Fast Finite Element Calculation of Effective Conductivity of Random Continuum Microstructures: the Recursive Poincaré-Steklov Operator Method

Thomas J. Hardin^{*1} and Christopher A. Schuh¹

¹*Department of Materials Science and Engineering, Massachusetts Institute of Technology*

Abstract

The effective conductivity of a block of composite can be extracted from the Dirichlet-to-Neumann Poincaré-Steklov operator (PSO) for that block. In this paper, a domain decomposition method for computing the PSO over a finite element mesh is discussed. A new numerical strategy is introduced to accelerate the computation of this operator, using the Schur complement to calculate the PSO for the smallest subdomains, then recursively merging subdomain PSOs up to the full domain. At each step of the algorithm, information extraneous to the PSO is discarded. The effective conductivity values computed by this method are identical to those obtained from a basic Finite Element Method, an order of magnitude faster and with much less computer memory consumed. As proof of concept, effective conductivity measurements are presented for a percolating random fractal-like microstructure across a range of phase fractions.

Keywords: homogenization, percolation, composite, effective medium, domain decomposition, reduction to the interface

^{*}hardin@mit.edu

1 Introduction

Homogenization is extrapolation of macro-scale properties of a composite from its microstructure—both the character and the spatial arrangement of its constituent phases. This paper deals with homogenization of random composites, focusing on properties related to transport of charge, diffusion of species, or conduction of heat. These phenomena are governed by flux laws of the type [1]:

$$\vec{J} = -\gamma_{\text{local}} \nabla \phi \quad (1)$$

where γ_{local} is a tensor which stands in for local electrical conductivity, diffusivity, or heat conductivity, and ϕ stands in for electrical potential, solute concentration, or temperature, depending on the context. This paper will, without loss of generality, use the language of electrostatics in referring to these quantities. At steady-state, the divergence of the flux field vanishes:

$$0 = \nabla \cdot (\gamma_{\text{local}} \nabla \phi) \quad (2)$$

for ϕ subject to prescribed boundary conditions [2,3]. In a composite where there is contrast between the respective γ values of constituent phases, it is desirable to determine an *effective conductivity tensor* γ_{eff} for the homogenized system, which (at steady-state) satisfies:

$$\langle \vec{J} \rangle = -\gamma_{\text{eff}} \langle \nabla \phi \rangle \quad (3)$$

where $\langle \cdot \rangle$ is a volume average [4]. In the limit of averaging over sufficiently large volumes of the composite microstructure, γ_{eff} converges to a single tensor representative of the composite as a whole, accounting for both the constituents' respective γ_{local} tensors and their spatial arrangement.

There are mathematically elegant ways to access the effective properties of simple microstructures [4–11]; however, for more arbitrary composite microstructures a straightforward approach to determining γ_{eff} is to:

1. generate explicit finite realizations of the microstructure
2. use numerical partial differential equation techniques to solve Eqn. (2) under a variety of boundary conditions
3. calculate $\langle \vec{J} \rangle$ and $\langle \nabla \phi \rangle$ for each instance
4. solve Eqn. (3) for γ_{eff} in a least-squares sense.

There are two broad classes of numerical methods for solving Eqn. (2) over composite microstructures with sharp boundaries between constituent phases: integral methods and finite element methods (FEMs).

Integral methods can be elegant, accurate, and fast, can extend to arbitrarily many dimensions, and can be specialized to periodic microstructures [8–10]. These methods use Green’s identities to transform Eqn. (2) from a volumetric partial differential equation to a surface integral equation [12, 13] where the relevant integral is over all the boundaries between phases in the composite. Unfortunately, evaluating this integral is a challenging proposition for microstructures not having obvious discretizations.

The finite element method is an alternative to integral methods for solving Eqn. (2). This method is extremely flexible and well-studied [14–20], with wide application in solving partial differential equations of many kinds. Using the finite element method to solve Eqn. (2) is straightforward. The technique is also tolerant of arbitrary random composite microstructures; this in addition to its flexibility make the finite element method the technique of interest in this paper.

Many approaches to solve the finite element method in nearly linear time have been developed. The main idea in most of these methods is to divide the finite element mesh into submeshes with continuity equations between them; the individual submeshes can be very efficiently analyzed independently of each other, so the speed of these methods hinges on rapidly solving the continuity equations [21–23]. Such approaches are categorized as multigrid, domain decomposition, or reduction to the interface methods [24–27].

This paper describes a numerical technique for computing the effective conductivity of a block of composite. The new method is based on the same equations used in the finite element method, and produces the same results, but with much less computational time and memory consumption. This is not only efficient, but also provides more accurate homogenized calculations by increasing the size of the finite element meshes that can be analyzed for given computer hardware.

This effort is motivated by an attempt to numerically compute the effective conductivity of continuum composite microstructures with phase fractions near the percolation threshold (a structural phase transition characterized by global connectivity of the reinforcing phase at phase fractions above the percolation threshold and only short-range connectivity below). When Eqn. 2 is solved over a high-contrast random composite near its percolation threshold, the resulting solution often has a somewhat fractal nature [28,29] with features on both long and short length-scales (see Figure 1). Consequently, when using the finite element method for such cases both the composite block size and the sampling density must be higher than would be necessary for composites away from the percolation threshold; the resulting finite element mesh is very large. Ultimately, the traditional finite element method is prohibitively slow and memory-intensive for some microstructures near the percolation threshold. The present development of a faster, less memory-intensive method for computing the effective conductivity of a composite speaks to this need.

2 Method

This section is built on a very simple first-order linear finite element framework (as found in the first three chapters of [15]) but it is readily extensible to higher-order finite element spaces. The figures in this section show a regular two-dimensional grid of square finite elements; however, the equations given here are general with respect to dimensionality and are immediately applicable to arbitrary finite element meshes.

