Subspace Mixed Finite Elements for Real-Time Heterogeneous Elastodynamics

Ty Trusty*
University of Toronto
Canada

Otman Benchekroun*
University of Toronto
Canada

Eitan Grinspun
University of Toronto
Canada

Danny M. Kaufman
University of Toronto and Adobe Research
U.S.A.

David I.W. Levin
University of Toronto and NVIDIA
Canada

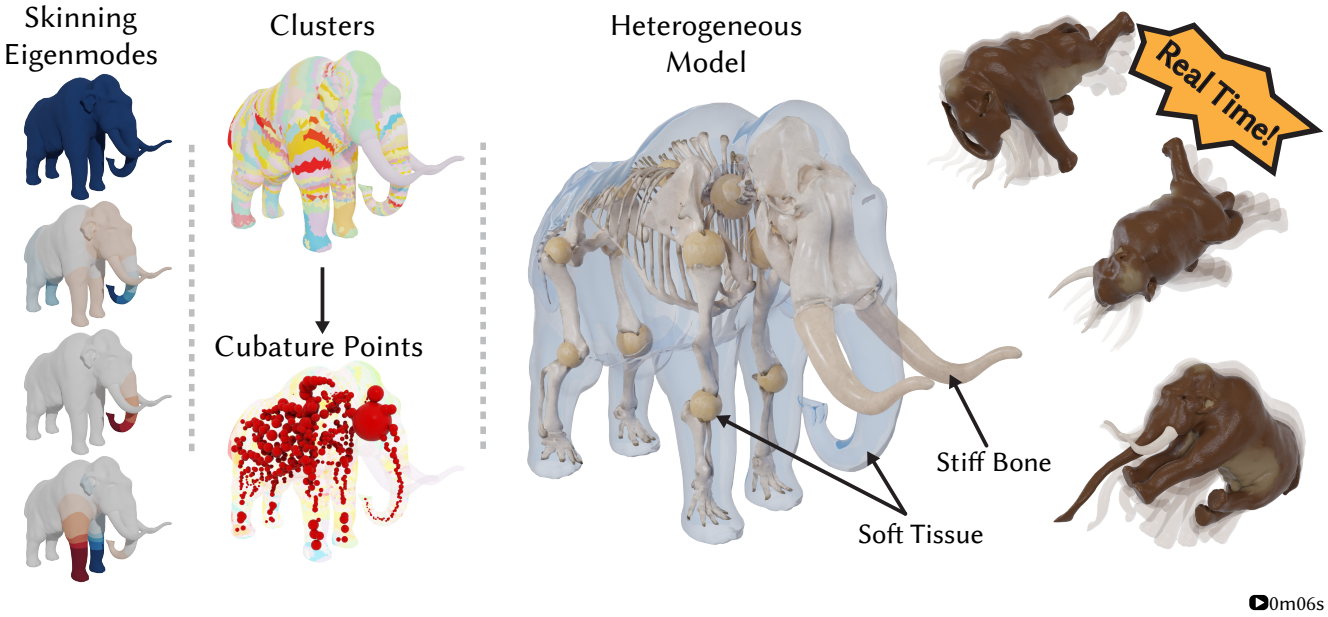

Figure 1: We propose a reduced space mixed finite element method (MFEM) built on a Skinning Eigenmode subspace and material-aware cubature scheme. Our solver is well-suited for simulating scenes with large material and geometric heterogeneities in real-time. This mammoth geometry is composed of 98,175 vertices and 531,565 tetrahedral elements and with a heterogenous composition of widely varying materials of muscles ($E = 5 \times 10^5$ Pa), joints ($E = 1 \times 10^5$ Pa), and bone ($E = 1 \times 10^{10}$ Pa). The resulting simulation runs at 120 frames per second (FPS).

ABSTRACT

Real-time elastodynamic solvers are well-suited for the rapid simulation of homogeneous elastic materials, with high-rates generally enabled by aggressive early termination of timestep solves. Unfortunately, the introduction of strong domain heterogeneities can make these solvers slow to converge. Stopping the solve short creates visible damping artifacts and rotational errors. To address these challenges we develop a reduced mixed finite element solver that preserves rich rotational motion, even at low-iteration regimes. Specifically, this solver augments time-step solve optimizations with auxillary stretch degrees of freedom at mesh elements, and maintains consistency with the primary positional degrees of freedoms at mesh nodes via explicit constraints. We make use of a Skinning Eigenmode subspace for our positional degrees of freedom. We accelerate integration of non-linear elastic energies with a cubature approximation, placing stretch degrees of freedom at cubature points. Across a wide range of examples we demonstrate that this subspace is particularly well suited for heterogeneous material simulation. Our resulting method is a subspace mixed finite element method completely decoupled from the resolution of the mesh that is well-suited for real-time simulation of heterogeneous domains.

*Indicates joint first authors.

Authors' addresses: Ty Trusty*, University of Toronto, Canada; Otman Benchekroun*, University of Toronto, Canada ; Eitan Grinspun, University of Toronto, Canada; Danny M. Kaufman, University of Toronto and Adobe Research, U.S.A.; David I.W. Levin, University of Toronto and NVIDIA, Canada.

KEYWORDS

Mixed FEM, Heterogeneous Materials

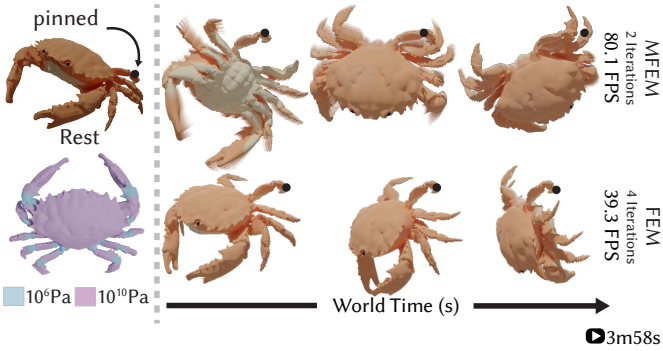

Figure 2: A crab with a hard shell (E=1e10 Pa) and soft joints (E=1e6 Pa) is simulated with our subspace MFEM and skinning subspace FEM. With only 2 solver iterations MFEM exhibits correct rotational and elastic behavior, whereas subspace FEM with 4 iterations – and consequently half the frame rate – exhibits noticeable damping.

1 INTRODUCTION

All elastic objects in the real world are heterogeneous. Yet many of our elastodynamic simulations, especially in the *real-time* regime, are evaluated on homogeneous materials. Applying them to *heterogeneous* materials makes these solvers slow to converge, leading to visual artifacts such as artifical damping. These convergence artifacts are exacerbated by a strict compute-time budget; a slowly converging solve will have to be cut short as new simulation frames are demanded. The mixed finite element method (MFEM) introduced by Trusty et al. [2022] shows success in preserving energetic motion for full space heterogeneous simulations. Unfortunately, their method scales in complexity with the full mesh resolution; larger meshes quickly become unavailable for real-time simulations. For example, the mammoth example shown in Fig. 1 runs at 263 seconds per iteration (maximum 0.003 FPS), far from the common real-time target of 60 FPS.

On the other hand, subspace methods have been popular in graphics for accelerating optimization problems since Pentland and Williams [1989]. However, subspace methods have very well known weaknesses in representing extreme rotational motion [Choi and Ko 2005], which is made worse by material or geometric heterogeneities. Recently Benchekroun et al. [2023] introduce Skinning Eigenmodes, a linear subspace that preserves rotation invariance during subspace simulation, and can represent rotational motion.

