the dinosaur model using* $|\tilde{\mathbf{U}}| = 10$. *The background shows some of the extreme deformations achieved using such* $|\tilde{\mathbf{U}}|$. $|\mathbf{B}|$ *is then found to be* $13, 18, 55$ *respectively using our criterion and* $16, 23, 80$ *using Kaiser criterion (orange). This roughly verified our hypothesis that* $|\mathbf{B}| = \mathcal{O}(|\tilde{\mathbf{U}}|^2)$.

5.2 Jacobian Evaluation

Even though the whole system resides in small subspaces, the time integration of Equation 10 is still inefficient because the Jacobian $\frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}} = \Phi \frac{\partial \tilde{\nabla}}{\partial \tilde{\mathbf{x}}}$ involved in the mass matrix non-linearly depends on all $|T|$ elements. Here, we exploit two methods to accelerate this procedure for real-time performance. If $|\mathbf{B}|$ is small ($|\mathbf{B}| \leq 100$ or $|\tilde{\mathbf{U}}| \leq 10$ in practice), we can leverage rational approximation of $\exp(\omega_i)$ to replace $\frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}}$ with a reduced rational function of $\tilde{\mathbf{x}}$ whose coefficients can be precomputed. This is sometimes called a Padé approximation [Moler and Loan 1978], which is known to be very efficient for matrix exponentials. However, when $|\mathbf{B}|$ grows larger, this approximation becomes incompetent compared with cubature approximation as is noted in [An et al. 2008]. Similarly, we introduce cubature approximation on the kinetic side as an alternative for larger subspaces. Note that, when floating frame is attached, the mass matrix takes a more involved form, see Appendix B for its derivation.

The idea of **Polynomial Approximation** is based on the observation that $\exp(\omega_i)$ has simple Taylor series at the origin. When this series is truncated at order p, we get a polynomial approximation of order $p+1$, which can be precomputed and evaluated efficiently like [Barbič and James 2005]. It corresponds to solving the partial differential equation: $\dot{\mathbf{Y}}(\mathbf{t}) = \omega_i \mathbf{Y}(\mathbf{t})$ using explicit Runge-Kutta method. In our experiments, we find that $p = 3$ gives a good approximation for reasonably large deformation. In this case, $\mathbf{x}(\tilde{\mathbf{x}})$ is a forth order polynomial in $\tilde{\mathbf{x}}$ whose coefficients can be precomputed.

However, it is well known that, to approximate matrix exponentials, rational functions are usually more efficient than polynomials [Moler and Loan 1978]. This motivates us to consider a reduced version of rational approximation. In the early stage of our research, we considered using general purpose multivariate rational approximation. But we soon found it impractical since the convergence of these algorithms has not been well studied. Therefore, we choose to apply rational approximation only to the matrix exponentials: each rotational part $\exp(\omega_i)$ can be approximated using a (p,q)-rational function $\mathbf{Q}(\omega_i)^{-1}\mathbf{P}(\omega_i)$ where $\mathbf{Q}(\omega_i), \mathbf{P}(\omega_i)$ are polynomial functions of ω_i (see [Moler and Loan 1978]):

$$\begin{aligned}\mathbf{P}(\omega_i) &= \sum_{j=0}^{p} \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!}(\omega_i)^j \\ \mathbf{Q}(\omega_i) &= \sum_{j=0}^{q} \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!}(-\omega_i)^j.\end{aligned}$$

They can then be plugged in Equation 4 to solve for an approximate $\mathbf{x}$ denoted as $\mathbf{x}_{pq}$. However, this would lead to a new right hand side involving $\mathbf{Q}^{-1}(\omega_i)$ which cannot be precomputed in a polynomial form. In order to cast the Poisson reconstruction Equation 4 into a reduced precomputable form, we instead look at the underlying overdetermined system:

$$\sqrt{D}\nabla\mathbf{B}\mathbf{x} = \sqrt{D}\left[\mathbf{Q}^{-1}\mathbf{P}(\boldsymbol{\epsilon} + \mathrm{Id}) - \mathrm{Id}\right], \tag{11}$$

with $\mathbf{Q}, \mathbf{P}, \boldsymbol{\epsilon}$ assembled from $\mathbf{Q}(\omega_i), \mathbf{P}(\omega_i), \epsilon_i$. What Equation 4 does is just approximating Equation 11 in a least square sense. We can now apply a metric change by multiplying $\mathbf{Q}$ from the left to get:

$$\mathbf{Q}\sqrt{D}\nabla\mathbf{B}\mathbf{x} = \sqrt{D}\left[\mathbf{P}(\boldsymbol{\epsilon} + \mathrm{Id}) - \mathbf{Q}\right].$$

This equation can be solved by further multiplying from the left a set of test functions. For whatever test function used, the derived left and right hand size would be precomputable as high order polynomials of $\tilde{\mathbf{x}}$. For example, if least square solution is desired, the test function is just $\mathbf{Q}\sqrt{D}\nabla\mathbf{B}$ itself. However, the polynomials derived this way would be of order as high as $\max(p+q+1, 2q)$ leading to a huge number of coefficients. So that we propose to exclude $\mathbf{Q}$ from our test functions, i.e. multiplying $(\sqrt{D}\nabla\mathbf{B})^T$ from the left of both sides, giving the following square system:

$$\begin{aligned}\bar{\mathbf{Q}}\mathbf{x}_{pq} &= \bar{\mathbf{P}} \\ \bar{\mathbf{Q}} &\triangleq \mathbf{B}^T\nabla^T\mathbf{Q}D\nabla\mathbf{B} \\ \bar{\mathbf{P}} &\triangleq \mathbf{B}^T\nabla^T D\left[\mathbf{P}(\boldsymbol{\epsilon} + \mathrm{Id}) - \mathbf{Q}\right],\end{aligned}$$

where the highest order of polynomial is just $\max(p+1, q)$ which can still be evaluated efficiently. The Jacobian $\frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}}$ approximated in this way becomes:

$$\frac{\partial \mathbf{x}_{pq}}{\partial \tilde{\mathbf{x}}} = \bar{\mathbf{Q}}^{-1}\left(\frac{\partial \bar{\mathbf{P}}}{\partial \tilde{\mathbf{x}}} - \frac{\partial \bar{\mathbf{Q}}}{\partial \tilde{\mathbf{x}}} \mathbf{:} \mathbf{x}_{pq}\right).$$

In Figure 6, largest allowable deformations using different (p,q) are illustrated. Note that the rational approximation degenerates to a Taylor approximation by choosing $q = 0$, which is usually enough for moderate deformation with $p = 3$. A choice of $q \neq 0$ should be used only in case of extreme deformation.

