



EigenFit for Consistent Elastodynamic Simulation Across Mesh Resolution

Yu Ju (Edwin) Chen
University of British Columbia
aq978063@cs.ubc.ca

David I.W. Levin
University of Toronto
diwlevin@cs.toronto.edu

Danny Kaufmann
Adobe
kaufman@adobe.com

Uri Ascher
University of British Columbia
ascher@cs.ubc.ca

Dinesh K. Pai
University of British Columbia
pai@cs.ubc.ca



Figure 1: Numerical stiffening can cause animations on a coarser FEM mesh (sequence (c)) to deviate from corresponding ones on a finer mesh (sequence (a)). To fix this error, EigenFit adjusts the magnitudes of the leading modes on the coarse mesh using the fine mesh at rest state, resulting in sequence (b). The meshes are coloured by normalized positional error with respect to the finer mesh. EigenFit often outperforms current leading alternatives.

ABSTRACT

Elastodynamic system simulation is a key procedure in computer graphics and robotics applications. To enable these simulations, the governing differential system is discretized in space (employing FEM) and then in time. For many simulation-based applications keeping the spatial resolution of the computational mesh effectively coarse is crucial for securing acceptable computational efficiency. However, this can introduce numerical stiffening effects that impede visual accuracy.

We propose and demonstrate, for both linear and nonlinear force models, a new method called EigenFit that improves the consistency and accuracy of the lower energy, primary deformation modes, as the spatial mesh resolution is coarsened. EigenFit applies a partial

spectral decomposition, solving a generalized eigenvalue problem in the leading mode subspace and then replacing the first several eigenvalues of the coarse mesh by those of the fine one at rest. EigenFit’s performance relies on a novel subspace model reduction technique which restricts the spectral decomposition to finding just a few of the leading eigenmodes. We demonstrate its efficacy on a number of objects with both homogenous and heterogenous material distributions.

CCS CONCEPTS

- Computing methodologies → Physical simulation.

KEYWORDS

Elastodynamic simulation, numerical stiffening

ACM Reference Format:

Yu Ju (Edwin) Chen, David I.W. Levin, Danny Kaufmann, Uri Ascher, and Dinesh K. Pai. 2019. EigenFit for Consistent Elastodynamic Simulation Across Mesh Resolution. In *SCA ’19: The ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA ’19)*, July 26–28, 2019, Los Angeles, CA, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3309486.3340248>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SCA ’19, July 26–28, 2019, Los Angeles, CA, USA

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6677-9/19/07...\$15.00

<https://doi.org/10.1145/3309486.3340248>

1 INTRODUCTION

The calibration, capture, animation and design of soft-body dynamics are critical for a wide range of domains spanning from robotics, automotive design and biomechanics to film, interactive animation and digital fabrication design. A fundamental computational bottleneck in all of these domains is the efficient forward simulation of elastodynamics. To simulate elastodynamics researchers often first discretize the governing differential system in space employing a finite element method (FEM) [Belytschko et al. 2013; Sifakis and Barbic 2012a], and then in time using a time-stepping method.

For simulation-based applications there are many reasons to alter the spatial and temporal resolution of the computational mesh. Critically, coarsening in both space and time is often required in order to make applications practical. Most notably, runtime costs for soft-body simulations scale superlinearly in the number of nodes in the FE mesh. However, both spatial and temporal coarsening introduce undesirable and often unexpected numerical artifacts that reduce the controllability, consistency, accuracy and effectiveness of resulting simulations. While understanding and reducing artifacts due to *temporal coarsening* is a long-standing and active area of investigation, the complementary problem of treating artifacts due to *spatial coarsening* has remained much less addressed. Here, we propose a new method, EigenFit, to mitigate spatial coarsening artifacts in elastodynamic simulation.

While aggressive spatial coarsening can significantly improve runtimes it has severe consequences for resulting soft-body simulations. First, as coarsening progresses the computational mesh loses geometric accuracy. Second, higher frequency modes that can only be expressed on finer meshes disappear as the mesh is progressively coarsened. Third, coarsening introduces *numerical stiffening*. The latter, in analogy to numerical dissipation, is the change in a simulated material's effective stiffness directly as consequence of spatial mesh coarsening.¹ This change in observed stiffness becomes more pronounced, despite material parameters remaining fixed, as we increase element sizes; see Figure 1.

Numerical stiffening thus remains a fundamental block to consistent and accurate simulations across changing mesh resolutions. Mesh resolution changes all the time, but, in principle, simulated physics should not change with it. Towards this goal we propose and demonstrate a new algorithm, EigenFit, that improves the consistency and accuracy of both linear and nonlinear model elastodynamic simulations. Specifically, EigenFit corrects the lower energy, primary deformation modes of spatially coarsened meshes to gain consistency in elastodynamic simulations across a broader range of mesh resolutions and motions. In particular we will show that consistent dynamic behaviour for the same shape and material is maintained as the spatial mesh resolution is varied in a range that preserves primary eigenmode shapes.

At its core EigenFit is conceptually direct. We apply partial spectral decomposition, solving a generalized eigenvalue problem to rescale leading eigenmodes of a coarsened model with initial-state ratios between fine- and coarse-resolution mesh eigenvalues. For

¹ Numerical stiffening is not to be confused with other notions of material stiffness that are independent of mesh discretization and often result from having very different scales in the underlying simulated differential system. Here it is the low eigenvalues that are at play, and numerical stiffness difficulties ebb away as the mesh resolution is increased.

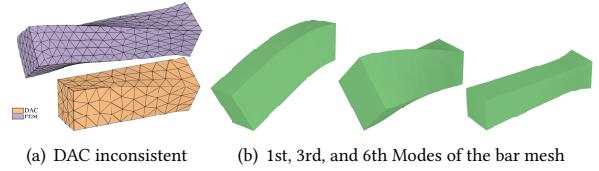


Figure 2: DAC cannot fix some important deformation modes. In this figure both bars are simulated with Young's modulus $Y = 4kPa$ and Poisson's ratio $\nu = 0.3$. However, the stretching and twisting motions are inconsistent. DAC only uses the first mode to adjust the dynamic behavior of the object. But in this example, the first mode of the elastic bar is a bending mode (see (b)). Consequently, the twisting (3rd mode) and stretching (6th mode) motions are inconsistent. Subfigure (a) also demonstrates that numerical stiffening can still occur away from locking at low Poisson ratio.

nonlinear models EigenFit then updates this fit as time progresses with a local re-linearization and decomposition that calculates just leading modes at each time step. By restricting this per-time step correction to subspaces formed by just the leading eigenmodes, EigenFit avoids prohibitively expensive, repeated full decompositions and so efficiently performs subspace correction followed by a low-rank update to map the model back to the full deformation space for time stepping.

EigenFit performs exceptionally well for linear constitutive materials; see Figure 1 and Section 4.1. Furthermore, unlike some other methods EigenFit is effectively oblivious to material heterogeneity and performs without change for heterogeneous materials where the Young modulus is a distributed parameter function. It naturally handles relatively stiff objects where motion is generated through local softer joints, as demonstrated in Section 4 and the supplementary video.

2 RELATED WORK AND NUMERICAL STIFFENING

In this work we focus on constructing coarse spatial-mesh elastodynamic simulations that improve accuracy and consistency of the primary, lowest energy eigenmodes with respect to a fine-resolution reference mesh. These leading modes generally exhibit the largest and most visible deformations in elastica and are thus critical for visual and functional applications in many domains including visual effects, e.g., film and animation, as well as functional design, e.g., robotics and biomechanics [Chen et al. 2017].

A key obstacle to this goal are spatial coarsening artifacts that arise in the form of numerical stiffening. In Section 2.1 we look further into the source of numerical stiffening. First, however, we briefly review existing remedies for it.