With the goal of simulating heterogeneous elastodynamic materials in real-time, we propose a subspace MFEM solver that makes use of a Skinning Eigenmode subspace and an accompanying heterogeneity-aware cubature approximation scheme. This solver inherits both the material-robust convergence benefits of its full space predecessor as well as the speed and reduced dimensionality provided by the subspace. The result is a convergent simulation for heterogeneous domains whose complexity is entirely decoupled from the resolution of the underlying mesh.

2 RELATED WORK

2.1 Subspaces for Heterogeneous Materials

Subspace simulation has been of interest in graphics since Pentland and Williams [1989], where a subspace is commonly formed via Linear Modal Analysis (LMA) of the elastic energy Hessian. These types of modes have well known drawbacks when used for accelerating elastodynamic simulation. Specifically, they struggle representing large non-linear deformations, such as rotations, which are a salient feature of heterogeneous material simulation.

Barbič and James [2005] use modal derivatives, whose aim is to supplement primary LMA modes with higher quality derivative modes to help the subspace stay up to date with the current deformation. Unfortunately, modal derivatives do not perfectly span rotations (See Fig. 5). Modal warping [Choi and Ko 2005], Rotation Strain coordinates [Huang et al. 2011], sub-structuring [Barbič and Zhao 2011; Kim and James 2011] and rigid-frame embedding [James and Pai 2002; Terzopoulos and Witkin 1988] all mitigate this issue by factoring out rotational motion and keeping track of it separately. These methods unfortunately scale in complexity with the number of rotations to be tracked, of which there may be many in a large-scale heterogeneous material. While non-linear subspaces via Deep Neural Networks have also been proposed [Shen et al. 2021], the resulting complexity of the subspace requires many optimization steps in order to reach a solution [Sharp et al. 2023].

Another option is to use linear skinning subspaces, which can represent rotations implicitly upon their construction. Many skinning subspaces are chosen to represent smooth, local deformations [Brandt et al. 2018; Jacobson et al. 2012; Lan et al. 2020; Wang et al. 2015]. Smoothness, however, is not an optimal prior when a material has sharp transitions in material properties (See Fig. 8 and Fig. 3). Faure et al. [2011] describe a local, material-sensitive set of skinning weights. Their construction rely on additional user parameters to control the smoothness of the subspace. Benchekroun et al. [2023] propose Skinning Eigenmodes, a method of constructing globally supported skinning weights that reflect material properties from an eigendecomposition of the elastic energy Laplacian. The globality of this subspace allows for a compact representation of fine scale motion. While material aware, Skinning Eigenmodes in standard finite element solvers still suffer degraded convergence with large heterogeneities. We show that the combination of this subspace with a mixed finite element method is the key to robust real-time heterogenous simulation.

2.2 Fast Elastic Solvers for Heterogeneous Materials

Standard discretizations struggle when applied to heterogeneous elastodynamics problems which motivates our use of a mixed discretization [Trusty et al. 2022]. Typically, work on efficient simulation of heterogenous materials is centered around homogenization or numerical coarsening [Chen et al. 2017, 2015; Kharevych et al. 2009] which uses a coarse (lower than material assignment resolution) mesh as a reduced space and homogenizes material properties within each coarse element. However these methods only simulate aggregate material behavior – by construction they cannot accurately represent the

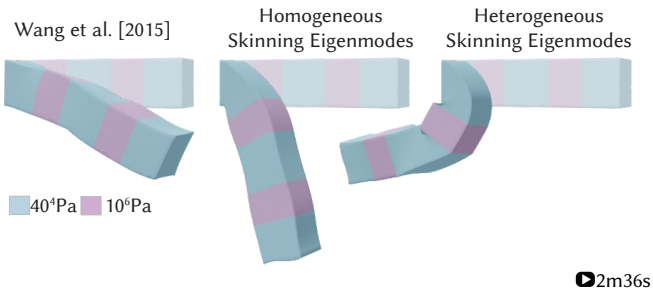

Figure 3: Material sensitive skinning modes directly lead to richer motion for heterogeneous materials.

heterogeneous strains induced by material or geometric heterogeneity as the shape functions themselves are typically polynomial within each element. While Chen et al. [2018] derive material-adaptive multi-resolution basis functions, their approach depends on a non-physical Rotation-Strain post-warping effect using Rotation Strain coordinates, which they show produces artifacts. Rather, our material-aware skinning weights allow more visually exacting reconstruction of animated motion using fewer degrees-of-freedom.

Another common method for accelerating non-linear elastic PDEs discretized via Finite Elements is by a cubature approximation [An et al. 2008; von Tycowicz et al. 2013] of the elastic energy. This approximates the total elastic energy with reweighed contributions from a set of sparsely sampled representative tetrahedra. The computation of these cubature points and weights is done in an expensive offline training phase, requiring the user to provide data with prior knowledge of the deformations they expect to encounter at run-time. Instead, Jacobson et al. [2012] accelerate an elastostatic solver by allowing tetrahedra to share strain quantities with other tetrahedra in their cluster. These clusters are found efficiently via a k-means clustering on the skinning weights, allowing the clusters to reflect the properties of the skinning weights. We combine both approaches: we find a strong set of cubature points as the centroid of the k-means clusters without requiring a training phase.

3 FULL SPACE MIXED FEM

Our starting point is the mixed finite element method (MFEM) of Trusty et al. [2022]. We discretize the domain with a tetrahedral mesh with $|\mathcal{V}|$ vertices and $|\mathcal{T}|$ elements.

We store positions as the coefficients $\boldsymbol{x} \in \mathbb{R}^{3|\mathcal{V}|\times 1}$ of Lagrange finite elements. We then introduce stretch degrees of freedom (DOFs) $\boldsymbol{s} \in \mathbb{R}^{6|\mathcal{T}|\times 1}$, corresponding to the symmetric stretch component of the polar decomposition of the deformation gradient ($F = RS$).

We maintain consistency between our positional and stretch DOFs $\boldsymbol{c}(\boldsymbol{x}, \boldsymbol{s}) = \boldsymbol{D}(\bar{\boldsymbol{s}}(\boldsymbol{x}) - \boldsymbol{s})$, where $\bar{\boldsymbol{s}}$ evaluates the stretch at each element as a function of $\boldsymbol{x}$. We make use of $D = \text{diag}([1\ 1\ 1\ 2\ 2\ 2])$ and $\boldsymbol{D} = \boldsymbol{I}_{|\mathcal{T}|} \otimes D$ to account for the symmetric off-diagonal terms in S.

This leads to the MFEM elastodynamic optimization problem,

$$\boldsymbol{x}^*, \boldsymbol{s}^*, \boldsymbol{\lambda}^* = \underset{\boldsymbol{x},\boldsymbol{s}}{\text{argmin}} \max_{\boldsymbol{\lambda}} \Psi_x(x) + \Psi_s(s) + \boldsymbol{\lambda}^T \boldsymbol{c}(\boldsymbol{x}, \boldsymbol{s}) \tag{1}$$

where $\Psi_{\boldsymbol{x}}(\boldsymbol{x})$ is the *quadratic* component of the elastodynamic energy that depends only on positional DOFs, $\Psi_{\boldsymbol{s}}(\boldsymbol{s})$ is the elastic strain energy written in terms of the stretch DOFs and the last term enforces the consistency constraint with Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{6|\mathcal{T}|}$. The solution is characterized by the KKT optimality conditions, which can be solved via a Newton-type method [Trusty et al. 2022].

