

# Numerical Coarsening of Inhomogeneous Elastic Materials

Lily Kharevych

Patrick Mullen

Houman Owhadi

Mathieu Desbrun

Caltech

## Abstract

We propose an approach for efficiently simulating elastic objects made of non-homogeneous, non-isotropic materials. Based on recent developments in homogenization theory, a methodology is introduced to approximate a deformable object made of arbitrary fine structures of various linear elastic materials with a dynamically-similar coarse model. This numerical coarsening of the material properties allows for simulation of fine, heterogeneous structures on very coarse grids while capturing the proper dynamics of the original dynamical system, thus saving orders of magnitude in computational time. Examples including inhomogeneous and/or anisotropic materials can be realistically simulated in realtime with a numerically-coarsened model made of a few mesh elements.

**Keywords:** Homogenization, model coarsening, model reduction.

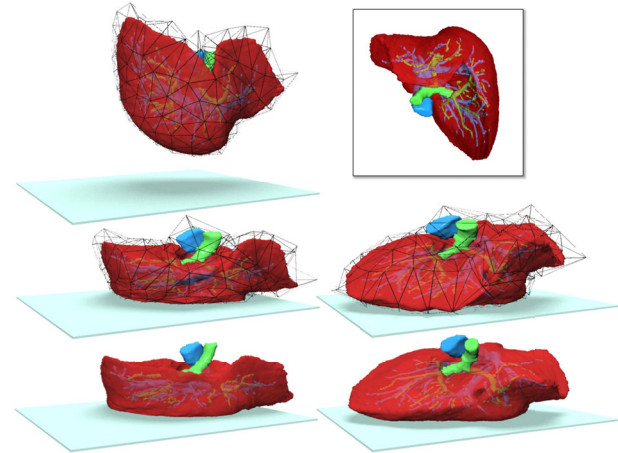
## 1 Introduction

Simulating ever larger and more complex dynamical systems requires ever more elaborate computational methods. While improved CPU speed and faster numerical solvers have allowed exquisitely detailed animation of complex deformable models, one outstanding limitation in computer animation (and in fact, in computational physics) is that simulation costs scale with structural complexity: capturing the proper dynamical behavior of a heterogeneous object requires a mesh fine enough to resolve the fine-scale heterogeneities. This sampling requirement can lead to prohibitive simulation times if the fine scales are geometrically complex, such as veinal structures in an organ. However, simply ignoring these fine scales can dramatically affect the overall dynamics of the object, rendering the object more or less rigid, or even failing to capture basic coarse deformation (see Figures 2 and 5).

In this paper, we present an approach to numerical coarsening of linear elastic objects to allow for interactive, realistic physical animation of structurally complex deformable objects. From a pair of meshes representing respectively a fine and a coarse geometric description of the elastic body, we devise a numerical procedure to turn the heterogeneous elastic properties of the fine mesh into possibly anisotropic elastic properties on the coarse mesh that effectively capture (in the  $H^1$  sense) the same physical behavior. Simulation of the coarse mesh equipped with these resulting coarse elastic properties is thus faithful to the dynamics of the original fine-detailed object, but at a fraction of the cost.

### 1.1 Previous Work

Simplifying a model while still capturing (numerically or visually) its coarse physical behavior has been investigated in many areas of science, of which we discuss the two most relevant to our context.

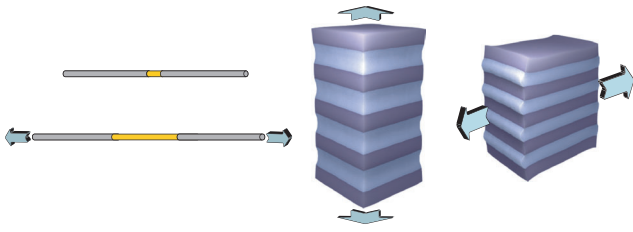


**Figure 1:** Numerical coarsening turns a fine mesh with heterogeneous elastic properties (here, a 200K-tet liver with veins, extracted from MRI data; courtesy of Dobrina Boltcheva, LSIT, France) into a coarse mesh (640 tets) with anisotropic elastic properties that effectively capture the same physical behavior. The coarse mesh (top) can thus be used as a proxy to animate the object (falling on the ground) about a hundred times faster than it would take to compute the elastic behavior on the fine mesh (bottom). Collision detection is done using the interpolated fine mesh boundary.

**Fast Deformable Models** As the complexity of objects required in computer animation grew, several strategies were devised to address the problem of efficiency and scalability. Solvers were made to scale nearly linearly in the number of nodes, through specific integration schemes providing fast, yet stable updates [Baraff and Witkin 1998] and/or multiresolution strategies for which adaptive refinement of the simulation is often based on the local amount of deformation [Debonne et al. 2001; Grinspun et al. 2002; Capell et al. 2002b; Nesme et al. 2006]. Physical models were also drastically simplified to reduce computational requirements [Rivers and James 2007], leading to a resurgence of (sometimes quasi-static) linear elasticity [James and Pai 1999] where the well-known limitations for large deformation are nicely compensated for by corotational methods [de Veubeke 1976; Müller et al. 2002; Hauth and Straßer 2004; Georgii and Westermann 2008].

Another string of contributions focused on *model reduction* to achieve further efficiency. These approaches reduced the state space dimension by limiting the space of possible deformations, typically through eigen-analysis of either the stiffness matrix [Pentland and Williams 1989] or of a set of observations [Krysl et al. 2000]. Model reduction, however, often leads to a significantly reduced set of admissible interactions [James and Fatahalian 2003], to non-local reduced (Ritz) basis functions and a mismatch between geometric and deformation DOFs [Pentland and Williams 1989; James and Pai 2002; Choi and Ko 2005; Treuille et al. 2006], and/or to non-linear runtime complexity [Li and Bai 2005; Barbič and James 2005; An et al. 2008].

A simple, yet efficient way to simulate complex objects at low cost is through *domain embedding*, where a complex geometry is embedded in a coarse mesh (sometimes referred to as a cage). Animating the coarse volumetric mesh induces deformation of the high resolution embedded geometry, providing a cheap and easy dynamic approximation of a complex object's behavior [Pentland and Williams 1989; Debonne et al. 2001; Capell et al. 2002a; Müller and Gross 2004; Wojtan and Turk 2008]. The main advantage of



**Figure 2: Inhomogeneous materials leads to Anisotropic Behavior:** In 1D (left), even a tiny amount of soft material between two rigid rods renders the resulting bar highly deformable when pulled; a cube of composite material in 3D (right) made out of two materials (the blue one being softer than the mauve one) exhibits significant anisotropy due to its composition: in this case, it stretches much more vertically than horizontally.

this approach is that the geometric and physical DOFs coincide, eliminating the non-sparsity issue of reduced spaces; one can reuse the exact same material simulator, but now on a coarse embedding mesh rather than on a fine, detailed geometry. However, methods of this last class focus almost exclusively on homogeneous materials. When a heterogeneous object is at play, the coarse mesh must now be assigned “averaged” material properties to best match the behavior of the original object. To our knowledge, the only work that tackled this issue in computer animation is [Nesme et al. 2006], where a spatial average of the elasticity tensor was proposed on cubical grids<sup>1</sup>. Yet, such a simple average does not accurately coarsen an elastic material: this procedure in 1D amounts to averaging a set of springs in series by their mean stiffness, while the correct equivalent stiffness is the inverse of the sum of the reciprocal of each fine stiffness (this relation is easily derived using the well-known electro-mechanical analogy; unfortunately, no such result in 2D or 3D is known). The difference between these two coarsening approaches can be quite significant: consider a 1D system made out of a very small and very soft section sandwiched between two large, extremely rigid sections; the resulting system is obviously very soft (see Figure 2(left)), whereas a spatial average would have made it very stiff. A proper numerical coarsening is thus crucial when approximating the dynamics of heterogeneous objects on coarse grids.

