Part I: Mathematical Formulation

by

Chetan Jhurani and L. Demkowicz
Part I: Mathematical Formulation

Chetan Jhurani       L. Demkowicz

Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Abstract

This paper is the first in a series to develop a numerical homogenization method for heterogeneous media and integrate it with goal-oriented finite element mesh adaptivity. Here we describe the physical application (Step and Flash Imprint Lithography) in brief and present the mathematical ideas behind the homogenization method. Numerical verification and detailed results will be shown in the second paper. The method requires Moore-Penrose pseudoinverse of element stiffness matrices. Algorithms for efficiently computing the pseudoinverse of sparse matrices will be the topic of the third one.

The purpose of homogenization is to reduce the number of degrees of freedom, find locally optimal effective material properties, and do goal-oriented mesh refinement. Traditionally, a finite element mesh is designed after obtaining material properties in different regions. The mesh has to resolve material discontinuities and rapid variations. In our approach, however, we generate a sequence of coarse meshes (possibly 1-irregular), and homogenize material properties on each coarse mesh element using a locally posed constrained convex quadratic optimization problem. This upscaling is done using Moore-Penrose pseudoinverse of the linearized fine-scale element stiffness matrices, and a material-independent interpolation operator. This requires solution of a continuous-time Lyapunov equation on each element. Using the adjoint solution, we compute local error estimates in the quantity of interest. The error estimates also drive the automatic mesh adaptivity algorithm. The results show that this method uses orders of magnitude fewer degrees of freedom to give fast and approximate solutions of the original fine-scale problem.

Contents

1 Introduction ................................................. 2
  1.1 Multiscale methods and numerical homogenization .......... 3
  1.2 Mesh adaptivity and error estimation ...................... 3
  1.3 Overview of this work ................................... 4
1 Introduction

Step and Flash Imprint Lithography (SFIL) is an imprint lithography process designed to transfer circuit patterns to fabricate microchips in low-pressure and room-temperature environments [6, 7, 17]. Photopolymerization is the main process to create polymeric patterns on a substrate. It is accompanied by densification which affects the shape of imprinted features [18].

In this research, we are interested in the post-polymerization step of the SFIL process. The object of interest is a heterogeneous glassy polymeric structure created on an organic
polymer layer which in turn is on a silicon substrate. The structure is modeled as monomers interacting with pair-potentials with neighbors in a lattice. An equilibrium configuration is found by minimizing the energy of the lattice. Numerical solution of such a molecular statics fine-scale base model is computationally very expensive due to the problem size, which is on the order of millions of degrees of freedom (DOFs). Rapid variation in material properties, ill-conditioning, nonlinearity, and non-convexity make this problem even more challenging to solve.

Typical dimensions of the patterns in the structure are much larger than individual molecules, but the discreteness plays an important role in modeling of such objects. In the context of SFIL, an approach for coupling of discrete polymer elasticity models with continuum hyperelasticity models has been presented in [10, 11].

Our objective is to approximate a nonlinear base model of the polymeric structure (based on molecular statics) by local numerical homogenization of fine-scale material properties and use goal-oriented adaptivity to change the models spatially. Before presenting the local numerical homogenization method, we give a brief review and background of multiscale methods and mesh adaptivity.

1.1 Multiscale methods and numerical homogenization

The desire to capture fundamental or more accurate small-scale models into large-scale models has given rise to the field of multiscale methods [26]. Such methods find the approximate solution on a coarse mesh but use the fine grid to construct the relevant information. The numerical techniques allow us to avoid the assumption of periodicity that is typically used in the analytical techniques of homogenization.

Hughes et al. use concepts of variational multiscale and residual-free bubbles to resolve the fine-scales [15, 31]. Engquist et al. use wavelet basis to compute effective homogenized operators and use truncation for a sparse approximation [25, 27]. Hou et al. compute operator-dependent basis functions by solving local auxiliary problems [30]. For two-phase flow in porous media, a numerical upscaling technique based on the assumption that net flux between coarse elements occurs only on the coarse-scale has been introduced by Arbogast [4]. In [34], Knapek introduced operator-dependent interpolation in the context of multigrid methods.

1.2 Mesh adaptivity and error estimation

Critical to the accuracy, reliability and mesh adaptivity in finite element methods is existence of good a posteriori error estimates. Such estimates not only provide confidence in the solution on the current mesh but also indicate elements to be refined further for an automatic
refinement strategy. Use of effective automatic refinement algorithms is essential to obtain an accurate solution of problems in complex domains.

Many researchers have contributed to the vast field of a posteriori error estimation and mesh refinement. For a comprehensive introduction and analysis of various methods, we refer to the monographs by Babuška and Strouboulis [5] and by Oden and Ainsworth [1].

For many applications, interest is restricted to part of the full domain or a goal represented by a functional of the solution. Usually, in the context of linear problems, the goal is a bounded linear functional on the Hilbert space containing the solution. Many algorithms have been developed for optimizing the mesh for reducing the error in a given quantity of interest rather than in some energy norm. Such algorithms provide the basis for the so-called goal-oriented adaptivity. The main tool behind such algorithms is characterization of the error in the goal in terms of the solution of the adjoint problem (which is driven by the goal). Amongst others, this approach was taken by Becker and Rannacher [12] and Oden and Prudhomme [38].

1.3 Overview of this work

In this work, we develop and implement a framework for numerical homogenization and goal-oriented adaptivity for nonlinear lattice elasticity problems. The method is developed with the polymer base model of SFIL in mind, but is quite general and can be applied to continuum problems with a given fine mesh that sufficiently resolves the fine-scale material properties.

The main research contribution lies in mathematical development and efficient software implementation of local numerical homogenization. The application area (SFIL) mathematical details are presented in this paper. Numerical verification and detailed results will be shown in the second paper. Algorithms for efficiently computing Moore-Penrose pseudoinverse of sparse matrices, which form the core of homogenization, will be the topic of the third one.

We describe the SFIL process and its modeling in Section 2. In Section 3, we present the local numerical homogenization method. A coarse mesh is selected first and homogenization is done on each individual element of the mesh. On each element, the homogenization method works with the Moore-Penrose pseudoinverse of the element stiffness matrix to produce pseudoinverse of the local homogenized stiffness matrix as the output. The homogenized stiffness matrices can be assembled in the usual manner and the resulting system solved to compute the coarse-scale solution. The work extends the existing goal-oriented $h$-refinement strategy [22] to numerical homogenization where coarse-scale and fine-scale operators are different. The adjoint solutions on coarse and fine meshes provide a basis of automatic goal-oriented adaptivity. This gives rise to 1-irregular meshes with hanging nodes. These are handled using the constrained approximation techniques [20, 21].
2 Description and Modeling of Step and Flash Imprint Lithography

Step and Flash Imprint Lithography (SFIL) is a viable low-cost alternative to existing lithography techniques. This technique was initially developed by the Willson Research Group at The University of Texas at Austin in the late 1990s [17]. It is designed for fabricating microchips in low-pressure and room-temperature environments [6, 7, 17]. It has enabled imprinting of features smaller than 20 nanometers (nm). Moreover, it has the inherent resolution necessary to define sub-10 nm geometries [39]. Roughly speaking, a template contains “negative” of the desired pattern. If liquid were to be trapped inside these negative features and polymerized, on removal of the template, we would obtain a polymer with the “positive” pattern. Figure 1 shows the basic idea behind the process and the resulting geometry. This section describes the process and the model of elasticity of polymeric lattices created in SFIL. Parameter estimation for bond potentials is discussed in Appendix A. The lattice elasticity model is described in detail in [10, 11].

2.1 Description of the SFIL process

![Figure 1: (a) A diagram of the SFIL process (not to scale) and (b) resulting patterns of size 40nm as seen using a scanning electron microscope.](image)

Multiple separate processes have to be carried out to complete the pattern transfer. To create a pattern layer, an organic polymer layer (transfer layer) is spin-coated on a silicon substrate. A low viscosity, photopolymerizable, organosilicon solution (etch barrier) is then distributed on the wafer. A transparent template, which has patterned relief structures, is placed over the coated silicon substrate. This displaces the etch barrier solution which gets...
trapped in the pattern. Once the pieces are in place, irradiation with UV light through the backside of the template cures the etch barrier into a cross-linked polymer film. A fluorocarbon release layer on the template allows separation from the substrate, leaving an organosilicon relief image that is a replica of the template pattern. A halogen etch is used to break through the undisplaced etch barrier material (residual layer) exposing the underlying transfer layer [7].

2.2 Modeling polymerization and densification in SFIL

We briefly describe the existing model of formation of polymeric lattices in SFIL. The details have been published elsewhere [10, 11, 35]. Free radical polymerization of the etch barrier solution results in a glassy polymer structure with a shape dictated by the patterns on the template. A mathematical polymerization model provides the topology of the polymer chains that form the etch barrier.