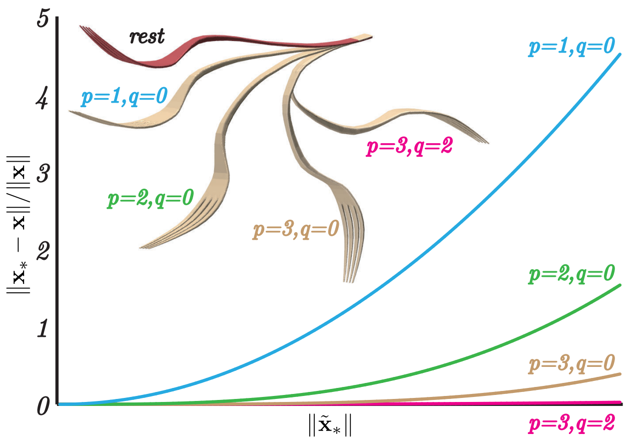


Figure 6: *We plotted relative error of polynomial approximation against deformation in RS space. Clearly, rational approximation is much more accurate than its Taylor counterpart. We also rendered the maximal allowable deformation within the relative error of* 10% *for the fork model using different* p *and* q*.*

Finally, we notice that, for practical issue, the expressions for high order derivative of $\bar{\mathbf{P}}$ and $\bar{\mathbf{Q}}$ can be very complex, making the analytical precomputation like [Barbič and James 2005] unavailable to us. We instead turn to the general purpose polynomial coefficient extraction algorithm [Cuyt and Wuytack 1986]. These methods only require evaluating a polynomial at specified points, which greatly simplifies the implementation of our method.

The approximation discussed above is enough for many applications, which is used in most of our examples. This is in part because the configuration space is rather small under RS coordinates. However, when large $|\tilde{\mathbf{U}}|$ or $|\mathbf{B}|$ is desired, **Cubature Optimization** would be more competitive as is indicated in [An et al. 2008]. Fortunately, this approximation is still available to our new kinetic term. Specifically, we can approximate $\mathbf{x}(\tilde{\mathbf{x}})$ by linearly combining $\Phi_j\tilde{\nabla}_j(\tilde{\mathbf{x}})$ with $j \in \mathcal{C}$, a small subset of elements. This essentially defines another reduced approximation to $\mathbf{x}$ denoted as:

$$\mathbf{x}_c(\tilde{\mathbf{x}}) = \sum_{i \in \mathcal{C}} \mathbf{w}_i \Phi_i \tilde{\nabla}_i(\tilde{\mathbf{x}}). \qquad (12)$$

In order to find $\mathcal{C}$, we solve the following sparse coding problem for a set of p training poses:

$$\mathbf{min}_{\mathbf{w}} \sum_{t=1}^{p} \frac{\|\mathbf{x}(\tilde{\mathbf{x}}_t) - \sum_i^{|T|} \mathbf{w}_i \Phi_i \tilde{\nabla}_i(\tilde{\mathbf{x}}_t)\|^2}{\|\mathbf{x}(\tilde{\mathbf{x}}_t)\|^2} + \phi(\mathbf{w}),$$

where ϕ is some sort of sparse regularizer on the weighting $\mathbf{w} = (\mathbf{w}_1, \cdots, \mathbf{w}_{|\mathcal{C}|})$ and Φ_i is the $|\mathbf{B}| \times 9$ block of Φ corresponding to T_i. As illustrated in Figure 13, our algorithm outperforms [An et al. 2008] by requiring less cubature points for similar accuracy. To make full use of this property, we need a sparse coding solver that finds as few cubature points as possible, given a user specified error bound. The details of our solver can be found in Appendix A.

5.3 Algorithmic Complexity

It is non-trivial to give a comparison of algorithmic complexity with previous methods because two subspaces are involved in our method. We first consider the case where $\mathbf{B}$ and $\tilde{\mathbf{U}}$ are chosen arbitrarily, linear elasticity is used for potential energy and only one Newton iteration is performed in each time step. Our algorithm is then composed of two substeps: the evaluation of LHS/RHS of a linear system and a linear equation solve. For the first substep, two alternatives are available. If $(p = 3, q = 2)$ rational approximation is used, the overhead is $\mathcal{O}(|\mathbf{B}|^2|\tilde{\mathbf{U}}|^2 + |\mathbf{B}||\tilde{\mathbf{U}}|^4 + |\mathbf{B}|^3)$. If cubature optimization is used, the overhead is $\mathcal{O}(|\mathcal{C}||\mathbf{B}||\tilde{\mathbf{U}}| + |\mathbf{B}|^2|\tilde{\mathbf{U}}| + |\mathbf{B}||\tilde{\mathbf{U}}|^2)$. We refer readers to Appendix D for a detailed derivation. Then the linear solve costs: $\mathcal{O}(|\tilde{\mathbf{U}}|^3)$, while previous methods take $\mathcal{O}(|\mathbf{B}|^3)$. Note that although cubature optimization seems to have lower complexity in terms of $|\mathbf{B}|$, the constant coefficient is much larger than that of rational approximation due to the evaluation of $\partial\tilde{\nabla}_i/\partial\tilde{\mathbf{x}}$ at each cubature point. Thus, for small $|\tilde{\mathbf{U}}|$, rational approximation is more efficient.

To compare this algorithm with [Barbič and James 2005; An et al. 2008], we can use our assumption that $|\mathbf{B}| = \mathcal{O}(|\tilde{\mathbf{U}}|^2)$. In this case, our algorithmic complexity using rational approximation is dominated by $\mathcal{O}(|\mathbf{B}|^3)$ while [Barbič and James 2005] is dominated by $\mathcal{O}(|\mathbf{B}|^4)$. For the version using cubature optimization, force and stiffness evaluation part of our method scales like $\mathcal{O}(|\mathcal{C}||\mathbf{B}|^{1.5} + |\mathbf{B}|^{2.5})$ while [An et al. 2008] scales like $\mathcal{O}(|\mathcal{C}||\mathbf{B}|^2)$. Although our method introduces a second term with higher order in $|\mathbf{B}|$, the first term is usually dominant due to the large constant factor and $|\mathcal{C}|$.

6 Hyperelastic Materials

To this point, we have assumed that W is linear elastic material. Indeed, most of the results are generated with W quadratic in $\tilde{\mathbf{x}}$. However, as mentioned in Section 3, since our method doesn't exploit any special structure in W, we are blessed with the flexibility to incorporate other materials, such as Mooney-Rivlin [Rivlin 1948], Arruda-Boyce [Arruda and Boyce 1993] or Fung [Fung 1981] to name just a few, using cubature optimization [An et al. 2008].

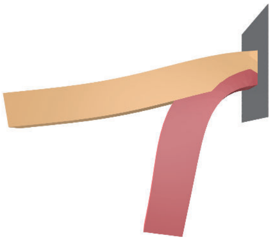


Figure 7: *The strain of* $\frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I})$ *would introduce undesired locking artifact to our model (brown) compared with infinitesimal strain in RS space (red).*

In principle, these hyperelastic laws are built on invariants derived from the strain tensor, see e.g. [Fung and Tong 2001]. Thus, all we need is a definition of strain tensor based on our RS space representation. Note that we can always compute the strain by first reconstructing the deformation gradient in Euclidean space, but that would bring extra computational overhead by involving the matrix exponentials. Giving the "pre-warped deformation gradient" $\mathbf{F} = \omega + \epsilon + \mathbf{I}$, it should be noticed that the strain definition $\mathbf{E} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I})$ is not equivalent to the usual Green's strain, which violates rotational invariance and leads to the locking artifact as illustrated in Figure 7. To properly define the Green's strain, $\mathbf{E} = \frac{1}{2}(\epsilon^T\epsilon) + \epsilon$ should be used. However, the infinitesimal strain (or Cauchy's strain tensor) in RS space $\mathbf{E} = \frac{1}{2}(\mathbf{F}^T + \mathbf{F}) - \mathbf{I} = \epsilon$ is indeed rotational invariant, and could be used in place of Green's strain.