Figure 1: (a): a two-dimensional composite microstructure at its percolation threshold. (b): solution to Eqn. (2) over the microstructure in (a) with a boundary condition of $\phi = 0$ at the top and $\phi = 1$ at the bottom, and isotropic phase γ values of 1 and 10^6 for the dark and light phases, respectively. Note fractal-like features in the solution on a wide range of length scales.

This section is broken into four subsections. In the first subsection, a mathematical object called the Poincaré-Steklov Operator (PSO) is reviewed and a method for representing and calculating the PSO for a finite element mesh is presented. In the second subsection, a case is considered where two adjacent blocks of finite element mesh respectively have known PSOs, and a method is presented for combining the two respective PSOs into a single PSO for the merged mesh. In the third subsection, an algorithm is presented to calculate the PSO for a large finite element mesh. This algorithm exploits the methods in the first two subsections recursively to achieve both memory and time efficiency. Finally, the fourth subsection deals with the boundary conditions to extract γ_{eff} from the PSO. Put together, these pieces comprise what we shall term the “Recursive Poincaré-Steklov Operator Method” (RPSOM).

2.1 PSO Representation & Calculation

One of the properties of Eqn. 2 is the existence of a function that maps every possible Dirichlet boundary condition to a corresponding Neumann boundary condition. That is, given a domain and a γ_{local} distribution across it, it is possible to construct an operator that

Figure 2: The elements in this mesh are partitioned into outer elements (lightly shaded) and interior elements. The nodes are partitioned into three sets: the boundary nodes have values $\vec{y}$, the set of interior nodes in outer elements have values $\vec{x}$, and the remaining interior nodes have values $\vec{z}$.

takes the boundary ϕ values and returns the corresponding boundary fluxes that would be observed upon solving Eqn. 2 with those boundaries. This is the (Dirichlet-to-Neumann) PSO [30,31].

The PSO can be approximated in a discrete setting using the Finite Element Method. Consider a finite element mesh with only Dirichlet boundaries (see Figure 2). Each element has a local γ_{local} tensor assigned to it. The elements containing boundary nodes are called *outer elements*. Let the values of boundary nodes be designated $\vec{y}$. Let the values of the non-boundary nodes in outer elements be designated $\vec{x}$. Let the values of all other nodes in the mesh—nodes not belonging to an outer element—be designated $\vec{z}$.

Now suppose that one wants to calculate the flux crossing the boundary between nodes c and d in the detail of Figure 2. The **mean** gradient of the potential field in element $abcd$ can in this case be **calculated**:

$$\nabla\phi \approx \left(\frac{(\phi(c) + \phi(d)) - (\phi(a) + \phi(b))}{2\lambda}, \frac{(\phi(a) + \phi(c)) - (\phi(b) + \phi(d))}{2\lambda} \right) \quad (4)$$

where λ is the element side length. The flux passing through the element can then be approximated:

$$\vec{J} \approx -\gamma_{\text{local}} \nabla \phi \quad (5)$$

from which the flux component crossing the boundary is readily obtained.

While the details of computing the flux across the boundary may vary depending on the finite element mesh in question (the square grid in Figure 2 is particularly simple), there is a key characteristic that will remain general: the computation required only γ_{local} and nodal potential values from outer elements. That is, to compute the boundary flux for the entire mesh, all that need be known are $\vec{x}$, $\vec{y}$, and γ_{local} from outer elements.

Consequently, if γ_{local} is known for the outer elements of a mesh, the PSO can be represented as simply a function mapping $\vec{y}$ (which, recall, represents the Dirichlet boundary condition) to $\vec{x}$. This function can be represented as a matrix $\mathbf{P}$ such that $\mathbf{P}\vec{y} = \vec{x}$. Each of the $n + p$ interior nodes corresponds (by the finite element method) to a linear equation linking the $n + p$ values of the interior nodes to the m boundary node values; these equations can be compiled into a block matrix equation [15]:

$$\begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xz} \\ \mathbf{K}_{zx} & \mathbf{K}_{zz} \end{bmatrix} \cdot \begin{bmatrix} \vec{x} \\ \vec{z} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{xy} \\ \mathbf{F}_{zy} \end{bmatrix} \cdot \vec{y} \quad (6)$$

where $\mathbf{K}$ and $(\mathbf{F}\vec{y})$ are respectively the familiar stiffness matrix and force vector from the finite element method.

At this point, one could potentially simply factor $\mathbf{K}$ and solve for $\mathbf{K}^{-1}\mathbf{F}$; however, since the PSO requires only the solution to $\vec{x}$ as a function of $\vec{y}$, there is no need to solve against rows of $\mathbf{F}$ corresponding to $\vec{z}$. The Schur complement [32–34] provides a way to avoid extraneous computations. Multiplying the bottom row of (6) by $\mathbf{K}_{xz}\mathbf{K}_{zz}^{-1}$ and subtracting from the top row yields:

$$\begin{bmatrix} \mathbf{K}_{xx} - \mathbf{K}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{K}_{zx} & \mathbf{K}_{xz} - \mathbf{K}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{K}_{zz} \end{bmatrix} \cdot \begin{bmatrix} \vec{x} \\ \vec{z} \end{bmatrix} = [\mathbf{F}_{xy} - \mathbf{K}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{F}_{zy}] \cdot \vec{y} \quad (7)$$

which reduces to

$$(\mathbf{K}_{xx} - \mathbf{K}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{K}_{zx})\vec{x} = (\mathbf{F}_{xy} - \mathbf{K}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{F}_{zy})\vec{y} \quad (8)$$