The polymers are formed by Radical Polymerization. A kinetic Monte Carlo algorithm simulates the chemical reactions that take place in the etch barrier solution [16, 35]. The inputs to the Monte Carlo algorithm are various reaction rates in form of probabilities and the ratio of individual constituents [23, 24]. Its output is a lattice-like stochastic cuboidal topology of monomers, cross-linker, initiator, substrate, and template molecules interacting with each other by central pair potentials. Empty lattice sites, also called voids, are introduced to facilitate motion of molecules. There is good agreement in the statistics of degree of polymerization between the simulations and experimentally observed quantities [33].

Polymerization is accompanied by densification due to the change in interaction potential between photopolymer precursors from Van der Waals to covalent. Typical change in the feature volume is around 9% [16, 18]. Densification affects the shape of the resulting structure. It is a very slow process compared to the reaction. Hence, polymerization is modeled separately from the subsequent densification.

2.3 Molecular statics base model and equilibrium

The result of the Monte Carlo algorithm is just a topology of molecules connected with different bonds. The molecules are treated as point masses in this model. We still do not know an equilibrium position of this network. Because of changes in bonds and the equilibrium lengths, pre-strain is built-in to the problem. The equilibrium configuration with Dirichlet boundary condition at the base (substrate) is found by letting the molecules relax to an equilibrium configuration [37].

The molecules are connected to nearest neighbors along edges and faces of the cubical lattice (Figure 2). Each molecule, unless it is on the boundary of the topology, is connected
to 18 other molecules (assuming no “void” molecules as neighbors). The face diagonal bonds simulate volume exclusion and provide shear stiffness to the lattice. All of them are modeled by Lennard-Jones potential. Edge bonds can be both covalent and non-covalent (Van der Waals bonds). The Monte Carlo step provides this information. Covalent bonds are modeled by a stiff harmonic potential.

Consider a cubical lattice with $N_1, N_2,$ and $N_3$ molecules in $x, y,$ and $z$ directions respectively. Let $\mathcal{A}$ be the set of all molecules, and

$$\mathcal{A} := \{ \{ x_{ijk} \}_{i=0}^{N_1-1} \}_{j=0}^{N_2-1} \}_{k=0}^{N_3-1}$$

be the set of all degrees of freedom (DOFs, coordinates of all the molecules). Similarly, $\mathcal{f}$ is the set of point forces on all molecules and $\mathcal{D}$ is the set of the initial guesses of the DOFs. Let $\mathcal{D}$ be the set of particles and directions with Dirichlet boundary conditions.

$$\mathcal{D} := \{ (i, j, k, p) : x_{ijk}^p = \bar{x}_{ijk}^p \text{ is fixed in direction } p \}$$

We can define the total potential energy $J$ for this lattice as a function of $\mathcal{x}$.

$$J(\mathcal{x}) := \frac{1}{2} \sum_{x_{ijk} \in \mathcal{x}} \sum_{x_{lmn} \in \mathcal{x}} E \left( \left| \left| x_{lmn} - x_{ijk} \right| \right| ; (i, j, k), (l, m, n) \right) - \sum_{x_{ijk} \in \mathcal{x}} \sum_{p=1}^{3} f_{ij}^p x_{ijk}^p \quad (1)$$

The symbol $\leftrightarrow$ means “is connected to”. Here $E$ denotes a central bond potential function that depends on the distance between any two given molecules and the bond parameters. For a lattice with covalent bonds forming the polymer chains, the bond parameters depend on the location of the molecules in the lattice. For the sake of a simple expression for $J$, this dependence of bond parameters is hidden in the expression for $E$ by making $E$ a function that also depends on the lattice indices $(i, j, k)$ and $(l, m, n)$. The factor of half takes care of double counting of each bond in the summation over neighbors.
The problem is to find an equilibrium position by minimizing the energy.

\[
\text{Minimize } J(\mathbf{x}) : x^{ijk}_p = \bar{x}^{ijk}_p \quad \forall (i, j, k, p) \in \mathcal{D}.
\]

We use a Newton trust-region method implemented in TAO/PETSc \cite{8, 13} to reach a local minimum starting from a given initial guess \( \bar{x} \). This requires derivatives of \( J \) up to second order. They are evaluated analytically.

### 2.4 Numerical solution of the base model

By the “base model” we mean the molecular statics model that uses the exact information about the lattice topology. Numerical solution of such a molecular statics base model, which is assumed to describe the microstructure completely, is computationally very expensive. This is due to a large problem size, on the order of millions of DOFs. Rapid variation in material properties, ill-conditioning, nonlinearity, and existence of multiple minima make this problem even more challenging to solve.

![Figure 3: Part of the etch barrier modeled as a lattice of size 21 × 101 × 21 in (a) pre-strained state and (b) equilibrium with fixed bottom layer. The colors correspond to different constituent molecules. The solution was computed by Bauman \cite{10}.](image)

Figure 3 shows a typical solution geometry visualized in VMD \cite{32} with the pre-strained fine-scale lattice and the lattice in equilibrium. The biggest problem run so far had 3 millions DOFs (for a 100-cubed lattice). The computation of equilibrium took 370 CPU hours (divided amongst 64 processors) and 25000 Newton iterations \cite{10}.

### 3 Local Numerical Homogenization

We describe a method for local numerical homogenization of nonlinear lattice elasticity. There is no assumption of periodicity of material properties or boundary conditions. The
method will be applied to the problem of molecular statics encountered in SFIL (Section 2). However, the method is general and applicable for problems with a given fine mesh that sufficiently resolves the fine-scale material properties.

Traditionally, a finite element mesh is designed after obtaining material properties in different regions. The mesh has to match material discontinuities. In our approach we do exactly the opposite. A coarse mesh is selected first and homogenization is done on each individual element of the mesh. In this way, the coarse mesh and the fine-scale structure become naturally compatible. On each element, the homogenization method works with the Moore-Penrose pseudoinverse of the element stiffness matrix to produce pseudoinverse of the local homogenized stiffness matrix as the output. This output depends on the local load as well as a chosen local interpolation from the coarse to the fine mesh. This process requires solving a continuous-time Lyapunov equation on each element. The pseudoinverse of the homogenized stiffness matrix is then pseudo-inverted to get the homogenized stiffness matrix for the assembly phase.

3.1 Using interpolation for dimensional reduction

We use a simple 1-D example of discrete linear springs with varying spring constants. Using this simple setting, we show that using a material-independent interpolation operator results in coarse-scale material properties that can be arbitrarily wrong.

\[
\begin{align*}
\begin{array}{cccccccc}
  & k_1 & k_i & k_{i+1} & k_N & F \\
 x_0 & x_1 & x_{i-1} & x_i & x_{i+1} & x_N
\end{array}
\end{align*}
\]

Figure 4: \(N\) springs with a fixed leftmost particle and a force \(F\) on the rightmost particle.

As shown in Figure 4, we have \(N\) harmonic springs, \(N \geq 2\), with identical equilibrium lengths and variable stiffnesses \(\{k_i\}_{i=1}^N\). Assume the left end-point is fixed at origin and that a force \(F\) is applied on the right end-point. It is readily seen that the tension in each spring is \(F\) and hence the displacement of the right end-point is \(F/k\), where the effective stiffness \(k\) is the harmonic mean of \(\{k_i\}_{i=1}^N\) times \(1/N\).

\[
\frac{1}{k} = \sum_{i=1}^{N} \frac{1}{k_i}.
\]

(2)

For this problem, the displacements \(\{u_i\}_{i=0}^N =: u\) can also be found by minimizing the total potential energy \(J\).

\[
\text{Minimize } J(u) = \frac{1}{2} \sum_{i=1}^{N} k_i (u_i - u_{i-1})^2 - Fu_N \quad \text{such that } u_0 = 0.
\]

(3)
We can minimize $J$ in a subspace. Let us consider all the springs to form a single element and assume a linear constraint $u_i = (i/N)u_N$, where $u_N$ is the only unknown. This interpolation scheme is independent of the specific values of $k_i$. The total potential energy in terms of a single unknown $u_N$ is

$$\frac{u_N^2}{2N^2} \sum_{i=1}^{N} k_i - F u_N.$$ 

The total energy is minimum when

$$u_N = \frac{N^2 F}{\sum_{i=1}^{N} k_i}.$$ 

This shows that the effective stiffness, given by $F/u_N$, is the arithmetic mean of $\{k_i\}_{i=1}^{N}$ times $1/N$.

$$k = \frac{1}{N^2} \sum_{i=1}^{N} k_i. \quad (4)$$

For positive quantities, the harmonic mean is always less than the arithmetic mean unless all the quantities are equal. This can be proved using induction on $N$. Hence, in 1-D, if we minimize in a subspace using the typical finite element shape functions, we always overestimate the effective stiffness. The relative error due to this approximation is not bounded. For example, take $N = 2$ and $k_1 = 1$. In this case, the exact effective stiffness is always less than 1 but the arithmetic mean stiffness can be made larger than any arbitrary value by increasing $k_2$.