Numerical stiffening in conformal FEM is treated directly by adapting the spatial mesh for accuracy until error is reduced and thus the artifact is mitigated [Babuska et al. 2012]. These refinements traditionally increase mesh-resolution (h -refinement) and/or element order (p -refinement) [Belytschko et al. 2013]. Many refinement approaches monitor error on the deformed mesh and so adapt during simulation [Bargteil and Cohen 2014; Grinspun et al. 2002; Mosler and Ortiz 2007]. Most recently Schneider et al. [2018]

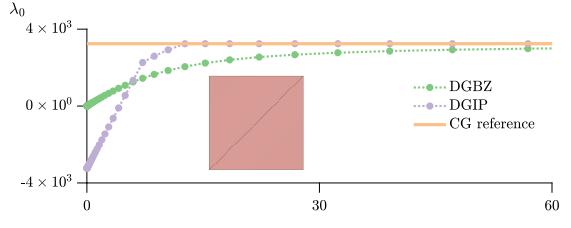


Figure 3: The lowest non-zero eigenvalue of a square mesh with two triangles under the Discontinuous Galerkin BZ formulation and IP formulation [Chen et al. 2018b; Kaufmann 2012; Kaufmann et al. 2009] with a reference continuous Galerkin (CG) solution. A penalty parameter η can be used to adjust the effective stiffness and alleviate numerical stiffening somewhat. However, the value of η is purely empirical.

proposed p -refinement on unstructured meshes based on analysis of the undeformed mesh at rest. This latter approach can be highly effective when a small number of ill-shaped elements are ruining accuracy. However, in the spatial coarsening setting we have a global problem where all elements may be problematically large. Thus, if we were to adopt a refinement solution to alleviate numerical stiffening, the entire domain would require refinement, leading to impractical meshes with high degree of freedom that one wishes to avoid computing with in the first place [Chen et al. 2017].

Alternatively, numerical homogenization methods [Chen et al. 2015; Kharevych et al. 2009; Nesme et al. 2009; Panetta et al. 2015; Torres et al. 2016] have also been proposed to better model energy density at coarse resolution. These methods work well in the static setting but do not account for inertia and fail to extend in the dynamic setting [Chen et al. 2017]. Moreover, the primary focus in these methods is specifically homogenizing static solutions for heterogenous materials. Here we focus on gaining accuracy for dynamics solutions without modifying the given constitutive material model.

Nonconforming FE methods, especially discontinuous Galerkin (DG), are another potential avenue for reducing numerical stiffening. In particular, DG methods can be applied to help reduce the related problem of numerical locking [Kaufmann 2012; Kaufmann et al. 2009]. By adding additional degrees of freedom to the system and weakly enforcing inter-element continuity with penalty terms, DG methods offer freedom to avoid degree-of-freedom locking. However, this additional flexibility unfortunately does not provide dynamical correction for numerical stiffening which occurs in addition to locking; see Figures 2(a) and 3.

Most recently, Chen et al. [2018b] combined homogenization and nonconforming FE for coarse static solutions to heterogenous materials. They also interestingly demonstrated modest improvement on a simple dynamics example, although they note that no explicit correction in this method is made towards fixing phase error.

The closest work to ours is that of Chen et al. [2017]. They observe that rather coarse meshes can accurately capture low-frequency mode geometry, and hence can capture the primary mode shapes of a simulated object. To a priori correct this model towards

handling numerical stiffening artifacts, they proposed a dynamics aware coarsening (DAC) method that precomputes and applies, per mesh, a one-time numerical rescaling of Young's modulus to match the lowest eigenmode frequency of a coarse mesh to an accurate sample. While this rescaling can be highly effective in many cases where a single primary mode dominates deformation dynamics, it is unable in many cases to fix a wide number of nonlocal and nonlinear errors and inconsistencies caused by numerical stiffening; see Figure 2. In particular, if deformations involve stretching, twisting and/or bending motions, then more than a single parameter adjustment will generally be needed. EigenFit iteratively re-fits a set of leading eigenmodes to achieve these corrections; see Figure 6 and Sections 3 and 4 for detailed discussion and comparisons. We refer to [Sifakis and Barbic 2012a; Xu and Barbic 2016] for a discussion of eigenmodes and model reduction. Furthermore, the inverse problem solved per mesh as part of DAC could become difficult in some applications, a difficulty that EigenFit avoids by not solving any inverse problems.

2.1 Motivating Example

To better understand numerical stiffening, let us consider the discretization of the simplest classical wave equation given by the partial differential equation (PDE)

$$u_{tt} = c^2 \nabla^2 u, \quad (1)$$

where c is the speed of wave. Further, assume that the PDE is in one space variable and subject to the boundary conditions $u(t, x = 0) = u(t, x = 1) = 0$.

Looking for solutions of the form $u(t, x) = e^{i\sqrt{\lambda}t}U(x)$ leads to the eigenvalue problem

$$-c^2 \nabla^2 U = \lambda U, \quad (2)$$

with eigenvalues

$$\lambda_j = (jc\pi)^2, \quad j \in \mathbb{N}. \quad (3)$$

Here the lowest eigenvalue is $\lambda_1 = \pi^2 c^2 \approx 9.86c^2$, with $\sqrt{\lambda_1}$ the frequency (in time) of the dominating dynamic mode.

Semi-Discretization. It is common to semi-discretize the PDE in Eq. (1) in space, and then step the resulting ODE in time to simulate dynamics. This semi-discretization in space is, of course, closely related to the eigenvalue problem Eq. (2). However, the spatial discretization error will pollute the overall time trajectory (dynamic behaviour) as it alters the analytic eigenvalues given in Eq. (3).

Finite Element Method. Applying conformal FEM, we first convert Eq. (2) to weak form. The eigenvalue problem (2) is then equivalent to requiring that for all appropriate test functions V ,

$$c^2 \int \nabla U \nabla V dx = \lambda \int V U dx. \quad (4)$$

We apply simplest Galerkin FEM by choosing piecewise linear “hat functions” satisfying the boundary conditions as the function space for U and V . The resulting element stiffness matrix from the left hand side and mass matrix from the right hand side of Eq. (4) are, respectively,

$$K_{e,h} = \frac{c^2}{h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad M_{e,h} = \frac{\lambda h}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

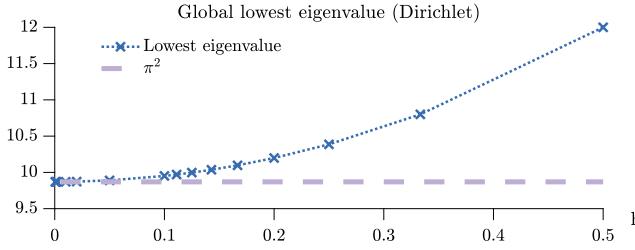


Figure 4: The lowest eigenvalue of Eq. (5) with $c = 1$ changes depending on the mesh resolution. This eigenvalue approaches the true eigenvalue π^2 of Eq. (3) monotonically from above as the mesh resolution improves.

After assembly and elimination of the boundary variables, we have converted Eq. (4) to the system

$$\begin{aligned} K_h \mathbf{u} &= \lambda M_h \mathbf{u}, \quad \text{where} \tag{5} \\ K_h &= -\frac{c^2}{h} \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & \end{bmatrix}, \\ M_h &= \frac{h}{6} \begin{bmatrix} 4 & 2 & & & & \\ 2 & 4 & 2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 2 & 4 & 2 & \\ & & & 2 & 4 & \end{bmatrix}. \end{aligned}$$

We have discretized Eq. (2) and converted it into a generalized eigenvalue problem. The lowest eigenvalue at a range of mesh resolutions is plotted in Figure 4. Notice that approximated eigenvalues using the FEM approach the exact one in Eq. (3) *from above* as $h \rightarrow 0$. This artifact is numerical stiffening, and here the simulated object is stiffer than the physical model. Moreover, at each progressively coarser mesh a corresponding simulation will be stiffer when compared to a finer mesh simulation.

This numerical stiffening is a direct consequence of an old result [Ciarlet et al. 1968] showing that the Galerkin approximation of the general eigenvalue problem will always lead to larger eigenvalues. In brief we are searching for the minimum of an approximated Ritz functional in an increasingly less inclusive function space as we coarsen the mesh in the weak form of a boundary value PDE problem. Note that this result will not hold in general, if we lump the mass matrix or use finite differences; although such method alterations will not improve the numerical stiffening: it is only the monotonic approach to the limit that would be lost.

3 METHOD

We now describe in detail a new method that mitigates numerical stiffening by matching primary vibration modes. As in [Chen et al. 2017] our setting requires not to coarsen the mesh beyond a point where the low-energy mode shapes differ between fine and coarse by more than some small tolerance. The method described below applies for linear and nonlinear force models.