**Scalar Homogenization** Homogenization theory [Bensoussan et al. 1978; Jikov et al. 1991] has been developed for this exact purpose of extracting information from fine scales to computational scales in order to perform efficient computations over composite, inhomogeneous materials (*i.e.*, with spatially varying physical properties, such as laminates, rebar-reinforced concrete, etc). In essence, this theory replaces the microscopic structure of a composite by an idealized, locally-homogeneous material with equivalent macroscopic physical properties—a procedure referred to as coarsening, homogenization, or upscaling. A variety of coarsening methods have been proposed, starting from well-known arithmetic and harmonic averages, to more involved ones like the renormalization method or the representative elementary volume; see [Farmer 2002] for a review. Unfortunately, the numerical homogenization techniques available so far can offer accurate results *only* if periodicity, ergodicity, or scale-separation assumptions on the material properties are satisfied, making them quite poorly adapted to our needs in computer animation.

Recently a homogenization technique of elliptic equations in divergence form requiring *no assumptions* on the material at hand was proposed in [Owhadi and Zhang 2007], providing a way to approximate a fine solution  $u$  of the static problem:

$$\begin{cases} \operatorname{div}(C(x)\nabla u) = f & \text{inside a domain } \mathbf{D} \\ u = u_0 & \text{on } \partial\mathbf{D} \end{cases} \quad (1)$$

with spatially-varying conductivity  $C(x)$ , by a coarse function  $u_h$

<sup>1</sup>See also the contemporaneous extension [Nesme et al. 2009].

that provably satisfies:  $\|u - u_h\|_{H^1} \leq Ch\|f\|_{L^2}$ . This coarse function  $u_h$  is thus a good match (in the  $H^1$  sense, thus even better in the  $L^2$  sense) for the real (fine) solution, even if it requires a much cheaper numerical solve. While seemingly appropriate for elastostatics given the similarities between Eq. (1) and the balance equation in elasticity, this technique is limited to the scalar case (as well other related recent methods, see [Babuška and Sauter 2008; Shu et al. 2008]). Additionally, general boundary conditions or non-conforming coarse meshes are neither discussed nor tackled.

## 1.2 Overview and Contributions

**Problem Statement** In this paper, we present a practical solution to the following numerical coarsening problem:

Given (a) a fine tetrahedral mesh  $\mathbf{D}$ , in which each tet  $T_p$  ( $p = 1 \dots |\mathbf{D}|$ ) has a different elasticity tensor  $\mathbf{C}_{T_p}$ , and (b) a coarse tetrahedral mesh  $\mathbf{D}$  (of much smaller element count, *i.e.*,  $|\mathbf{D}| \ll |\mathbf{D}|$ ) approximating the same geometry as  $\mathbf{D}$ , find an “effective” elasticity tensor  $\mathbf{C}_{T_q}$  per coarse tet  $T_q$  such that the overall dynamics obtained by an off-the-shelf elasticity simulator applied to either of them matches well.

Our coarsening procedure is achieved by first computing on  $\mathbf{D}$  a set of global harmonic displacements to analyze the heterogeneous fine-scale properties, then by deducing the effective coarse-scale property for each coarse mesh element. We show how our approach can be seen as a mollification of the displacement field to allow for proper averaging of the physical properties, and how it matches the simple, known case of elasticity in 1D. Note that even if the object described by the fine mesh is made out of different isotropic materials (*i.e.*, with the same stiffness independent of the directional orientation of the applied force), the resulting elasticity tensors at the coarse level are often *anisotropic* as they reflect the object’s fine, inhomogeneous composition (see Figure 2).

Our approach contrasts with previous work in several ways. Unlike methods based on Krylov spaces (using various definitions of “eigen” deformations), our coarse model is not limited to a linear space of deformations, and does not involve reduced coordinates not matching the geometric description of the object. Instead, our coarse model is simulated with a traditional finite-element solver but on a coarser grid approximating the object’s global geometry. The resulting dynamical system can thus be deformed arbitrarily with a computational complexity proportional to the size of the coarse mesh. Additionally, the accuracy of our approach decays gracefully with the maximum edge length of the coarse mesh.

## 1.3 Tensor Notation

Our exposition will make heavy use of tensors of various ranks as required in elasticity theory. To facilitate the direct implementation of our approach, we will often employ an index-based notation, where a rank-1 tensor  $X$  (*i.e.*, vector, or 1D array for coding purposes) has its components denoted as  $X_i$  (where  $i$  takes on the values 1, 2, or 3), a rank-2 tensor  $Y$  (*i.e.*, a matrix, or 2D array) will have its components referred to as  $Y_{ij}$ , and so forth. The “Einstein” summation convention (where summation is implied by repeated indices [Barr 1989]) will be assumed. We will also employ the concise notion of tensor *contraction*. A single contraction, where a summation over a single index is used, will be denoted as “.”;  $X \cdot Y$  will therefore refer to either a matrix-vector product  $[X \cdot Y]_i = X_{ia}Y_a$  or a matrix-matrix product  $[X \cdot Y]_{ij} = X_{ia}Y_{aj}$ . A double contraction, where now a summation over two indices is performed, will be denoted as “..”. It will be used for products of a rank-4 tensor by a rank-2 tensor, or products of two rank-4 tensors— $[F : G]_{ij} = F_{ijab}G_{ab}$ , and  $[F : G]_{ijkl} = F_{ijab}G_{abkl}$  respectively. Finally, a comma will indicate differentiation with respect to one or several of the coordinates as customary, while  $\nabla$  will refer to the gradient operator; *e.g.*,  $(\nabla Y)_{ij} = Y_{i,j} = \partial Y_i / \partial x_j$  for a rank-1 tensor,  $(\nabla Y)_{ijk} = Y_{i,jk} = \partial Y_{ij} / \partial x_k$  for a rank-2 tensor, etc.

## 2 Background on Linear Elasticity

Linear elasticity has received extensive attention, and many detailed explanations of its foundations can be found throughout the literature. We recap the main notions nonetheless, by way of introduction to our notation and to the delicate issues of tensor symmetries that will arise in our coarsening approach.

**Strain Tensor** Given a displacement field  $\mathbf{u}$  defined over the undeformed configuration of an object, the symmetric part of the deformation gradient  $\boldsymbol{\epsilon}$ , computed as

$$\boldsymbol{\epsilon} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T) \quad (\text{i.e., } \epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})), \quad (2)$$

represents the strain undergone by the object. Indeed, the remaining part is antisymmetric, and thus only represents a pure (infinitesimal) rotation—which does not induce a deformation. Note that by definition, we have  $\epsilon_{ij} = \epsilon_{ji}$ . This strain tensor is a linearized version of the more general Cauchy strain tensor considered in non-linear elasticity.

**Potential Energy** Hookean materials are assumed to have, just like a simple spring, a potential function  $W(\mathbf{u})$  that is quadratic in the strain tensor:

$$W(\mathbf{u}) = \frac{1}{2} \boldsymbol{\epsilon} : \mathbf{C} : \boldsymbol{\epsilon} = \frac{1}{2} \epsilon_{ij} C_{ijkl} \epsilon_{kl}, \quad (3)$$

where  $\mathbf{C} = \{C_{ijkl}\}$  is a rank-4 tensor called the *elasticity tensor* (sometimes referred to as the compliance tensor, or tensor of elastic compliances).

**Stress Tensor** The stress tensor  $\boldsymbol{\sigma}$ , describing the forces acting inside the body, is then defined as:  $\boldsymbol{\sigma} = \partial W / \partial \boldsymbol{\epsilon}$ . Each component  $\sigma_{ij}$  of this tensor represents, pointwise, the  $i$ -component of the force on a surface with unit area whose normal is in the  $j$ -direction. Consequently, body forces are expressed by the divergence of the stress tensor, resulting in the usual equation of motion for linear elasticity:

$$\rho \ddot{\mathbf{u}} = \text{div } \boldsymbol{\sigma} + \mathbf{f},$$

where  $\mathbf{f}$  represents the external forces applied to the material.