Although this 1-D problem is simple, its analysis clearly shows the pitfalls associated with homogenization. Problems in higher dimensions are more complex and there may not be a simple effective property like in 1-D. In fact, in the case presented above, the effective material property was just one of the many possible ones. The one chosen was to match approximate end-point displacement with the exact one. It does not tell us what happens as we move away from the end-point. Thus, in a general case, homogenized properties will also depend on volumetric forces and types of boundary conditions and their distribution. In addition, the choice of a coarse mesh plays an important role for the numerical homogenization.

### 3.2 Local numerical homogenization

We change our focus from the best energy norm solution in a simple subspace. Our focus is on getting locally best effective material properties from the linearized operator of the nonlinear lattice elasticity problem of SFIL (2). For an iterative numerical scheme, the material properties are the element stiffness matrices that depend on the current coarse mesh and the current solution guess.
We begin by introducing a coarse grid compatible with the lattice, as illustrated in Figure 5. In each coarse element, the DOFs are displacements of the corners. We find the effective stiffness for each element (Sections 3.3 and 3.4) in which every coarse DOF interacts with each other. The local effective stiffness matrices are then assembled to form the global stiffness matrix and the problem is solved using the standard Finite Element technology. The DOFs corresponding to the hanging nodes are eliminated before the assembly stage.

![Figure 5: A 2-D lattice with fixed bottom layer is approximated with a coarse mesh. Effective local stiffness is computed on coarse elements and used to create a global effective lattice. The diagonal bonds in the fine lattice are not shown for clarity.](image)

Consider a lattice with a load provided by the built-in pre-strain, the known external loads, and unknown reactions. Without solving the fine-scale problem exactly we cannot know the forces on the masses forming a detached coarse element. However, given the location of the masses, we can compute the pre-strain in each element. To homogenize locally, we can use the pre-strain as a particular load for which we want accurate homogenization. We may also choose to homogenize without considering any particular load. We will formulate the homogenization method using both choices (separately). For the moment, we work with a known non-zero self-equilibrated load denoted by $f, f \in \mathbb{R}^N$. In Section 3.4 we consider the case $f = 0$ which can be understood as homogenization for all loads.

Let $K$ be a symmetric element stiffness matrix of size $N$. The local equation for $u \in \mathbb{R}^N$ is

$$Ku = f.$$
The exact solution is given by

\[ u = K^\dagger f + u_0 \]

where \( K^\dagger \) is the Moore-Penrose pseudoinverse [14] of \( K \) and \( u_0 \) is an arbitrary vector in the null space \( \mathcal{N}(K) \) of \( K \).

We mark the coarse-scale quantities with a hat. For \( M \leq N \), let \( \hat{K} \in \mathbb{R}^{M \times M} \), as yet unknown, represent the effective element stiffness for the coarse element DOFs. Let \( \hat{f} \in \mathbb{R}^M \) be the coarse-scale load. \( A \in \mathbb{R}^{N \times M} \) is a chosen interpolation operator. It is local to the element. Vector \( \hat{f} = A^T f \) is the natural interpolation of the fine-scale load \( f \).

In terms of the unknown \( \hat{K} \), the coarse-scale solution is

\[ \hat{u} = \hat{K}^\dagger A^T f + \hat{u}_0 \]

where \( \hat{K}^\dagger \) is the Moore-Penrose pseudoinverse of \( \hat{K} \) and \( \hat{u}_0 \in \mathcal{N}(\hat{K}) \). When compared to the fine-scale solution, the error is

\[ u - A\hat{u} = (K^\dagger - A\hat{K}^\dagger A^T) f + (u_0 - A\hat{u}_0). \]

Hence, up to an arbitrary constant, the error \( e \in \mathbb{R}^N \) is

\[ e := (K^\dagger - A\hat{K}^\dagger A^T) f. \]

Let \( B \) be a symmetric positive-definite matrix of size \( N \). We will use \( B \) to measure the error \( e \) and in process define a homogenization problem to compute \( K^\dagger \).

\( B \) should not approximate energy norm. Suppose \( K \) is positive definite, and \( B \) is chosen to be \( K \). Then we are minimizing the error in the energy norm in a round-about way, and as we have seen this results in an incorrect homogenization when the interpolation operator \( A \) is a material independent.

Section 3.3 defines the local homogenization problem for a given self-equilibrated load. If the load is not known, or is zero, we modify the local homogenization procedure to compute best effective properties for arbitrary loads. This takes care of forces on an element due to other elements. Homogenization for all loads (by not using any given load) is explained in Section 3.4.

### 3.3 Locally best effective properties for a given load

We now define and solve the local homogenization problem for a given non-zero and self-equilibrated load \( f \).
3.3.1 Definition of local homogenization

Given $K \in \mathbb{R}^{N \times N}$, $K$ symmetric, $A \in \mathbb{R}^{N \times M}$, $B \in \mathbb{S}_+^N$, $f \in \mathbb{R}^N - \{0\}$, and $\epsilon > 0$, the local homogenization problem is to find a symmetric $\hat{K}^\dagger$ that minimizes $\mathcal{E}$, where

$$\mathcal{E}(\hat{K}^\dagger) := \frac{1}{2} \left\| (K^\dagger - A\hat{K}^\dagger A^T) f \right\|_B^2 + \frac{\epsilon}{2} \left\| K^\dagger - A\hat{K}^\dagger A^T \right\|_{F,B}^2 \left\| f \right\|_2^2. \quad (5)$$

The first term in the sum is $\frac{1}{2} \left\| e \right\|_B^2 = \frac{1}{2} e^T B e$ and measures the error in the solution. Symbol “$F, B$” in second term stands for Frobenius matrix norm weighted with the matrix $B$. For $X \in \mathbb{R}^{N \times N}$,

$$\left\| X \right\|_{F,B}^2 := \text{trace}(X^TBX).$$

‘trace’ of a matrix is the sum of its diagonal elements. If $B$ is the identity matrix, then $\left\| \cdot \right\|_{F,B}$ is the standard Frobenius matrix norm.

In the limit $\epsilon \to 0$, this minimization problem can be interpreted as a method to define a local effective stiffness $\hat{K}$ that provides the best solution for arbitrary loads but subject to the constraint that for a particular load $f$ the error is the smallest possible.

We choose $\epsilon$, a dimensionless parameter, a couple of magnitudes smaller than 1. The second term is a regularization term and is included to obtain a unique $K^\dagger$. We require regularization for two reasons.

- Firstly, the overall solution method is iterative due to mesh refinements and Newton iteration steps. Hence $f$ used for homogenization in a single step represents only an approximation to an “exact” $f$.

- Secondly, without regularization, the minimization problem given by Equation (5) is ill-posed (with infinitely many solutions). This happens because the action of operators $K$ and $\hat{K}$ is used only for a single load $f$. Thus regularization is necessary even if an “exact” $f$ is known.

The “closure assumption” in this homogenization method is that local (to an element) effective material properties are not affected by other elements [4].

3.3.2 Solution of the local homogenization problem

Equation (5) is a typical minimization problem with linear constraints and a convex quadratic objective function in the entries of $\hat{K}^\dagger$. Note that the objective function is not quadratic in the entries of $\hat{K}$. In fact, because of the perturbation properties of the Moore-Penrose
pseudoinverse, it is a highly nonlinear and discontinuous function in the entries of $\hat{K}$ \cite{14}. We will solve for $\hat{K}^\dagger$ and obtain $\hat{K}$ by computing its pseudoinverse. We will see later in this section that $\hat{K}^\dagger$ solves a continuous-time Lyapunov equation.

Symmetry of $\hat{K}$ implies (and is implied by) symmetry of $\hat{K}^\dagger$. We use a Lagrange multiplier matrix $\Lambda$ to impose the symmetry condition. To keep the notation simple we let the unknown $\hat{K}^\dagger$ be called $X$ in the rest of this section. The Lagrangian is

$$L(X, \Lambda) = E(X) + \text{trace}(\Lambda^T (X - X^T)).$$

(6)

Stationarity of $L(X, \Lambda)$ implies $X = X^T$ and

$$(A^T BA)X (A^T (ff^T + \epsilon \|f\|^2 I)A) - A^T BK^\dagger (ff^T + \epsilon \|f\|^2 I)A = \Lambda^T - \Lambda.$$  

(7)

$I$ is the identity matrix of size compatible with $f$. The details of the derivation of this equation are in Appendix B.