As an example, for isotropic Fung's model [Fung 1981] used in our examples, the density function is:

$$\begin{aligned} W_{Fung}(\nabla\mathbf{q}(p)) \quad &= \quad W_{StVK}(\nabla\mathbf{q}(p), \mu_1, \lambda_1) + \\ &\quad\;\; c(\mathbf{e}^{W_{StVK}(\nabla\mathbf{q}(p), \mu_2, \lambda_2)} - 1), \end{aligned}$$

where we use two sets of Lame's coefficients depicting stress under small and large strain respectively and an additional material coefficient c. As illustrated in Figure 8, this energy is useful for graphic applications, mimicking the effect of strain limiting. By replacing Green's strain with infinitesimal strain in RS space, W_{StVK} becomes quadratic in $\tilde{\mathbf{x}}$ and we have:

$$\tilde{W}_{Fung}(\tilde{\mathbf{x}}) = \frac{1}{2}\tilde{\mathbf{x}}^T\mathbf{K}_1\tilde{\mathbf{x}} + c(\mathbf{e}^{\frac{1}{2}\tilde{\mathbf{x}}^T\mathbf{K}_2\tilde{\mathbf{x}}} - 1),$$

where the two StVK terms become two precomputable linear elastic terms with reduced stiffness matrices $\mathbf{K}_1, \mathbf{K}_2$. Other energy functions can be simplified in a similar way.

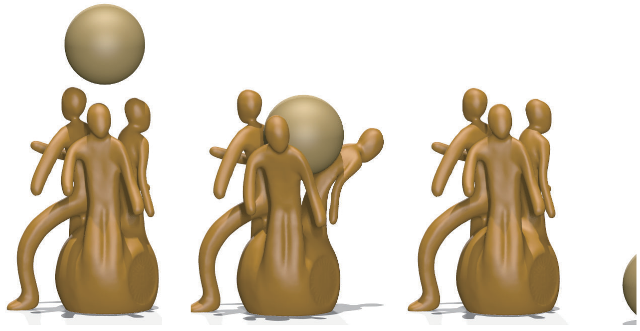


Figure 8: *Three men backing up the heavy sphere. Initial configuration is on the left. With linear elastic material the model is too soft to hold the sphere (right), while hyperelastic materials can effectively limit the strain (middle).*

When using non-linear material model, one can use another set of cubature points for the kinetic term (another possible way is to use polynomial approximation). In our experiments, we found it enough to optimize the cubature set for kinetic term only and reuse it for potential energy by solving an additional NNLS problem [Lawson and Hanson 1974].

7 Results and Discussion

In this section, we will investigate the dynamic behavior of our model in terms of energy distribution and system non-linearity. We then evaluate its efficiency and accuracy with several benchmark tests.

A first point we want to make is that the proposed model presents similar dynamic behaviors as conventional corotated/StVK model. To show this we present a comparison with traditional "full space" (non)linear elastic models in the Euclidean space, i.e. linear elastic and corotated/StVK elastic models. From Figure 10, it is clear that our method achieves almost the same maximal kinetic energy as the corotated/StVK model (these two are indistinguishable in this test), while the linear elastic model can reach higher kinetic energy due to its gross distortion artifact. It should be noticed that our method has lower frequency of vibration compared with the corotated/StVK model under the same Young's modulus. This frequency shift can be explained by the compatibility error introduced in Equation 4 which is solved in a least square sense. As a result, our model appears slightly softer. This problem can be alleviated by increasing Young's modulus by 12.5% as illustrated by the blue curve in Figure 10.

A more informative comparison is with previous subspace method [An et al. 2008] using corotated linear elastic material. We will show that, when expressed in RS coordinates, the dynamic system

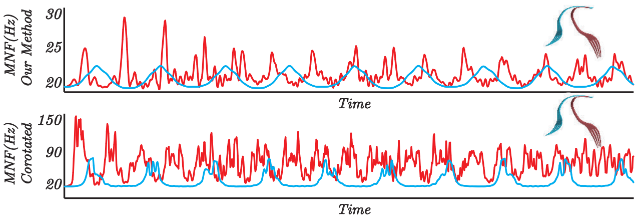


Figure 9: *The MNF change over a* $2s$ *simulation run on the fork model under small gravity (blue,* $g = 9.81m/s^2$*) and large gravity (red,* $g = 49.05m/s^2$*). Both simulations use the same initial configuration and material properties (Young's modulus=*$1e7$*, Poisson ratio=*0.4*). The MNFs are found from minimal* $\sqrt{\lambda}$ *of the problem:* $\lambda\mathbf{M}\mathbf{v} = \mathbf{K}\mathbf{v}$.

can be largely linearized. Our analysis is based on the observation that, for a linear system of type $\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{0}$, the minimal natural frequency (MNF) is an invariant solely determined by $\mathbf{M}, \mathbf{K}$. It thus suggests that, for a non-linear system with changing $\mathbf{M}$ in our case or changing $\mathbf{K}$ in the case of [An et al. 2008; Barbič and James 2005], the fluctuation of MNF is an effective indicator of non-linearity. In Figure 9, we plotted the MNF change over time for small (in blue) and large (in red) deformations. Starting from an initial MNF of 20Hz, previous methods present much higher non-linearity under this criterion than our RS space method. Note that we use a large set $|\mathcal{C}| = 1000$ cubature points for this test to rule out possible noise from cubature approximation error, and the jittering of the MNF is very likely from the complex behavior of mode coupling.

The snapshots of dynamic simulations from different methods are shown in Figure 11. Under small gravity, the simple linearization in Euclidean space (Linear Modal Analysis) introduces significant artifacts. The linear elastic methods in RS space [Choi and Ko 2005; Huang et al. 2011; Li et al. 2014] are able to achieve plausible result in this case (upper row), but fails as well under large gravity (bottom row). As for our method, Taylor approximation $(p = 3, q = 0)$ is enough under small gravity but we have to resort to rational approximation under large gravity. Finally, both our method and [Barbič and James 2005] need to solve a non-linear problem in each time step for accuracy. To compare the non-linearity, just one Newton step is used for these two methods. They present comparable dynamic behavior under small gravity, but our method with $(p = 3, q = 2)$ generates better results in terms of less energy dispassion (see Figure 12 and the accompanying video). The corotated method, another Euclidean space non-linear method [Müller and Gross 2004], also has excessive damping in this experiment. Less non-linearity in the RS space makes our method more efficient than the Euclidean ones since less Newton steps are required for plausible results, which is very attractive for many interactive applications.