The quantity $\mathbf{S} = \mathbf{K}_{xx} - \mathbf{K}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{K}_{zx}$ is the Schur complement of block $\mathbf{K}_{zz}$; since $\mathbf{K}$ is singular positive definite, it can be shown (see [35] page 834) that $\mathbf{S}$ is also singular positive definite. Eq. (8) can be numerically solved to obtain:

$$\vec{x} = [\mathbf{S}^{-1}(\mathbf{F}_{xy} - \mathbf{K}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{F}_{zy})]\vec{y} \quad (9)$$

So the matrix $\mathbf{P} = \mathbf{S}^{-1}(\mathbf{F}_{xy} - \mathbf{K}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{F}_{zy})$, in addition to γ_{local} for the outer elements, represents the PSO for the mesh in Figure 2.

2.2 Merging two PSOs

Consider now a finite element mesh whose elements have been partitioned into two submeshes—suppose, into a left and a right half. There will be nodes shared between the two submeshes (on the interface, see Figure 3); put the values of those nodes into $\vec{w}$. The remaining nodes on the left will have values in a vector $\vec{u}$ and the remaining nodes on the right will have values in a vector $\vec{v}$.

As shown in Figure 3, we further subdivide the nodes as follows. Let $\vec{w}_q$ be the subset of $\vec{w}$ representing nodes on the boundary of the combined supermesh. Let $\vec{w}_r$ be the part of $\vec{w}$ not in $\vec{w}_q$. Let $\vec{u}_y$ be the subset of $\vec{u}$ representing nodes on the boundary of the combined supermesh. Let $\vec{u}_x$ be the part of $\vec{u}$ representing interior nodes belonging to outer elements of the left submesh. Let $\vec{v}_y$ and $\vec{v}_x$ be defined similarly.

Now suppose that a PSO is known for each of the two halves individually; that is, γ_{local} is known for all the outer elements of the two submeshes and the two parent PSOs are $\mathbf{P}^u$ and $\mathbf{P}^v$, respectively:

$$\vec{u}_x = \mathbf{P}_{xy}^u \vec{u}_y + \mathbf{P}_{xq}^u \vec{w}_q + \mathbf{P}_{xr}^u \vec{w}_r \quad (10)$$

$$\vec{v}_x = \mathbf{P}_{xy}^v \vec{v}_y + \mathbf{P}_{xq}^v \vec{w}_q + \mathbf{P}_{xr}^v \vec{w}_r \quad (11)$$

Figure 3: Scheme for partitioning nodes when combining two PSOs (sharing an edge) into one.

The finite element equations corresponding to r nodes may be expressed similarly to (6):

$$\mathbf{K}_{rr}\vec{w}_r = \mathbf{F}_{rq}^w\vec{w}_q + \mathbf{F}_{ry}^u\vec{u}_y + \mathbf{F}_{rx}^u\vec{u}_x + \mathbf{F}_{ry}^v\vec{v}_y + \mathbf{F}_{rx}^v\vec{v}_x \quad (12)$$

Note that $\mathbf{F}_{ry}$ and $\mathbf{F}_{rx}$ will be very sparse and can be manipulated into a block structure by judicious node numbering. Substitute Eqs. (10) and (11) into (12):

$$\begin{aligned} \mathbf{K}_{rr}\vec{w}_r = & \mathbf{F}_{rq}^w\vec{w}_q + \mathbf{F}_{ry}^u\vec{u}_y + \mathbf{F}_{rx}^u \left(\mathbf{P}_{xy}^u\vec{u}_y + \mathbf{P}_{xq}^u\vec{w}_q + \mathbf{P}_{xr}^u\vec{w}_r \right) \\ & + \mathbf{F}_{ry}^v\vec{v}_y + \mathbf{F}_{rx}^v \left(\mathbf{P}_{xy}^v\vec{v}_y + \mathbf{P}_{xq}^v\vec{w}_q + \mathbf{P}_{xr}^v\vec{w}_r \right) \end{aligned} \quad (13)$$

which simplifies to:

$$\begin{aligned} (\mathbf{K}_{rr} - \mathbf{F}_{rx}^u\mathbf{P}_{xr}^u - \mathbf{F}_{rx}^v\mathbf{P}_{xr}^v)\vec{w}_r = & (\mathbf{F}_{rq}^w + \mathbf{F}_{rx}^u\mathbf{P}_{xq}^u + \mathbf{F}_{rx}^v\mathbf{P}_{xq}^v)\vec{w}_q + \\ & (\mathbf{F}_{ry}^u + \mathbf{F}_{rx}^u\mathbf{P}_{xy}^u)\vec{u}_y + (\mathbf{F}_{ry}^v + \mathbf{F}_{rx}^v\mathbf{P}_{xy}^v)\vec{v}_y \end{aligned} \quad (14)$$

Define the following:

$$\Sigma = \mathbf{K}_{rr} - \mathbf{F}_{rx}^u\mathbf{P}_{xr}^u - \mathbf{F}_{rx}^v\mathbf{P}_{xr}^v \quad (15)$$

$$\mathbf{\Pi}_{rq}^w = \mathbf{\Sigma}^{-1} (\mathbf{F}_{rq}^w + \mathbf{F}_{rx}^u \mathbf{P}_{xq}^u + \mathbf{F}_{rx}^v \mathbf{P}_{xq}^v) \quad (16)$$

$$\mathbf{\Pi}_{ry}^u = \mathbf{\Sigma}^{-1} (\mathbf{F}_{ry}^u + \mathbf{F}_{rx}^u \mathbf{P}_{xy}^u) \quad (17)$$

$$\mathbf{\Pi}_{ry}^v = \mathbf{\Sigma}^{-1} (\mathbf{F}_{ry}^v + \mathbf{F}_{rx}^v \mathbf{P}_{xy}^v) \quad (18)$$