We define some intermediate matrices for simplification.

$$U := A^T BA$$  

(8)

$$V := A^T (ff^T + \epsilon \|f\|^2 I)A$$  

(9)

$$W := A^T BK^\dagger (ff^T + \epsilon \|f\|^2 I)A$$  

(10)

Matrices $U$ and $V$ are symmetric. Using the definitions of $U, V,$ and $W$, we can write Equation (7) as

$$UXV - W = \Lambda^T - \Lambda.$$  

Since $\Lambda^T - \Lambda$ is skew-symmetric, so is $UXV - W$. Hence, we can eliminate $\Lambda$.

$$UXV - W + (UXV - W)^T = 0.$$  

(11)

Assume $\epsilon > 0$. We then use the Sherman-Morrison-Woodbury formula \cite{40} to express $V^{-1}$.

$$V^{-1} = \frac{1}{\epsilon \|f\|^2} \left( (A^T A)^{-1} - \frac{(A^T A)^{-1} A^T f f^T A (A^T A)^{-1}}{\epsilon \|f\|^2 + f^T A (A^T A)^{-1} A^T f} \right).$$  

(12)

This exists if the following three conditions are true. Term $\epsilon \|f\|^2$ is positive, $(A^T A)^{-1}$ exists, and the denominator of the second term is non-zero. The second condition holds because columns of $A$ are linearly independent. The first and third conditions are true because $\epsilon \|f\|^2 > 0$ and $A(A^T A)^{-1} A^T$ is positive semi-definite. In addition, $V$ is positive-definite since it is invertible and is sum of a positive definite matrix and a rank-1 positive semi-definite matrix.

Using $X = X^T$, invertibility of $V$, and symmetry of $U$ and $V$, Equation (11) can be
simplified to

\[(V^{-1}U)X + X(V^{-1}U)^T = V^{-1}(W + W^T)V^{-1}.\]

We define two more intermediate matrices.

\[
C := V^{-1}U \quad (13)
\]

\[
D := V^{-1}(W + W^T)V^{-1} \quad (14)
\]

Hence, \(X\) solves

\[CX + XC^T = D. \quad (15)\]

This is a continuous-time Lyapunov equation, a linear matrix equation with \(X\) as the unknown matrix. \(D\) is symmetric. If the sum of any two eigenvalues of \(C\) is non-zero (which is satisfied if \(C\) is positive definite), then a unique solution \(X\) exists and it is symmetric [3, 41].

### 3.4 Locally best effective properties for all loads

We now define and solve the local homogenization problem when the load \(f\) is not given, or is 0, or we want to compute effective properties independent of any self-equilibrated load.

Given \(K \in \mathbb{R}^{N \times N}, \) \(K\) symmetric, \(A \in \mathbb{R}^{N \times M}, \) and \(B \in \mathbb{S}^N_{++}, \) the local homogenization problem is to find a symmetric \(\hat{K}^{\dagger}\) that minimizes \(\mathcal{E},\) where

\[
\mathcal{E}(\hat{K}^{\dagger}) := \frac{1}{2} \left\| K^{\dagger} - A\hat{K}^{\dagger}A^T \right\|_{F,B}^2. \quad (16)
\]

The notation is same as used in Equation (5). We can proceed to form a Lagrangian and minimize \(\mathcal{E}(\hat{K}^{\dagger})\) for a symmetric \(\hat{K}^{\dagger}\) as explained in Appendix B. The only difference is that compared to the definitions of the intermediate matrices in Section 3.3.2, the definitions of \(V\) and \(W\) are modified. \(U\) remains the same.

\[
U := A^TBA, \quad (17)
\]

\[
V := A^TA, \quad (18)
\]

\[
W := A^TBA. \quad (19)
\]

Matrices \(C\) and \(D\) are defined as they were defined in Equations (13)–(14) but in terms of these \(U, V,\) and \(W.\) We still have to solve a Lyapunov equation (Equation (15)).

#### 3.4.1 A special case: minimum error in \(\ell^2\) norm

If we want minimum error in the \(\ell^2\) norm and thus choose \(B = I,\) it turns out that we don’t have to solve a Lyapunov equation because \(C\) becomes identity. Note that in this case,
$U = V$, and $W = W^T$. Thus $\hat{K} = X = V^{-1}WV^{-1}$.

\[
\hat{K} = ((A^T A)^{-1} A^T K A (A^T A)^{-1})^\dagger
= (A^\dagger K^\dagger (A^\dagger)^T)^\dagger
\tag{20}
\]

It can be shown that if we derive the effective stiffness matrix for least error in energy norm as done in Section 3.1, then we get $\hat{K} = A^T K A$. These two expressions can be arbitrarily different in general.

It is readily seen that if the fine-scale stiffness matrix $K$ is positive semi-definite, then so is the coarse-scale stiffness matrix $\hat{K}$. This can be shown as follows. If $K$ is positive semi-definite, then so is $K^\dagger$. This implies that $x^T K^\dagger x \geq 0$ for any compatible vector $x$, in particular for $u = (A^\dagger)^T y$. Thus, $((A^\dagger)^T y)^T K^\dagger ((A^\dagger)^T y) \geq 0$. Rearranging, we get $y^T (A^\dagger K^\dagger (A^\dagger)^T) y \geq 0$. Thus, $A^\dagger K^\dagger (A^\dagger)^T$ is positive semi-definite. This implies that its pseudoinverse $\hat{K} = (A^\dagger K^\dagger (A^\dagger)^T)^\dagger$ is positive semi-definite too.

### 3.5 Constrained convex optimization for local homogenization

In Section 3.3, the local problem to homogenize a single element was posed using a regularization approach. We created a single error function, which was a weighted sum of the error (due to homogenization) for a specific non-zero load $f$ and a norm of the difference of fine-scale compliance matrix ($K^\dagger$) and interpolated compliance matrix ($A\hat{K}^\dagger A^T$). This required choosing a small weighing parameter $\epsilon > 0$. The other constraint, of symmetry of $X$, was imposed exactly using Lagrange multipliers.

In this section we formulate local homogenization as a (fully) constrained convex optimization problem that does not require the approximation due to regularization.

There are two sets of constraints that the entries of the unknown matrix $X = \hat{K}^\dagger$ must satisfy exactly. Firstly, $X$ must be symmetric. Secondly, for a specific non-zero load $f$ the error due to homogenization should be the minimum possible. The constrained optimization formulation is derived using a two-step procedure. In the first step, the constraint equations are derived. In the second step, while satisfying the constraints of first step, $X$ must minimize a norm of the difference of fine-scale compliance matrix ($K^\dagger$) and interpolated compliance matrix ($A\hat{K}^\dagger A^T$).

The constraint equation for minimizing the error for a specific load $f$ can be derived from the first-order necessary optimality condition of the following problem.

\[
\min_{X=X^T} \frac{1}{2} \| (K^\dagger - AXA^T) f \|_B^2
\]

Using a Lagrange multiplier approach to impose $X = X^T$ as shown in Section 3.3 and
Appendix B we get the optimality condition.

$$A^T B A X A^T f f^T A - A^T B K^\dagger f f^T A$$ must be skew-symmetric.

Thus,

$$(A^T B A X A^T f - A^T B K^\dagger f)(A^T f)^T$$ must be skew-symmetric.

It can be shown that if the outer product of two vectors is skew-symmetric, then at least one of the two vectors is zero. The proof is in Appendix C. Since $A$ is full-rank and $f$ is assumed non-zero, it implies that $A^T f$ is non-zero. Thus the first vector, $A^T B A X A^T f - A^T B K^\dagger f$, is 0. Obviously this equation cannot be used to fully determine $X$ since it fixes the action of $X$ only on a single vector, $A^T f$.

This leads us to the second stage of the optimization procedure.

$$\min_X \frac{1}{2} \left\| K^\dagger - A X A^T \right\|^2_{F,B}$$ such that $A^T B A X A^T f = A^T B K^\dagger f$ and $X = X^T$ (21)

We define auxiliary matrices $U, V, W,$ and $\tilde{f}$, where $U = A^T B A, V = A^T A, W = A^T B K^\dagger,$ and $\tilde{f} = A^T f$. Using Lagrange multipliers to impose the two constraints, we can deduce the following first-order optimality condition.

$$\begin{bmatrix}
U \otimes V + V \otimes U & U \otimes \tilde{f} + \tilde{f} \otimes U \\
U \otimes \tilde{f}^T + \tilde{f}^T \otimes U & 0
\end{bmatrix}
\begin{bmatrix}
\vec(X) \\
\mu
\end{bmatrix}
= 
\begin{bmatrix}
\vec(WA + A^T W^T) \\
2 W f
\end{bmatrix}
$$

Here $\otimes$ is the Kronecker product operation, “vec” is the vectorization operation and $\mu$ is an unknown vector of Lagrange multipliers that constrain $X$ to be the best possible for the specific load $f$. The Lagrange multipliers for imposing the symmetry of $X$ have been eliminated. The $(1, 2)$ block is the transpose of $(2, 1)$ block. Since the $(1, 1)$ block $(U \otimes V + V \otimes U)$ is symmetric positive definite and the $(2, 1)$ block $(U \otimes \tilde{f}^T + \tilde{f}^T \otimes U)$ has full row-rank, the system is invertible and characterizes the unique minimum $X$. The Kronecker products are not computed. One should transform the system to a Lyapunov equation like structure for fast solution. This is slightly more complicated than the regularization approach of Section 3.3. Hence, we will present all the numerical results in subsequent papers using the simpler approach. However, this derivation and the results show that a unique solution can be found for “exact” local homogenization.