The performance of offline precomputation and online evaluation for some representative examples is summarized in Table 1. In our experience, our method is always faster than [Barbič and James 2005]. The computational advantage is remarkable especially when $|\tilde{\mathbf{U}}|$ grows larger. As is illustrated in the inset, we use $|\tilde{\mathbf{U}}| = 20$ and $|\mathbf{B}| = 230$ allowing each handle of the vase to be deformed nearly individually. Both rational and cubature approximation can still achieve real-time performance. [Barbič and James 2005] runs out of memory after two days of precomputation, while our method can still handle it within moderate time. Note that we implemented [Barbič and James 2005] using our general purpose polynomial coefficient extraction algorithm. The original analytical method may consume less memory without requiring to solve a coefficient ma-

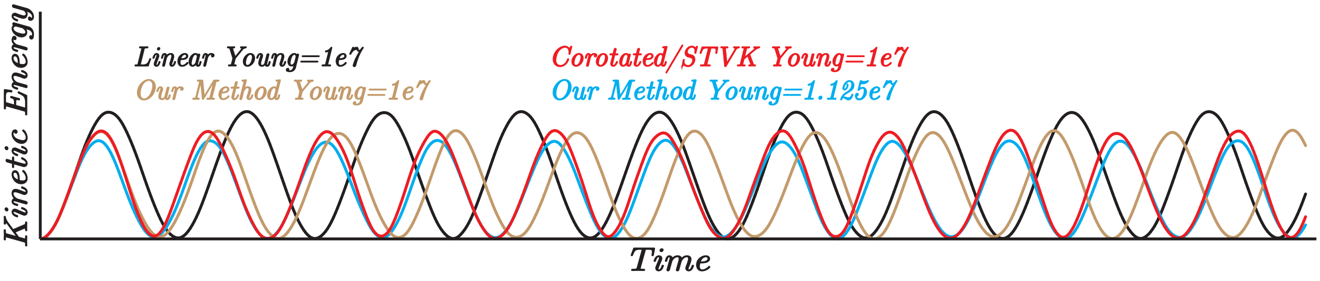


Figure 10: *This figure plots the kinetic energy change over time. The simulation runs on the fork model with one end fixed driven solely by gravitational forces. Both the linear and non-linear models reside in "full space" while our method runs with $|\tilde{\mathbf{U}}| = 5, |\mathbf{B}| = 20$ and rational approximation for transfer function. Since we used variational integrator where the total energy is almost conserved, we can omit potential energy plot to save space.*

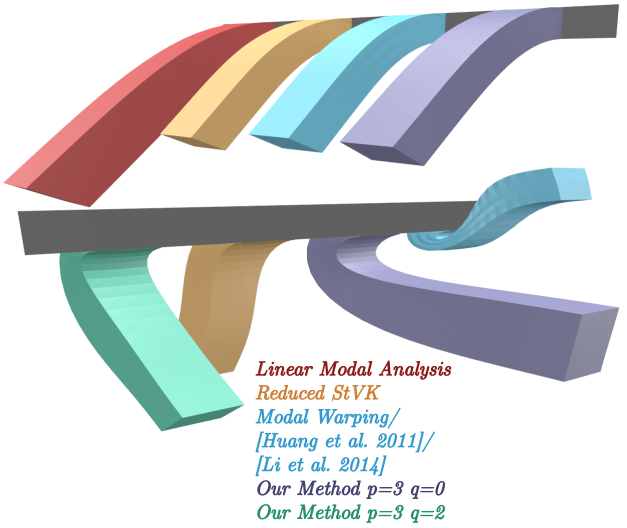


Figure 11: *Comparison between different methods. Under small gravity (upper row), our method using Taylor approximation runs the fastest and presents similar non-linear deformation with previous methods. But under large gravity (bottom row), Taylor approximation cannot eliminate gross distortion and Modal Warping presents non-physical deformation. In this case, rational approximation achieves the best compromise between speed and quality.*

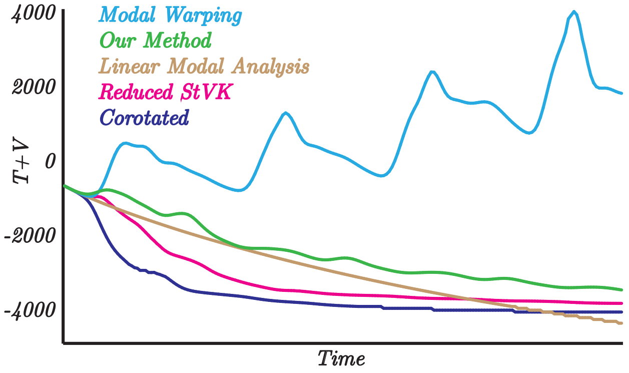


Figure 12: *A comparison of total energy change over time. We use Backward Euler with only one iteration. In this case the linear modal analysis is exact. But corotated and StVK models suffer from excessive damping, while total energy of modal warping changes non-physically.*

trix. But even after the precomputation it is still impossible for it to handle problem of this size in real-time. Also, we observed that, under similar accuracy, our method is always faster than [An et al. 2008] using corotated linear elasticity. In Figure 13, both our method and [An et al. 2008] are compared with a groundtruth "full space" StVK simulation. The set of cubature points in this experiment are selected prudently, where we use simple greedy cubature optimizer for both methods with Monte-Carlo acceleration switched off to rule out possible bias. In this case, 15 cubature points achieve 5 – 6% relative error on both models and 45 cubature points achieve 2 – 3% relative error. If both methods use 45 points, our method achieves 2× speedup. However, with only 15 points our method is able to achieve comparable accuracy as [An et al. 2008] using 45 points, but runs 5× faster.

In examples with multiple bodies we use [Hirota et al. 2000] for collision detection and handling. In these cases, the overhead of collision processing would quickly dominate the algorithm especially when large areas of bodies are in contact. This bottleneck should be removed in our future work.

8 Limitation and Future Work

As a new dynamic model, this work leaves much space for further exploration. To begin with, by using such a small configuration space, some local deformations may be left out. Currently, all the examples illustrate only salient global deformation. But we expect that methods such as [Harmon and Zorin 2013] can be used as a patch. Another potential problem is related to basis construction, where we propose to build $\tilde{\mathbf{U}}$ first and then $\mathbf{B}$. An important alternative is to build $\mathbf{B}$ directly from animation snapshots as is done in [Kim and James 2009]. In that case, we still have no idea what is an effective small $\tilde{\mathbf{U}}$. Also, compared with [Idelsohn and Cardona 1985], our basis extension algorithm doesn't respect nonlinear stiffness matrix when hyperelastic materials are used. This problem should be addressed especially when working with anisotropic materials. Besides, when compared with [Barbič and James 2005]