So, substituting into (14) one obtains:

$$\vec{w}_r = \mathbf{\Pi}_{rq}^w \vec{w}_q + \mathbf{\Pi}_{ry}^u \vec{u}_y + \mathbf{\Pi}_{ry}^v \vec{v}_y \quad (19)$$

This can in turn be introduced into (10):

$$\begin{aligned} \vec{u}_x &= \mathbf{P}_{xy}^u \vec{u}_y + \mathbf{P}_{xq}^u \vec{w}_q + \mathbf{P}_{xr}^u (\mathbf{\Pi}_{rq}^w \vec{w}_q + \mathbf{\Pi}_{ry}^u \vec{u}_y + \mathbf{\Pi}_{ry}^v \vec{v}_y) \\ &= (\mathbf{P}_{xy}^u + \mathbf{P}_{xr}^u \mathbf{\Pi}_{ry}^u) \vec{u}_y + (\mathbf{P}_{xq}^u + \mathbf{P}_{xr}^u \mathbf{\Pi}_{rq}^w) \vec{w}_q + \mathbf{P}_{xr}^u \mathbf{\Pi}_{ry}^v \vec{v}_y \end{aligned} \quad (20)$$

$$\begin{aligned} \vec{v}_x &= \mathbf{P}_{xy}^v \vec{v}_y + \mathbf{P}_{xq}^v \vec{w}_q + \mathbf{P}_{xr}^v (\mathbf{\Pi}_{rq}^w \vec{w}_q + \mathbf{\Pi}_{ry}^u \vec{u}_y + \mathbf{\Pi}_{ry}^v \vec{v}_y) \\ &= (\mathbf{P}_{xy}^v + \mathbf{P}_{xr}^v \mathbf{\Pi}_{ry}^v) \vec{v}_y + (\mathbf{P}_{xq}^v + \mathbf{P}_{xr}^v \mathbf{\Pi}_{rq}^w) \vec{w}_q + \mathbf{P}_{xr}^v \mathbf{\Pi}_{ry}^u \vec{u}_y \end{aligned} \quad (21)$$

By selecting the relevant rows of (19), (20), and (21), one may finally construct a new PSO for the combined mesh block. One may discard the rows of (19), (20), and (21) corresponding to nodes that do not belong to edge elements of the merged block; these rows (equations) are totally extraneous to the PSO of the merged block of mesh. Similarly, one may discard the γ_{local} values for mesh elements that are not outer elements of the merged block.

2.3 A Recursive Algorithm for PSO Calculation

Conceptually, the approach in this section is to break a large finite element mesh into small blocks, calculate the PSO for each block, pairwise merge the PSOs of the small blocks into larger blocks, then pairwise combine those into even larger blocks, and so forth, until the PSO is known for the full mesh. After each merge step, information relating to γ_{local} and nodal

Algorithm 1: Calculate PSO Recursively

```
1: function PSOCALCRECURSIVE(Mesh  $M$ ) ▷ Returns the PSO of  $M$ 
2:   if the size of  $M$  is less than a threshold then
3:     return PSOCALCDIRECT( $M$ )
4:   else
5:     Mesh  $L, R \leftarrow$  PARTITION( $M$ )
6:     PSO  $psoL \leftarrow$  PSOCALCRECURSIVE( $L$ )
7:     PSO  $psoR \leftarrow$  PSOCALCRECURSIVE( $R$ )
8:     return PSOMERGE( $psoL, psychoR$ )
9:   end if
10: end function

11: function PSOCALCDIRECT(Mesh  $M$ ) ▷ Returns the PSO of  $M$  using Sec. 2.1
12: function PARTITION(Mesh  $M$ ) ▷ Returns two half-meshes partitioned from  $M$ 
13: function PSOMERGE(PSO  $psoL$ , PSO  $psoR$ ) ▷ Returns a single PSO merged from  $psoL$  and  $psoR$  using Sec. 2.2
```

values not belonging to outer elements of the mesh block are discarded. The new method is outlined in pseudocode in Algorithm 1. Note that the “partition” step must produce submeshes sharing at least one boundary node, and that for performance the partitions should be of roughly equal size and the boundary should be as small as possible; note also that due to this requirement, this algorithm might not be particularly helpful for some finite element meshes.

This algorithm bears some obvious similarity to the multigrid, domain decomposition, and reduction to the interface methods mentioned in the introduction. However, it achieves improved performance and memory efficiency by discarding information on unnecessary degrees of freedom.

2.4 Boundary Conditions and Extracting the Effective Conductivity

Having constructed the PSO for a finite element mesh, all that remains is configuring the boundary conditions in a way that is conducive to extracting γ_{eff} . So, consider again the mesh in Figure 2. Suppose that γ_{local} is known for the outer elements of the mesh, and that a matrix $\mathbf{P}$ is known such that $\vec{x} = \mathbf{P}\vec{y}$.