Discretizing the equations of motion for an elastodynamic system using the finite element method (FEM) results in a large system of ordinary differential equations in time t given by

$$M\ddot{\mathbf{q}}(t) = f_{\text{tot}}(\mathbf{q}, \dot{\mathbf{v}}),$$

where the unknowns $\mathbf{q} = \mathbf{q}(t)$ are nodal displacements of the FEM mesh with corresponding velocities $\mathbf{v}(t) = \dot{\mathbf{q}}(t)$. The mass matrix M is symmetric positive definite. The total force is further written as

$$f_{\text{tot}} = f_{\text{els}}(\mathbf{q}) + f_{\text{dmp}}(\mathbf{q}, \dot{\mathbf{v}}) + f_{\text{ext}}(\mathbf{q}),$$

with the elastic forces given by

$$f \equiv f_{\text{els}}(\mathbf{q}) = -\frac{\partial}{\partial \mathbf{q}} W(\mathbf{q}(t)),$$

where $W(\mathbf{q}(t))$ is the elastic potential of the corresponding hyperelasticity model. We further define the tangent stiffness matrix $K = -\frac{\partial}{\partial \dot{\mathbf{q}}} f_{\text{els}}(\mathbf{q}(t))$. This matrix is constant and symmetric positive definite if f is linear, but for nonlinear elastic forces it depends on the unknown $\mathbf{q}(t)$ and may occasionally become indefinite. See, for instance, [Chen et al. 2017, 2018a; Ciarlet 1988; Sifakis and Barbic 2012b; Teran et al. 2005]. To model internal friction of elastic objects and to improve stability of the system, we have employed the Rayleigh damping force

$$f_{\text{dmp}}(\mathbf{q}, \dot{\mathbf{v}}) = (\alpha M + \beta K)\dot{\mathbf{v}}, \tag{6}$$

with $\alpha \geq 0$, $\beta > 0$.

Below we first motivate our approach in the linear setting (Section 3.2). We then extend our approach to the nonlinear setting using an iteratively updated subspace correction (Section 3.3). Our full dimensional method with subspace correction is described in Section 3.3.3.

With our spatial treatment method now in place, it can be applied in conjunction with any suitable time-stepping integrator. For simplicity we use semi-implicit backward Euler (SI) [Baraff and Witkin 1998].

3.1 Mesh eigenmodes

To introduce our ideas gradually, suppose at first that we perform an FEM simulation of an elastic object motion under a linear elastic force at two distinct mesh resolutions. We then have respective time-independent mass matrices M_c, M_f and stiffness matrices K_c, K_f corresponding to the coarse and fine meshes. We can find the eigenmodes $u_{c,i}$ and $u_{f,i}$ (modes of deformation) of the coarse and fine meshes by carrying out the principal component analysis

$$\begin{aligned} K_c u_{c,i} &= \lambda_{c,i} M_c u_{c,i}, \quad i = 1, \dots, N_c, \\ K_f u_{f,i} &= \lambda_{f,i} M_f u_{f,i}, \quad i = 1, \dots, N_f, \end{aligned}$$

where $N_c < N_f$. While the eigenmodes $u_{c,i}$ and $u_{f,i}$ have different sizes, they correspond to the same i th deformation mode of the object for low indices i , $i = 1, 2, \dots, m$, where $m \leq N_c$. However, as discussed in Section 2, their corresponding eigenvalues $\lambda_{c,i}$ and $\lambda_{f,i}$ will be different due to numerical stiffening. Although their relative difference is only by $O(1)$, this can change the object deformation properties significantly; see Figures 1 and 6. In particular, avoiding mass lumping we have $\lambda_{c,i} > \lambda_{f,i}$, and since these eigenvalues directly relate to the oscillation frequency of the corresponding

mode, simulating on a coarse mesh has been observed to make the object look stiffer and oscillate faster.

To enable such an eigenvalue adjustment, the first idea that may spring to mind is to use the fact that, if the scalar λ and the vector u are an eigenpair of a matrix A satisfying $Au = \lambda u$, then for any integer j , also $A^j u = \lambda^j u$. Thus, one could construct an interpolating polynomial $p(A)$ for $A = M_c^{-1} K_c$ such that $P(A)u_{c,i} = \lambda_{f,i} u_{c,i}$ for the first few modes, $i = 1, 2, \dots, m$, and replace $\lambda_{c,i}$ by $p(\lambda_{c,i})$. However, this idea does not work out, even for small fittings of $m = 3$. This is because the eigenvalue spread is typically wide and unequal. The resulting, *extrapolated* values of $p(\lambda_{c,i})$ for $i > m$ are often rather far from the corresponding $\lambda_{f,i}$, and the process is highly ill-conditioned even when using a Lagrange basis for the polynomial interpolation process. We thus proceed with a direct strategy for modifying the primary coarse-mesh eigenvalues.

3.2 Direct eigenvalue modification

Continuing our gradual development, in this section we assume that the stiffness matrix K is constant and symmetric positive definite. To reduce the numerical stiffening effect we modify the first m eigenvalues directly within the eigendecomposition, where the parameter m is not large (typically in our calculations, $m < 20$). First, using the shorthand $M = M_c$, $K = K_c$, we write the generalized eigenvalue problem from Section 3.1 above as

$$KU = MUD, \quad (7)$$

where D is a diagonal matrix having the eigenvalues on its main diagonal, and U is the matrix having the eigenvectors as columns:

$$\begin{aligned} D &= \text{diag} [\lambda_{c,1}, \lambda_{c,2}, \dots, \lambda_{c,N_c}], \\ U &= \left[\begin{array}{cccc} | & | & \cdots & | \\ u_{c,1} & u_{c,2} & \cdots & u_{c,N_c} \\ | & | & \cdots & | \end{array} \right]. \end{aligned} \quad (8)$$

Solving such problem also ensures mass-orthogonality, $U^T MU = I$ and $MUU^T = I$.²

Next, correction of frequency and thus eigenvalues is direct in the linear setting. We simply replace D in Eq. (8) with

$$D^\dagger = \text{diag} [\lambda_{f,1}, \lambda_{f,2}, \dots, \lambda_{f,m}, \lambda_{c,m+1}, \dots, \lambda_{c,N_c}]. \quad (9)$$

We select the parameter m using the Hausdorff distance between the eigenshapes of the coarse and fine meshes, following Chen et al. [2017]. Now, using mass orthogonality [Barbić and James 2005], we can approximate the generalized eigenvalue equations (7) by $K^\dagger U = MUD^\dagger U^T$, and obtain our modified stiffness matrix K^\dagger directly from this expression:

$$K^\dagger = MUD^\dagger U^T M. \quad (10a)$$

For linear models, we then construct the corresponding modified force by

$$f^\dagger = -K^\dagger x, \quad (10b)$$

where x is the deformation.

²Note, however, that U is not an orthogonal matrix in general.

3.3 Nonlinear Materials

The approach described in Section 3.2 is direct and simple. However, it is only applicable to linear force models, where the stiffness matrix and the modal analysis remain constant throughout the simulation.

In this section we extend our observations from the linear setting to nonlinear force models, e.g., neo-Hookean, StVK, and the spline model of Xu et al. [2015], as well as to co-rotational FEM.

The obvious step to begin with is to consider a local linearization, i.e., re-initialization, at each time step. However, there are several obstructions to this approach. One is that the Jacobian matrix that arises at each given time step depends on the solution there which, in turn, depends in general on all modes. So there is some mixing of modes in the nonlinear case that does not happen in the linear one. Moreover, approximations for the stiffness matrix, for instance when using neo-Hookean force, might give rise to negative eigenvalues (corresponding possibly to material compression). Finally, the sheer cost of carrying out an eigendecomposition at each time step can quickly become prohibitive.

3.3.1 Rescaling a subset of eigenvalues. Rather than a one-time precomputed rescaling as performed in Chen et al. [2017], we update our fitting iteratively with a local re-linearization at each time step. We first designate our set of m target eigenvalue ratios $r_i = \frac{\lambda_{f,i}}{\lambda_{c,i}}$ for $i = 1, 2, \dots, m$, either precomputed from the rest shape of the meshes or else set by hand, e.g., for animation design.

At each time step we then

- (1) Build a local tangent stiffness matrix K and perform generalized eigenvalue decomposition on the coarse mesh as in Eqs. (7)–(8). Note that, unlike for linear forces, here both the eigenvectors and eigenvalues depend on the state $u_{c,i} = u_{c,i}(x_t)$ and $\lambda_{c,i} = \lambda_{c,i}(x_t)$.
- (2) Set the ratio matrix

$$R = \text{diag} [r_1, \dots, r_m, 1, \dots, 1], \quad (11)$$

and then $D^\dagger = RD$. This adjusts the leading eigenvalues to their target values.