**Material Law** From these basic definitions, one can deduce that, for a Hookean (linear elastic) material, the elasticity tensor  $\mathbf{C}$  defines a linear relationship between stress and strain since  $\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\epsilon}$ . In slot-naming index convention, it yields:

$$\sigma_{ij} = C_{ijkl} \epsilon_{kl} \quad (4)$$

**Symmetries of Elasticity Tensor** While being a rank-4 tensor (thus with 81 components in 3D), the elastic tensor  $\mathbf{C}$  possesses several symmetries. The symmetry of the stress tensor implies that  $\mathbf{C}$  is symmetric in its first pair of indices ( $C_{ijkl} = C_{jikl}$ ), while the symmetry of the strain tensor results in  $\mathbf{C}$  being symmetric in its second pair of indices ( $C_{ijkl} = C_{ijlk}$ ). Finally, since the strain energy is a quadratic form, we also have symmetry under an interchange of the first and second pairs of slots ( $C_{ijkl} = C_{klij}$ ) [Feynman et al. 2006]. This leaves only 21 independent components in 3D.

However, if the material is further assumed to be *isotropic*, as is pervasive in graphics, then  $\mathbf{C}$  only possesses 2 independent components, usually expressed as Lamé coefficients  $\lambda$  and  $\mu$ , or as Young modulus  $E$  and Poisson ratio  $\nu$ . Note that this isotropic assumption amounts to assuming that deformations within the medium have no preferred direction. While this is a valid simplification for a homogenous material, this clearly breaks in our coarsening context: anisotropy has to be accounted for as Figure 2 demonstrates.

## 3 Coarsening Methodology

We now present how we derive, from an elastic, inhomogeneous object, a coarse approximation that possesses a similar physical behavior. We assume that the procedure is performed in dimension  $d$ , where  $d = 1, 2$ , or 3 for generality. Additionally, for consistency and clarity, we use ROMAN characters to refer to quantities living on the fine mesh  $\mathbf{D}$ , and BLACKBOARD characters to refer to quantities on the coarse mesh  $\mathbf{D}$ .

### 3.1 Coarsening Procedure Setup and Overview

We start from a fine mesh  $\mathbf{D}$ , in which each tet  $T_p$  ( $p = 1 \dots |\mathbf{D}|$ ) has a different elasticity tensor  $\mathbf{C}_{T_p}$  (i.e., a different set of Lamé coefficients if we assume isotropy of each fine element). We wish to approximate its dynamics on a given coarser mesh  $\mathbf{D}$  (with  $|\mathbf{D}| \ll |\mathbf{D}|$ ) that describes the same geometry. That is, we need to find, as a precomputation, an effective elasticity tensor  $\mathbf{C}_{T_q}$  on each coarse mesh element  $T_q$  so that the dynamics of the resulting coarse system closely matches the original fine object.

$$(\mathbf{D} = \{T_p\}, \mathbf{C} = \{C_{T_p}\}) \xrightarrow{\text{coarsening}} (\mathbf{D} = \{T_q\}, \mathbf{C} = \{C_{T_q}\})$$

Our approach first “probes” the fine material by computing  $d(d+1)/2$  harmonic displacements to capture how the fine mesh behaves when linear forces are applied to the boundary of the material. This set of displacements will in turn be used to derive a coarsening procedure to enforce that the potential energy of the coarse mesh ( $\mathbf{D}, \mathbf{C}$ ) exactly matches the integral of the potential of the fine mesh ( $\mathbf{D}, \mathbf{C}$ ) within each coarse tet  $T_q$ . This coarsening procedure can be seen as an extension of the upscaling procedure with discontinuous elements introduced in Section 1.3 of [Owhadi and Zhang 2007] for scalar equations.

### 3.2 Downsampling Fields

Downsampling a field from the fine mesh  $\mathbf{D}$  to the coarse mesh  $\mathbf{D}$  is easily achieved. Each coarse-mesh vertex position  $\mathbf{X}_i$  is expressed as a linear combination (through barycentric coordinates) of the fine-mesh vertices  $\mathbf{x}_i$  defining the fine tetrahedron in which  $\mathbf{X}_i$  lies at rest. In other words, the vertex positions  $\mathbf{x}_i$  of the fine mesh are first interpolated by linear finite elements on the fine mesh  $\mathbf{D}$ , then the coarse nodes are defined as samples of this linear reconstruction. Boundary nodes that lie outside of the fine domain  $\mathbf{D}$  require special treatment: for these nodes that do not have a bounding fine tet, we find one (or more) fine element(s) closest to it and use barycentric extrapolation instead (i.e., negative barycentric coordinates). More details of this procedure and special handling of boundaries will be discussed in Section 4.

Notice however that such a downsampling is accurate *only* if the field we downsample is sufficiently *smooth*. While displacements of the objects throughout an animation can be assumed to be fairly smooth, this is far from true for a field like the elasticity tensor  $\mathbf{C}$ , as we assume the material to be inhomogeneous on fine scales.

### 3.3 Numerical Coarsening Rationale

While displacements are easily downsampled, the elasticity tensor and mass matrix require more care to enforce that the coarse dynamics closely approximates the fine dynamics.

**Potential Energy** We first need to derive a tensor  $\mathbf{C}_{T_q}$  per coarse tet  $T_q$ . In order to correctly reproduce the force field within the object, we should enforce that the potential energy of each coarse tet matches the integral of the potential energy over the fine tets contained within the coarse tet; i.e., given our setup, we should target the following equality:

$$\int_{T_q} \boldsymbol{\epsilon}(\mathbf{u}) : \mathbf{C} : \boldsymbol{\epsilon}(\mathbf{u}) dV = \int_{T_q} \boldsymbol{\epsilon}(\mathbf{U}) : \mathbf{C} : \boldsymbol{\epsilon}(\mathbf{U}) dV$$

on each coarse tet  $T_q$  for *all* possible deformation fields  $\mathbf{u}$ . This is a tall order, as even if each (potentially anisotropic) coarse elasticity tensor has 21 degrees of freedom, the space of possible fine deformations is significant. Therefore the best we can hope to achieve is to perfectly capture this equality on a few displacements: if these displacements are characteristic of the typical deformations that the fine mesh can endure, we will have achieved our goals. As we will show below, we will introduce characteristic displacements  $\mathbf{h}_{\alpha\beta}$  (with  $1 \leq \alpha \leq \beta \leq d$ , for a total of  $d(d+1)/2$  displacement fields)

and derive coarse elasticity tensors that enforce:

$$\int_{\mathbf{T}_q} \boldsymbol{\epsilon}(\mathbf{h}_{\alpha\beta}) : \mathbf{C} : \boldsymbol{\epsilon}(\mathbf{h}_{\delta\gamma}) dV = \int_{\mathbf{T}_q} \boldsymbol{\epsilon}(\mathbf{H}_{\alpha\beta}) : \mathbf{C} : \boldsymbol{\epsilon}(\mathbf{H}_{\delta\gamma}) dV \quad (5)$$

on each coarse tet  $\mathbf{T}_q$ , and for all  $\alpha \leq \beta$  and  $\delta \leq \gamma$  (note that  $\mathbf{H}_{\alpha\beta}$  is the upscaled displacement based on  $\mathbf{h}_{\alpha\beta}$  as defined in Section 3.2). This results in 21 independent equations in 3D after accounting for the major symmetry of  $\mathbf{C}$ , and their enforcement is equivalent to enforcing potential energy equality for all linear combinations of the test displacements. We will also demonstrate that our particular choice of characteristic displacements leads to a variational interpretation, giving another justification of our approach. Note that for clarity we will continue to use Greek letter indices for indexing among characteristic displacements, while Roman letter indices will still be used for coordinates and gradients.

**Kinetic Energy** As one of our goals in this paper is to be able to reuse a conventional finite-element solver on the coarse model, we do not allow the coarse mass matrix to be anisotropic since most implementations that we are aware of in graphics assume a lumped, diagonal mass matrix. Therefore, we define the mass matrix  $\mathbf{M}$  to be a diagonal matrix, for which the diagonal elements represent the usual lumped mass around each node of the coarse mesh. Note that to obtain a better coarsening of the mass matrix such that the kinetic energies match well:

$$\frac{1}{2} \int_{\mathbf{D}} \dot{\mathbf{U}}^T \cdot \mathbf{M} \cdot \dot{\mathbf{U}} \approx \frac{1}{2} \int_{\mathbf{D}} \dot{\mathbf{u}}^T \cdot \mathbf{M} \cdot \dot{\mathbf{u}},$$

our treatment of the potential energy could also be used—where now we need to compute “characteristic frequencies” as it involves time derivatives. We omit this treatment here because unless the mass density contrast in the object is significant, phase errors on the final behavior of the coarsened system are unlikely to be visually crucial to be worth the extra computational time required by non-lumped (non-diagonal) mass matrices.