Now suppose that one wants to switch one of the Dirichlet boundary edges to an insulated boundary — let the values of nodes that are to be transformed be contained in $\vec{a}$ and let $\vec{b}$ contain the node values in $\vec{y}$ but not in $\vec{a}$. So:

$$\vec{x} = \mathbf{P}_{xa}\vec{a} + \mathbf{P}_{xb}\vec{b} \quad (22)$$

Now, going through each of the nodes that are to be transformed, the finite element method can construct a system of linear equations linking the ex-Dirichlet node values to the rest of the nodes in $\vec{y}$ and $\vec{x}$. That is, these equations can be expressed similarly to Eqn. (6):

$$\mathbf{K}_{aa}\vec{a} = \mathbf{F}_{ab}\vec{b} + \mathbf{F}_{ax}\vec{x} \quad (23)$$

Substituting Eqn. (22) into Eqn. (23) and solving for $\vec{a}$, one finds:

$$\vec{a} = (\mathbf{K}_{aa} - \mathbf{F}_{ax}\mathbf{P}_{xa})^{-1} (\mathbf{F}_{ab} + \mathbf{F}_{ax}\mathbf{P}_{xb})\vec{b} \quad (24)$$

which can be substituted back into Eqn. (22) to obtain a new PSO for the new boundary configuration. Similarly, to construct a periodic boundary condition, one need simply construct the necessary finite element equations, substitute Eqn. (22) into them, and solve as above.

To extract the effective conductivity from a composite block for which the PSO is known, one strategy is to establish a “global” potential gradient by constraining one side to $\phi = 0$, the opposing side to $\phi = \Phi$, and insulating all other sides. This fixes $\langle \nabla \phi \rangle \approx (\Phi/L)\hat{n}$ (where $\hat{n}$ is a unit vector in the direction of the imposed potential gradient). This configuration also ensures that $\langle \vec{J} \rangle \cdot \hat{n} = \langle \vec{J} \cdot \hat{n} \rangle_\Gamma$ where Γ is either of the two Dirichlet boundary edges. Substitution of these expressions into Eqn. (3) yields a term of γ_{eff} . Unfortunately, when the block is insulated on its sides, it is not possible to recover a meaningful mean flux measurement in directions other than that of the global potential gradient. This reduces the amount of information that can be extracted from each simulation; nonetheless, by rotating the composite domain relative to the global potential gradient, it is possible to construct the

γ_{eff} tensor.

Note that the above-described boundary scheme is one of several that accesses effective conductivity; another might be full Dirichlet conditions on the block boundary. Ideally, boundary effects can be corrected for by extrapolating across increasing domain sizes.

3 Results

3.1 Performance

The RPSOM was benchmarked against a simple traditional finite element method for two-dimensional square-grid meshes with a variety of side lengths. In all cases where a traditional finite element solution was computed, it matched the boundary solution produced by the RPSOM to within floating-point error. Both methods were coded in C++11 and compiled using the Intel compiler with full optimization enabled, and were linked against the Intel MKL library’s single-thread linear algebra routines. Both methods were tested on a Linux workstation with an Intel Xeon E5-2640 V3 processor, clocked at 2.6GHz. The code was written for strictly serial execution, though the RPSOM algorithm does suggest the possibility of at least partial parallelization. The traditional finite element method used an optimized-bandwidth sparse stiffness matrix, factored using a Simplicial Cholesky LDLt algorithm from the Eigen 3 package [36]. The RPSOM implementation utilized both Eigen 3’s matrix manipulation utilities and MKL’s single-threaded BLAS and LAPACK algorithms. The benchmarks shown here were generated using a Python wrapper running the algorithms from the shell; the wrapper program measured memory consumption (taking 100 samples per second) and reported the peak memory consumed by the program along with the total execution time.

As shown in Figure 4a, the RPSOM realized computation times an order of magnitude faster than the traditional finite element method. The performance gains observed from this method stem from subdividing the mesh and discarding extraneous degrees of freedom from

Figure 4: Time and peak memory consumption to compute the effective conductivity for a square grid of $n \times n$ elements, using the benchmark described in the body of this paper.

the calculation. Consider a mesh of $N \times N$ nodes. Using a banded Cholesky solver, factoring the stiffness matrix of the mesh would cost $O(N^5)$ operations [37]. Conversely, suppose that the mesh is subdivided into $\frac{N \times N}{n \times n}$ submeshes of dimension $n \times n$. Computing the PSO for a single submesh using the Schur complement method outlined in Section 2.1 costs $O(n^6)$, assuming fully dense matrix arithmetic. Therefore, to compute all the submeshes' PSO costs $O(n^6 \times \frac{N \times N}{n \times n}) = O(N^2 n^4)$, which (for fixed n) is a dramatic improvement on factoring the full stiffness matrix. Of course, the downside is the necessity of then stitching together the submeshes' PSOs. The operation count for stitching two PSOs together that share an edge of p nodes is $O(p^3)$. Since at least the last merge operation occurs with a merge edge length of N , the asymptotic performance of the algorithm is $O(N^3)$. Nonetheless, the dramatic speedup shown in Figure 4a speaks to the utility of the method.

As shown in Figure 4b, the memory consumed by the new method is less than a quarter of that consumed by the standard finite element method (which, as previously mentioned, leveraged a fully sparse matrix framework for its calculations).

The reduction in memory achieved by this method stems entirely from discarding extraneous degrees of freedom. Returning to the $N \times N$ square-element mesh, for the standard finite element method to store its Cholesky factorization costs $O(N^3)$. Conversely, since the new method only ever stores information regarding mesh and submesh boundaries, the storage requirement for the new method is **practically** closer to $O(N^2)$.

3.2 Calculations on a Random Composite Microstructure

As described in Section 1, this study is motivated by a desire to measure γ_{eff} near the percolation threshold of random composites. This section shows some preliminary results along those lines for a random two-dimensional composite microstructure.

The test microstructure selected is shown at three phase fraction values in Figure 5 and its exact specifications are provided in the Appendix. The microstructure is constructed around “seed points” randomly distributed in the plane, and has irregularly-shaped features at three

Figure 5: Test microstructure at three phase fractions; each sample edge has a length of five units.

distinct length scales (similar to a fractal). The irregularity, randomness, and fractal-like qualities of this microstructure make it a challenging candidate for homogenization by many existing methods. This section shows how the RPSOM nonetheless enables extraction of the test microstructure’s effective conductivity.