Model	Algor.	$(p, q)/\lvert\mathcal{C}\rvert$	$\lvert\tilde{\mathbf{U}}\rvert$	$Pre.$	$Frm.[Coll.]$
Beam	$\mathbf{x}_{pq}$	(3,0)	5	17	0.45
Beam	$\mathbf{x}_c$	9	5	55	0.6
Beam	StVK	N/A	5	19	1.2
Vase	$\mathbf{x}_{pq}$	(3,2)	20	2530	38
Vase	$\mathbf{x}_c$	194	20	1613	12
Vase	StVK	N/A	20	> $50h$	N/A
Dinosaur+FF	$\mathbf{x}_{pq}$	(3,2)	10	59	40(per body)[1537]
Letter+FF	$\mathbf{x}_{pq}$	(3,2)	10	304	21(per body)[2413]
Dinosaur	$\mathbf{x}_c$	15	10	N/A	0.71
Dinosaur	$\mathbf{x}_c$	45	10	N/A	2.3
Dinosaur	$\mathbf{x}_{pq}$	(3,2)	10	59	1.6
Dinosaur	[An et al. 2008]	45	10	N/A	4.5

Table 1: *Performance of representative examples using different methods. All the tests are done on a desktop computer with a 2.3GHz E5-2630 CPU. From left to right: Name of model (with optional floating frame: +FF), algorithm used, (p, q) if $\mathbf{x}_{pq}$ is used or number of cubature points if $\mathbf{x}_c$ is used, size of configuration space (we always use $|\mathbf{B}| = |\tilde{\mathbf{U}}| + |\tilde{\mathbf{U}}|(|\tilde{\mathbf{U}}| + 1)/2$), time for precomputation (in sec) and time for system build and equation solve [time for collision detection] per step (in ms).*

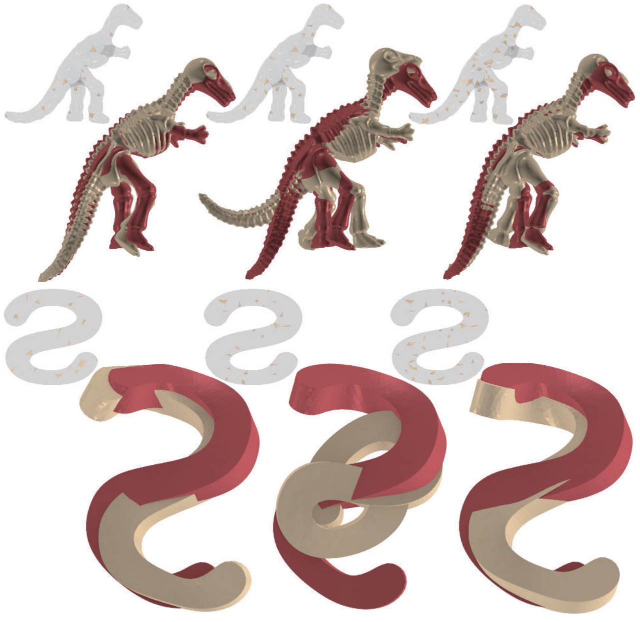


Figure 13: *Comparison of reduced simulation with "full space" groundtruth (red). Our method requires only* 15 *cubature points to achieve high consistency (left). However, with* 20 *cubature points, previous method still exhibits large error (middle). In their case,* 45 *cubature points are needed for similar consistency (right).*

which is exact, our rational approximation scheme would introduce additional error in kinetic energy and two more additional parameters (p, q) to be tuned. For applications considered in this work, we find the best parameter setting is always $p = 3, q = 0/2$. Other choices lead to either visible dynamic error or slow online evaluation. In these cases, cubature approximation should be used. A final problem is that, in order to achieve reasonable consistency with "fullspace" methods, the Young's modulus used in our method has to be rescaled since our model appears softer. Unfortunately, we still don't have an automatic algorithm for determining this scaling factor. This problem essentially reflects the fact that, by using Equation 3, we reconstructed $\mathbf{q}$ from $\tilde{\mathbf{q}}$ only in a least square sense, which is very similar to the discontinuous galerkin method [Kaufmann et al. 2008].

Acknowledgements.

Partial funding was provided by NSFC (No.61170139, No.61210007), 973 Program of China (No. 2015CB352503), Fundamental Research Funds for the Central Universities (No. 2015FZA5018, 2015XZZX005-05). Some of the models are provided by AIM@SHAPE shape repository and INRIA Gamma dataset.

References

AN, S. S., KIM, T., AND JAMES, D. L. 2008. Optimizing cubature for efficient integration of subspace deformations. *ACM Transactions on Graphics 27*, 5, 165:1–165:10.

ANDO, R., THÜREY, N., AND WOJTAN, C. 2015. A dimension-reduced pressure solver for liquid simulations. In *Computer Graphics Forum (Eurographics)*, vol. 34, 10.

ARRUDA, E. M., AND BOYCE, M. C. 1993. A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials. *Journal of the Mechanics and Physics of Solids 41*, 2, 389–412.

BARBIČ, J., AND JAMES, D. L. 2005. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Transactions on Graphics 24*, 3, 982–990.

BARBIČ, J., SIN, F., AND GRINSPUN, E. 2012. Interactive editing of deformable simulations. *ACM Transactions on Graphics 31*, 4, 70:1–70:8.

BOUAZIZ, S., MARTIN, S., LIU, T., KAVAN, L., AND PAULY, M. 2014. Projective dynamics: fusing constraint projections for fast simulation. *ACM Transactions on Graphics 33*, 4, 154.

CAPELL, S., GREEN, S., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2002. Interactive skeleton-driven dynamic deformations. *ACM Transactions on Graphics 21*, 3, 586–593.

CHOI, M. G., AND KO, H.-S. 2005. Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Transactions on Visualization and Computer Graphics 11*, 1, 91–101.

CUYT, A., AND WUYTACK, L. 1986. *Nonlinear Methods in Numerical Analysis*. North-Holland Publishing Co.

FUNG, Y.-C., AND TONG, P. 2001. *Classical and computational solid mechanics*, vol. 1. World scientific.

FUNG, Y. 1981. *Biomechanics: mechanical properties of living tissues*. Biomechanics / Y. C. Fung. Springer-Verlag.

HAHN, F., MARTIN, S., THOMASZEWSKI, B., SUMNER, R., COROS, S., AND GROSS, M. 2012. Rig-space physics. *ACM Transactions on Graphics 31*, 4, 72.

HARMON, D., AND ZORIN, D. 2013. Subspace integration with local deformations. *ACM Transactions on Graphics 32*, 4, 107.

HAUSER, K. K., SHEN, C., AND OBRIEN, J. F. 2003. Interactive deformation using modal analysis with constraints. In *Proceedings of Graphics Interface*, vol. 3, 16–17.

HIROTA, G., FISHER, S., AND LIN, M. 2000. Simulation of non-penetrating elastic bodies using distance fields. Tech. rep., Chapel Hill, NC, USA.

HUANG, J., TONG, Y., ZHOU, K., BAO, H., AND DESBRUN, M. 2011. Interactive shape interpolation through controllable dynamic deformation. *IEEE Transactions on Visualization and Computer Graphics 17*, 7, 983–992.

IDELSOHN, S. R., AND CARDONA, A. 1985. A load-dependent basis for reduced nonlinear structural dynamics. *Computers & Structures 20*, 13, 203 – 210. Special Issue: Advances and Trends in Structures and Dynamics.

IRVING, G., TERAN, J., AND FEDKIW, R. 2006. Tetrahedral and hexahedral invertible finite elements. *Graphical Models 68*, 2, 66–89.