### 3.4 Global Harmonic Displacements

We first compute a few defining displacements  $\mathbf{h}_{\alpha\beta}$  to study how the fine mesh behaves under a set of chosen conditions. For our purposes of simulating elastic objects, we compute these displacements by solving the following set of static boundary value problems for  $\{\mathbf{h}_{\alpha\beta}\}_{1 \leq \alpha \leq \beta \leq d}$ :

$$\begin{cases} \operatorname{div}(\mathbf{C} : \boldsymbol{\epsilon}(\mathbf{h}_{\alpha\beta})) = 0 & \text{inside } \Omega \\ (\mathbf{C} : \boldsymbol{\epsilon}(\mathbf{h}_{\alpha\beta})) \cdot \mathbf{n} = \boldsymbol{\epsilon}(x_\alpha \mathbf{e}_\beta) \cdot \mathbf{n} & \text{for } \mathbf{x} \in \partial\Omega, \end{cases} \quad (6)$$

where  $x_\alpha$  denotes the  $\alpha$ -th coordinate of space and  $\mathbf{e}_\beta$  is the unit vector in the  $\beta$ -th coordinate direction. Note that we can thus rewrite  $\boldsymbol{\epsilon}(x_\alpha \mathbf{e}_\beta)$  as  $\frac{1}{2}(\mathbf{e}_\alpha \otimes \mathbf{e}_\beta + \mathbf{e}_\beta \otimes \mathbf{e}_\alpha)$ . We fix the last six degrees of freedom (translation and rotation) of Eq. (6) by fixing the zero-th and first moments, resulting in a unique solution. The reader may recognize the typical requirement of  $\mathbf{C}$ -harmonicity, along with Neumann boundary conditions prescribing surface tractions equal to  $\boldsymbol{\epsilon}(x_\alpha \mathbf{e}_\beta) \cdot \mathbf{n}$ . We will thus refer to this family of  $d(d+1)/2$  static solutions as “global harmonic displacements” (see Figure 3 for basic examples on both homogeneous and inhomogeneous materials). These static solutions represent characteristic displacements resulting from a global “probing” of the object by a set of linear traction fields on the boundary. For notational simplicity, we will denote by  $\mathbf{H}$  the rank-3 tensor whose components are the coordinates of every harmonic displacement  $\mathbf{h}_{\alpha\beta}$ , *i.e.*,  $\mathbf{H}_{k\alpha\beta} = (\mathbf{h}_{\alpha\beta})_k$ . We finally symmetrize  $\mathbf{H}$  through  $\mathbf{H}_{k\alpha\beta} = \mathbf{H}_{k\beta\alpha}$  for simplicity, as it avoids having the restriction  $\alpha \leq \beta$  in further equations.

### 3.5 Harmonic Mollifier

The *symmetric part* of the gradient of the tensor  $\mathbf{H}_{k\alpha\beta}$  will play a crucial role in coarsening. This rank-4 tensor  $\mathbf{G}$  is defined as

$$\mathbf{G}_{k\alpha\beta} = \frac{1}{2} (\mathbf{H}_{k\alpha\beta,l} + \mathbf{H}_{l\alpha\beta,k}).$$

Note that this last expression is a generalization of the symmetrized gradient operator  $\boldsymbol{\epsilon}$  for rank-2 tensors, and therefore the resulting  $\mathbf{G}$  has the minor symmetry  $\mathbf{G}_{k\alpha\beta} = \mathbf{G}_{l\alpha\beta}$  as well as  $\mathbf{G}_{k\alpha\beta} = \mathbf{G}_{k\beta\alpha}$  thanks to the symmetry of  $\mathbf{H}$ . Although of higher-order in our case, this tensor can be shown to help mollify solutions of the elastic equation just as [Owhadi and Zhang 2007] demonstrated in the scalar case of anisotropic Poisson equations: we also observe that for any displacement  $\mathbf{u}$  of our fine object, the field  $\mathbf{G}^{-1} : \boldsymbol{\epsilon}(\mathbf{u})$  becomes Hölder continuous, *i.e.*, quite smooth, though not necessarily Lipschitz. This will be particularly useful: this “mollified” field *can be approximated (for any reasonably smooth displacement field) on the coarse level without significant loss of information, by subsampling each term*:

$$\mathbf{G}^{-1} : \boldsymbol{\epsilon}(\mathbf{u}) \approx \mathbf{G}^{-1} : \boldsymbol{\epsilon}(\mathbf{U}) \quad (7)$$

where  $\mathbf{G}$  is the coarse mesh analogue of  $\mathbf{G}$ . This property is crucial in getting accurate coarsening.

### 3.6 Homogenization of Fine Scales

We finally “downsample” the elasticity tensor  $\mathbf{C}$  as follows, so as to preserve the symmetries of the coarse elasticity tensor mentioned in Section 2:

$$\mathbf{C}_{\mathbf{T}_q} := \mathbf{G}_{\mathbf{T}_q}^{-T} : \langle \mathbf{G}^T : \mathbf{C} : \mathbf{G} \rangle_{\mathbf{T}_q} : \mathbf{G}_{\mathbf{T}_q}^{-1}, \quad (8)$$

or rewritten using tensor notation,

$$[\mathbf{C}_{\mathbf{T}_q}]_{ijkl} := [\mathbf{G}_{\mathbf{T}_q}^{-T}]_{i\alpha\beta} [\langle \mathbf{G}^T : \mathbf{C} : \mathbf{G} \rangle_{\mathbf{T}_q}]_{\alpha\beta\gamma\delta} [\mathbf{G}_{\mathbf{T}_q}^{-1}]_{\gamma\delta kl}.$$

This coarsening is achieved by first averaging quantities on the fine mesh  $\mathbf{D}$  through:

$$[\langle \mathbf{G}^T : \mathbf{C} : \mathbf{G} \rangle_{\mathbf{T}_q}]_{\alpha\beta\gamma\delta} := \sum_{\substack{\mathbf{T}_p \in \mathbf{D} \\ \mathbf{T}_p \cap \mathbf{T}_q \neq \emptyset}} \frac{|\mathbf{T}_p \cap \mathbf{T}_q|}{|\mathbf{T}_q|} [\mathbf{G}_{\mathbf{T}_p}^T]_{\alpha\beta ij} [\mathbf{C}_{\mathbf{T}_p}]_{ijkl} [\mathbf{G}_{\mathbf{T}_p}]_{k\ell\gamma\delta},$$

then by computing the inverse of the tensor  $\mathbf{G}$  on the coarse mesh. We stress that this inverse needs to be done with care: this is an inverse in the (reduced) space of tensors acting on *symmetric* tensors. However, as we will represent this tensor in the reduced space, this will be a standard 6x6 matrix inverse. Please refer to Appendix B if unaccustomed to this matrix representation of tensors.