The fundamental unit of this computational experiment is an “instantiation” of the microstructure. Each instantiation is a square block with a characteristic edge length (L) and a sampling resolution (R) equal to the number of elements per unit length. Each instantiation also has a unique random spatial phase configuration and can assume any phase fraction in $[0, 1]$. For this study, the conductivity of the dark phase was set to 1 and that of the light phase to 10^6 .

For each instantiation, the RPSOM was used to calculate effective conductivity at a variety of phase fractions (p); the phase fractions were sampled between 0.05-0.95 at intervals of 0.05, and between 0.27-0.33 at intervals of 0.005. The resulting $p \mapsto \gamma_{\text{eff}}$ curve was subsequently least-squares fitted to the Generalized Effective Medium (GEM) equation [38, 39]. The GEM equation has three parameters: p_c , s , and t . The parameter p_c corresponds with the percolation threshold, and s and t are critical exponents that respectively describe the curvature of the $p \mapsto \gamma_{\text{eff}}$ function immediately below and above the percolation threshold. The least-squares fitting procedure varied these three parameters to minimize p -direction error:

L	R					
	29	32	37	43	51	64
29	2051	2052	2052	2052	2052	2052
32	1820	1820	1820	1820	1820	1820
37	1362	1362	1362	1362	1362	1362
43	1135	1135	1135	1135	1135	1135
51	681	681	681	681	681	681
64	454	454	454	454	454	454
85	227	227	227	227	227	227

Table 1: Number of replicates for each combination of L and R levels.

Figure 6: For each of the 46385 microstructure instantiations, (p, γ_{eff}) points were fitted by the GEM equation, and then the term of the residual with the largest magnitude was found (ϵ). This figure shows a histogram of the ϵ statistics; in about 99% of cases, the worst error in the fit is less than 0.02.

$$\min_{p_{c,s,t}} \sum_i (p_{p_{c,s,t}}(\gamma_{\text{eff}}^i) - p^i)^2 \quad (25)$$

where $(p^i, \gamma_{\text{eff}}^i)$ are points sampled in the $p \mapsto \gamma_{\text{eff}}$ curve, and $p_{p_{c,s,t}}$ is the GEM-fit function. This approach varies from traditional least-squares, but avoids difficulties resultant from the fact that the range of γ_{eff} covers several orders of magnitude. A histogram of the fit error is shown in Figure 6 and the 50th and 99th percentile (nearly the worst) fits are shown in Figure 7. These indicate that the GEM is a satisfactory fit for each instantiation's $p \mapsto \gamma_{\text{eff}}$ curve.

The full experiment produced microstructure instantiations across a grid in (L, R) space, with the number of replicates for each (L, R) combination shown in Table 1. Each instanti-

Figure 7: The 50th and 99th percentile (so, the median and nearly the worst) fits of the GEM equation to effective conductivity measurements of an instantiation.

ation produced a value of p_c , s , and t ; for each (L, R) combination, the p_c , s , and t values were averaged, and the standard error and 95% confidence intervals were calculated. These are shown in Figure 8; note that in the plots in that figure, the abscissa axes are inverted. For each of p_c , s , and t , the following bilinear model was fit to the means using weighted least-squares (using inverse sample standard deviations as weights):

$$\{p_c|s|t\} \approx a_{11} + a_{L1}\frac{1}{L} + a_{1R}\frac{1}{R} + a_{LR}\frac{1}{LR} \quad (26)$$

which enabled extrapolation to $L \rightarrow \infty$ and $R \rightarrow \infty$ to eliminate edge and finite sampling effects from the experiment. The resulting measurements for the percolation threshold and critical exponents for the test microstructure are: $p_c = 0.3080 \pm 0.0013$, $s = 1.423 \pm 0.013$, and $t = 1.3707 \pm 0.0045$.

To validate these measurements, a “traditional” p_c estimate was calculated using methods roughly analogous to [40]; details of this estimate are presented in the Appendix. The “traditional” percolation threshold measurement was 0.30998 ± 0.00087 . This estimate is statistically distinct from the RPSOM+GEM-measured percolation threshold (z -test $p = 0.011$), but it is notable that the RPSOM+GEM-estimated percolation threshold varies from the traditionally-estimated threshold by a mere 0.002, which inspires some confidence in the RPSOM+GEM-measured s and t parameters.

4 Conclusion

Whereas a simple finite element method provides a way of calculating the effective electrical conductivity, diffusivity, or heat conductivity of a block of composite (these properties are generalized by the symbol γ), the method in this paper does so more efficiently by discarding all unnecessary internal computation and focusing on the domain boundary. The Poincaré-Steklov Operator (PSO) (which maps Dirichlet to corresponding Neumann boundary conditions) captures the information necessary to calculate γ_{eff} for a finite block of composite.

Figure 8: GEM-fitted parameter data: (from top to bottom) p_c , s , and t . Sample means are marked with $\bullet$, along with 95% confidence intervals in solid gray. The colored surface fits the sample means; the vertical black lines descend from the sample means to the fit surface. The abscissa axes are inverted in these plots.

Here we show that it can be represented for a finite element mesh using only nodal and γ_{local} values from elements containing boundary nodes; nodal and γ_{local} information from interior elements is extraneous to the PSO. By recursively subdividing a large finite element mesh, calculating individual PSOs for the smallest subdivisions, and then recursively merging the PSOs to construct the full mesh’s PSO, the “Recursive Poincaré-Steklov Operator Method” (RPSOM) discards the internal degrees of freedom that are not necessary to represent the PSO at each step and consumes less than a quarter the memory of a traditional finite element algorithm. Similarly, by disregarding extraneous internal degrees of freedom, the RPSOM achieves a speedup of an order of magnitude over the traditional finite element method.