3.6 Homogenization of pre-stress

In case of polymer lattices of SFIL, even if there is no external force on molecules, the lattice is not in equilibrium in general. This is because of the residual forces due to a mismatch in inter-molecular distance and the length at which the inter-molecular potential is minimum. Using the terminology from continuum elasticity, we call the residual forces pre-stress. If
a single element is removed from the lattice, as shown in Figure 5, the pre-stress in that element is self-equilibrated. This is because if a spring leads to a force $F$ on a particle, then it leads to a force $-F$ on the corresponding neighbor. Taking the sum and moment around any point, the total force as well as the total moment is zero. This pre-stress leads to a pre-strain.

To homogenize correctly, we need to transfer the pre-stress (or pre-strain) information in the fine-scale lattice to coarse-scale mesh. This is done after determining the coarse-scale ‘material properties’ ($\hat{K}$). Let $p$ and $u$ denote the fine-scale pre-stress and pre-strain, respectively. The corresponding coarse-scale quantities are $\hat{p}$ and $\hat{u}$. Figure 6 shows the transfer operation.

![Figure 6: The homogenized pre-stress is the coarse-scale pre-stress that would give a deformation such that the fine-scale and coarse-scale lattices are approximately similar.](image)

For a single iteration, we have

$$Ku = p \implies u = K^\dagger p$$

and

$$\hat{K}\hat{u} = \hat{p} \implies \hat{u} = \hat{K}^\dagger \hat{p}.$$  

We find $\hat{u}$ (and hence $\hat{p}$ too) by minimizing the error

$$||A\hat{u} - u||_B^2,$$

where $A$ is the interpolation operator and $B$ is a symmetric positive definite matrix that forms the norm in which we measure the error. We will use $B = I$ here. Since $A$ has full column-rank, the solution is simply $\hat{u} = A^\dagger u$. Using the expressions for $u$ and $\hat{u}$, the coarse-scale pre-stress is

$$\hat{p} = \hat{K}A^\dagger K^\dagger p.$$  \hspace{1cm} (22)
3.7 Remarks on local homogenization

The interpolation operator $A$ is a parameter to play with and it can be chosen to get better homogenized properties. Operator-dependent interpolation [30] should be a better choice but this is a speculation in context of the current approach. We use the simple bilinear interpolant for lattice based problems in 2-D (and trilinear in 3-D). We can also make the error $\mathcal{E}$ a function of $A$ as well and look for the best $A$. This turns out to be a complicated nonlinear problem with many more unknowns.

We have a choice of the matrix norm in the regularization term. Spectral norm may be better but its use leads to a nonlinear problem with non-smooth objective function. In contrast, weighted Frobenius norm leads to a convex quadratic problem with linear equality constraints (due to the imposed symmetry of $\hat{K}^\dagger$).

Load-independent homogenization (Section 3.4) may not be correct for problems involving layered media. Boundary loads and local material distribution influence the homogenized properties [2, p. 24]. For example, in conductivity problems for layered media, if the flux is in parallel to the layers, the effective conductivity is given by the arithmetic average. However, if the flux is in the perpendicular direction, the effective conductivity is given by the harmonic average [29, p. 13]. A local self-equilibrated load can be computed by projecting the global known load.

The choice of $B$ is critical. If we choose $B = K$ ($K$ should be positive definite for this) then we don’t gain anything when compared to the minimization in a subspace. The approximate solution of Equation (5) leads to the arithmetic average rather than the harmonic average. Since our interpolation operator is independent of the material properties, and we are not resolving the variations, we should look for a “weaker” convergence. Here the theory of homogenization for PDEs guides us to believe that $B$ should be independent of the operator and correspond to a discrete approximation of the $L^2$ norm.

These alternatives, load dependent or independent homogenization and possible values of the matrix $B$, are numerically explored and justified in the next paper in the series.

3.8 An analytical example of local homogenization

The local homogenization method of Section 3.3.1 is exemplified with homogenization of 3 harmonic springs forming a single element of a 1-D lattice.

We homogenize the springs to create a single effective spring with spring constant $\hat{k}$ (Figure 7). To keep the analysis simple, we first choose the matrix $B$ (used in Equation (5)) to be identity. Later on we use a different $B$ to reduce errors where we want. We recover the classical effective spring constant given by the harmonic average.
The solution is $u$

We will determine the best $\hat{\epsilon}$

We use standard linear interpolation for the inner points. The interpolation operator is $\epsilon$

The fine-scale system is described by displacements $u \in \mathbb{R}^4$, stiffness matrix $K \in \mathbb{R}^{4 \times 4}$, and load $f \in \mathbb{R}^4$ with $f_4 = -f_1 - f_2 - f_3$.

$$Ku = \begin{bmatrix}
  k_1 & -k_1 & 0 & 0 \\
  -k_1 & k_1 + k_2 & -k_2 & 0 \\
  0 & -k_2 & k_2 + k_3 & -k_3 \\
  0 & 0 & -k_3 & k_3
\end{bmatrix}
\begin{bmatrix}
  u_1 \\
  u_2 \\
  u_3 \\
  u_4
\end{bmatrix}
= \begin{bmatrix}
  f_1 \\
  f_2 \\
  f_3 \\
  f_4
\end{bmatrix}
= f$$

The solution is $u = K^Tf + (u_0, u_0, u_0, u_0)^T$ where $u_0$ is arbitrary and

$$K^T = \frac{1}{16}
\begin{bmatrix}
  \frac{9}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} \\
  -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & \frac{9}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} \\
  -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & \frac{9}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} \\
  -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & -\frac{3}{k_1} + \frac{4}{k_2} + \frac{1}{k_3} & \frac{9}{k_1} + \frac{4}{k_2} + \frac{1}{k_3}
\end{bmatrix}.$$

We use standard linear interpolation for the inner points. The interpolation operator is $A = \begin{bmatrix} 1 & \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix}^T$.

For the homogenized element with a single spring, the effective stiffness matrix is

$$\hat{K} = \begin{bmatrix} \hat{k} & -\hat{k} \\ -\hat{k} & \hat{k} \end{bmatrix}.$$

Its pseudoinverse is

$$\hat{K}^+ = \frac{1}{4}
\begin{bmatrix}
  \frac{1}{k} & -\frac{1}{k} \\
  -\frac{1}{k} & \frac{1}{k}
\end{bmatrix}.$$

We will determine the best $\hat{k}$ by minimizing the error $E$ (Equation (5)). We choose $B = I$.

For an arbitrary load $f$ and $\epsilon > 0$, we can analytically solve for the minimum. In the limit $\epsilon \to 0$ we get

$$\hat{k} = \frac{10}{9}
\frac{(3f_1 + 2f_2 + 3f_3)k_1k_2k_3}{3k_1k_2(f_1 + f_2 + f_3) + 3k_2k_3(f_1) + 4k_3k_1(f_1 + f_2)}.$$
To simplify the expression, let us use loads only at the end-points. Thus, $f_1 = 1, f_2 = 0,$ and $f_3 = 0$. Then

$$\hat{k} = \frac{10 k_1 k_2 k_3}{9 k_1 k_2 + 12 k_1 k_3 + 9 k_2 k_3}.$$ 

Compare $\hat{k}$ with the classical effective spring constant

$$k = \frac{k_1 k_2 k_3}{k_1 k_2 + k_1 k_3 + k_2 k_3}.$$

The expressions for $\hat{k}$ and $k$ are not the same. However, we do get a value very close to the one that is given by the harmonic average. Since we chose $B = I$, the errors at the end-points were given equal emphasis as the errors at interior points. In derivation of the classical effective constant, we are only concerned with matching end-point displacements. This explains the discrepancy between the two effective spring constants.

This discrepancy can be resolved by using $B = \text{diag}(M, 1, 1, M)$, where $M \gg 1$, so that errors at end-points are penalized more. The best effective stiffness can be computed to be

$$\hat{k} = \frac{k_1 k_2 k_3}{(1 - m)k_1 k_2 + (1 + 2m)k_1 k_3 + (1 - m)k_2 k_3}$$

where $m := \frac{1}{9M + 1} \to 0$ as $M \to \infty$. Hence, we recover the classical effective stiffness by choosing an appropriate norm to measure the local error.