- (3) In the locally linear setting the modified tangent stiffness matrix is now given by the expression in Eq. (10a). The force is correspondingly adjusted to match:

$$f^\dagger = M \left(URU^{-1} \right) f = MURU^T f. \quad (12)$$

In the general nonlinear material model setting the eigenvalues are not always positive [Ciarlet 1988]. This can mix the rescaling effect on different deformation modes. To mitigate this issue, during assembly of the tangent stiffness matrix, we project any indefinite element stiffness matrices to the PD cone, by replacing offending negative eigenvalues with a small positive value ϵ (we chose $\epsilon = 1e-3$), in turn guaranteeing positive definiteness of the global stiffness matrix; see Teran et al. [2005].

Although the generalized eigenvalue decomposition is performed on the coarse mesh at the beginning of each time step, the corresponding decomposition on the fine mesh is performed only once, at the beginning. Thus, in (11) we keep the rescaling ratios fixed, to avoid calculating ratios of the eigenvalues of coarse and fine at different times. The updated coarse modes, however, are of course used in Eqs. (12) and (10a).

3.3.2 Subspace correction. As noted earlier, a fundamental drawback of our strategy so far is the computation of the full eigen-decomposition of the system matrix at every time step. Observe, however, that our goal is just the re-fitting of the lowest m modes. Thus, we may focus much of our attention on a subspace of the eigendecomposition. This is especially important as computation of a low number of eigenpairs is significantly less expensive (see, e.g., [Stewart 2002]). Starting with this observation we can instead perform our rescaling in the reduced space spanned by the lowest s modes, where s is a small integer in the range $m \leq s \leq N_c$. Its value depends on the task complexity and the computational cost.

Concretely, at each time step we compute only the lowest s eigenpairs. Denote the corresponding reduced-space eigenvalue and eigenmode matrices D_s and U_s . In contrast to standard linear and nonlinear modal analysis and dimension reduction methods, here we build a small, *local* subspace about each time step. In our setting the relevant eigendecomposition can be efficiently computed by performing a sequence of inverse power iterations to compute at each time step the lowest few eigenpairs corresponding to the dominant motions. Note that in the subspace of dimension $s < N_c$, $U_s^T M U_s = I_s$, but $M U_s U_s^T \neq I$.

Within each time step's local subspace, the corresponding mass matrix is the $s \times s$ identity matrix, while the subspace tangent stiffness matrix K_s is diagonal. The corresponding reduced stiffness matrix and force are then

$$K_s = U_s^T K U_s = \text{diag}[\lambda_{c,1}, \dots, \lambda_{c,s}], \quad f_s = U_s^T f. \quad (13)$$

Next, following Section 3.3.1, we build the diagonal subspace rescaling matrix of size s in the reduced space,

$$R_s = \text{diag}[r_1, \dots, r_s].$$

The corresponding modified subspace tangent stiffness matrix and forces, rescaled by r_i , are then

$$K_s^\dagger = R_s U_s^T K U_s = R_s \text{diag}[\lambda_{c,1}, \dots, \lambda_{c,s}], \quad (14a)$$

$$f_s^\dagger = R_s U_s^T f. \quad (14b)$$

3.3.3 The EigenFit method. Finally, our local subspace corrections to the primary dynamic modes must be added back to the full system for simulation of complete dynamics.

To move our subspace correction into the full system we start by observing that, for an arbitrary diagonal scaling matrix A_s , the matrix $A_s U_s^T$ rescales and projects down to the s -mode basis, as in Eqs. (14) above. In turn, $M U_s^T A_s$ lifts the projected quantity back to the full space of the FEM system. To jointly remove the current, uncorrected force contributions spanning the subspace, and to add their corrected counterparts back in, we then use $(R_s - I_s)$. Thus, the full-space tangent stiffness matrix, with our subspace correction, is

$$K^\ddagger = K + M U_s (R_s - I_s) U_s^T K U_s U_s^T M, \quad (15a)$$

$$= K + M U_s (R_s - I_s) \text{diag}[\lambda_{c,1}, \dots, \lambda_{c,s}] U_s^T M, \quad (15b)$$

and the corrected force is

$$f^\ddagger = f + M U_s (R_s - I_s) U_s^T f. \quad (15c)$$

Note that with the above model we have regained the target correction sought in Section 3.3.1, while only incurring the cost of the small s -mode eigendecomposition.

3.4 Implementing an EigenFit integration step

The matrix K^\ddagger in Eq. (15b) consists of the (relatively) large but sparse positive definite $N_c \times N_c$ matrix K plus an added full (dense) matrix of size $N_c \times N_c$ and rank s . This may make computations using any of our time integration methods expensive if one is not careful.

Let us define

$$Y = M U_s (R_s - I_s), \quad Z^T = \text{diag}[\lambda_{c,1}, \dots, \lambda_{c,s}] U_s^T M. \quad (16a)$$

Then Y and Z are both $N_c \times s$ (i.e., “long and skinny”: $s \ll N_c$), and

$$K^\ddagger = K + Y Z^T. \quad (16b)$$

For the SI time discretization method we have to solve a system of the form

$$(M + h^2 K^\ddagger) v_+ = z$$

for the velocities v_+ at the next time level (where h is the step size). Here z is a known right hand side. If we use a preconditioned conjugate gradient method for this, then an oracle for an efficient matrix-vector product can be readily constructed, taking into account the thinness of Y and Z .

For some other cases, however, when N_c is not extremely large and the physical system is stiff, i.e., in the presence of large Young modulus values, we may well wish to solve the SI linear system (or any other such algebraic system arising from an implicit time difference method) by a direct method based somehow on Gaussian elimination. For this purpose we invoke the famous Sherman-Morrison-Woodbury formula [Nocedal and Wright 2006],

$$(A + Y Z^T)^{-1} = A^{-1} - A^{-1} Y (I + Z^T A^{-1} Y)^{-1} Z^T A^{-1}. \quad (17)$$

In our case we have Y and Z defined as above in Eq. (16a) and $A = h^{-2} M + K$. The formula is then applied to the right hand side $h^{-2} z$, requiring one Cholesky decomposition of the sparse A followed by $s + 1$ forward-backward solves. (Note that the matrix $I + Z^T A^{-1} Y$ is only $s \times s$, and its inversion is thus assumed cheap.) In our context, the resulting direct inversion method is typically significantly more efficient, even when using SI, than iterative methods such as conjugate gradient.

For *linear force models*, we note that while Section 3.2 is there for didactical purposes, there is no reason to perform a full eigen-decomposition even in the linear case. In practice we thus apply EigenFit as is also for linear forces, noting that the low rank correction matrices Y and Z^T are calculated just once.

3.5 Practical considerations

In many of the experiments, video clips and figures of this paper we naturally compare our method's performance on a coarse mesh to similar results on a fine mesh. But in practice there will be no detailed fine mesh calculations, or else the purpose of using the coarse mesh in the first place would be nullified. We therefore must be able to predict, to a reasonable degree of assurance, if applying EigenFit on a particular coarse mesh would lead to reasonable approximations for a similar simulation on the finer mesh.

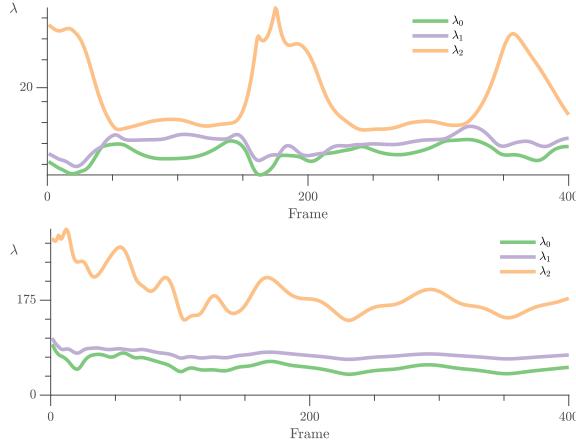


Figure 5: Eigenvalues corresponding to the first three leading modes of a bar with a neo-Hookean force. In the top subfigure (corresponding to a softer body $Y=1.e4\text{Pa}$) there is a mode crossing. No such unfortunate behaviour is observed for the stiffer object with $Y=1.e5\text{Pa}$. in the bottom figure.

The essential reason why there is hope is the general observation, shared by both the mechanical engineering and the computer graphics communities, that the first s modes, $s \leq 20$, already essentially determine the visual result. This allows for the development of a model reduction technique such as we have just presented. It allows us to always obtain rather acceptable results for linear forces, as demonstrated throughout this paper and especially in the next section.

At the other end there is always an inherent restriction when applying at each time step a technique that is essentially a linearization of a nonlinear problem, especially when the time step is large and there is a lot of deformation action across it. This is what our work as well as all leading others are often doing. A key to our success is to keep the leading modes from tangling up: if a mode on the coarse mesh no longer corresponds to the one on the fine mesh, then of course the ratio of the corresponding eigenvalues becomes meaningless. See Figure 5.