4 SUBSPACE MIXED FEM

We introduce a linear subspace $\boldsymbol{B} \in \mathbb{R}^{3|\mathcal{V}|\times r}$ for our positional DOFs, and approximate them with $\boldsymbol{x} \approx \boldsymbol{B}\boldsymbol{u}$, where $\boldsymbol{u} \in \mathbb{R}^r$, $r \ll 3|\mathcal{V}|$, are subspace coefficients. With some precomputations, this subspace can be used to evaluate the quadratic $\Psi_x(\cdot)$ at run-time explicitly in terms of reduced dimensions. By contrast, fast evaluation of the non-linear stretch energy, $\Psi_s(\cdot)$, and the consistency constraint term requires the use of numerical cubature [An et al. 2008]. Evaluating these corresponding energy densities over a subset of all tetrahedra, C, and reweighing their contributions according to a precomputed cubature weight yields

$$\Psi_s(\boldsymbol{s}) \approx \sum_c^{|C|} w_c \psi_z(z_c) = \Psi_z(\boldsymbol{z}), \tag{2}$$

$$\boldsymbol{\lambda}^T \boldsymbol{c}(\boldsymbol{x}, \boldsymbol{s}) \approx \sum_c^{|C|} w_c \boldsymbol{\mu}_c^T D(\bar{z}_c(\boldsymbol{u}) - z_c) = \boldsymbol{\mu}^T \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{z}) \tag{3}$$

where we have introduced $\boldsymbol{z} \in \mathbb{R}^{6|C|}$, the stretch DOFs at the cubature tetrahedra, and $\boldsymbol{\mu} \in \mathbb{R}^{6|C|}$, the Lagrange multipliers enforcing the consistency constraint at the cubature points.

We can finally rewrite the optimization problem entirely in terms of reduced space DOFs:

$$\boldsymbol{u}^*, \boldsymbol{z}^*, \boldsymbol{\mu}^* = \underset{\boldsymbol{u},\boldsymbol{z}}{\text{argmin}} \max_{\boldsymbol{\mu}} \Psi_u(\boldsymbol{u}) + \Psi_z(\boldsymbol{z}) + \boldsymbol{\mu}^T \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{z}) \tag{4}$$

We solve this optimization using Sequential Quadratic Programming (SQP), where search directions for the $(k+1)$-th iteration are found by solving the KKT system

$$\begin{bmatrix} \boldsymbol{H}_u & \boldsymbol{0} & \boldsymbol{G}_u^T \\ \boldsymbol{0} & \boldsymbol{H}_z & \boldsymbol{G}_z \\ \boldsymbol{G}_u & \boldsymbol{G}_z & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{du} \\ \boldsymbol{dz} \\ \boldsymbol{\mu} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_z \\ \boldsymbol{f}_\mu \end{bmatrix}, \tag{5}$$

where all quantities are evaluated using DOFs from the previous iteration, $\{\boldsymbol{u}^k,\ \boldsymbol{z}^k\}$. $\boldsymbol{H}_u = \boldsymbol{B}^T \boldsymbol{H}_x \boldsymbol{B} \in \mathbb{R}^{r\times r}$ and $\boldsymbol{H}_z = \frac{\partial^2 \Psi_z}{\partial^2 \boldsymbol{z}} \in \mathbb{R}^{6|C|\times 6|C|}$ are the reduced Hessians (with $\boldsymbol{H}_x$ being the full-space Hessian); $\boldsymbol{f}_u = \boldsymbol{B}^T \boldsymbol{f}_x \in \mathbb{R}^r$, $\boldsymbol{f}_z = \frac{\partial \Psi_z}{\partial \boldsymbol{z}} \in \mathbb{R}^{6|C|}$, and $\boldsymbol{f}_\mu = \boldsymbol{g}(\boldsymbol{u}^k, \boldsymbol{z}^k) \in \mathbb{R}^{6|C|}$ are the reduced forces (with $\boldsymbol{f}_x$ being the full-space force); $\boldsymbol{G}_u = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}} \in \mathbb{R}^{6|C|\times r}$ and $\boldsymbol{G}_z = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{z}} \in \mathbb{R}^{6|C|\times 6|C|}$ are the reduced space constraint Jacobians. The transpose is omitted from $\boldsymbol{G}_z$ since it is a diagonal matrix of cubature weights.

We condense this system by applying a series of Schur complements so that for $\boldsymbol{du}$ we instead solve

$$(\boldsymbol{H}_u + \boldsymbol{K})\boldsymbol{du} = -\boldsymbol{f}_u + \boldsymbol{G}_u^T \boldsymbol{G}_z^{-1}(\boldsymbol{f}_z - \boldsymbol{H}_z \boldsymbol{G}_z^{-1} \boldsymbol{f}_\mu), \tag{6}$$

where $\boldsymbol{K} = \boldsymbol{G}_u \boldsymbol{G}_z^{-1} \boldsymbol{H}_z \boldsymbol{G}_z^{-1} \boldsymbol{G}_u^T$, and for $\boldsymbol{dz}$ and $\boldsymbol{\mu}$ we solve

$$\boldsymbol{dz} = -\boldsymbol{G}_z^{-1}(\boldsymbol{f}_\mu + \boldsymbol{G}_u \boldsymbol{du}), \tag{7}$$

$$\boldsymbol{\mu} = -\boldsymbol{G}_z^{-1}(\boldsymbol{f}_z + \boldsymbol{H}_z \boldsymbol{dz}). \tag{8}$$

The updates for the next SQP iteration are then $\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \alpha \boldsymbol{du}$ and $\boldsymbol{z}^{k+1} = \boldsymbol{z}^k + \alpha \boldsymbol{dz}$, where α is a step size given by backtracking line search over the Lagrangian, $\mathcal{L}(\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{\mu}) = \Psi_u(\boldsymbol{u}) + \Psi_z(\boldsymbol{z}) + \boldsymbol{\mu}^T \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{z})$. In this final form, none of the terms depend on a full space quantity, so the update for $\boldsymbol{du}$ is efficiently solved with a direct dense linear solver,

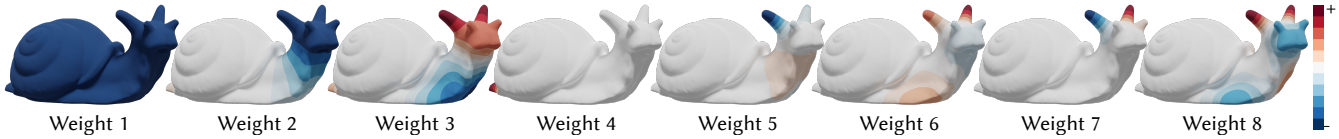

Figure 4: The skinning weights we get from Skinning Eigenmodes are naturally material aware. High frequency modes are concentrated on soft parts of the snail, which are more likely to exhibit rich deformation. In contrast, the stiff shell only has access to a constant skinning weight (shared by all parts of the snail), allowing rigid motion to be producible within our skinning subspace.

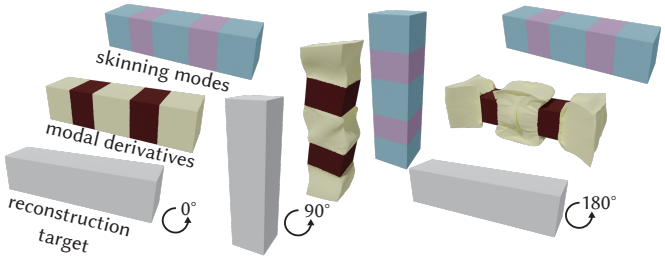

Figure 5: Modal derivatives are not suited for reconstructing rotations on the input shape. Fixing these artifacts typically requires explicitly tracking a rigid frame [Terzopoulos and Witkin 1988].

and the updates for $\boldsymbol{dz}$ and $\boldsymbol{\mu}$ are local and performed in parallel, making their cost negligible.