JAMES, D. L., AND PAI, D. K. 2002. Dyrt: dynamic response textures for real time deformation simulation with graphics hardware. *ACM Transactions on Graphics (TOG) 21*, 3, 582–585.

KAUFMANN, P., MARTIN, S., BOTSCH, M., AND GROSS, M. 2008. Flexible Simulation of Deformable Models Using Discontinuous Galerkin FEM. In *Eurographics/SIGGRAPH Symposium on Computer Animation*, The Eurographics Association, M. Gross and D. James, Eds.

KIM, T., AND DELANEY, J. 2013. Subspace fluid re-simulation. *ACM Transactions on Graphics 32*, 4, 62.

KIM, T., AND JAMES, D. L. 2009. Skipping steps in deformable simulation with online model reduction. In *ACM Transactions on graphics*, vol. 28, ACM, 123.

LAWSON, C. L., AND HANSON, R. J. 1974. *Solving least squares problems*, vol. 161. SIAM.

LI, S., HUANG, J., DE GOES, F., JIN, X., BAO, H., AND DESBRUN, M. 2014. Space-time editing of elastic motion through material optimization and reduction. *ACM Transactions on Graphics 33*, 4, 108:1–108:10.

MARTIN, S., THOMASZEWSKI, B., GRINSPUN, E., AND GROSS, M. 2011. Example-based elastic materials. *ACM Transactions on Graphics 30*, 4, 72:1–72:8.

MOLER, C., AND LOAN, C. V. 1978. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review 20*, 801–836.

MÜLLER, M., AND GROSS, M. 2004. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, Canadian Human-Computer Communications Society, 239–246.

PENTLAND, A., AND WILLIAMS, J. 1989. Good vibrations: modal dynamics for graphics and animation. *SIGGRAPH Computer Graphics 23*, 3, 207–214.

RIVLIN, R. 1948. Large elastic deformations of isotropic materials. iv. further developments of the general theory. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences 241*, 835, 379–397.

SCHULZ, C., VON TYCOWICZ, C., SEIDEL, H.-P., AND HILDEBRANDT, K. 2014. Animating deformable objects using sparse spacetime constraints. *ACM Transactions on Graphics 33*, 4, 109:1–109:10.

TERZOPOULOS, D., AND WITKIN, A. 1988. Physically based models with rigid and deformable components. *Computer Graphics and Applications, IEEE 8*, 6, 41–51.

TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. In *SIGGRAPH Computer Graphics*, vol. 21, ACM, 205–214.

VON TYCOWICZ, C., SCHULZ, C., SEIDEL, H.-P., AND HILDEBRANDT, K. 2013. An efficient construction of reduced deformable objects. *ACM Transactions on Graphics 32*, 6, 213.

YANG, J., AND ZHANG, Y. 2011. Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM journal on scientific computing 33*, 1, 250–278.

A ℓ_1-Sparse Cubature Solver

In this section, we present the cubature optimizer used in this work. The solver features user controllable relative error bound and near optimal solutions. We start from the following standard form of non-negative sparse coding problem:

$$\begin{aligned} \mathbf{argmin}_{\mathbf{w}} \quad & \|\mathbf{A}\mathbf{w}-\mathbf{b}\|^2+\phi(\mathbf{w}) \\ \mathbf{s.t.} \quad & \mathbf{w}\geq\mathbf{0}. \end{aligned}$$

In order to control the relative error, $\|\mathbf{A}\mathbf{w}-\mathbf{b}\|\leq\delta$ is added as hard constraint and we optimize for $\phi(\mathbf{w})=\|\mathbf{w}\|_0$. In order to make the non-convex problem tractable, the ℓ_0 term is approximated using iteratively reweighted ℓ_1. After these modifications, we end up with the following convex problem:

$$\begin{aligned} \mathbf{argmin}_{\mathbf{w}} \quad & \mathbf{c}^T\mathbf{w} \\ \mathbf{s.t.} \quad & \|\mathbf{A}\mathbf{w}-\mathbf{b}\|\leq\delta \quad \mathbf{w}\geq\mathbf{0}, \end{aligned}$$

where $\mathbf{c}_i=(\mathbf{w}_i^{old})^{p-1}$ is the weighting. Fortunately, this problem has been well studied and can be solved very efficiently using ADMM method [Yang and Zhang 2011]. It works by introducing slack variables $\mathbf{x}_r,\mathbf{z}_r$ to reformulate it as:

$$\begin{aligned} \mathbf{argmin}_{\mathbf{w}} \quad & \mathbf{c}^T\mathbf{w} \\ \mathbf{s.t.} \quad & \mathbf{A}\mathbf{w}+\mathbf{z}_r=\mathbf{b} \quad \mathbf{w}=\mathbf{x}_r \\ & \|\mathbf{z}_r\|\leq\delta \quad \mathbf{x}_r\geq\mathbf{0}. \end{aligned}$$

ADMM then works by iteratively optimizing the following Augmented Lagrangian:

$$\begin{aligned} \mathbf{argmin}_{\mathbf{w}} \quad & \mathbf{c}^T\mathbf{w} \\ & -\lambda_z^T(\mathbf{A}\mathbf{w}+\mathbf{z}_r-\mathbf{b})+\frac{\beta_z}{2}\|\mathbf{A}\mathbf{w}+\mathbf{z}_r-\mathbf{b}\|^2 \\ & -\lambda_x^T(\mathbf{w}-\mathbf{x}_r)+\frac{\beta_x}{2}\|\mathbf{w}-\mathbf{x}_r\|^2 \\ \mathbf{s.t.} \quad & \|\mathbf{z}_r\|\leq\delta \quad \mathbf{x}_r\geq\mathbf{0}. \end{aligned}$$

In each iteration, it first updates $\mathbf{x}_r,\mathbf{z}_r,\mathbf{w}$ respectively and then adjust λ_x,λ_z. For $\mathbf{z}_r,\mathbf{x}_r$, the subproblem can be solved analytically to find $\mathbf{z}_r=\mathcal{P}_\delta\left(\lambda_z/\beta_z-(\mathbf{A}\mathbf{w}-\mathbf{b})\right)$, where $\mathcal{P}_\delta$ is the projection to $\|\mathbf{z}_r\| \leq \delta$ and $\mathbf{x}_r = \mathbf{max}(\mathbf{w} - \lambda_x/\beta_x, \mathbf{0})$. While for $\mathbf{w}$, the subproblem is just a linear solve:

$$(\beta_x + \beta_z \mathbf{A}^T\mathbf{A})\mathbf{w} = \beta_z \mathbf{A}^T(\mathbf{b} - \mathbf{z}_r) + \mathbf{A}^T\lambda_z + \beta_x \mathbf{x}_r + \lambda_x - \mathbf{c}.$$

Finally, λ_x, λ_z are updated according to:

$$\begin{aligned}\lambda_z &= \lambda_z - \gamma\beta_z(\mathbf{A}\mathbf{w} + \mathbf{z}_r - \mathbf{b})\\ \lambda_x &= \lambda_x - \gamma\beta_x(\mathbf{w} - \mathbf{x}_r)\\ \gamma &\in \left(0, (\sqrt{5}+1)/2\right).\end{aligned}$$

The convergency of this method is proved in [Yang and Zhang 2011]. The algorithm pipeline is summarized in Algorithm 1. Our experiment shows that even after one outer iteration, the solution becomes quite sparse. By discarding components of $\mathbf{w}$ very close to zero, subsequence ADMM solves can be much faster.