Following extensive experimentation, we have arrived at the following criterion. Our algorithm verifies that at least half of the current eigenvectors match to the rest-state eigenvectors. The matching process simply uses the mass orthogonality from Eqs. (7) and (8). This is a functional distance measurement between two meshes. Since true orthogonality is not achievable from this matching process, we use a loss $tol = 0.4$ throughout the simulation. Practically, we compute $U_s^T M U_{s,r}$, where $U_s, U_{s,r}$ are the s current and rest state eigenvectors, and verify that at least half of the columns contain an entry larger than 0.6. At the first frame, a similar calculation is performed to verify that the resolution of the coarse mesh is capable of simulating the dominating motion of the fine mesh. That is, we calculate $U_{s,r,c}^T B M_f U_{s,r,f}$, where $U_{s,r,c}, U_{s,r,f}$ are the coarse and fine rest state eigenvectors, M_f is the mass matrix from the fine mesh, and B is a mapping matrix from the coarse mesh to the fine mesh through barycentric coordinates. Again, we verify that at least half of the columns contain an entry larger than 0.6. The

Table 1: List of Meshes

Mesh ID	#DOF ($3 \times$ #Vertices)	#Tetrahedrons
Arm fine	15,372	19,378
Arm coarse	7,800	8,948
Armadillo fine	40,539	54,233
Armadillo coarse	4,425	4,902
Bar fine	42,069	72,989
Bar coarse	3,846	5,020
Fert fine	36,324	47,813
Fert coarse	5,385	6,020
Rampant fine	59,382	78,686
Rampant coarse	2,820	3,070
Skater fine	32,169	42,161
Skater coarse	8,394	9,584

present simulation is deemed acceptable only if it passes both tests. In short, we use the criterion above to make sure the nonlinearity is not out of hand. Note that there are many types of nonlinearity that can appear in different applications, and different security measures can be used accordingly. For example the alignment process in [Xu and Barbić 2016] finds a set of transformation angles by solving an orthogonal Procrustes problem, and this set of angles can also be used as a measurement for nonlinearity.

4 RESULTS

In this section we demonstrate the efficacy of EigenFit on homogeneous and heterogeneous material deformable objects of various shapes simulated under both linear and nonlinear forces. For simplicity, for all examples in this section, we use time step size $h = 1e - 2$. In the comparisons below, we always use a fine spatial mesh simulated using plain FEM as the ground truth. We compare this fine mesh trajectory against various coarse mesh trajectories simulated by plain FEM, EigenFit or DAC. In this section we use a number of meshes for the reported simulations. Table 1 summarizes the mesh information used, while Table 2 summarizes our simulation times for each of the coarse mesh, fine mesh, and the EigenFit simulations. We found that for coarse mesh nonlinear material, 20% of the time increase comes from eigendecomposition, while the rest comes from the low rank update. Due to the nature of dynamic simulation, some results are easier to visualize and understand in animated form. For this we refer readers to our supplementary video. In all of the figures, the color code is calibrated using the maximum error from each simulation. We rendered an embedded fine mesh together with a wireframe from the simulated coarse mesh. Note further that many of the examples in this section and the supplementary video involve positional constraints.

In this project, we use nested cages [Sacht et al. 2015] and TetWild [Hu et al. 2018] to generate meshes for our simulation inputs. We implemented our algorithm in C++ with GAUSS library, and use Spectra for eigenvalue/eigenvector computation. We will make our code available with the published paper.

4.1 Linear hyperelastic constitutive models

While nonlinear models are commonplace in animation tasks nowadays, engineering and fabrication often rely on linear dynamical analysis. Furthermore, many animation techniques rely on underlying linear models which are modified via warping to yield visually acceptable results. Linear force models remain popular due to their simplicity and rapid execution, and have been used in some interesting and varied applications such as acoustic transfer [Li et al. 2015]. In our specific context this section is a good place to demonstrate how EigenFit leverages the underlying principle behind numerical coarsening. We can also clearly see how EigenFit differs from DAC and improves upon it.

Under the setting of linear elasticity, the eigensystem is constant. We can perform the full eigendecomposition as described in Section 3.1. However, since the high energy modes rarely have observable amplitude, we only carry out a partial eigendecomposition to adjust the first few eigenvalues and eigenvectors, as described in Section 3.2. Notice that we only need to perform this decomposition and store Y, Z in Eq. (16a) once for the entire simulation.

4.1.1 Bar. In this example, we simulate a twisted bar using linear material with constant Young's modulus $Y = 1e5\text{Pa}$, Poisson's ratio $\nu = 0.45$, and IM with $\alpha = 0, \beta = 0.01$ in Eq. (6). In this case, the fine mesh has $N_f = 14,023$ and the coarse mesh has $N_c = 1,282$ free vertices. In the EigenFit algorithm, we pick $m = s = 10$ modes to match from coarse mesh to the fine mesh. We track a corner point of the bar and plot the trajectories and errors in Figure 6. We also show the simulation results at $t = 2.9\text{sec}$ with different view angles to provide comparison with other methods.

Note that the Euclidean error of EigenFit at the corner point is consistently lower than those of the plain FEM and DAC. Interestingly, if we only look at the y trajectory of the corner point in Figure 6(c), DAC does a good job at matching the ground truth. By looking at the first few eigen-deformations in Figure 7, we can see why this is the case. The first eigen-deformation is a bending mode that affects the y coordinate only; see Figure 7(a). Since DAC uses the first eigenvalue ratio to adjust the Young modulus, it fixes the stiffening effect in this motion. However, since we are simulating a twisted bar, there are other deformation modes that could not be captured by such eigen-deformation alone. In our EigenFit algorithm, the 3rd eigenvalue ratio is used to compensate the stiffening effect in the rotational motion; see Figure 7(c). This suggests that EigenFit can be viewed as a high-order improvement to DAC.

4.1.2 Armadillo. Our algorithm can of course handle more complex geometries than a rectangular bar. In this example we simulate an armadillo using Armadillo meshes described in Table 1. Figure 1 demonstrates that EigenFit can match the corresponding modes to the correct frequency in the fine mesh. This results in a consistent simulation trajectory across different resolutions.

Summary for linear forces: We have carried out dozens of additional simulations for many different deformable objects under linear force. For all examples tried EigenFit performs very well indeed. For DAC, the observations above also hold consistently: it performs well (i.e., comparable to EigenFit) if there is one dominating mode, but when several different dominating modes are present DAC falls behind.

4.2 Mildly nonlinear elastodynamics models

To further demonstrate the capacity of EigenFit, we have applied it for nonlinear material force models, including as rigid as possible (ARAP) [Chao et al. 2010] and Neo-Hookean.

We have used ARAP energy, a popular nonlinear energy model in computer graphics, to simulate the Armadillo and Skater meshes with stiffness parameter $1e6\text{Pa}$, simulated with the SI integrator. We fixed and shook the feet of both objects for 30 frames and observed the resulting oscillation afterwards. For EigenFit, we used 10 modes and applied the 10 calculated ratios at every frame, as described in Section 3.3. Although the result is not as impressively perfect as in the linear case, EigenFit still tracks the fine mesh motion much better than the regular coarse FEM simulation; see Figure 8 and Figure 10, respectively. In particular, notice that the position error and velocity are oscillatory and out of phase in the coarse FEM simulation (yellow line in the error plots), which means that its dominating motion has the wrong frequency. EigenFit (green line in the error plots) matches the frequency of this motion, and thus has a much lower error.

In the armadillo mesh, we also compared to DAC. Notice that EigenFit also performs better than DAC in this case, and it is not hard to see why by looking at the first 3 dominating modes. From our EigenFit calculation we found the dominating three modes have eigenvalue ratios $r_1 = 0.536, r_2 = 0.711$, and $r_3 = 0.675$, and the corresponding motions are forward bending (Fig. 8(c)), side shifting (Fig. 8(d)), and slight twisting about the z-axis (Fig. 8(e)). Effectively, DAC applied one single ratio, 0.536, to all of these motions, whereas EigenFit correctly matched their corresponding ratios.

4.2.1 Dynamic constraint. EigenFit also works when a constraint changes during the simulation. In Figure 9, we simulated a shaken armadillo, and released one leg during the simulation. The EigenFit ratios are recalculated when the boundary conditions are changed. The plots show that using EigenFit reduced the error significantly.