5 SUBSPACE CONSTRUCTION

5.1 Skinning Eigenmode Subspace

There are many ways to construct our positional subspace $\boldsymbol{B}$. We opt for using a skinning subspace [Brandt et al. 2018; Hahn et al. 2012]. As discussed by Benchekroun et al. [2023], these subspaces span rotations (as shown in Fig. 5), a particularly salient feature for heterogeneous stiff materials.

Other linear subspaces, such as modal derivatives [Barbič and James 2005], do not generally span rotations [Benchekroun et al. 2023]. For free flying motion, this limitation may be addressed by embedding a rigid frame that is tracked explicitly during the simulation [Terzopoulos and Witkin 1988]. For heterogeneous materials with multiple independent stiff components that do not necessarily rotate in unison (such as the bar in Fig. 5), keeping track of potentially many rigid frames becomes increasingly inconvenient, and scales in complexity with the heterogeneity of the material.

We opt to build our skinning subspace using Skinning Eigenmodes [Benchekroun et al. 2023], which span rotations and furthermore provide a straightforward automatic method to generating material-aware skinning weights (see Fig. 4). Specifically, we obtain a set of skinning weights $\boldsymbol{W} \in \mathbb{R}^{|\mathcal{V}|\times m}$ by solving the weight space generalized eigenvalue problem,

$$\boldsymbol{H}_w \boldsymbol{W} = \boldsymbol{M}_w \boldsymbol{W} \boldsymbol{\Gamma}. \tag{9}$$

Above, $\boldsymbol{H}_w = \frac{\partial^2 \Psi}{\partial^2 \boldsymbol{x}_1} + \frac{\partial^2 \Psi}{\partial^2 \boldsymbol{x}_2} + \frac{\partial^2 \Psi}{\partial^2 \boldsymbol{x}_3} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the elastic energy Laplacian (subscripts $\{1, 2, 3\}$ denote each of the 3 dimensions) and $\boldsymbol{M}_w \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the scalar mass matrix. The use of the elastic energy Laplacian is what provides this subspace with its material-aware properties.

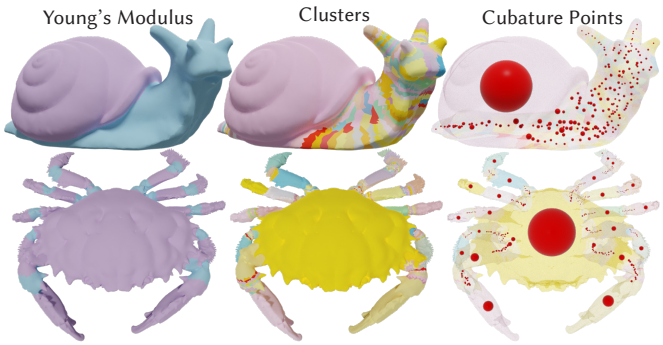

Figure 6: Our cubature points are found as the centroids of each k-means cluster. Note that our centroids are sensitive to the heterogeneity of the Young's modulus. Stiffer regions can have their strain be approximated with fewer cubature points.

The decomposition provides us with eigenvalues $\boldsymbol{\Gamma}$ and eigenvectors $\boldsymbol{W}$, the latter of which correspond to linear blend skinning weights. The use of the elastic energy Laplacian is what provides this subspace with its material-aware properties. These skinning weights $\boldsymbol{W}$ can then be used to construct our subspace basis $\boldsymbol{B}$ using the standard linear blend skinning Jacobian formula,

$$\boldsymbol{B} = \boldsymbol{I}_3 \otimes ((\boldsymbol{1}_m^T \otimes \bar{\boldsymbol{X}}) \odot (\boldsymbol{W} \otimes \boldsymbol{1}_4^T)), \tag{10}$$

where $\bar{\boldsymbol{X}} \in \mathbb{R}^{|\mathcal{V}|\times 4}$ are the rest positions in homogeneous coordinates. We can relate m, the number of skinning weights to our subspace DOFs via $r = 12m$.

5.2 Cubature Construction

A cubature scheme is characterized by a set of cubature points, and their corresponding cubature weights. A good cubature scheme is crucial for the quality of our subspace approximation. Undersampling leads to spurious deformations [McAdams et al. 2011], whereas excessive sampling introduces unnecessary cost.

A cubature scheme can be optimized to fit a training dataset [An et al. 2008]. Cubature weights are computed via a Non-Negative Least Squares (NNLS) fitting of the forces observed in the training set. Cubature points are then greedily added at elements where the current cubature fitting most poorly reconstructs the training forces. This approach is well suited for scenarios where the user knows a priori the types of deformation they want to approximate. However, for scenarios where a user is exploring deformations for potentially many meshes at a time, the requirement of building a good cubature

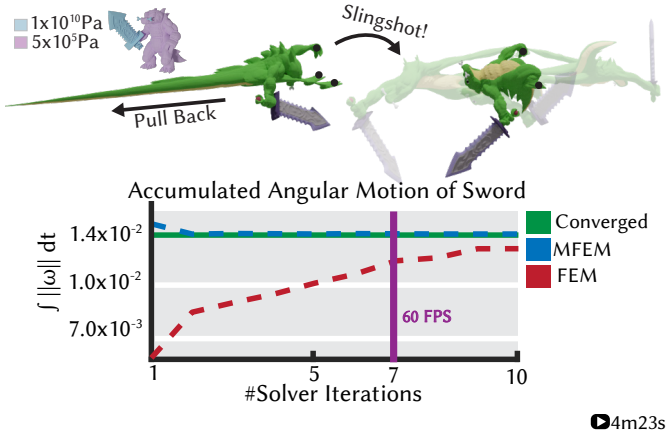

Figure 7: Our subspace MFEM solver almost perfectly reproduces the angular motion of the sword over the first 25 simulation timesteps, whereas FEM consistently underestimates it.

training set as well as the time it takes to iteratively solve a large NNLS problem for the cubature weights can overly constrain the creative process. Yang et al. [2015] propose an avenue for acceleration based on a Preconditioned Conjugate Gradient method.

We propose an alternative, fast and simpler method for constructing our cubature approximation that is well suited for heterogeneous materials, inspired by the clustering scheme of Jacobson et al. [2012]. To sample cubature tetrahedra, we cluster our domain and choose the tetrahedra closest to the centroid of each cluster. We construct these clusters from a k-means clustering on our skinning weights.

$$l = \text{kmeans}(\boldsymbol{W}_{\mathcal{T}}\boldsymbol{\Gamma}^{-2}, |C|), \tag{11}$$

where $\boldsymbol{W}_{\mathcal{T}}$ are our skinning weights averaged from the vertices to the elements. We weigh each skinning weight by its inverse squared eigenvalue $\boldsymbol{\Gamma}^{-2}$ in order to favor weights that correspond to low energy deformations, which are more likely to occur at run-time. We then compute cluster centroids and choose our cubature points as the tets closest to each centroid. The cubature weights are then trivially computed as the mass of each cluster. Using the skinning weights as our clustering features allows the cubature scheme to reflect the properties of our skinning weights, such as material and geometric heterogeneity, or any pinning constraints that may have been imposed on our skinning weights. In particular, note from Fig. 6 and Fig. 16 that our cubature sampling parallels the anticipated strain heterogeneity of the domain: regions more likely to deform, such as soft regions or thin regions, will be sampled relatively densely; regions less likely to deform, such as stiff or thick regions, will be sampled relatively sparsely, as shown in Fig. 16.