### 3.9 Computational aspects

Computing the homogenized material properties requires computation of Moore-Penrose pseudoinverse of multiple matrices with different structural properties. We discuss this topic briefly here in Section 3.9.1 and in detail in the third paper in this series. In addition, we have to solve a dense Lyapunov equation of a small size (number of DOFs in the coarse-scale element). This is discussed in Section 3.9.2 below.

#### 3.9.1 Computation of Moore-Penrose pseudoinverse

The pseudoinverse of a real matrix $K$ can be defined and computed in terms of the SVD [14, 28]. Let $K$ have an SVD $K = U \Sigma V^T$, then the pseudoinverse is $K^\dagger = V \Sigma^\dagger U^T$ where $\Sigma^\dagger$ is formed by taking the reciprocal of all non-zero singular values (diagonal entries in $\Sigma$) and applying a transpose operation.

SVD is the most general and reliable procedure for computing pseudoinverses. However,
it is an $O(N^3)$ procedure. It also ignores the sparsity of the matrix. For these reasons, it is not the best method in the context of numerical homogenization. We will present and analyze faster algorithms for computing the pseudoinverse.

- We need pseudoinverses of the large and sparse fine-scale stiffness matrix $K$. Actually, only the action of the pseudoinverse is needed on $f$ and columns of $A$ (Equations (10) and (19)).
- In case of Equation (20), we need the action of $K^\dagger$ on columns of $(AT)^\dagger$.
- In case of Equation (20), we need the pseudoinverse of a dense $A$.
- Whether we homogenize for a particular load or not, we have to compute the homogenized stiffness matrix from its pseudoinverse. Thus we need to explicitly compute pseudoinverse of a symmetric matrix of a relatively small fixed size (since it is on the coarse-scale). For example, for a 3-D lattice elasticity problem with a cube forming the coarse-scale, the size of effective stiffness matrices is $24 \times 24$. In 2-D the size is $8 \times 8$.

For a rectangular matrix of full column rank, for example the interpolation matrix $A$, the pseudoinverse can be expressed using the normal equations.

$$A^\dagger = (A^TA)^{-1}A^T$$

Since the number of columns of $A$ is fixed and small, we can either use the normal equations for computing $A^\dagger$ or preferably use $QR$ factorization of $A$.

### 3.9.2 Solution of the Lyapunov equation

Computing the pseudoinverse of the best local stiffness matrix requires solving a Lyapunov equation (except for the special case mentioned above for load-independent homogenization and when $B$ is identity). This equation arises in many different areas, for example control theory and stability analysis. Many algorithms and implementations exist for its efficient solution [3, 36]. We use the Bartels-Stewart algorithm [9] that uses lower real Schur decomposition of the matrix $C$. Let $C = QC'Q^T$, $D = QD'Q^T$, and $X = QX'Q^T$, where $Q \in \mathbb{R}^{M \times M}$ is a real orthonormal matrix, $C'$ is real block lower triangular with maximum block size 2. With these definitions, Equation (15) becomes

$$C'X' + X'C'^T = D'.$$

Since $C'$ has a special structure, the transformed equation can be solved using a “forward-substitution” algorithm. We successively solve for $X'_{11}, X'_{12}, \ldots, X'_{1m}, X'_{21}, X'_{22}$ and so on where $m$ is the number of blocks. Finally, we can compute $X = QX'Q^T$.  


3.10 Error estimation for goal-oriented adaptivity

Many times, we are interested in only a linear functional of the full solution instead of the full solution itself [22]. For example, in case of the lattice, we may be interested in the average deformation of an edge. In such a case, local error estimates to refine the mesh should reflect the error in the goal (the linear functional) and not the energy. In the usual analysis, the fine-scale and coarse-scale operators come from the same underlying physical model and are directly related using the interpolation operator. When we locally homogenize to get a different coarse-scale model, we have to derive error estimates for goal-oriented adaptivity again.

We work with abstract variational forms in this section. Let the solution be in \( u_D + V \), where \( u_D \) is a particular solution satisfying the given Dirichlet boundary condition, and \( V \) is a Hilbert space with elements satisfying homogeneous boundary condition. Let \( X \) be the Hilbert space on which the problem is posed (\( V \subset X \)).

Let the goal \( G : X \rightarrow \mathbb{R} \) be a linear and continuous functional of the solution. Let \( B : X \times X \rightarrow \mathbb{R} \) be a continuous, coercive bilinear form and \( L : X \rightarrow \mathbb{R} \) be a continuous linear form. We keep track of the interpolation operator \( A : \hat{X} \rightarrow X \) also.

The Galerkin orthogonality condition is slightly different if the fine-scale and coarse-scale operators are different. This is the case if the coarse-scale operator is derived by local homogenization. As before, quantities relevant on the coarse-scale are marked with a hat. We assume that the coarse-scale is chosen in such a way that the Dirichlet boundary conditions are satisfied exactly.

The fine-scale primal problem is

\[
\text{Find } u \in u_D + V \text{ such that } B(u, v) = L(v) \quad \forall \ v \in V.
\]

The coarse-scale primal problem is

\[
\text{Find } \hat{u} \in u_D + \hat{V} \text{ such that } \hat{B}(\hat{u}, \hat{v}) = \hat{L}(\hat{v}) \quad \forall \ \hat{v} \in \hat{V}
\]

where the coarse-scale load is defined by

\[
\hat{L}(\hat{v}) = L(A\hat{v}) \quad \forall \ \hat{v} \in \hat{X}.
\]
Corresponding to a goal $G \in \mathcal{X}'$, we have the fine-scale and coarse-scale adjoint problems. The fine-scale adjoint problem is

\[
\text{Find } w \in \mathcal{V} \text{ such that } \mathcal{B}(v, w) = G(v) \quad \forall \, v \in \mathcal{V}.
\]

The coarse-scale adjoint problem is

\[
\text{Find } \hat{w} \in \hat{\mathcal{V}} \text{ such that } \hat{\mathcal{B}}(\hat{v}, \hat{w}) = \hat{G}(\hat{v}) \quad \forall \, \hat{v} \in \hat{\mathcal{V}}
\]

where the goal on the coarse-scale is defined by

\[
\hat{G}(\hat{v}) = G(A\hat{v}) \quad \forall \, \hat{v} \in \hat{\mathcal{X}}.
\]

The error in the goal is

\[
G(u) - \hat{G}(\hat{u}) = G(u) - G(A\hat{u}) = G(u - A\hat{u}) = \mathcal{B}(u - A\hat{u}, w) = \mathcal{B}(u - A\hat{u}, w - A\hat{w}) + \mathcal{B}(u - A\hat{u}, A\hat{w}) = \mathcal{B}(u - A\hat{u}, w - A\hat{w}) + \mathcal{B}(u, A\hat{w}) - \mathcal{B}(A\hat{u}, A\hat{w}).
\]

In case of compatible fine-scale and coarse-scale bilinear forms, the last two terms would have canceled each other. This is not true for our case. Instead, we prove that $\mathcal{B}(u, A\hat{w}) = \hat{\mathcal{B}}(\hat{u}, \hat{w})$ and simplify the error expression.

\[
\begin{align*}
\mathcal{B}(u, v) &= \mathcal{L}(v) \quad \forall \, v \in \mathcal{V} \\
\Rightarrow \mathcal{B}(u, A\hat{v}) &= \mathcal{L}(A\hat{v}) \quad \forall \, \hat{v} \in \hat{\mathcal{V}} \\
\Rightarrow \mathcal{B}(u, A\hat{v}) &= \hat{\mathcal{L}}(\hat{v}) \quad \forall \, \hat{v} \in \hat{\mathcal{V}} \\
\Rightarrow \mathcal{B}(u, \hat{w}) &= \hat{\mathcal{B}}(\hat{u}, \hat{v}) \quad \forall \, \hat{v} \in \hat{\mathcal{V}} \\
\Rightarrow \mathcal{B}(u, \hat{w}) &= \hat{\mathcal{B}}(\hat{u}, \hat{w})
\end{align*}
\]

Finally, the error in the goal, $G(u) - \hat{G}(\hat{u})$, is

\[
\frac{\mathcal{B}(u - A\hat{u}, w - A\hat{w})}{\text{Standard characterization}} + \frac{\hat{\mathcal{B}}(\hat{u}, \hat{w}) - \mathcal{B}(A\hat{u}, A\hat{w})}{\text{Incompatible bilinear forms}}.
\]

We can split both the terms into quantities defined on individual elements. As before, the expression above uses the exact primal and adjoint solutions. To approximate the error estimates, we work with a current coarse mesh and a uniformly refined finer mesh as an approximation of the fine-scale mesh. These local error estimates provide indicators for a further mesh refinement.
3.11 Integration of homogenization, mesh adaptivity, and Newton iterations for nonlinear problems

We summarize the basic structure for homogenizing and solving the problem on a coarse mesh. Figure 9 depicts this structure in a flowchart.

![Flowchart](image)

Figure 9: Overall structure of integrating homogenization, mesh adaptivity, and Newton iterations for nonlinearity.