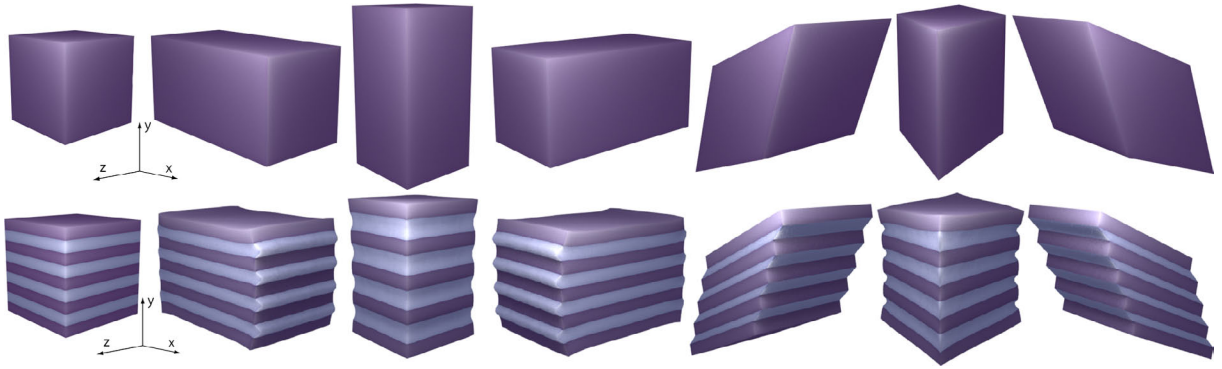
This procedure for deriving an effective tensor not only respects the symmetries that any elastic tensor should have, but also satisfies Eq. (5).

### 3.7 Variational (Finite Element) Interpretation

Since the traditional finite-element variational treatment of elasticity considers the weaker form of the divergence term by pairing it with another arbitrary “test” deformation  $\mathbf{z}$ , we can now write (discarding boundary terms for clarity):

$$\begin{aligned} \int_{\mathbf{D}} \operatorname{div}(\mathbf{C} : \boldsymbol{\epsilon}(\mathbf{u})) \cdot \mathbf{z} &= \int_{\mathbf{D}} \boldsymbol{\epsilon}(\mathbf{z}) : \mathbf{C} : \boldsymbol{\epsilon}(\mathbf{u}) \\ &= \int_{\mathbf{D}} \boldsymbol{\epsilon}(\mathbf{z}) : \mathbf{G}^{-T} : \mathbf{G}^T : \mathbf{C} : \mathbf{G} : \mathbf{G}^{-1} : \boldsymbol{\epsilon}(\mathbf{u}) \\ &\stackrel{*}{\approx} \int_{\mathbf{D}} \boldsymbol{\epsilon}(\mathbf{Z}) : \mathbf{G}^{-T} : \mathbf{G}^T : \mathbf{C} : \mathbf{G} : \mathbf{G}^{-1} : \boldsymbol{\epsilon}(\mathbf{U}) \\ &= \int_{\mathbf{D}} \boldsymbol{\epsilon}(\mathbf{Z}) : \mathbf{C} : \boldsymbol{\epsilon}(\mathbf{U}) = \int_{\mathbf{D}} \operatorname{div}(\mathbf{C} : \boldsymbol{\epsilon}(\mathbf{U})) \cdot \mathbf{Z} \end{aligned}$$

where the step marked by the asterisk is a consequence of the mollification property in Eq. (7) used on both  $\mathbf{u}$  and  $\mathbf{z}$ . Therefore, our definition of the upscaled elasticity tensors can be seen as a particular choice of a test function  $\mathbf{z}$  for which the upscaled test function  $\mathbf{Z}$  is a linear basis function of the coarse mesh  $\mathbf{D}$ , so that a *typical linear finite-element treatment of the coarse mesh closely corresponds to a finite-element treatment of the fine mesh*.



**Figure 3: Harmonic Displacements:** The six harmonic displacements obtained from a homogenous material (top), and a heterogeneous material made of layers of 2 different elastic materials (bottom). The deformations correspond to respectively:  $\mathbf{h}_{11}$ ,  $\mathbf{h}_{22}$ ,  $\mathbf{h}_{33}$ ,  $\mathbf{h}_{12}$ ,  $\mathbf{h}_{23}$ , and  $\mathbf{h}_{13}$ .

### 3.8 Discussion

Our specific procedure to accurately downsample the elasticity tensor field of a fine elastic object can be understood either from the variational point of view (through mollification of the displacement, Section 3.7), or from the exact matching of the potential energy for a set of characteristic displacements. Other variants can also be derived, potentially at the cost of losing one of these two properties. For instance, characteristic functions satisfying an alternate set of boundary value problems could be chosen. In particular, the boundary conditions of the harmonic equation should be changed to mixed Dirichlet/Neumann conditions *if* we know that some vertices will be fixed during simulation: this will capture harmonic displacements that are more appropriate to this particular use. Other extensions could relax the exact enforcement of Eq. (5), and consider a least square solution for a larger family of carefully-tuned characteristic displacements instead if prior knowledge on the use of the coarse simulation is available. Also, computing *local* harmonic characteristic fields would become attractive if the mesh topology is allowed to change over time: we will leave this local approach for future work, as a careful study of the consequences of these multiple local solves versus a global solve for coarsening is delicate to perform, by lack of a proper metric to use for fair comparison.

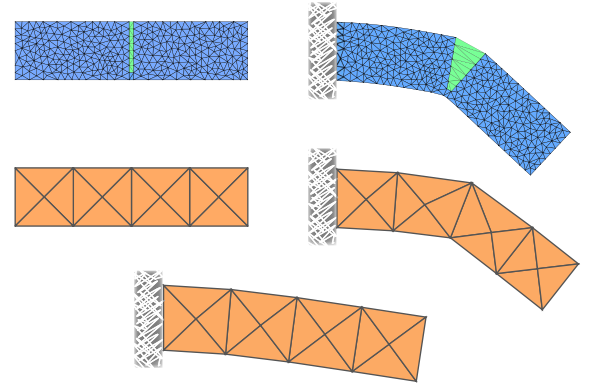
With our proposed approach, we can piggyback on the analysis provided in [Owhadi and Zhang 2007] (with further details in [Berlyand and Owhadi 2009]) to conclude that our coarse simulation using  $\mathbb{U}$  will satisfy:  $\|\mathbf{u} - \mathbb{U}\|_{\mathcal{H}^1} \leq \alpha h \|\mathbf{f}\|_{\mathcal{L}^2}$  (where  $h$  is the maximum size of a coarse element), thus by duality,  $\|\mathbf{u} - \mathbb{U}\|_{\mathcal{L}^2} \leq C\alpha h^2 \|\mathbf{f}\|_{\mathcal{L}^2}$ . In practice, this implies that the error made by the coarsening procedure is of the order of the size of the coarse mesh.

## 4 Implementation Details

Although 3D numerical coarsening is mostly achieved by solving six harmonic displacement fields and a few linear algebra operations as we explained in the previous section, several components deserve more details.

### 4.1 Symmetric Tensor Representation

While our presentation has consistently used tensor notation, implementation can be done using 6D vectors to represent symmetric rank-2 (3x3) tensors, and 6x6 matrices to encode rank-4 tensors: this memory-efficient representation often used in computer graphics exploits the symmetries of the tensor we have to deal with. For completeness, we spell out this conversion in Appendix B. In this representation, double contractions can be performed by 6x6 matrix products, and the entity  $\mathbb{G}^{-1}$  is exactly the inverse of the 6x6 matrix  $\mathbb{G}$  used to encode the harmonic mollifier. Alternatively, one can implement coarsening by copying literally the formulae provided in our explanations using arrays and their indices—although  $\mathbb{G}^{-1}$  will then require special care.



**Figure 4: Coarsening of Cracks:** (left) In this 2D example, coarsening is used to turn a bar-like object (blue) containing a thin slice of soft material (green) into a very coarse mesh (peach-colored mesh); (right) when deformed under gravity, both models present similar deformations; (bottom) a simple spatial averaging of the material elasticity coefficients or stiffness tensors does not capture this bending behavior, not accounting properly for the weak material in the middle.