Summary for mildly nonlinear scenarios: The results in Figures 8–10, as well as those in Figures 11–13, all indicate that when the nonlinear effect is sufficiently moderate so that a quasi-linearization at the beginning of each time step captures the essence of what happens throughout it, the conclusions drawn before for linear forces can be extended, albeit in an imperfect sense. We do observe mild mismatches, but both EigenFit and DAC are expected to perform better than regular FEM, and situations where DAC is worse than EigenFit arise here as well.

4.3 Large deformation for nonlinear numerical material

For more general nonlinear scenarios it is important to understand that there is not a panacea. The methods proposed in the literature all rely on some localization that may not always hold, and none of them performs satisfactorily in the large context. Figure 14 is a case in point, where both EigenFit and DAC give poorer results than the regular coarse mesh FEM.

In practice, of course the utility of any method of the sort considered here depends on not having to simulate the “ground truth” fine mesh trajectory. If the ultimate fine mesh is much finer than

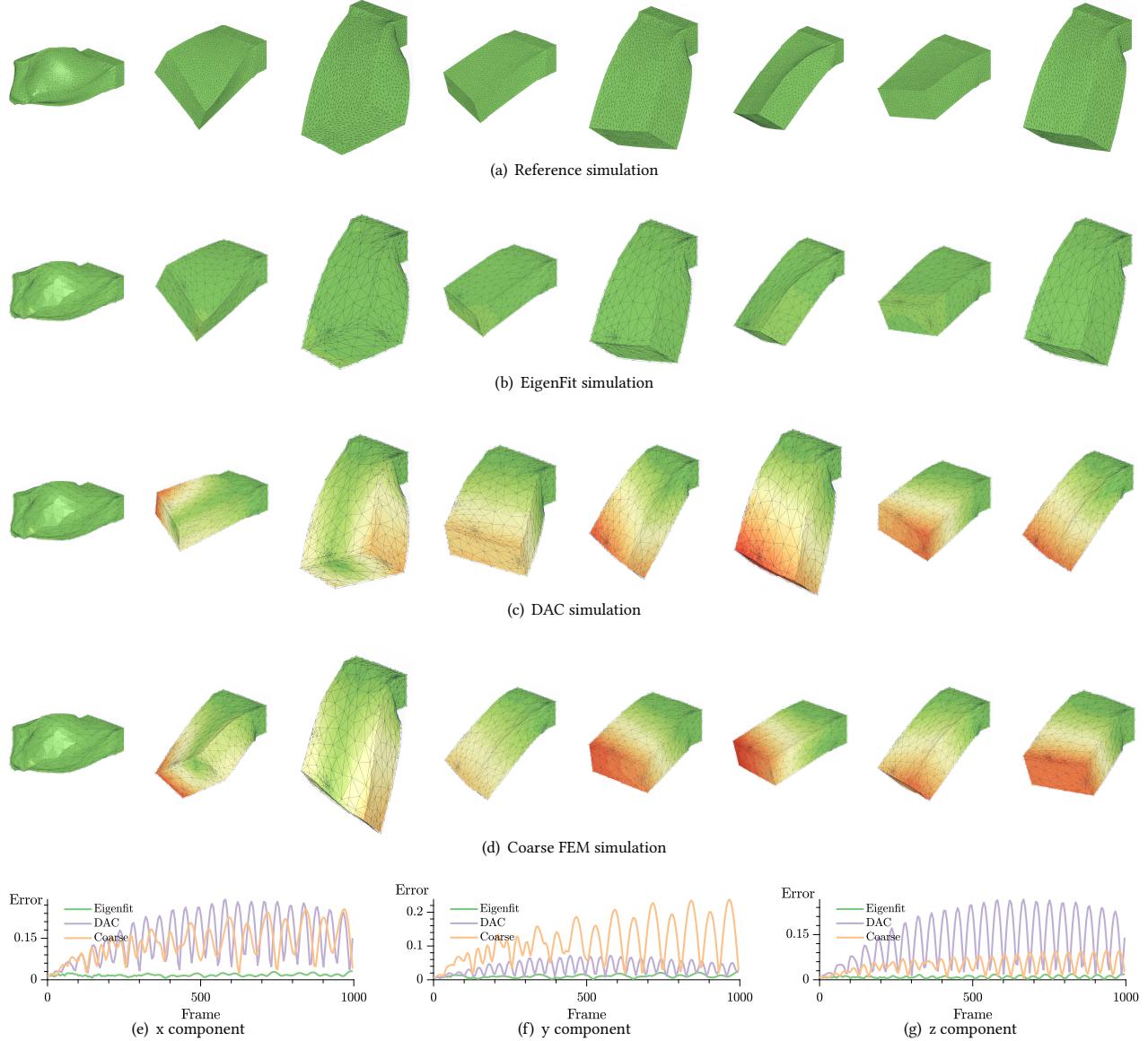


Figure 6: Animation shots for a twisted bar under a linear force, progressing in time from left to right. The fine mesh animation (top row) is faithfully reproduced by Eigenfit on the coarse mesh (2nd row), displaying robustness to irregular meshes. On the other hand, DAC (3rd row) and the raw FEM coarse mesh simulations (bottom row) produce significant, visible disagreements with the fine mesh.

the coarse mesh (hence directly simulating on it could be prohibitively expensive), then the simplest cure may be to make the coarse mesh finer but still not as fine as the finest mesh. This is a common practice in scientific computing; see, e.g., [Trottenberg et al. 2000]. The condition devised in Section 3.5 goes towards ensuring that EigenFit is not applied where its chances to perform well are deemed too low.

Remaining still with homogeneous objects for simplicity sake, we also observe that for the same forces applied to the same object except that the Young modulus varies, the error is smaller the stiffer

the object gets; see Table 3. It's for rather soft bodies exhibiting large deformation where a finer coarse mesh is particularly desirable.

4.4 Heterogeneous deformable objects

Thus far in this narration we have restricted our attention to homogeneous objects in order to concentrate on the many other aspects of the numerical stiffness problem and to enable direct comparison to DAC. However, of course there are many deformable objects in practice, both natural and designed, where the Young modulus is a non-constant function over the object's domain; see Figure 15.

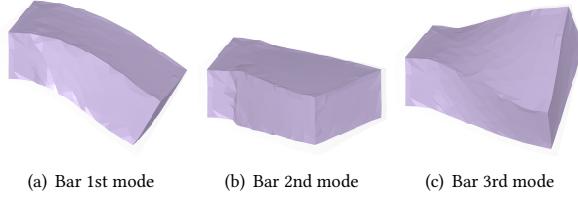


Figure 7: The first three modes of the bar example in Figure 6.

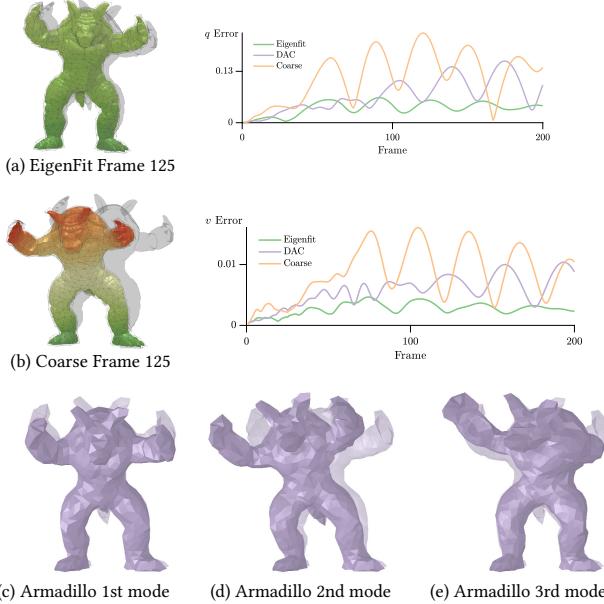


Figure 8: ARAP armadillo with homogeneous material. The SI integrator was used to simulate 200 frames. Grey silhouette shows the reference simulation from a fine mesh. The two plots are maximum position error and velocity error.

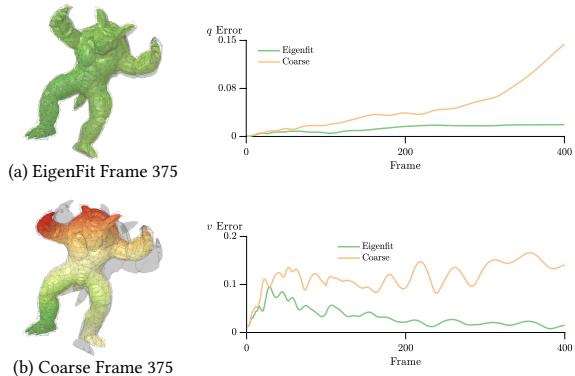


Figure 9: ARAP Armadillo mesh with changing constraints. Color bar was calibrated to fit the max error. The two plots are max position error and velocity error.