We create a compatible coarse mesh from the fine lattice data. Here compatibility means that all fine lattice Dirichlet boundary conditions are reproducible by the coarse mesh Dirichlet boundary conditions. After making an initial guess for the displacements, we compute the gradient of energy on the fine lattice and restrict (using the transpose of the interpolation operator) it to the coarse lattice. We continue if the norm of the gradient is large, otherwise we stop and report convergence. The next step is to compute the Hessian on the coarse lattice by homogenizing individual coarse elements. Using the coarse gradient and Hessian we perform the Newton step for coarse displacements. The new displacements are prolonged to the fine lattice, and we iterate for the next step. Once the step size is below a threshold, we use energy-oriented or goal-oriented error estimates and refine elements with
large errors. We repeat the Newton iterations on the refined lattice until convergence. The mesh is refined again if needed.

A Parameter Estimation of Molecular Potentials

We describe an inverse problem approach to determine the bond stiffness parameters of the molecular lattice from stress-strain experiments on poly(methyl methacrylate) (PMMA).

A.1 Experimental stress-strain relationship

PMMA, sold by trade name Plexiglas amongst others, is easily available in any desired form. The main reason why it is used to compute parameters for the SFIL lattice is that the monomer unit is an acrylate except for two methyl groups. Acrylates form the principle components of the compounds used to produce the etch barrier [23], Figure 10.

The PMMA sample is a small dog-bone shaped object with length 1.25 cm and width 0.4 cm (Figure 11). The sample is pulled at both ends by forces such that the strain increases by 0.25% at each step. The maximum strain is 2.25% (Table A.1). The experiments were conducted by Elizabeth Collister from the Willson Research Group [19].

<table>
<thead>
<tr>
<th>Force (kN)</th>
<th>0.35</th>
<th>0.65</th>
<th>1.1</th>
<th>1.35</th>
<th>1.75</th>
<th>1.9</th>
<th>2.2</th>
<th>2.3</th>
<th>2.45</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stress (MPa)</td>
<td>7</td>
<td>13</td>
<td>22</td>
<td>27</td>
<td>35</td>
<td>38</td>
<td>44</td>
<td>46</td>
<td>49</td>
</tr>
<tr>
<td>Axial strain (%)</td>
<td>0.25</td>
<td>0.5</td>
<td>0.75</td>
<td>1</td>
<td>1.25</td>
<td>1.5</td>
<td>1.75</td>
<td>2</td>
<td>2.25</td>
</tr>
</tbody>
</table>

Table 1: Experimental data from the stress-strain experiments on PMMA.

A.2 Formulation of the inverse problem

We determine the spring potential parameters for a cubical lattice by using the data from the experimental stress-strain curve.

We use the superscript $i$ to mark the $i^{th}$ experiment. The forces used are $\{F_i\}_{i=1}^{N_E}$. To keep the notation simple, we use the superscript 0 for the quantities when no force is applied. Thus, $i = 0, 1, 2, \ldots, N_E$ and $F^0 = 0$. 

26
Figure 10: Molecular structure of three constituents of the etch barrier and PMMA. Acrylate groups are marked [23].

Figure 11: PMMA sample used for the stress-strain experiments.

We take a cubical lattice with $N_1$, $N_2$, and $N_3$ molecules in $x$, $y$, and $z$ directions respectively. $A$ is the set of all molecules, and

$$
\mathbf{x} := \left\{ \left\{ x^{lmn}_{i} \right\}_{l=0}^{N_1-1} \right\}_{m=0}^{N_2-1} \right\}_{n=0}^{N_3-1}
$$

is the set of all molecular coordinates. Let $\mathbf{x}^i$ denote the molecular coordinates when force $F^i$ is used on the molecular lattice. We use the vector $\mathbf{p}$ to denote the unknown lattice potential parameters. Thus, $\mathbf{x}^i = \mathbf{x}^i(\mathbf{p})$.

Let $L$ represent the length of the lattice in the direction of the force. $L$ is a linear functional of $\mathbf{x}$ defined using the average distance between the cube surfaces perpendicular
to direction of force. If the force is in $x$ direction,

$$
\mathcal{L}(x) = \frac{1}{N_2 N_3} \sum_{m=0}^{N_2-1} \sum_{n=0}^{N_1-1} \left( x^{(N_1-1)mn} - x^{(0)mn} \right). \tag{23}
$$

The observed quantities are the strains $\epsilon^i$. In terms of the lengths, the observables are

$$
\epsilon^i = \frac{\mathcal{L}(x^i) - \mathcal{L}(x^0)}{\mathcal{L}(x^0)}.
$$

Let $q^i$ be the experimentally observed strains. The misfit as a function of the unknown parameters $p$ is defined as

$$
Q(p) = \frac{1}{2} \sum_{i=1}^{N_E} (\epsilon^i - q^i)^2.
$$

Expanding the expression, we get

$$
Q(p) = \frac{1}{2} \sum_{i=1}^{N_E} \left( \frac{\mathcal{L}(x^i(p)) - \mathcal{L}(x^0(p))}{\mathcal{L}(x^0(p))} - q^i \right)^2.
$$

We want to minimize the misfit as a function of $p$. Looking ahead, to compute $\frac{\partial Q}{\partial p}$, we need $\frac{\partial \mathcal{L}}{\partial x^i}$ and $\frac{\partial x^i}{\partial p}$. We can compute the first quantity from the definition of $\mathcal{L}$ in terms of $x$ (Equation (23)). To compute the second quantity, we need to know the equilibrium equation for the lattice. Let $J = J(x, p)$ be the total potential energy (Equation (1)). At minimum, $\frac{\partial J}{\partial x} = 0 \ \forall p$.

Assuming sufficient regularity, we differentiate this constraint with respect to $p$. Thus,

$$
\frac{\partial^2 J}{\partial x^2} \frac{\partial}{\partial p} + \frac{\partial^2 J}{\partial x \partial p} = 0.
$$

We assume $\frac{\partial^2 J}{\partial x^2}$ is positive-definite, which implies that the equilibrium configuration $x$ is a local minimum. Thus we can solve this linear system of equations for $\frac{\partial x}{\partial p}$. Finally, the gradient of $Q$ can be computed as

$$
\frac{\partial Q}{\partial p} = \sum_{i=1}^{N_E} \left[ (\frac{\partial \mathcal{L}}{\partial x} \left( \frac{\partial^2 J}{\partial x^2} \right)^{-1} \frac{\partial^2 J}{\partial x \partial p} ) \bigg|_{x^0} \mathcal{L}(x^i) - (\frac{\partial \mathcal{L}}{\partial x} \left( \frac{\partial^2 J}{\partial x^2} \right)^{-1} \frac{\partial^2 J}{\partial x \partial p} ) \bigg|_{x^i} \mathcal{L}(x^0) \right].
$$

All the quantities on the right hand side are computable. We use the steepest descent method to get the optimum $p$. 28
A.3 Numerical results

We scaled the sample to match the area of a lattice with 51 cells on each size. We determined two spring constants for a lattice with harmonic springs for the covalent bond and the non-covalent bonds. Hence, $p = (k_1, k_2)$. For this lattice, the optimum values were 1320 N/m for the covalent bond and 1085 N/m for the non-covalent bond. The experimental and lattice stress-strain data-points are shown in Figure 12.

![Figure 12: Experimental stress-strain curve for a PMMA sample and the best stress-strain curve for a lattice with 52 points in each side.](image)

A.4 Remarks on the inverse problem approach

The numerical results show that the bond parameters determined via the inverse problem approach result in a stress-strain relationship that matches well the experimentally determined stress-strain relationship. However, this does not imply that the optimal bond parameters themselves are accurate. This is because the experiments are carried out at length scale of centimeters whereas the size of polymerized material is on a length scale less than micrometers. This is a physics based reason behind the possible inaccuracy. The second reason, a mathematical one, is that the inputs to the inverse problem are averaged forces and resulting strains (over the measurement surface). This makes any inference of the microstructure inherently ill-conditioned.

To clarify using a simple 1-D example, consider two linear springs (of possibly different stiffness) joined in series. It is impossible to determine the two individual stiffness constants by applying forces on the end-points only and measuring the relative displacement of the end-points. One can only determine a quantity that is a function of the harmonic average.
Thus, it would be best to get the values of the bond parameters from an ab initio approach. This is a challenging problem by itself and outside the scope of this work.

We also note that the inverse problem gives deterministic values of the bond parameters. The lattice, however, is stochastic in nature since it is generated by a Monte Carlo simulation of the chemical reactions. Thus, one may want to quantify this uncertainty using a probability distribution function for unknown parameters. This requires multiple solution of the inverse problem for a brute-force determination. We do not consider the stochastic case.

B Local Numerical Homogenization – Stationary Point of the Lagrangian

We derive conditions for stationarity of the Lagrangian, which is defined in Equation (6), for the local homogenization problem (Section 3.3.2).