The RPSOM is demonstrated here by calculating the effective conductivity of a random composite microstructure at a variety of phase fractions. The $p - \gamma$ curves thus generated are well fitted by the Generalized Effective Medium equation and can be extrapolated to eliminate finite size and finite sampling effects. Although the RPSOM is illustrated here for two-dimensional square-grid meshes, it straightforwardly is extensible to three dimensions and general finite element meshes. While this paper focused on phenomena governed by Ohm’s Law-like equations, other elliptic partial differential equations feature PSOs, including elasticity. The concepts in this paper may generalize to, for example, fast calculation of forces at mesh boundaries in response to boundary displacement.

Acknowledgements

TJH gratefully acknowledges funding from the Department of Defense (DoD) through the National Defense Science & Engineering Graduate (NDSEG) Fellowship Program. CAS acknowledges support from the US National Science Foundation under contract CMMI-1332789.

Appendix A: Test Microstructure Specification

A block D of the test microstructure is constructed using a multi-step numerical process as follows:

1. The region D , scaled by a factor of four, (call it $4D$) and the region immediately surrounding $4D$ are randomly seeded with points P using a Poisson process with parameter $\Lambda = 1$.
2. Define two functions from $4D \mapsto [0, \infty)$ as the distances to the first and second nearest seed points; call those distances respectively $r_1(x)$ and $r_2(x)$ (where $x \in 4D$; note that $r_1(x) \leq r_2(x)$ in general).
3. A scalar field $z : 4D \mapsto [1, \infty)$ is defined: $z(x) = r_2(x)/r_1(x)$ (see Figure 9a).
4. Define L as the coordinates of the corner of D opposite the origin. Define a new scalar field $Z : D \mapsto [1, \infty)$: $Z(x) = z(x) + z(2L - 2x)/2 + z(4x)/4$ (see Figure 9b).
5. Let $\lambda(p)$ be a function $\lambda \mapsto [0, \infty)$ be a function such that for random $x \in D$, the probability that $Z(x) < \lambda(p)$ is p .
6. Define the microstructure as follows: for point $x \in D$ and $p \in [0, 1]$, if $Z(x) < \lambda(p)$, then x is phase 0. Otherwise, x is phase 1. The resulting microstructure has phase fraction p of phase 0.

The resulting microstructure is pictured in Figure 5 at three phase fraction levels.

Appendix B: Test Microstructure Percolation Threshold Measurement

For validation purposes, the percolation threshold is calculated using methods similar to those in [40]. The test microstructure is generated for blocks of side length L , sampled in a

Figure 9: (a): A sample of field z , and (b) a sample of field Z , as described in Appendix A. Both fields as shown in this figure have been scaled to $[0, 1]$ such that by thresholding at p will produce a structure with phase fraction p .

square grid of side length n with a sampling density of $R = n/L$. The number of replicates for each combination of L and R levels is found in Table 2. For each microstructure block, the percolation threshold was calculated to within 0.1% using a graph-searching algorithm, assuming both edge-only connectivity and edges plus corners connectivity. For each (L, R) treatment, the mean percolation threshold and standard error is calculated. These mean percolation thresholds were fitted using weighted least-squares on the following model:

$$p_c \approx a_{11} + a_{L1} \frac{1}{L} + a_{1R} \frac{1}{R} + a_{LR} \frac{1}{LR} + a_{1R^2} \frac{1}{R^2} + a_{LR^2} \frac{1}{LR^2} \quad (27)$$

The fits of the percolation threshold were satisfactory. The model is then extrapolated out to $L = \infty$ and $R = \infty$; the model fits are shown plotted against R in the $L = \infty$ plane in Figure 10. Both the edges-only and the edges+corners case produce essentially the same extrapolated solution: 0.31003 ± 0.00086 and 0.30998 ± 0.00087 , respectively. As mentioned in this paper's body, these measurements of p_c show satisfactory agreement with the GEM-calculated percolation threshold.

L	R							
	29	32	37	43	51	64	85	128
26	15100	16129	15100	16129	15100	16129	15100	16129
32	10784	11652	10784	11652	10784	11652	10784	11652
43	5850	6412	5850	6412	5850	6412	5851	6413
64	2509	2638	2509	2638	2509	2638	2509	2638
128	607	561	607	561	607	561	607	561

Table 2: Number of replicates for each combination of L and R levels.

Figure 10: Cross-sections of the model fit for both the edges-only and the edges+corners case, sectioned at the $L = \infty$ plane and plotted with respect to $\frac{1}{R}$. The point where the curves cross $R = \infty$ is the final estimate for the percolation threshold of the microstructure.