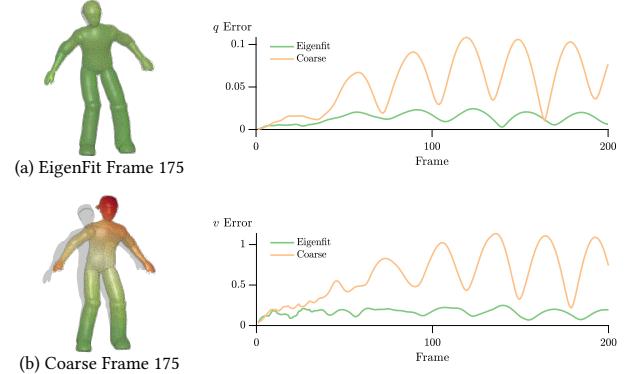


Figure 10: ARAP skater with homogeneous material. The SI integrator was used to simulate 200 frames. Grey silhouette shows the reference simulation from a fine mesh. The two plots are maximum position error and velocity error.

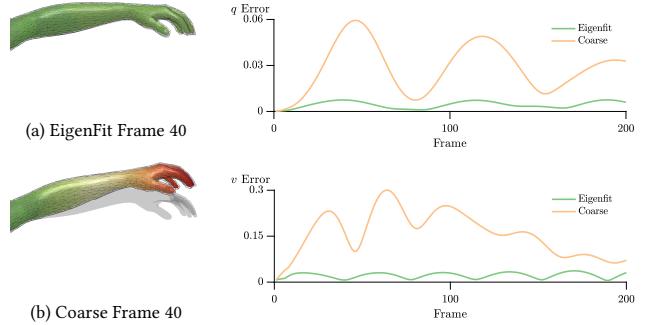


Figure 11: Neo-Hookean arm with heterogeneous material. Color scheme is calibrated to fit the max error. EigenFit and regular coarse FEM are displayed. The error plots are for max position and max velocity error.

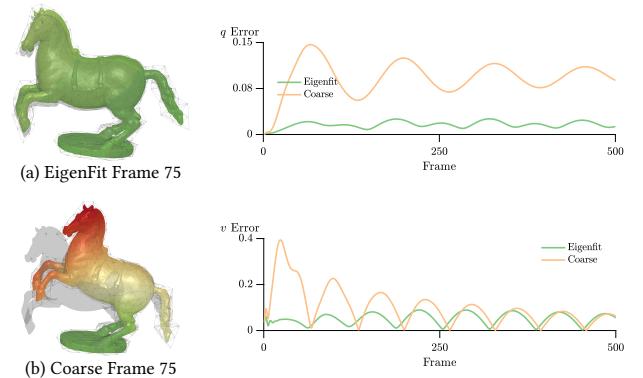


Figure 12: ARAP Rampant mesh with heterogeneous material. Color scheme is calibrated to fit the max error, and EigenFit is compared to regular coarse FEM. The two plots are for max position error and max velocity error.

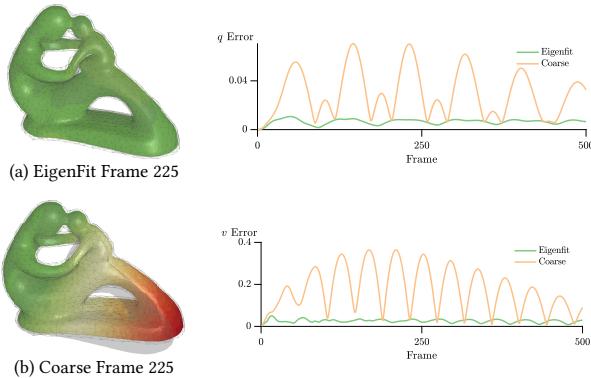


Figure 13: Neo-Hookean Fert mesh with heterogeneous material. Color bar calibrated to fit the max error. The two plots are max position error and velocity error.

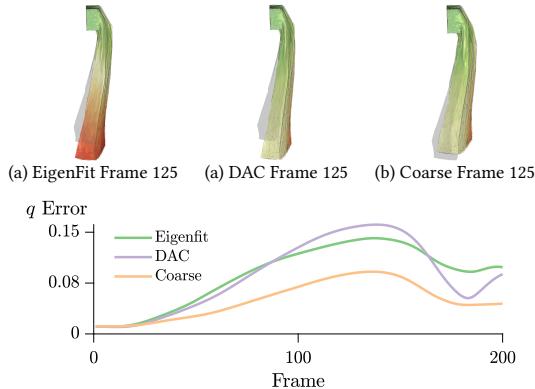


Figure 14: Under large deformation both EigenFit and DAC may perform poorly. Here both methods fail to improve on the original coarse FEM simulation. This is due to the fact that the underlying frequency matching (for DAC) and mode matching (for EigenFit) conditions fail to hold.

Table 2: Relative Simulation Time. A set of coarse and fine meshes were simulated. We report the CPU time, with the relative simulation time given in brackets, using the fine mesh simulation time as the reference.

	Coarse	EigenFit	Fine
Arm (ARAP)	219 (0.26)	444 (0.53)	839
Armadillo (Linear)	224 (0.16)	300 (0.22)	1,387
Armadillo (ARAP)	198 (0.08)	546 (0.23)	2,399
Bar (Linear)	335 (0.20)	386(0.22)	1,683
Fert (Neo-Hookean)	633(0.15)	1,914(0.44)	4,331
Rampant (ARAP)	279(0.06)	634 (0.14)	4,496
Skater (ARAP)	298 (0.23)	865 (0.65)	1,311

Table 3: Error and Stiffness

Stiffness Parameter (Pa)	1e4	1e5	5e5	1e6	1e7
Arm (ARAP)	0.91	1.39	0.615	0.611	0.438
Armadillo (ARAP)	0.838	0.755	0.622	0.307	0.192
Bar (Neo-Hookean)	1.431	0.7281	0.22	0.247	0.292
Hand (ARAP)	1.35	1.04	0.991	0.977	1.09
Skater (ARAP)	0.92	0.594	0.316	0.273	0.182

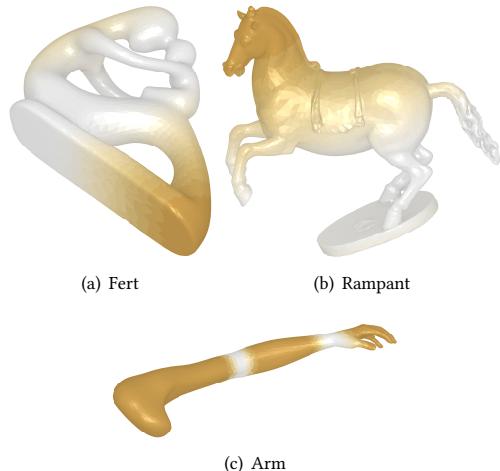


Figure 15: Examples of heterogeneous objects: lighter colour corresponds to lower Young modulus.

To find the distributed parameter function that is Young's modulus in such a case, especially for a natural object, can become significantly more difficult [Wang et al. 2015]. However, the application of EigenFit is independent of finding this function for different meshes. Thus, assuming the availability of Young's modulus and an adequate coarse spatial mesh, EigenFit applies to heterogeneous objects as readily as to homogeneous ones!

Furthermore, EigenFit naturally handles relatively stiff objects where motion is generated through local softer joints, as demonstrated to various degrees in Figure 15. In such cases the leading modes corresponding to the softer joints can be captured well, and larger deformations in the stiffer part of the object can be successfully followed on the coarse mesh.

In Figures 11–13 we used heterogeneous material on the Arm, Rampant, and Fert meshes. Color coding in Figure 15 shows how the stiffness parameter varies spatially in the examples we used; darker color means stiffer, and lighter color is softer. For the Arm mesh, the peak stiffness is $1e8$, and $2e4$ at the minimum. For the Rampant mesh, the peak stiffness is $1e8$, and $2e6$ for the minimum. For the Fert mesh the peak stiffness is $1e9$ and $1e5$ at the minimum. Observe from these figures and the supplementary video that with nonlinear heterogeneous constitutive material, EigenFit improves the simulation results for non-mild deformations from the coarse FEM mesh.