For $K^{\dagger} \in \mathbb{R}^{N \times N}$, $K^{\dagger}$ symmetric, $A \in \mathbb{R}^{N \times M}$, $X \in \mathbb{R}^{M \times M}$, $B \in \mathbb{S}_{++}^{N}$, $f \in \mathbb{R}^{N} - \{0\}$, $\Lambda \in \mathbb{R}^{M \times M}$, and $\epsilon > 0$ the Lagrangian is

$$
\mathcal{L}(X, \Lambda) = \frac{1}{2} \left( (K^{\dagger} - AXA^{T})f \right)^{2}_{B} + \frac{\epsilon}{2} \left( K^{\dagger} - AXA^{T} \right)^{2}_{F,B} ||f||_{2}^{2}
+ \text{trace}(\Lambda^{T}(X - X^{T}))
$$

\[=
\begin{align*}
\frac{1}{2} f^{T}K^{\dagger}BK^{\dagger}f + \frac{\epsilon}{2} ||f||_{2}^{2} \text{trace} \left( K^{\dagger}BK^{\dagger} \right) \\
+ \text{trace}(\Lambda^{T}(X - X^{T})) - f^{T}K^{\dagger}BAXA^{T}f - \epsilon ||f||_{2}^{2} \text{trace} \left( K^{\dagger}BAXA^{T} \right) \\
+ \frac{1}{2} f^{T}AX^{T}(A^{T}BA)XA^{T}f + \frac{\epsilon}{2} ||f||_{2}^{2} \text{trace} \left( AX^{T}(A^{T}BA)XA^{T} \right).
\end{align*}
\]

Here 1, 2, and 3 denote the constant, linear, and quadratic terms respectively.
B.1 Derivatives of the linear terms

We differentiate $L(X, \Lambda)$ with respect to $\Lambda_{ij}$. $\delta$ denotes the Kronecker delta symbol.

\[
\frac{\partial L}{\partial \Lambda_{ij}} = \frac{\partial}{\partial \Lambda_{ij}} \left( \text{trace}(\Lambda^T(X - X^T)) \right) \\
= \frac{\partial}{\partial \Lambda_{ij}} \left( \sum_{k,l} \Lambda_{lk}(X_{lk} - X_{kl}) \right) \\
= \sum_{k,l} \delta_{il}\delta_{jk}(X_{lk} - X_{kl}) \\
= X_{ij} - X_{ji}
\]

Thus, $\frac{\partial L}{\partial \Lambda} = X - X^T$ and we recover the symmetry constraint on $X$.

We now differentiate the linear terms of $L(X, \Lambda)$ with respect to $X_{ij}$.

\[
\frac{\partial}{\partial X_{ij}} \left( \text{trace}(\Lambda^T(X - X^T)) \right) = \frac{\partial}{\partial X_{ij}} \left( \sum_{k,l} \Lambda_{lk}(X_{lk} - X_{kl}) \right) \\
= \sum_{k,l} \Lambda_{lk}(\delta_{li}\delta_{kj} - \delta_{ki}\delta_{lj}) \\
= \Lambda_{ij} - \Lambda_{ji}
\]

Thus, $\frac{\partial}{\partial X} \left( \text{trace}(\Lambda^T(X - X^T)) \right) = \Lambda - \Lambda^T$.

For $u, v \in \mathbb{R}^M$, we have

\[
\frac{\partial}{\partial X_{ij}} (u^T X v) = \frac{\partial}{\partial X_{ij}} \left( \sum_{k,l} X_{kl} v_l u_k \right) \\
= \sum_{k,l} \delta_{ki}\delta_{lj} v_l u_k \\
= u_i v_j.
\]

Thus, $\frac{\partial}{\partial X} (u^T X v) = uv^T$. 

31
For $C \in \mathbb{R}^{N \times M}$ and $D \in \mathbb{R}^{M \times N}$, we have
\[
\frac{\partial}{\partial X_{ij}} \left( \text{trace}(CXD) \right) = \frac{\partial}{\partial X_{ij}} \left( \sum_{k,l,m} C_{kl}X_{lm}D_{mk} \right) = \sum_{k,l,m} C_{kl}\delta_{li}\delta_{mj}D_{mk} = \sum_{k} C_{ki}D_{jk} = (DC)_{ji} = (DC)_{ij}^T.\]

Thus, $\frac{\partial}{\partial X} \left( \text{trace}(CXD) \right) = (DC)^T = C^TD^T$.

### B.2 Derivatives of the quadratic terms

We now differentiate the quadratic terms of $L(X, \Lambda)$ with respect to $X_{ij}$.

For $v \in \mathbb{R}^M$ and $F \in \mathbb{R}^{M \times M}$, we have
\[
\frac{\partial}{\partial X_{ij}} \left( v^T X^T F X v \right) = \frac{\partial}{\partial X_{ij}} \left( \sum_{k,l,m,n} F_{kl}v_mv_nX_{lm}X_{kn} \right) = \sum_{k,l,m,n} F_{kl}v_mv_n \left( \delta_{il}\delta_{jm}X_{kn} + \delta_{ki}\delta_{jn}X_{lm} \right) = 2 \sum_{k,n} F_{ki}v_jv_nX_{kn} = 2(FXv)_iv_j.
\]

Thus, $\frac{\partial}{\partial X} \left( v^T X^T F X v \right) = 2FXvv^T$.

For $G \in \mathbb{R}^{N \times M}$ and $H \in \mathbb{R}^{M \times M}$, we have
\[
\frac{\partial}{\partial X_{ij}} \left( \text{trace}(GXTHXGT) \right) = \frac{\partial}{\partial X_{ij}} \left( \sum_{k,l,m,n,p} G_{kl}G_{kp}H_{mn}X_{ml}X_{np} \right) = \sum_{k,l,m,n,p} G_{kl}G_{kp}H_{mn} \left( \delta_{mi}\delta_{lj}X_{np} + \delta_{ni}\delta_{pj}X_{ml} \right) = 2 \sum_{k,n,p} G_{kj}G_{kp}H_{in}X_{np} = 2(HXG^TG)_{ij}.
\]
Thus, \( \frac{\partial}{\partial X} \left( \text{trace}(GX^THX^TG) \right) = 2HXG^TG. \)

### B.3 The stationarity conditions

Choosing

\[
\begin{align*}
  u &= A^TBK^f \\
  v &= A^Tf \\
  C &= K^\dagger BA \\
  D &= A^T \\
  F &= A^TBA \\
  G &= A \\
  H &= A^TBA
\end{align*}
\]

in the equations above, we derive the stationarity conditions for the Lagrangian. We get \( X = X^T \) and Equation (7), which is reproduced here.

\[
(A^TBA)X(A^T(ff^T + \epsilon ||f||_2^2 I)A) - A^T BK^\dagger(ff^T + \epsilon ||f||_2^2 I)A = \Lambda^T - \Lambda
\]

### C On Skew-symmetry of Outer Product of Two Vectors

We prove that if the outer product of two vectors is skew-symmetric, then at least one of the two vectors is zero. We do not require the outer product to be identically zero to deduce that at least one of the vectors is zero. We only require a weaker condition (that the outer product be skew-symmetric).

Let \( u, v \in \mathbb{R}^N \). Let \( uv^T \) be skew-symmetric. Thus \( uv^T + vu^T = 0 \). We show that either \( u = 0 \), or \( v = 0 \), or both can be zero, which is the trivial case. Without loss of generality, assume \( u \neq 0 \). We have the following implications.

\[
\begin{align*}
  uv^T + vu^T &= 0 \\
  \Rightarrow (uv^T + vu^T)u &= 0 \\
  \Rightarrow u(v^T u) + v ||u||_2^2 &= 0 \\
  \Rightarrow u(u^T v) + ||u||_2^2 v &= 0 \\
  \Rightarrow (uv^T + ||u||_2^2 I)v &= 0
\end{align*}
\]
Now $||u||^2 I$ is symmetric positive definite (because $u \neq 0$) and $uu^T$ is symmetric positive semi-definite. Thus, their sum is symmetric positive definite and its action on a vector can be zero iff the vector is 0. Hence $v = 0$ and we're done.

A different (and more elementary) way to prove this result is to explicitly write the elements of $uv^T + vu^T$ (in terms of $u_i$ and $v_i$) and equate each of them to 0 to deduce progressively that either all $u_i = 0$, or all $v_i = 0$, or both vectors are 0. We skip the details.

**Acknowledgements**

This work was supported by the Department of Energy under Grant No. DE-FG02-05ER25701. The authors would also like to thank Dr. J. Tinsley Oden, Dr. C. Grant Willson, Dr. Jon Bass, Dr. Serge Prudhomme, Dr. Paul T. Bauman, and Dr. Elizabeth Collister for many helpful discussions.

**References**