Algorithm 1 Reweighted ADMM Cubature Solver

```
$p = 1, k = 1, \mathbf{w}^k = \mathbf{1}$
while $p > 0.3$ do                                  ▷ Outter loop
    for all $i$ do                                  ▷ Discard small $\mathbf{w}_i$
        if $\mathbf{w}_i^k$ not small then
            set $\mathbf{c}_i = (\mathbf{w}_i^k)^{p-1}$
        else
            Set $\mathbf{c}_i = 0$
        end if
    end for
    while $\|\mathbf{w}^k - \mathbf{w}^{k-1}\|/\|\mathbf{w}^k\| > \epsilon$ do      ▷ ADMM Loop
        Update $\mathbf{z}_r, \mathbf{x}_r$
        Solve linear system for $\mathbf{w}^{k+1}$
        Update $\lambda_x, \lambda_z$
        $k = k + 1$
    end while
    $p = p - 0.1$
end while
```

Scaling to large mesh is one issue for an implementation of the above method, since it requires solving a dense system to update $\mathbf{x}$. We propose to remove this bottleneck by using a hierarchical version of ADMM solver. Specifically, when the number of mesh elements N exceeds a threshold ($N = 3000$ in all our examples), we cluster the mesh into approximately $\sqrt{N}$ groups and solve a group-wise ADMM first. Then, only the elements in the selected groups need to be considered. Specifically, we use a nearest neighbor clustering seeded with Poisson Disk Sampling within the volume. This procedure can be applied recursively to deal with mesh of arbitrarily large size while restricting the matrix size of underlying solver. More importantly, this treatment won't violate the user specified error bound δ.

To validate our method, Figure 14 illustrates the cubature computed for some of our examples. And in Table 2, we compare our method with the Greedy solver [An et al. 2008]. Clearly, our solver is advantageous in both time and number of cubature points needed to reach a specified relative error. For comparison, the Greedy method is used as an internal solver in the hierarchical framework. The original version [An et al. 2008] using randomly sampled candidates almost always results in even more cubature points.

Besides, a much more efficient solver based on iterative hard thresholding (NN-HTP) has been recently proposed in [von Tycowicz et al. 2013]. A comparison with their method is shown in the last column of Table 2, where all the parameters and stopping criteria of NN-HTP are chosen according to [von Tycowicz et al. 2013]. Since their method requires an additional parameter: the maximal number of cubature points which is hard to determine, we cannot compare the two algorithms based on relative error. Instead, we set the maximal number of cubature points equals to the number of cubature points returned by our algorithm and see if they can achieve the same level of accuracy. The result shows that NN-HTP is usually less accurate but more efficient. In conclusion, if user wants to maximize the online cubature performance, our method should be a better choice than NN-HTP because ℓ_1-sparse solver automatically determines the suitable number of cubature points without over-estimation and the optimized cubature set usually achieves lower relative error than all existing methods.

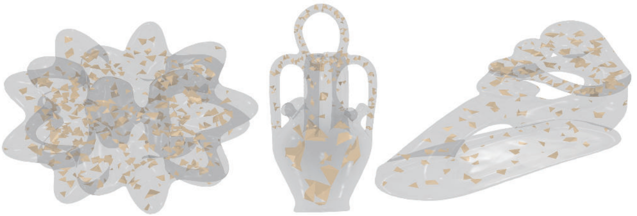


Figure 14: *Cubature points computed for some large meshes using our ℓ_1 Cubature Solver.*

B Attaching to Floating Frame

When our new dynamic system is attached to a floating frame, the configuration space is extended to $(\tilde{\mathbf{x}}, \mathbf{t}, \Theta)$. From these generalized coordinates, the Euclidean space deformation is recovered by Equation 8. One drawback of these coordinates is a much more involved mass matrix. But it can still be evaluated in a subspace. Assuming that lumped "full space" mass matrix $\mathbf{M}_{\text{Id}}$ is used, we want to evaluate:

$$\frac{\partial \mathbf{u}^T}{\partial(\tilde{\mathbf{x}}, \mathbf{t}, \Theta)} \mathbf{M}_{\text{Id}} \frac{\partial \mathbf{u}}{\partial(\tilde{\mathbf{x}}, \mathbf{t}, \Theta)} = \begin{pmatrix} \tilde{\mathbf{M}} & & \\ \mathbf{A} & \sum_{i=1}^m \rho_i |T_i| \,\text{Id} & \\ \mathbf{B} & \mathbf{C} & \mathbf{D} \end{pmatrix},$$

where we have omitted the symmetric upper-triangular part. The two diagonal submatrices are invariant to the frame and the remaining blocks $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ can be assembled as follows:

$$\begin{aligned}\mathbf{A}_{ij} &= \Xi^1_{imkl}\mathbf{x}^m_{,j}\mathbf{R}^{kl}\\ \mathbf{B}_{ij} &= (\Xi^2_{mkl} + \Xi^3_{mnkl}\mathbf{x}^n)\mathbf{x}^m_{,j}\mathbf{R}^{pk}_{,i}\mathbf{R}^{pl}\\ \mathbf{C}_{ij} &= (\Xi^1_{jmkl}\mathbf{x}^m + \Xi^4_{jkl})\mathbf{R}^{kl}_{,i}\\ \mathbf{D}_{ij} &= ((\Xi^3_{mnkl}\mathbf{x}^n + \Xi^2_{mkl} + \Xi^2_{mlk})\mathbf{x}^m + \Xi^5_{kl})\mathbf{R}^{pk}_{,i}\mathbf{R}^{pl}_{,j},\end{aligned}$$

where the precomputed tensors Ξ^* are:

$$\begin{aligned}\Xi^1_{ijkl} &= \frac{\partial \mathbf{u}}{\partial \mathbf{t}_i}^T \mathbf{M}_{\text{Id}}\mathbf{e}_{kl}\mathbf{B}_j & \Xi^2_{ikl} &= \mathbf{r}^T\mathbf{M}_{\text{Id}}\mathbf{e}_{kl}\mathbf{B}_i\\ \Xi^3_{ijkl} &= \mathbf{B}_i^T\mathbf{M}_{\text{Id}}\mathbf{e}_{kl}\mathbf{B}_j & \Xi^4_{ikl} &= \frac{\partial \mathbf{u}}{\partial \mathbf{t}_i}^T \mathbf{M}_{\text{Id}}\mathbf{e}_{kl}\mathbf{r}\\ \Xi^5_{kl} &= \mathbf{r}^T\mathbf{M}_{\text{Id}}\mathbf{e}_{kl}\mathbf{r} & &,\end{aligned}$$

$\mathbf{e}_{ij}$ is a block diagonal matrix with e_{ij} in each 3×3 diagonal block and $\mathbf{R} = \exp(\Theta)$. Note that all these derivations are based on the assumption that $\mathbf{M}_{\text{Id}}$ is a lumped "full space" mass matrix.