6 IMPLEMENTATION

We implement our method in both Matlab and C++, with geometry processing utilities provided by libigl [Jacobson et al. 2018] and gptoolbox [Jacobson et al. 2021] and physics utilities provided by Bartels [Levin 2018]. Our C++ code is parallelized with OpenMP [Chandra et al. 2001]. For modelling and rendering we use Blender [R Core Team 2013]. To solve the Generalized Eigenvalue Problem in Eq. (9), as well as the k-means clustering, we use Matlab's eigs() and kmeans() functions. To solve for the search direction (Eq. (6)) in each Newton iteration we use Eigen's [Guennebaud et al. 2010] SimplicialLLT direct solver.

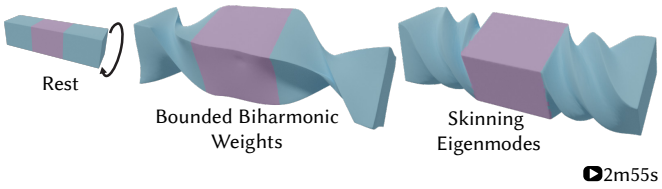

Figure 8: Smooth local skinning weights (left), such as Bounded Biharmonic Weights [Jacobson et al. 2011] are not optimal for modeling materials with sharp heterogeneities. Skinning Eigenmodes (right) are material sensitive and lead to a sharp resolution of extreme twisting motions. Both simulations use MFEM.

Algorithm 1 provides pseudocode for a single simulation step of our subspace Mixed-FEM solver.

Matrices coloured in blue remain constant throughout the simulation, whereas matrices in red change every timestep, but remain fixed throughout Newton iterations.

At the end of each simulation step, mesh geometry is usually queried for visualization purposes. Standard subspaces require full space projection, $\boldsymbol{x} = \boldsymbol{Bu}$, to in order to capture the deformed mesh geometry. This is a full space operation that can easily become the bottleneck for any subspace simulation application. Instead, we perform this step entirely on the GPU [Barbič and James 2005]. As our subspace is a skinning subspace, it's especially convenient to perform this step in the vertex shader [Benchekroun et al. 2023]: we pass the skinning weights forming our subspace $\boldsymbol{W}$ to our vertex shader as vertex attributes in a preprocessing step, and send our reduced space coordinates $\boldsymbol{u}$ as uniforms each draw call. As Table 1 shows, this effectively makes the computation time for this step negligible compared to the other stages of our pipeline.

Algorithm 1: Performs one simulation step of our subspace Mixed-FEM solver

Function `simulationStep`$(\boldsymbol{u}, z)$**:**
  while not converged **do**
    $\boldsymbol{H}_z, \boldsymbol{H}_u \leftarrow$ hessians$(\boldsymbol{u}, z)$
    $\boldsymbol{f}_z, \boldsymbol{f}_u, \boldsymbol{f}_\mu \leftarrow$ gradients$(\boldsymbol{u}, z)$
    $\boldsymbol{G}_z, \boldsymbol{G}_u \leftarrow$ constraintGradients$(\boldsymbol{u}, z)$
    $\boldsymbol{K} \leftarrow \boldsymbol{G}_u\boldsymbol{G}_z^{-1}\boldsymbol{H}_z\boldsymbol{G}_z^{-1}\boldsymbol{G}_u^T$ // assemble stiffness matrix

    // Global linear solve
    $\boldsymbol{du} \leftarrow (\boldsymbol{H}_u + \boldsymbol{K})^{-1}(\boldsymbol{G}_u^T\boldsymbol{G}_z^{-1}(\boldsymbol{f}_z - \boldsymbol{H}_z\boldsymbol{G}_z^{-1}\boldsymbol{f}_\mu) - \boldsymbol{f}_u)$
    // Local solves
    $\boldsymbol{dz} = -\boldsymbol{G}_z^{-1}(\boldsymbol{f}_\mu + \boldsymbol{G}_u\boldsymbol{du})$
    $\boldsymbol{\mu} = -\boldsymbol{G}_z^{-1}(\boldsymbol{f}_z + \boldsymbol{H}_z\boldsymbol{dz})$

    $\alpha \leftarrow$ lineSearch$(\boldsymbol{du}, \boldsymbol{dz}, \boldsymbol{\mu})$
    $\boldsymbol{u} \leftarrow \boldsymbol{u} + \alpha\boldsymbol{du}$
    $z \leftarrow z + \alpha\boldsymbol{dz}$
  return $\boldsymbol{u}, z$

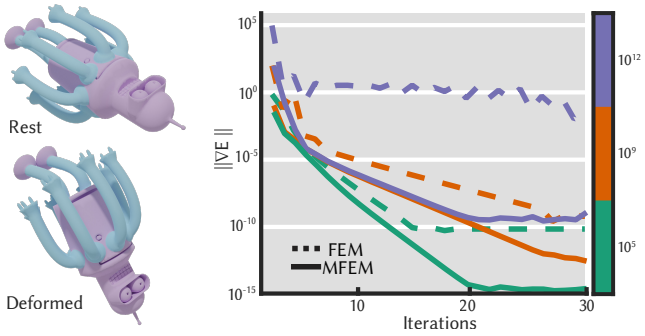

Figure 9: We start the octobot mesh in the deformed state (bottom left). We then run 30 iterations of FEM and MFEM iterations as we vary the Young's modulus of the stiff region (shown in purple). The soft region (shown in blue) remains fixed at $1 \times 10^5 Pa$. For large material heterogeneities, FEM takes much longer to converge than our MFEM.

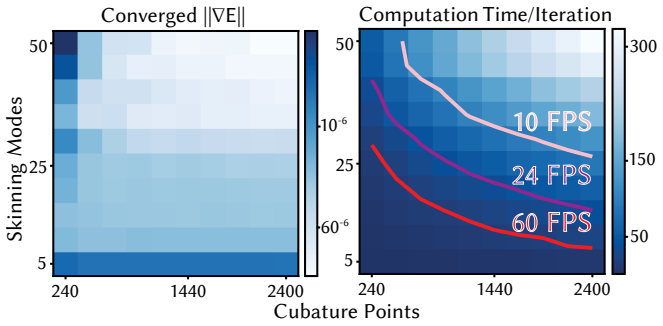

Figure 10: Pareto-search exploring the cost/benefit tradeoff of varying our two subspace parameters, the number of skinning modes m and the number of cubature points $|C|$. We visualize the resulting full space energy gradient after convergence (left) as well as the computation time per newton iteration (right).

7 RESULTS & DISCUSSION

In the following examples, without loss of generality, we apply implicit Euler time stepping and use the fixed corotational (FCR) elasticity model [Stomakhin et al. 2012] (any hyperelastic model is applicable Fig. 15). MFEM denotes our subspace MFEM solver and FEM denotes a solver which uses the same skinning subspace, but applied in standard FEM.

7.1 Iteration Ablation

The advantages of our subspace MFEM solver become especially apparent for truncated real-time simulations with large heterogeneities. Fig. 2 shows a crab model with a stiff shell and soft joints pinned at one of its hind legs and falling under gravity. The subspace simulation is carried out with 16 skinning modes and 342 cubature points. We allow only two solver iterations every timestep and compare results between our subspace MFEM solver, and a traditional subspace FEM solver. The FEM example manifests a very common solver truncation artifact which *heavily* damps motion. By contrast, our MFEM solver easily allows the crab to exhibit rich rigid motion.