5 CONCLUSIONS, LIMITATIONS AND FUTURE

Mitigating numerical stiffening for coarsened meshes will open the door for elastodynamic simulation in many domains not previously possible. Automated fabrication design requires outer optimization loops that can call expensive FE simulations hundreds of times about each design sample. Simply decreasing the resolution needed for sufficient accuracy would open the door to much faster optimization. Furthermore, as geometric design parameters change so does the computational mesh. If changes in the meshing are allowed to change the effective material stiffness of the simulations, then the entire, exceedingly expensive design optimization is invalidated. Current optimization design tools, e.g., in the automotive industry, apply mesh-warping techniques in these settings that deform the rest mesh in attempt to maintain mesh consistency over changing design parameters. However, large changes in the design-space necessarily require large and often abrupt changes in mesh resolution, again forcing tools to conservatively apply high-resolution meshes.

In this work we have assumed that the material constitutive model is given and have not attempted to change it, e.g., through homogenization techniques. When changing the spatial mesh resolution the apparent physical behaviour of the simulated motion of a flexible object may change. We have presented, analyzed and demonstrated a method that significantly reduces this unwanted mesh dependence, making the simulation consistent across mesh resolutions. The method involves matching the leading eigenvalues of a given coarse mesh with those of a fine reference mesh at rest. This necessitates a partial spectral decomposition, which for nonlinear constitutive laws must be carried out at each time step. For this we have proposed and implemented a model reduction method, EigenFit, where the essential work is carried out in a small subspace of the eigenmodes. We demonstrated EigenFit in action on a range of different meshes for both homogeneous and heterogeneous material objects. In addition to the figures in this paper we refer the reader to the supplementary video, where the gained motion consistency is clearly demonstrated.

We have applied EigenFit to a class of heterogeneous and nonlinear materials. To our knowledge, the dynamic aspect of numerical stiffening and numerical homogenization has not been discussed in detail in the literature, and our assumption that the coarse mesh is given and is “sufficiently fine” is not out of place in many heterogeneous simulation applications (e.g., for animating a teddy bear or a plant).

Notice that “large deformation” in the present context is the amount of deformation that would modify the ordering of the eigenmodes significantly. It is possible for certain mesh configurations to have visually small deformation that still leads to a significant degree of mode crossing. This is indeed the case with the Hand mesh listed in Table 3, where a cluster of eigenvalues/eigenmodes corresponding to the motion of each finger can easily exhibit mode crossing, making the EigenFit mode matching effort ineffective.

For general nonlinear forces with large deformations applied to heterogeneous material an effort is required, extending [Chen et al. 2017], to decide which coarse mesh is “sufficiently fine” to enable a reasonable treatment of the numerical stiffening phenomenon. To that end a practical direction is to start with a coarse mesh as

in [Chen et al. 2017] and gradually refine it for a representative scenario until it is deemed fine enough in the present context. Then use the obtained mesh as the new “coarse mesh” to which EigenFit can be applied.

ACKNOWLEDGMENTS

The authors thank NSERC and Adobe for funding this project.

REFERENCES

- Ivo Babuska, Joseph E Flaherty, William D Henshaw, John E Hopcroft, Joseph E Oliger, and Tayfun Tezduyar. 2012. *Modeling, mesh generation, and adaptive numerical methods for partial differential equations*. Vol. 75. Springer.
- David Baraff and Andrew Witkin. 1998. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 43–54.
- Jernej Barbič and Doug L James. 2005. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM transactions on graphics (TOG)* 24, 3 (2005), 982–990.
- Adam W. Bargteil and Elaine Cohen. 2014. Animation of Deformable Bodies with Quadratic Bézier Finite Elements. *ACM Trans. Graph. (TOG)* 33, 3 (June 2014).
- Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. 2013. *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons.
- Isaac Chao, Ulrich Pinkall, Patrick Saman, and Peter Schröder. 2010. A simple geometric model for elastic deformations. *ACM Trans. Graph. (TOG)* 29, 4 (2010), 38.
- 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 (July 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 (July 2015), 10 pages.
- Jiong Chen, Huijun Bao, Wang Tianyu, Mathieu Desbrun, and Jin Huang. 2018b. Numerical coarsening using discontinuous shape functions. *ACM Trans. Graphics (TOG)* 37, 4 (August 2018).
- Yu Ju Chen, Uri Ascher, and Dinesh Pai. 2018a. Exponential Rosenbrock-Euler Integrators for Elastodynamic Simulation. *IEEE Transactions on Visualization and Computer Graphics* (2018).
- Philippe G Ciarlet. 1988. *Three-dimensional elasticity*. Vol. 20. Elsevier.
- P. G. Ciarlet, M. H. Schultz, and R. S. Varga. 1968. Numerical Methods of High-order Accuracy for Nonlinear Boundary Value Problems IV. Periodic Boundary Conditions. *Numer. Math.* 12, 4 (Nov. 1968), 266–279. <https://doi.org/10.1007/BF02162508>
- Eitan Grinspun, Petr Krysl, and Peter Schröder. 2002. CHARMS: A Simple Framework for Adaptive Simulation. *ACM Trans. Graph. (SIGGRAPH 2002)* 21, 3 (July 2002).
- Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4 (2018).
- Peter Kaufmann. 2012. *Discontinuous Galerkin FEM in computer graphics*. Ph.D. Dissertation. ETH Zurich.
- Peter Kaufmann, Sebastian Martin, Mario Botsch, and Markus Gross. 2009. Flexible simulation of deformable models using discontinuous Galerkin FEM. *Graphical Models* 71, 4 (2009), 153–167.
- Lily Kharayvych, Patrick Mullen, Houman Owhadi, and Mathieu Desbrun. 2009. Numerical Coarsening of Inhomogeneous Elastic Materials. 28, 3 (2009), 51:1–51:8.
- Dingzeyu Li, Yun Fei, and Changxi Zheng. 2015. Interactive Acoustic Transfer Approximation for Modal Sound. *ACM Trans. Graph.* 35, 1 (2015). <https://doi.org/10.1145/2820612>
- J Mosler and M Ortiz. 2007. Variational h-adaption in finite deformation elasticity and plasticity. *Internat. J. Numer. Methods Engrg.* 72, 5 (2007), 505–523.
- Matthieu Nesme, Paul G. Kry, Lenka Jeřábková, and François Faure. 2009. Preserving Topology and Elasticity for Embedded Deformable Models. *ACM Trans. Graph.* 28, 3, Article 52 (July 2009), 9 pages.
- Jorge Nocedal and Stephen Wright. 2006. *Numerical Optimization*. New York: Springer. 2nd Ed.
- Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic Textures for Additive Fabrication. 34, 4 (2015), 135:1–135:12.
- Leonardo Sachet, Etienne Vouga, and Alec Jacobson. 2015. Nested Cages. *ACM Transactions on Graphics (TOG)* 34, 6 (2015).
- Teseo Schneider, Yixin Hu, Jérémie Dumas, Xifeng Gao, Daniele Panozzo, and Denis Zorin. 2018. Decoupling Simulation Accuracy from Mesh Quality. In *SIGGRAPH Asia 2018 Technical Papers (SIGGRAPH Asia '18)*. ACM, New York, NY, USA, Article 280, 14 pages. <https://doi.org/10.1145/3272127.3275067>
- Eftychios Sifakis and Jernej Barbić. 2012a. FEM simulation of 3D deformable solids: a practitioner’s guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*. ACM, 20.

- Eftychios Sifakis and Jernej Barbic. 2012b. FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*. ACM, 20.
- Gilbert W Stewart. 2002. A Krylov–Schur algorithm for large eigenproblems. *SIAM J. Matrix Anal. Appl.* 23, 3 (2002), 601–614.
- Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 181–190.
- Rosell Torres, Alejandro Rodríguez, José M. Espadero, and Miguel A. Otaduy. 2016. High-resolution Interaction with Corotational Coarsening Models. *ACM Trans. Graph.* 35, 6, Article 211 (Nov. 2016), 11 pages.
- Ulrich Trottenberg, Cornelius W Oosterlee, and Anton Schuller. 2000. *Multigrid*. Academic press.
- Bin Wang, Longhua Wu, KangKang Yin, Uri M Ascher, Libin Liu, and Hui Huang. 2015. Deformation capture and modeling of soft objects. *ACM Trans. Graph.* 34, 4 (2015), 94–1.
- Hongyi Xu and Jernej Barbić. 2016. Pose-space subspace dynamics. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 35.
- Hongyi Xu, Funshing Sin, Yufeng Zhu, and Jernej Barbić. 2015. Nonlinear material design using principal stretches. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 75.