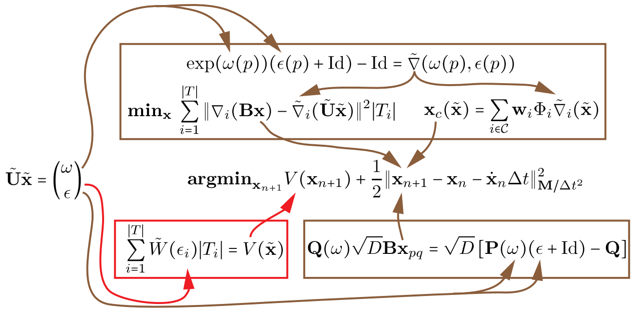
C A Summary of Equations

We provide here a summary of notations and equations used in our method. As illustrated in the figure, one needs to define the RS

Model	m	$\lvert\mathbf{B}\rvert$	$\mathbf{A}$	ℓ_1	Greedy	NN-HTP
Vase	71082	230	802	194/1373	351/9568	1.3%/709
Three Men	87973	65	69	50/164	103/421	1.6%/26
Bumpy	106014	135	193	407/1167	>500/31990	2.3%/419
Fertility	90486	135	158	234/558	301/7759	2.0%/301
Homer	249309	135	429	170/1658	278/11820	2.0%/1007

Table 2: *From left to right: name of model, number of elements, number of extended basis vectors, time taken (in sec) for generating matrix* $\mathbf{A}$*, number of cubature points found/time taken (in sec) using our* ℓ_1 *optimizer and the greedy solver, relative error achieved with a same number of cubature points/time taken (in sec) using NN-HTP solver. We set relative error to be* 1% *and use* 1000 *training poses for all our examples.*

space kinetic energy (brown line) and potential energy (red line) in order to reach the final timestepping equation. We have discussed three ways to evaluate our kinetic term: the accurate method $\mathbf{x}$; the high order polynomial approximation $\mathbf{x}_{pq}$; the cubature approximation $\mathbf{x}_c$.


$\exp(\omega(p))(\epsilon(p)+\mathrm{Id})-\mathrm{Id}=\tilde{\nabla}(\omega(p),\epsilon(p))$
$\mathbf{min}_{\mathbf{x}}\sum_{i=1}^{|T|}\|\nabla_i(\mathbf{Bx})-\tilde{\nabla}_i(\tilde{\mathbf{U}}\tilde{\mathbf{x}})\|^2|T_i|$
$\mathbf{x}_c(\tilde{\mathbf{x}})=\sum_{i\in\mathcal{C}}\mathbf{w}_i\Phi_i\tilde{\nabla}_i(\tilde{\mathbf{x}})$
$\tilde{\mathbf{U}}\tilde{\mathbf{x}}=\begin{pmatrix}\omega\\ \epsilon\end{pmatrix}$
$\mathbf{argmin}_{\mathbf{x}_{n+1}}V(\mathbf{x}_{n+1})+\frac{1}{2}\|\mathbf{x}_{n+1}-\mathbf{x}_n-\dot{\mathbf{x}}_n\Delta t\|^2_{\mathbf{M}/\Delta t^2}$
$\sum_{i=1}^{|T|}\tilde{W}(\epsilon_i)|T_i|=V(\tilde{\mathbf{x}})$
$\mathbf{Q}(\omega)\sqrt{D}\mathbf{Bx}_{pq}=\sqrt{D}\,[\mathbf{P}(\omega)(\epsilon+\mathrm{Id})-\mathbf{Q}]$

D The Algorithmic Complexity

Algorithm 2 Substep Using Polynomial Evaluation

```
Evaluate $\bar{\mathbf{Q}}$, $\frac{\partial\bar{\mathbf{Q}}}{\partial\tilde{\mathbf{x}}}$ with overhead $\mathcal{O}(|\mathbf{B}|^2|\tilde{\mathbf{U}}|^2)$
Evaluate $\bar{\mathbf{P}}$, $\frac{\partial\bar{\mathbf{P}}}{\partial\tilde{\mathbf{x}}}$ with overhead $\mathcal{O}(|\mathbf{B}||\tilde{\mathbf{U}}|^4)$
Evaluate $\bar{\mathbf{Q}}^{-1}$ with overhead $\mathcal{O}(|\mathbf{B}|^3)$
Solve for $\mathbf{x}=\bar{\mathbf{Q}}^{-1}\bar{\mathbf{P}}$ with overhead $\mathcal{O}(|\mathbf{B}|^2)$
Solve for $\frac{\partial\mathbf{x}}{\partial\tilde{\mathbf{x}}}=\bar{\mathbf{Q}}^{-1}(\frac{\partial\bar{\mathbf{P}}}{\partial\tilde{\mathbf{x}}}-\frac{\partial\bar{\mathbf{Q}}}{\partial\tilde{\mathbf{x}}}\mathbf{:x})$ with overhead $\mathcal{O}(|\mathbf{B}|^2|\tilde{\mathbf{U}}|)$
Evaluate $\tilde{\mathbf{M}}=\frac{\partial\mathbf{x}}{\partial\tilde{\mathbf{x}}}^T\mathbf{M}\frac{\partial\mathbf{x}}{\partial\tilde{\mathbf{x}}}$ with overhead $\mathcal{O}(|\mathbf{B}|^2|\tilde{\mathbf{U}}|+|\mathbf{B}||\tilde{\mathbf{U}}|^2)$
```

Algorithm 3 Substep Using Cubature Optimization

```
Evaluate $\mathbf{x}$, $\frac{\partial\mathbf{x}}{\partial\tilde{\mathbf{x}}}$ from Equation 12 with overhead $\mathcal{O}(|\mathcal{C}||\mathbf{B}||\tilde{\mathbf{U}}|)$
Evaluate $\tilde{\mathbf{M}}=\frac{\partial\mathbf{x}}{\partial\tilde{\mathbf{x}}}^T\mathbf{M}\frac{\partial\mathbf{x}}{\partial\tilde{\mathbf{x}}}$ with overhead $\mathcal{O}(|\mathbf{B}|^2|\tilde{\mathbf{U}}|+|\mathbf{B}||\tilde{\mathbf{U}}|^2)$
```

We derive the algorithmic complexity of our method in this section. We start from the version using $(p=3,q=2)$ polynomial approximation. In this case, the overhead of each step is illustrated in Algorithm 2, where the dominant terms are $\mathcal{O}(|\mathbf{B}|^2|\tilde{\mathbf{U}}|^2+|\mathbf{B}||\tilde{\mathbf{U}}|^4+|\mathbf{B}|^3)$. And for the version using cubature optimization for Jacobian approximation, the procedure is illustrated in Algorithm 3, where the dominant terms are: $\mathcal{O}(|\mathcal{C}||\mathbf{B}||\tilde{\mathbf{U}}|+|\mathbf{B}|^2|\tilde{\mathbf{U}}|+|\mathbf{B}||\tilde{\mathbf{U}}|^2)$.