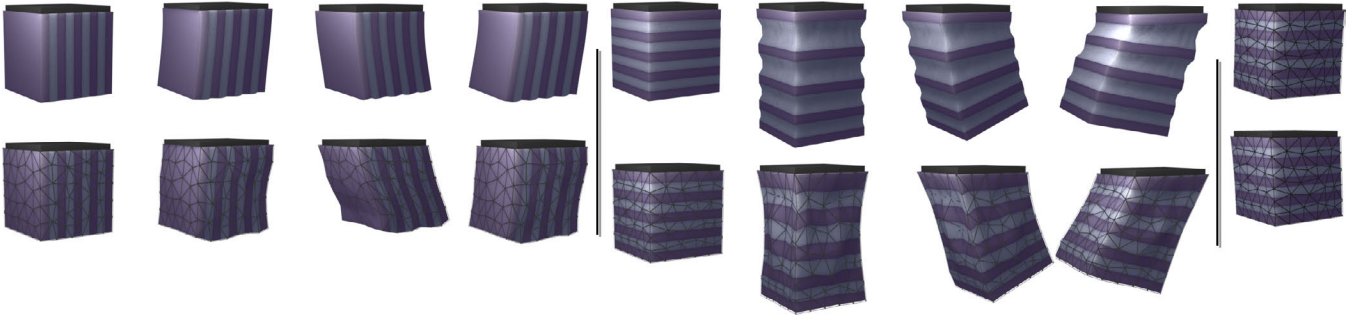
### 4.2 Boundary Treatment

As briefly mentioned earlier, we treat coarse nodes that are outside the fine domain through barycentric extrapolation, *i.e.*, when a field needs to be evaluated on this coarse node, we rely on the values of a few closest fine boundary nodes (between one and three in our implementation, depending on the local curvature of the object) to extrapolate the field based on the positions of the respective positions at rest. Additionally, we found it beneficial to alter the local definition of the normal  $\mathbf{n}$  used in the Neumann condition of Eq. (6) to become the normal  $\mathbf{N}$  of the *coarse bounding tet* instead. This change is more in line with the variational interpretation described in Section 3.7, as the test function near the boundary should be reverted to the coarse element basis function, hence improving boundary coarsening. Note that *only* the external boundary of the object should be subjected to traction: holes inside the domain must be left without traction to be treated as such when computing harmonic displacements.

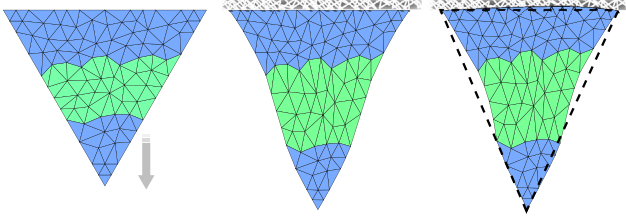
### 4.3 Coarse-to-Fine Mapping for Display

The tensors used for numerical coarsening can also be reused to deduce fine-mesh vertex positions deduced from coarse deformations. This coarse-to-fine interpolation is a quasi-static approximation, as it assumes equilibrium inside each coarse tetrahedron, so it is devoid of higher temporal frequencies of the fine mesh. However, it can be an effective way to reuse some of the information gathered about the anisotropy of the object being simulated.

For each coarse tet  $T_q = \{\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3, \mathbf{v}^4\}$  with undeformed vertex po-



**Figure 5: Physical Accuracy:** On the inhomogeneous layered cube used in Fig 3, a fine simulation (top, left and middle) is well captured by our coarsening approach (bottom, left and middle), despite the anisotropy of the object; if, however, the material coefficients (right, showing the most extreme positions reached during the motion) or the stiffness matrices of the original object are simply averaged, coarse simulations do not match the fine behavior.



**Figure 6: Fine Mesh Display:** A 2-material composite object (left) is subjected to gravity with its top vertices fixed, resulting in an elongated deformation (middle). From a coarse mesh deformation made of a single triangle (right, dashed), we can reconstruct a quasi-static fine solution (right) using precomputed harmonic displacements: this cheap linear map from coarse to fine deformation enhances visual impact at low cost.

sitions  $\{\mathbf{x}_0^0, \mathbf{x}_0^1, \mathbf{x}_0^2, \mathbf{x}_0^3\}$  and current deformed positions  $\{\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\}$  we solve for a rotation matrix  $\mathbf{R}$  and a symmetric matrix  $\mathbf{S}$  such that

$$\begin{pmatrix} \mathbf{R}(\mathbf{x}^1 - \mathbf{x}^0) \\ \mathbf{R}(\mathbf{x}^2 - \mathbf{x}^0) \\ \mathbf{R}(\mathbf{x}^3 - \mathbf{x}^0) \end{pmatrix} = \begin{pmatrix} \mathbf{x}_0^1 - \mathbf{x}_0^0 \\ \mathbf{x}_0^2 - \mathbf{x}_0^0 \\ \mathbf{x}_0^3 - \mathbf{x}_0^0 \end{pmatrix} + \begin{pmatrix} (\mathbf{H}^1 - \mathbf{H}^0) : \mathbf{S} \\ (\mathbf{H}^2 - \mathbf{H}^0) : \mathbf{S} \\ (\mathbf{H}^3 - \mathbf{H}^0) : \mathbf{S} \end{pmatrix} \quad (9)$$

where  $\mathbf{H}^i$  is the component of  $\mathbf{H}$  corresponding to the vertex  $\mathbf{x}^i$ . The interpolated current deformed position for any fine vertex  $\mathbf{x}$  inside this tet can then be computed as

$$\mathbf{x} = \mathbf{x}^0 + \mathbf{R}^T(\mathbf{x}_0 - \mathbf{x}_0^0) + (\mathbf{H}^x - \mathbf{H}^0) : \mathbf{S} \quad (10)$$

where  $\mathbf{H}^x$  is the component of  $\mathbf{H}$  for  $\mathbf{x}$ . This amounts to finding a rotation and linear combination of the harmonic displacements which matches the coarse tet vertices exactly, and then using this same rotation and linear combination to place the fine vertices. Blending of the displacements across adjacent coarse tets can also be added to avoid derivative discontinuities. As this interpolation relies on linear elasticity, it may not be appropriate for large coarse deformations, and a robust solution based on projection onto the shape space spanned by harmonic displacements is left as future work. While Figure 6 shows how this coarse-to-fine map behaves on a simple example, all other figures use barycentric interpolation.

#### 4.4 Animating Coarse Models

We hereafter assume that the reader is familiar with the finite element implementation of linear elasticity, where displacements are stored in a vector  $\mathbf{u}_k$  for each time step  $t_k$  and a stiffness matrix  $\mathbf{K}$  is assembled so that the potential energy becomes  $\mathbf{u}_k^T \mathbf{K} \mathbf{u}_k / 2$ . We will make no distinction between coarse and fine meshes, as the time integration of elasticity applies to both equivalently. We will also limit our discussion to the topics for which our implementation differs from usual practices.

**Corotational Method for Potential Evaluation** Corotational methods all share the same intent: they try to render the linear strain tensor (Eq. (2)) more accurate for large deformation by removing as much of the *current local rotation* as possible. For each tetrahedral element a corotated reference frame is chosen, “with respect

to which the relative displacements [...] due to the current deformations are minimum in some global sense” [de Veubeke 1976]. Instead of expressing the position of the deformed body always in the same coordinate frame (by adding the current displacements  $\mathbf{u}$  to the rest state  $\mathbf{x}_0$ ), a corotational approach defines a matrix  $\mathbf{R}^e$  per element  $e$  (expressing a rotation around the current barycenter  $\mathbf{b}^e$  of the element) such that the current deformed position  $\varphi^e$  of the node elements satisfies:

$$\mathbf{R}^e \varphi^e = \mathbf{x}_0^e + \mathbf{u}^e - \mathbf{b}^e,$$

where  $\mathbf{R}^e$  removes the purely rotational part of the current deformation. This amounts to defining a “corotated displacement”  $\widehat{\mathbf{u}}^e$  per element as:

$$\widehat{\mathbf{u}}^e = \mathbf{R}^{eT}(\mathbf{x}_0^e + \mathbf{u}^e - \mathbf{b}^e) - (\mathbf{x}_0 - \mathbf{b}_0^e),$$

where  $\mathbf{b}_0$  denotes the barycenter of the element in the rest configuration. The potential  $W = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u}$  is then replaced by  $\frac{1}{2} \widehat{\mathbf{u}}^T \widehat{\mathbf{K}} \widehat{\mathbf{u}}$ , thus turning into a sum over all elements now function of both the displacement field  $\mathbf{u}$  and the rotation field  $\mathbf{R} = \{\mathbf{R}^1, \dots, \mathbf{R}^T\}$ . The force field in the elastic body thus becomes:

$$\nabla W = \frac{1}{2} \sum_e \left[ \mathbf{R}^e \mathbf{K}^e \mathbf{R}^{eT} (\mathbf{x}_0^e + \mathbf{u}^e - \mathbf{b}^e) - 2 \mathbf{R}^e \mathbf{K}^e (\mathbf{x}_0^e - \mathbf{b}_0^e) \right].$$