7.2 Complex Deformation

As shown in Fig. 8, our subspace solver can reproduce extreme twisting motions for a heterogeneous candy with a hard middle (1×10^{10} Pa) and soft extremities (1×10^6 Pa). The twist is enforced via a spring force, and the whole simulation is carried out in a subspace of 16 skinning weights and 192 cubature points. We compare our subspace's result to one created with bounded biharmonic weights [Jacobson et al. 2011; Lan et al. 2020] with weight handles located about samples found via farthest point sampling. While bounded biharmonic weights provide a smooth basis for simulation, this subspace is not aware of the heterogeneity present within the candy's domain, resulting in most of the modes locking their motions to ensure the stiff regions remain undeformed, leaving little degrees of freedom available to accomodate the twist. By contrast, our skinning eigenmode subspace is sensitive to the material properties of the candy, which allows our simulation to better capture this sharp transition in the material properties of the domain.

7.3 Material Heterogeneity

Fig. 9 investigates how the extent of the heterogeneity affects the convergence of our subspace Mixed-FEM solve and specifically compares against the convergence of a traditional FEM simulation.

We start with the Octobot mesh in a deformed state and run both subspace FEM and MFEM solvers for a single timestep. The subspace used is composed of 16 skinning modes and 800 cubature points. We perform this experiment for 3 different Young's moduli and plot the iteration progression of the Newton decrement for each solve.

Fig. 1 stress tests our solver's ability to simulate large-scale models with a high number of material discontinuities. Here, a Mammoth with stiff skeleton bones (1×10^{10} Pa), soft joints (5×10^5 Pa), and softer muscle (1×10^5 Pa) is excited by an external periodic force applied on its back bone, moving it up and down and thrashing it around the scene. We observe energetic rag-doll rotational motion of the limbs and body, a detail noticeably absent from the unconverged subspace FEM simulation provided in the supplemental video.

7.4 Geometric Heterogeneity

Heterogenity of elastic moduli is just one possible source of large variations in elemental strains. Another possible source is the *geometry* of the domain; heterogeneous thickness, for instance, can lead to comparatively small and large strains in slender and thick regions, respectively (see Fig. 13).

In this example, we wind up a pendulum, twist it back a few times, and release it, allowing it to unwind and come to rest. We carry out the simulation in a subspace composed of 16 Skinning Modes and 400 cubature points (Fig. 16). Starting both methods at the twisted state, we simulate the unwinding with MFEM and FEM with one iteration per timestep. We observe that MFEM maintains the same energy preserving benefits, while FEM again exhibits rotation damping artifacts. This example uses homogeneous material properties, emphasizing that our method offers an advantage when the *strain* is heterogeneous, whether induced by constitutive or geometric properties.

7.5 Mode-Cubature Pareto Fronts

In Fig. 10, we investigate how both our subspace parameters, the number of skinning modes and the number of cubature points, affect both the accuracy of our subspace approximation as well as the time

Table 1: We report average times (in milliseconds) for one iteration of subspace MFEM/FEM and full-space simulations for meshes of various complexity. MFEM corresponds to a simulation step time for our subspace mixed FEM solver, FEM is the time for a subspace FEM solve step, and m and $|C|$ are, respectively, the number of skinning modes and cubature points used in both subspace solvers. Proj is the time for the full-space projection used in the subspace solvers. Lastly, Full MFEM is the time for a full-space MFEM iteration (Trusty et al. [2022]).

Mesh	$\|\mathcal{V}\|$	$\|\mathcal{T}\|$	m	$\|C\|$	**MFEM** (ms)	**FEM** (ms)	**Proj** (ms)	**Full MFEM** (ms)
Octobot (Fig. 9)	32 591	132 124	5	227	1.19	1.10	0.42	3,099.1
Gatorman (Fig. 7)	54, 235	227, 035	10	192	2.01	2.04	0.41	11,442.7
Mammoth (Fig. 1)	98, 175	531, 565	16	581	7.37	7.56	0.54	263,545
Crab (Fig. 2)	57,529	223, 565	16	342	5.87	5.51	0.49	7,483.75

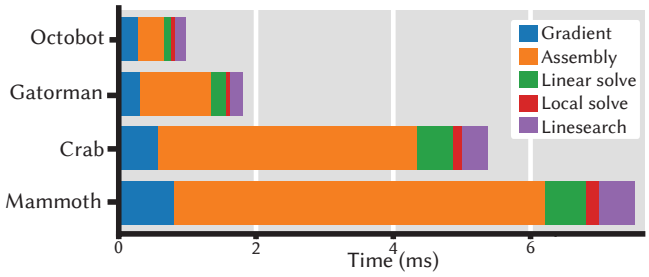

Figure 11: A timing breakdown of the core components for a single subspace MFEM simulation step of the Octobot (Fig. 9), Gatorman (Fig. 7), Crab (Fig. 2), and Mammoth (Fig. 1) simulations.

it takes to run. With the octobot starting in the bent position shown in Fig. 9 and allowing one simulation timestep to occur. We carry out this experiment assuming a homogeneous material with a Young's Modulus of 10^5 Pa.

To measure the accuracy of the converged solution (Fig. 10, left), we project our subspace solution back to the full space and evaluate the gradient of the full space elastodynamic optimization problem, which should be 0 for an accurate, converged result. In particular, note that the top left part of the grid-search makes use of many skinning modes, but still incurs a lot of error. This may seem unintuitive, but in fact stems from the introduction of a null space in our cubature approximation. Because we have so few cubature points in this regime, but many degrees of freedom for motion, spurious 0-energy oscillations manifest, a known cubature pitfall [McAdams et al. 2011]. In practice, we've found that setting the number of cubature points to 20× the number of skinning modes allows us to safely steer clear of this regime (Fig. 14), and all the examples reported in the rest of this paper do not exhibit these spurious deformations.

7.6 Timing Comparison and Discussion

Table 1 provides timings per MFEM and FEM iteration respectively. We compare timings for the Octobot (Fig. 9 and Fig. 10), the Gatorman (Fig. 7), the Mammoth (Fig. 1) and the Crab (Fig. 2) with varying subspace sizes. We also compare our solver's performance to the full-space MFEM solver of Trusty et al. [2022] and attain an average speed up of over 3 orders of magnitude. The additional computation required of our MFEM solver, when compared to FEM, is only the local stretch and Lagrange multiplier solves Eq. (7), which incurs an added $O(k)$ operations. This step only incurs a marginal difference as shown clearly in the timing breakdown of Fig. 11, which shows the MFEM simulation time is dominated by the $O(m^2k)$ dense $\boldsymbol{K}$ matrix assembly. With an asymptotically equivalent runtime as FEM, as well as more favorable energetic behavior at low-iterations, our solver enables real-time heterogenous domain simulation. In contrast, an equivalent subspace size requires FEM to perform more iterations (Fig. 2 and Fig. 7), making real-time simulation unattainable in many cases.

7.7 Artifact Tradeoffs between MFEM and FEM

While FEM exhibits extreme damping artifacts at low iteration counts, our solver can exhibit overly-energetic motion at low-iteration counts. Fig. 7 shows a subspace simulation on a case with extreme deformation and a localized external force.

Here, a soft gatorman (5×10^5 Pa) wielding a stiff (1×10^{12} Pa) sword is pulled back from its tail (using a soft penalty constraint) and slingshotted towards its enemies. The subspace for the simulation is composed of 10 skinning modes and 192 cubature points. We compare our subspace simulation results with those of a traditional FEM solver as we increase the number of solver iterations. Note that our subspace allows us to capture the localized rotational motion of the sword which is absent from the FEM solution. We find that MFEM is overly energetic at low iterations, causing an initial overestimation of angular motion. This results in jittering artifacts, which quickly disappear when taking more than one solver iteration. In contrast, FEM requires many more iterations to recover the correct rotational behavior (see the supplemental video for a demonstration).