References

- [1] R. W. Balluffi, S. Allen, and W. C. Carter, *Kinetics of materials*. John Wiley & Sons, 2005.
- [2] D. Gilbarg and N. S. Trudinger, *Elliptic partial differential equations of second order*. Springer, 2015.
- [3] R. B. Guenther and J. W. Lee, *Partial differential equations of mathematical physics and integral equations*. Courier Corporation, 1988.
- [4] N. B. Murphy, E. Cherkaev, C. Hohenegger, and K. M. Golden, “Spectral measure computations for composite materials,” *Communications in Mathematical Sciences*, vol. 13, no. 4, pp. 825–862, 2015.
- [5] K. Golden and G. Papanicolaou, “Bounds for effective parameters of heterogeneous media by analytic continuation,” *Communications in Mathematical Physics*, vol. 90, no. 4, pp. 473–491, 1983.
- [6] G. W. Milton, *The Theory of Composites (Cambridge Monographs on Applied and Computational Mathematics)*. Cambridge University Press, 2002.
- [7] N. Murphy and K. Golden, “The ising model and critical behavior of transport in binary composite media,” *Journal of Mathematical Physics*, vol. 53, no. 6, p. 063506, 2012.
- [8] J. Helsing, “The effective conductivity of arrays of squares: large random unit cells and extreme contrast ratios,” *Journal of Computational Physics*, vol. 230, no. 20, pp. 7533–7547, 2011.
- [9] ———, “The effective conductivity of random checkerboards,” *Journal of Computational Physics*, vol. 230, no. 4, pp. 1171–1181, 2011.
- [10] J. Helsing, R. C. McPhedran, and G. W. Milton, “Spectral super-resolution in meta-material composites,” *New Journal of Physics*, vol. 13, no. 11, p. 115005, 2011.
- [11] M. Schneider, D. Merkert, and M. Kabel, “Fft-based homogenization for microstructures discretized by linear hexahedral elements,” *International Journal for Numerical Methods in Engineering*, vol. 109, no. 10, pp. 1461–1489, 2017.
- [12] L. Greengard and M. Moura, “On the numerical evaluation of electrostatic fields in composite materials,” *Acta numerica*, vol. 3, pp. 379–410, 1994.
- [13] L. Greengard and J.-Y. Lee, “Electrostatics and heat conduction in high contrast composite materials,” *Journal of Computational Physics*, vol. 211, no. 1, pp. 64–76, 2006.
- [14] G. Strang and G. J. Fix, *An analysis of the finite element method*. Prentice-Hall Englewood Cliffs, NJ, 1973, vol. 212.
- [15] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

- [16] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*. Springer Science & Business Media, 2012, vol. 15.
- [17] P. G. Ciarlet, *The finite element method for elliptic problems*. Siam, 2002, vol. 40.
- [18] V. Girault and P.-A. Raviart, *Finite element methods for Navier-Stokes equations: theory and algorithms*. Springer Science & Business Media, 2012, vol. 5.
- [19] S. G. Mikhlin and T. Boddington, *Variational methods in mathematical physics*. Pergamon Press Oxford, 1964, vol. 1.
- [20] A. A. Samarskij and E. S. Nikolaev, *Numerical Methods for Grid Equations: Volume II Iterative Methods*. Birkhäuser, 2012.
- [21] A. Quarteroni and A. Valli, *Numerical approximation of partial differential equations*. Springer Science & Business Media, 2008, vol. 23.
- [22] W. Hackbusch, *Multi-grid methods and applications*. Springer Science & Business Media, 2013, vol. 4.
- [23] S. Brenner and R. Scott, *The mathematical theory of finite element methods*. Springer Science & Business Media, 2007, vol. 15.
- [24] B. N. Khoromskij and G. Wittum, *Numerical solution of elliptic differential equations by reduction to the interface*. Springer Science & Business Media, 2012, vol. 36.
- [25] J. H. Bramble, *Multigrid methods*. CRC Press, 1993, vol. 294.
- [26] B. Smith, P. Bjorstad, and W. Gropp, *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge university press, 2004.
- [27] A. Toselli and O. Widlund, *Domain decomposition methods: algorithms and theory*. Springer, 2005, vol. 34.
- [28] D. Ben-Avraham and S. Havlin, “Diffusion on percolation clusters at criticality,” *Journal of Physics A: Mathematical and General*, vol. 15, no. 12, p. L691, 1982.
- [29] B. Sapoval, M. Rosso, and J.-F. Gouyet, “The fractal nature of a diffusion front and the relation to percolation,” *Journal de Physique Lettres*, vol. 46, no. 4, pp. 149–156, 1985.
- [30] H. Poincaré, “La méthode de neumann et le probleme de dirichlet,” *Acta mathematica*, vol. 20, no. 1, pp. 59–142, 1897.
- [31] V. Steklov, “General methods for the solution of principal mathematical physics problems,” *Kharkov Mathematical Society Publishing, Kharkov*, 1901.
- [32] E. V. Haynsworth, “On the schur complement.” DTIC Document, Tech. Rep., 1968.
- [33] —, “Determination of the inertia of a partitioned hermitian matrix,” *Linear algebra and its applications*, vol. 1, no. 1, pp. 73–81, 1968.

- [34] J. Schur, “Über potenzreihen, die im innern des einheitskreises beschränkt sind.” *Journal für die reine und angewandte Mathematik*, vol. 147, pp. 205–232, 1917.
- [35] T. H. Cormen, *Introduction to algorithms*. MIT press, 2009.
- [36] G. Guennebaud, B. Jacob *et al.*, “Eigen v3,” <http://eigen.tuxfamily.org>, 2010.
- [37] J. W. Demmel, *Applied numerical linear algebra*. Siam, 1997.
- [38] D. McLachlan, “Measurement and analysis of a model dual conductivity medium using a generalized effective medium theory,” *Physica A: Statistical Mechanics and its Applications*, vol. 157, no. 1, pp. 188–191, 1989.
- [39] D. S. McLachlan, M. Blaszkiewicz, and R. E. Newnham, “Electrical resistivity of composites,” *Journal of the American Ceramic Society*, vol. 73, no. 8, pp. 2187–2203, 1990.
- [40] V. E. Brunini, C. A. Schuh, and W. C. Carter, “Percolation of diffusionally evolved two-phase systems,” *Physical Review E*, vol. 83, no. 2, p. 021119, 2011.