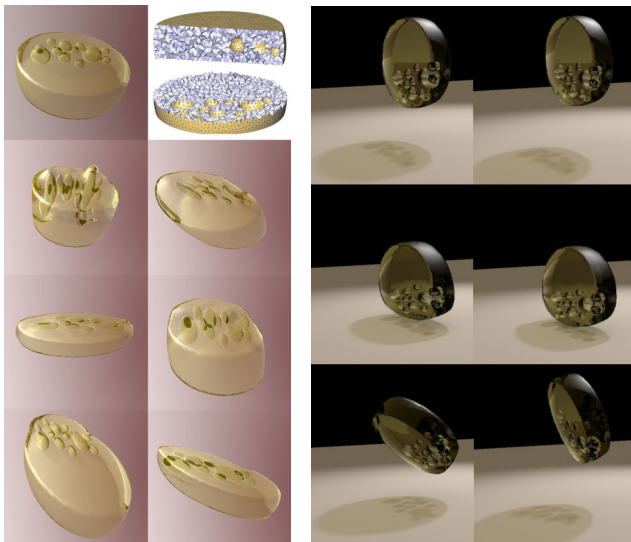
Notice that the material velocity  $\dot{\varphi}$  is still equal to  $\dot{\mathbf{u}}$ : the corotational treatment presented above *only* affects the potential energy computations (and thus, the internal forces).

While corotational methods define various procedures to derive the rotation matrix  $\mathbf{R}^e$  of each element [Müller et al. 2002; Hauth and Straßer 2004; Nesme et al. 2006], we follow the treatment of [de Veubeke 1976]. That is, we pick the corotational frame of each element by minimizing  $|\widehat{\mathbf{u}}^e|^2$ : our tests show much improved behavior compared to QR or polar decomposition, as reported in [Georgii and Westermann 2008]. We followed the implementation of this latter reference to solve the small non-linear system for each tetrahedral element.

**Time Integration** Once the rotation matrix for each element is solved for, we use a midpoint integration scheme for the time update of our elasticity models. While this integrator is symplectic, the corotational treatment affects the numerics and energy is not as accurately preserved as with non-corotational simulations; however, this scheme was used in all our examples, without noticeable visual artifacts. A truly variational integrator for corotational methods would be a valuable tool in the future.

## 5 Results

In order to demonstrate the efficacy of numerical coarsening, we tested our approach on models of varying size, shape, and material composition. A first sanity check was to test that a homogenous object is coarsened into the same material—for coarse tets entirely inside the model. We then tried a layered object (Figures 2(right), 3(bottom), and 5) made out of two distinct materials. As expected,



Fine mesh and cross sections (35K tets, top), along with its harmonic deformations.

Fine (left) and Coarse (right) animations. The coarse mesh has only 200 tets.

**Figure 7: Cheese:** Elastic properties of a wheel of cheese with holes of various sizes in half of the wheel are turned into anisotropic elastic properties on a coarse mesh (200 tets). Animating the coarse object (right) takes only a fraction ( $\sim 1/150$ ) of the cost it would take to compute the elastic behavior on the fine mesh. Notice that the side containing the holes behaves softer, even though the coarse mesh does not spatially capture these cavities.

we witness an “accordion” effect when the object is deformed perpendicular to its layers, while lateral deformation are much less pronounced. This example is simple yet anisotropic enough to convincingly prove that other forms of coarsening (average of stiffness matrices, or of material coefficients) are just not enough to capture the proper dynamics on the coarse mesh.

We also tested more subtle geometric details that can significantly affect the dynamics. In particular, Figure 4 shows that a fine, but deep crack is properly taken into account, resulting in a coarsened motion exhibiting much larger deformation due to the local “weakening” of the material. In Figure 7, we coarsen a cheese wheel model, with half of the wheel containing gruyere-like holes. As the harmonic deformations clearly exhibit, the coarsened material properties are significantly affected by the inhomogeneity of the model. We also compare the motion of the coarsened model (200 tets) to the much finer original tetrahedral mesh (35K tets needed in order to represent the holes) of the wheel, indicating good visual agreement while reducing the computational complexity: the coarse animation runs 150 times faster than the fine one. Finally, we applied our numerical coarsening technique to a medical model, consisting of a liver and its two interior veins (portal vein, and inferior vena cava; see Figure 1). The veins act as reinforcement, rendering the liver stiffer. We started from a MRI dataset made out of 200K tets tagged as either belonging to the liver, or one of the two veins. After assigning material properties to these three components, we numerically coarsened this model to obtain an anisotropic coarse material made out of 210 vertices behaving dynamically similar to the original model.

**Limitations.** It should be reemphasized that our coarsening is currently limited to linear elasticity. The use of corotational methods injects geometric nonlinearity to coarse simulations, thus limiting the visual drawbacks of linear elasticity. However, the same use of corotated elements for fine meshes will add non-linear details that low tet-count meshes will be unable to match, even after proper coarsening. An extension to non-linear coarsening is thus desirable.

## 6 Conclusions

In this paper, we proposed a formulation of elasticity on coarse resolutions, where the influence of known fine scales is modeled through the derivation of effective elasticity tensors. The effective tensors are obtained by solving inhomogeneous Laplace equations with Neumann boundary conditions, leading to non-trivial, symmetry-respecting averaging of the material properties.

Directions of future work are manifold. As we mentioned earlier, an extension for which coarsening is performed through a series of *local* computations over each coarse tetrahedron would be highly desirable: this would allow for fast updates of fine structures during tearing for example. Geometric approaches to reconstruct fine deformations from the coarse ones would be valuable complements to further improve visual impact at low cost. Using non-diagonal mass matrices on the coarse mesh would also most likely bring benefits, albeit at a nonnegligible computational cost. We could also restrict our coarse meshes to be logically rectangular to simplify computations. Finally, medical applications such as virtual surgery on complex inhomogeneous organs are being further explored.

**Acknowledgments.** The authors wish to thank Yiyong Tong for his comments and Keenan Crane for rendering assistance. This research was partially funded by the NSF (CCF-0811373, DMS-0453145, and CMMI-0757106), the DOE (DE-FG02-04ER25657), and Pixar Animation Studios.

## References

- AN, S., KIM, T., AND JAMES, D. L. 2008. Optimizing cubature for efficient integration of subspace deformations. *ACM Transactions on Graphics (SIGGRAPH Asia)* 27, 4 (Dec.).
- BABUŠKA, I., AND SAUTER, S. A. 2008. Efficient solution of anisotropic lattice equations by the recovery method. *SIAM J. Sci. Comput.* 30, 5, 2386–2404.
- BARAFF, D., AND WITKIN, A. P. 1998. Large steps in cloth simulation. In *ACM SIGGRAPH Proceedings*, 43–54.
- BARBIČ, J., AND JAMES, D. 2005. Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models. *ACM Transactions on Graphics* 24, 3 (Aug.), 982–990.
- BARR, A. H. 1989. The Einstein Summation Notation: Introduction and Extensions. ACM SIGGRAPH Course Notes #30 “Topics in Physically-based Modeling”, J1–J12.
- BENSOUSSAN, A., LIONS, J. L., AND PAPANICOLAOU, G. 1978. *Asymptotic analysis for periodic structure*. North Holland, Amsterdam.
- BERLYAND, L., AND OWHADI, H. 2009. Finite dimensional approximation of solutions of divergence form systems of equations with rough and high contrast coefficients. *To appear*.
- CAPELL, S., GREEN, S., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2002. Interactive skeleton-driven dynamic deformations. *ACM SIGGRAPH* 21, 3 (July), 586–593.
- CAPELL, S., GREEN, S., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2002. A multiresolution framework for dynamic deformations. In *Symposium on Computer Animation*, 41–48.
- CHOI, M.-G., AND KO, H.-S. 2005. Modal warping: Real-time simulation of large rotational deformation. *IEEE Trans. on Visualization and Computer Graphics* 11, 1, 91–101.
- DE VEUBEKE, B. F. 1976. The dynamics of flexible bodies. *International Journal of Engineering Science* 14, 895–913.