7.8 Limitations of Global Subspaces

The global support of the skinning eigenmode subspace allows our simulation to efficiently capture complex ranges of motion. This can lead to noticeable global artifacts when a user is exciting a local region of the mesh. For example Fig. 12 shows a user bending the mammoth's hind leg, causing a jerk motion in the mammoth's trunk. We measure the amount of deformation induced by summing accumulated vertex displacements throughout the simulation. We increase the size of the subspace and observe that this artifact goes away as the number of skinning modes increases.

8 CONCLUSION AND FUTURE WORK

We have presented a new subspace mixed finite element method that offers real-time elastodynamic simulation for large-scale heterogeneous domains. Typical subspace methods experience degraded performance and jarring artifacts in these settings. We show that coupling a skinning eigenmode subspace with a mixed finite element

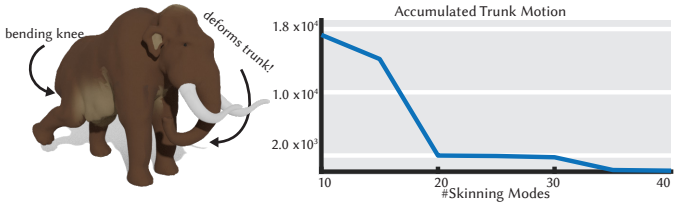

Figure 12: Because of our globally supported subspace, bending the knee of the mammoth causes the trunk to deform. This artifact goes away as we increase the size of the subspace.

method and applying a heterogeneity-aware cubature scheme yields a solver robust to extreme heterogeneities with performance decoupled from the resolution of the underlying mesh. Our method provides exciting opportunities for future work. There exists a complex cost/quality tradeoff between dense globally supported subspaces, and sparse locally supported subspaces. Understanding this tradeoff would help resolve the artifacts in Fig. 12 and would pave the way to robust reduced-space contact simulation – a difficult open problem for reduced space methods [Lan et al. 2020]. Finally, we believe our subspace MFEM solver could be extended for use in physics based inverse design in engineering and biomechanics where domain heterogeneities are commonplace.

ACKNOWLEDGEMENTS

We would like to thank Aravind Ramakrishnan, Abhishek Madhan, Chief Trusty for proof-reading and Kevin Caldwell and Kevin Wang for providing simulation meshes. This research was made possible with the administrative help of our lab system administrator John Hancock and financial officer Xuan Dam. This research was funded by two NSERC Discovery grants, an Ontario Early Researchers Award and the Canada Research Chairs Program.

REFERENCES

Steven S. An, Theodore Kim, and Doug L. James. 2008. Optimizing Cubature for Efficient Integration of Subspace Deformations. In *ACM SIGGRAPH Asia 2008 Papers* (Singapore) *(SIGGRAPH Asia '08)*. Association for Computing Machinery, New York, NY, USA, Article 165, 10 pages. https://doi.org/10.1145/1457515.1409118

Jernej Barbič and Doug L. James. 2005. Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models. *ACM Trans. Graph.* 24, 3 (jul 2005), 982–990. https://doi.org/10.1145/1073204.1073300

Jernej Barbič and Yili Zhao. 2011. Real-Time Large-Deformation Substructuring. In *ACM SIGGRAPH 2011 Papers* (Vancouver, British Columbia, Canada) *(SIGGRAPH '11)*. Association for Computing Machinery, New York, NY, USA, Article 91, 8 pages. https://doi.org/10.1145/1964921.1964986

Otman Benchekroun, Jiayi Eris Zhang, Siddhartha Chaudhuri, Eitan Grinspun, Yi Zhou, and Alec Jacobson. 2023. Fast Complementary Dynamics via Skinning Eigenmodes. arXiv:2303.11886 [cs.GR]

Christopher Brandt, Elmar Eisemann, and Klaus Hildebrandt. 2018. Hyper-Reduced Projective Dynamics. *ACM Trans. Graph.* 37, 4, Article 80 (jul 2018), 13 pages. https://doi.org/10.1145/3197517.3201387

Rohit Chandra, Leo Dagum, David Kohr, Ramesh Menon, Dror Maydan, and Jeff McDonald. 2001. *Parallel programming in OpenMP*. Morgan kaufmann.

Desai Chen, David I. W. Levin, Wojciech Matusik, and Danny M. Kaufman. 2017. Dynamics-Aware Numerical Coarsening for Fabrication Design. *ACM Trans. Graph.* 36, 4, Article 84 (jul 2017), 15 pages. https://doi.org/10.1145/3072959.3073669

Desai Chen, David I. W. Levin, Shinjiro Sueda, and Wojciech Matusik. 2015. Data-Driven Finite Elements for Geometry and Material Design. *ACM Trans. Graph.* 34, 4, Article 74 (jul 2015), 10 pages. https://doi.org/10.1145/2766889

Jiong Chen, Hujun Bao, Tianyu Wang, Mathieu Desbrun, and Jin Huang. 2018. Numerical Coarsening Using Discontinuous Shape Functions. *ACM Trans. Graph.* 37, 4, Article 120 (jul 2018), 12 pages. https://doi.org/10.1145/3197517.3201386

Min Gyu Choi and Hyeong-Seok Ko. 2005. Modal Warping: Real-Time Simulation of Large Rotational Deformation and Manipulation. *IEEE Transactions on Visualization and Computer Graphics* 11, 1 (jan 2005), 91–101. https://doi.org/10.1109/TVCG.2005.13

François Faure, Benjamin Gilles, Guillaume Bousquet, and Dinesh K. Pai. 2011. Sparse Meshless Models of Complex Deformable Solids. *ACM Trans. Graph.* 30, 4, Article 73 (jul 2011), 10 pages. https://doi.org/10.1145/2010324.1964968

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

Fabian Hahn, Sebastian Martin, Bernhard Thomaszewski, Robert Sumner, Stelian Coros, and Markus Gross. 2012. Rig-Space Physics. *ACM Trans. Graph.* 31, 4, Article 72 (jul 2012), 8 pages. https://doi.org/10.1145/2185520.2185568

Jin Huang, Yiying Tong, Kun Zhou, Hujun Bao, and Mathieu Desbrun. 2011. Interactive Shape Interpolation through Controllable Dynamic Deformation. *IEEE Transactions on Visualization and Computer Graphics* 17, 7 (jul 2011), 983–992. https://doi.org/10.1109/TVCG.2010.109

Alec Jacobson et al. 2021. gptoolbox: Geometry Processing Toolbox. http://github.com/alecjacobson/gptoolbox.

Alec Jacobson, Ilya Baran, Ladislav Kavan, Jovan Popović, and Olga Sorkine. 2012. Fast Automatic Skinning Transformations. *ACM Trans. Graph.* 31, 4, Article 77 (jul 2012), 10 pages. https://doi.org/10.1145/2185520.2185573

Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. 2011. Bounded Biharmonic Weights for Real-Time Deformation. *ACM Trans. Graph.* 30, 4, Article 78 (jul 2011), 8 pages. https://doi.org/10.1145/2010324.1964973

Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. https://libigl.github.io/.