DEBUNNE, G., DESBRUN, M., CANI, M.-P., AND BARR, A. H. 2001. Dynamic real-time deformations using space & time adaptive sampling. In *ACM SIGGRAPH Proceedings*, 31–36.

FARMER, C. L. 2002. Upscaling: a review. *International Journal for Numerical Methods in Fluids* 40, 1-2, 63–78.

FEYNMAN, R., LEIGHTON, R., AND SANDS, M. 2006. *The Feynman Lectures on Physics*. Addison-Wesley.

GEORGII, J., AND WESTERMANN, R. 2008. Corotated finite elements made fast and stable. In *Proceedings of the 5th Workshop on Virtual Reality Interaction and Physical Simulation*.

GRINSPUN, E., KRYSL, P., AND SCHRÖDER, P. 2002. Charms: A simple framework for adaptive simulation. *ACM Transactions on Graphics* 21, 3 (July), 281–290.

HAUTH, M., AND STRASSER, W. 2004. Corotational simulation of deformable solids. In *Winter School on Computer Graphics*, 137–144.

JAMES, D. L., AND FATAHALIAN, K. 2003. Precomputing interactive dynamic deformable scenes. *ACM Transactions on Graphics* 22, 3 (July), 879–887.

JAMES, D. L., AND PAI, D. K. 1999. ARTDEFO: Accurate real time deformable objects. In *ACM SIGGRAPH Proceedings*, 65–72.

JAMES, D. L., AND PAI, D. K. 2002. DyRT: Dynamic response textures for real time deformation simulation with graphics hardware. *ACM Trans. on Graphics* 21, 3 (July), 582–585.

JKOV, V. V., KOZLOV, S. M., AND OLEINIK, O. A. 1991. *Homogenization of Differential Operators and Integral Functionals*. Springer-Verlag.

KRYSL, P., LALL, S., AND MARSDEN, J. 2000. Dimensional Model Reduction in Non-linear Finite Element Dynamics of Solids and Structures. *I.J.N.M.E.* 51, 479–504.

LI, R.-C., AND BAI, Z. 2005. Structure preserving model reduction using a Krylov subspace projection formulation. *Comm. Math. Sci.* 3, 2, 179–199.

MÜLLER, M., AND GROSS, M. 2004. Interactive virtual materials. In *Proceedings of Graphics Interface*, 239–246.

MÜLLER, M., DORSEY, J., MCMILLAN, L., JAGNOW, R., AND CUTLER, B. 2002. Stable real-time deformations. In *Proceedings of the Symposium on Computer Animation*, 49–54.

NESME, M., PAYAN, Y., AND FAURE, F. 2006. Animating shapes at arbitrary resolution with non-uniform stiffness. In *Eurographics Workshop in Virtual Reality Interaction and Physical Simulation (VRIPHYS)*.

NESME, M., KRY, P. G., JEŘÁBKOVÁ, L., AND FAURE, F. 2009. Preserving topology and elasticity for embedded deformable models. *ACM Trans. on Graphics* 28, 3 (Aug.).

OWHADI, H., AND ZHANG, L. 2007. Metric-based upscaling. *Communications on Pure and Applied Math.* 60, 675–723.

PENTLAND, A., AND WILLIAMS, J. 1989. Good vibrations: modal dynamics for graphics and animation. In *ACM SIGGRAPH Proceedings*, 215–222.

RIVERS, A. R., AND JAMES, D. L. 2007. Fastlsm: Fast lattice shape matching for robust real-time deformation. *ACM Trans. on Graphics* 26, 3 (July), 82:1–82:6.

SHU, S., BABUŠKA, I., XIAO, Y., XU, J., AND ZIKATANOV, L. 2008. Multilevel preconditioning methods for discrete models of lattice block materials. *SIAM Journal on Scientific Computing* 31, 1, 687–707.

TREUILLE, A., LEWIS, A., AND POPOVIĆ, Z. 2006. Model reduction for real-time fluids. *ACM Trans. on Graphics* 25, 3 (July), 826–834.

WOJTAN, C., AND TURK, G. 2008. Fast viscoelastic behavior with thin features. *ACM Trans. on Graphics* 27(3), 47 (Aug.).

## A Applying Coarsening in 1D

We briefly demonstrate how our numerical coarsening applied in 1D leads to the known stiffness equivalence rule for springs in series. (Remember that a spring’s stiffness  $k$  is equal to the spring’s elastic compliance  $C$  divided by its rest length.) For simplicity, consider just two springs, one with stiffness  $k_1$  between  $x_0$  and  $x_1$ , and the other with stiffness  $k_2$  between  $x_1$  and  $x_2$  (see below).



The harmonic displacement  $h$ , for which we set  $h(x_0) = h_0 = 0$  to fix the gauge of the system, must satisfy Eq. (6), i.e.,

$$\begin{cases} k_1 h_1 = k_2 [h_2 - h_1] & (\text{balance of forces at } x_1) \\ k_2 [h_2 - h_1] = k_2 (x_2 - x_1) & (\text{Neumann condition at } x_2). \end{cases}$$

We deduce that:  $h_1 = k_2(x_2 - x_1)/k_1$  and  $h_2 = (x_2 - x_1)(k_1 + k_2)/k_1$ . We can then compute the mollifier per element:

$$G_{[x_0, x_1]} = \frac{h_1}{x_1 - x_0} = \frac{k_2}{k_1} \frac{x_2 - x_1}{x_1 - x_0} \quad G_{[x_1, x_2]} = \frac{h_2 - h_1}{x_2 - x_1} = 1.$$

If we now wish to replace the two springs by a single one, we get:

$$\mathbf{G}_{[x_0, x_2]} = \frac{h_2}{x_2 - x_0} = \frac{k_1 + k_2}{k_1} \frac{x_2 - x_1}{x_2 - x_0}.$$

Using Eq. (8), the equivalent stiffness  $k$  satisfies:

$$k(x_2 - x_0) = \mathbf{G}_{[x_0, x_2]}^{-1} \left[ k_1(x_1 - x_0) \mathbf{G}_{[x_0, x_1]}^2 \frac{x_1 - x_0}{x_2 - x_0} + k_2(x_2 - x_1) \mathbf{G}_{[x_1, x_2]}^2 \frac{x_2 - x_1}{x_2 - x_0} \right] \mathbf{G}_{[x_0, x_2]}^{-1}.$$

Therefore we get the traditional stiffness equivalence between two springs in series:

$$\frac{1}{k} = \frac{1}{k_1} + \frac{1}{k_2}.$$

## B Tensor-Matrix Conversion

Tensors  $\mathbf{T} = T_{abcd}$  of rank-4 in 3D (i.e.,  $1 \leq a, b, c, d \leq 3$ ) contain 81 components, and are thus traditionally converted into a  $9 \times 9$  matrix  $M$  through, assuming zero-based indexing:

$$T_{abcd} = M[3(a-1) + b, 3(c-1) + d]. \quad (11)$$

When  $\mathbf{T}$  has minor symmetries ( $T_{abcd} = T_{bacd}$  and  $T_{abcd} = T_{abdc}$ , like all the tensors used in this paper), one can further reduce the size of the representation by introducing a  $6 \times 6$  matrix  $N$  such that:

$$N = RMR^T \quad \text{with } R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

where  $R$  is called the *reduction matrix*. Converting back from this reduced matrix space to a full tensor is easily achieved as well: from a  $6 \times 6$  matrix  $N$ , the equivalent  $9 \times 9$  matrix  $M$  is found through:

$$M = E^T N E \quad \text{with } E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

where  $E$  is called the *expansion matrix*, and the final conversion to a tensor uses Eq. (11).