Doug L. James and Dinesh K. Pai. 2002. DyRT: Dynamic Response Textures for Real Time Deformation Simulation with Graphics Hardware. *ACM Trans. Graph.* 21, 3 (jul 2002), 582–585. https://doi.org/10.1145/566654.566621

Lily Kharevych, Patrick Mullen, Houman Owhadi, and Mathieu Desbrun. 2009. Numerical Coarsening of Inhomogeneous Elastic Materials. *ACM Trans. Graph.* 28, 3, Article 51 (jul 2009), 8 pages. https://doi.org/10.1145/1531326.1531357

Theodore Kim and Doug L. James. 2011. Physics-Based Character Skinning Using Multi-Domain Subspace Deformations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Vancouver, British Columbia, Canada) *(SCA '11)*. Association for Computing Machinery, New York, NY, USA, 63–72. https://doi.org/10.1145/2019406.2019415

Lei Lan, Ran Luo, Marco Fratarcangeli, Weiwei Xu, Huamin Wang, Xiaohu Guo, Junfeng Yao, and Yin Yang. 2020. Medial Elastics: Efficient and Collision-Ready Deformation via Medial Axis Transform. *ACM Trans. Graph.* 39, 3, Article 20 (apr 2020), 17 pages. https://doi.org/10.1145/3384515

David I.W. Levin. 2018. Bartels. https://libigl.github.io/.

Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. 2011. Efficient Elasticity for Character Skinning with Contact and Collisions. *ACM Trans. Graph.* 30, 4, Article 37 (jul 2011), 12 pages. https://doi.org/10.1145/2010324.1964932

A. Pentland and J. Williams. 1989. Good Vibrations: Modal Dynamics for Graphics and Animation. *SIGGRAPH Comput. Graph.* 23, 3 (jul 1989), 207–214. https://doi.org/10.1145/74334.74355

R Core Team. 2013. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/

Nicholas Sharp, Cristian Romero, Alec Jacobson, Etienne Vouga, Paul G Kry, David IW Levin, and Justin Solomon. 2023. Data-Free Learning of Reduced-Order Kinematics. (2023).

Siyuan Shen, Yin Yang, Tianjia Shao, He Wang, Chenfanfu Jiang, Lei Lan, and Kun Zhou. 2021. High-Order Differentiable Autoencoder for Nonlinear Model Reduction. *ACM Trans. Graph.* 40, 4, Article 68 (jul 2021), 15 pages. https://doi.org/10.1145/3450626.3459754

Alexey Stomakhin, Russell Howes, Craig Schroeder, and Joseph M. Teran. 2012. Energetically Consistent Invertible Elasticity. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Lausanne, Switzerland) *(SCA '12)*. Eurographics Association, Goslar, DEU, 25–32.

Demetri Terzopoulos and Andrew Witkin. 1988. Physically Based Models with Rigid and Deformable Components. *IEEE Comput. Graph. Appl.* 8, 6 (nov 1988), 41–51. https://doi.org/10.1109/38.20317

Ty Trusty, Danny Kaufman, and David I.W. Levin. 2022. Mixed Variational Finite Elements for Implicit Simulation of Deformables. In *SIGGRAPH Asia 2022 Conference Papers* (Daegu, Republic of Korea) *(SA '22)*. Association for Computing Machinery, New York, NY, USA, Article 40, 8 pages. https://doi.org/10.1145/3550469.3555418

Christoph von Tycowicz, Christian Schulz, Hans-Peter Seidel, and Klaus Hildebrandt. 2013. An Efficient Construction of Reduced Deformable Objects. *ACM Trans. Graph.* 32, 6, Article 213 (nov 2013), 10 pages. https://doi.org/10.1145/2508363.2508392

Yu Wang, Alec Jacobson, Jernej Barbič, and Ladislav Kavan. 2015. Linear Subspace Design for Real-Time Shape Deformation. *ACM Trans. Graph.* 34, 4, Article 57 (jul 2015), 11 pages. https://doi.org/10.1145/2766952

Yin Yang, Dingzeyu Li, Weiwei Xu, Yuan Tian, and Changxi Zheng. 2015. Expediting Precomputation for Reduced Deformable Simulation. *ACM Trans. Graph.* 34, 6, Article 243 (nov 2015), 13 pages. https://doi.org/10.1145/2816795.2818089

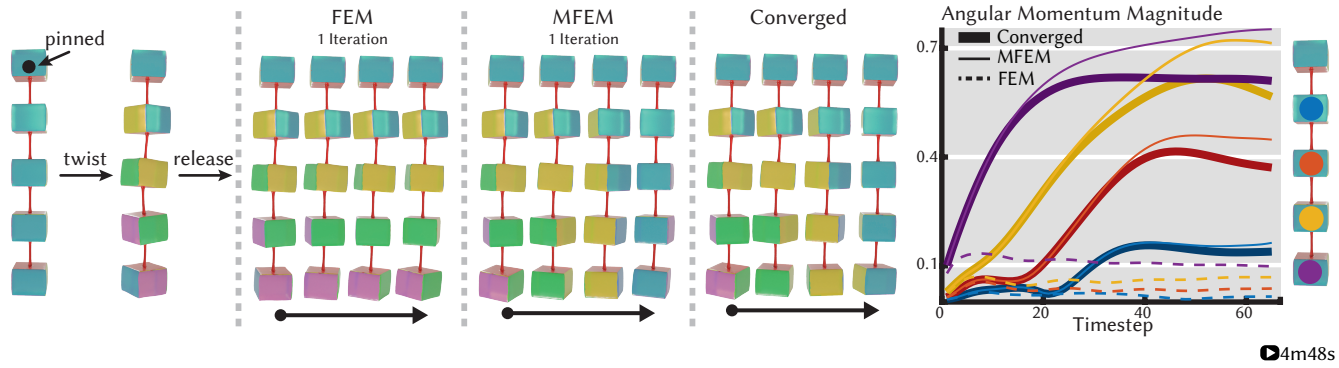

Figure 13: We pin the pendulum from the top, twist the bottom end, and simulate the unwinding. We compare results from FEM and MFEM with one solver iteration per timestep against a converged subspace FEM solution. Even at low iterations our MFEM solvers show much better agreement, which is reflected on the plot on the right where total angular momentum for each pendulum block is plotted over time.

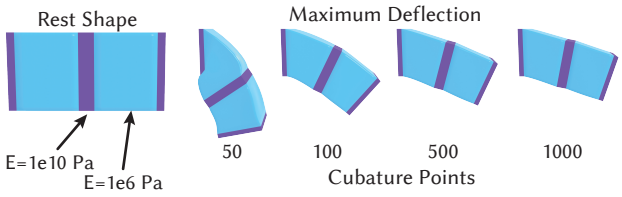

Figure 14: Under-integration with our clustering-based integration leads to artificial softening in softer regions. Here we simulate a 48,000 tetrahedra heterogeneous cantilevered beam and visualize the maximum deflection with different numbers of cubature points. 5 skinning modes are used for this simulation.

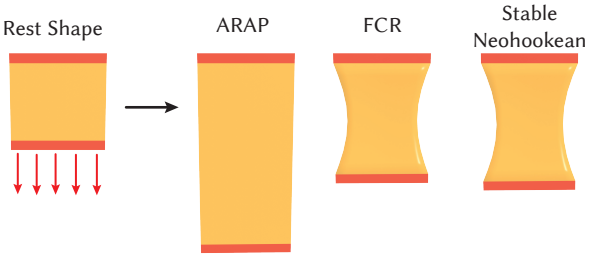

Figure 15: Our method is compatible with any hyperelastic material model. Here we apply a load and simulate beams to equilibrium with As-Rigid-As-Possible (ARAP), fixed corotational (FCR), and stable Neohookean material models

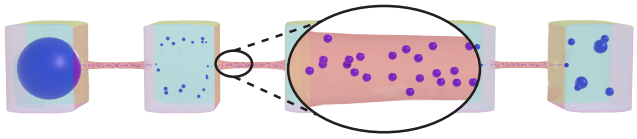

Figure 16: Our cubature sampling scheme is geometry aware and constraint aware. Note that our schemes samples more densely in the thin regions and only samples a single point on the far left where the pendulum is pinned.