Computation of effective dielectric tensor of two-phase composites using boundary element method and $\mu$-CT scans

M-J. Yvonne Ou  
Department of Mathematical Sciences  
University of Delaware  
Newark, DE 19711

Sylvain Nintcheu Fata  
Division of Computational Mathematics  
Oak Ridge National Laboratory  
Oak Ridge, TN 37831

Tsz-Kin Mak & Ming Xu  
Department of Mathematics  
Chinese University of Hong Kong  
Hong Kong

Curtis Kisielius  
Department of Computer Engineering,  
University of Delaware,  
Newark, DE 19716

1 Introduction

In the study of two-phase composites with fine microarchitecture, the relation between effective properties, the constituent material properties and the microstructure plays a central role. For special microstructure such as rank-N laminates and composites with ellipsoidal inclusions, explicit formulas of effective properties as a function of microstructure have been derived [Mil02]. However, for composites with more complex microstructure such as cancellous bone, explicit formulas for bulk properties are nonexistent, hence it relies on numerical computation to predict the effective properties of composites from informations on the microgeometry and the constituent material properties. For the case of cancellous bones, the main challenge arises from the fact that cancellous bone consists of very thin trabeculi with a large void space filled with marrow. This means the mesh has to be fine enough to capture this delicate microstructure. Besides, handling of the transmission conditions on the interface between two constituent materials requires good estimation of the normal vectors to the interface.

The goal of this paper is to present the preliminary results on using the package we developed for computing the effective properties of two-phase composites. This package is named BIBLAP, which stands for Box-in-Box Laplace solver. BIBLAP is based on the package CBEM_LAP (Collocation BEM for the Laplace equation) developed by Dr. Sylvain Nintcheu Fata at Oak Ridge National Laboratory and can be downloaded from http://intetec.org

What we have in mind for the future is to use this numerical scheme to verify the theoretical results regarding how the effective properties at different frequencies of a composite can be use for retrieving statistical informations on the microstructure such as volume fraction and 2-point correlation functions. This process is referred to as ”dehomogenization” [CO08, Ou11].
The results presented here are for dielectric properties of composites made of two isotropic dielectric materials. The generalization to composites of two isotropic elastic materials is straightforward and is currently under development. The paper is organized as follows. Section 2 describes the mathematical formulation of the problem. In Section 3, the Laplace solve CBEM_LAP, which plays a central role in our scheme is explained and the iteration steps for solving the problem is presented. We include in Section 4 a problem whose solution can be analytically computed for verifying our numerical scheme. The method we used for generating the BEM mesh from $\mu$-CT scans is described in Section 5. Numerical results are given in Section 6, followed by Discussion/Future work in Section 7.

2 Formulation of the problem

Consider a two-phase composite that occupies the region $\Omega \in \mathbb{R}^3$. Assume the dielectric tensors of constituent materials are $\epsilon_1$ and $\epsilon_2$, occupying regions $\Omega_1$ and $\Omega_2$, respectively. The effective dielectric properties are defined as functionals of the solution to the boundary value problem

$$(\text{BVP}) \left\{ \begin{array}{ll}
\nabla \cdot \left[ (\epsilon_1 \chi_1(x) + \epsilon_2 \chi_2(x)) \nabla \phi \right] = 0, \\
\phi|_{\partial \Omega} = \sum_{j=1}^{d} E_0^j x_j
\end{array} \right. \quad (1)$$

where $\chi_i$ is the characteristic function of $\Omega_i$, $i = 1, 2$. We will refer to $\partial \Omega$ as “outer surface” and the set which separate $\Omega_1$ from $\Omega_2$ as “interface” and denote it by $I$. Note that $\Omega = \Omega_1 \cup I \cup \Omega_2$. We define $\phi_i := \phi|_{\Omega_i}$, $i = 1, 2$ and consider the transmission conditions

$$\left\{ \begin{array}{l}
\phi_1|_I = \phi_2|_I \\
(\epsilon_1 \nabla \phi_1) n_1|_I = -(\epsilon_2 \nabla \phi_2) n_2|_I
\end{array} \right. \quad (2)$$

(3)

where $n_i$ is the outer unit normal vector of $\partial \Omega_i$, $i = 1, 2$. For simplification, we assume the dielectric tensor of each phase to be isotropic, i.e. $(\epsilon_1)_{ij} = \epsilon_1 \delta_{ij}$, $(\epsilon_2)_{ij} = \epsilon_2 \delta_{ij}$, with $\epsilon_1$ and $\epsilon_2$
being two constant scalars and δ_{ij} Kronecker Delta. In this case, the boundary value problem and transmission conditions simplify to

\begin{align*}
\text{(BVP1)} \quad \left\{ \begin{array}{l}
\nabla \cdot \left[ (\epsilon_1 \chi_1(x) + \epsilon_2 \chi_2(x)) \nabla \phi \right] = 0, \\
\phi|_{\partial \Omega} = \sum_{j=1}^{d} E_0^j x_j 
\end{array} \right.
\end{align*} (4)

\begin{align*}
\begin{cases}
\phi_1|_I = \phi_2|_I \\
\epsilon_1 \frac{\partial \phi_1}{\partial \mathbf{n}_1}|_I = -\epsilon_2 \frac{\partial \phi_2}{\partial \mathbf{n}_2}|_I
\end{cases} \quad (5)
\end{align*}

\begin{align*}
\begin{cases}
\phi_1|_I = \phi_2|_I \\
\epsilon_1 \frac{\partial \phi_1}{\partial \mathbf{n}_1}|_I = -\epsilon_2 \frac{\partial \phi_2}{\partial \mathbf{n}_2}|_I
\end{cases} \quad (6)
\end{align*}

For fixed i, k, i, k = 1 \sim 3, the (i, k) component of the effective dielectric tensor \( \epsilon_{ik}^* \) is calculated by first solving (BVP1) with the constant vector \( E_0^j = e_k \), the k-th unit vector in Cartesian coordinates \( (e_k)_j = \delta_{jk} \) followed by evaluating the integral [Ber78]

\begin{align*}
\epsilon_{ik}^* \overset{\text{def}}{=} \frac{1}{|\Omega|} \int_{\Omega} \left[ \epsilon_1 \chi_1(x) + \epsilon_2 \chi_2(x) \right] \nabla \phi \, dx
\end{align*} (7)

Therefore, we can compute the tensor \( \epsilon^* \) a column of at a time. For example, to find the first column \( \epsilon_{1,1}^* \), we set \( E_0 = e^1 = <1, 0, 0>^t \), solve (BVP1) for \( \phi \), and

\begin{align*}
\epsilon_{1,1}^* = \frac{1}{|\Omega|} \int_{\Omega} \left[ \epsilon_1 \chi_1(x) + \epsilon_2 \chi_2(x) \right] \nabla \phi \, dx
\end{align*}

3 Numerical Scheme

Noting that in (BVP1) the solution \( \phi \) satisfies the Laplace equations inside each phase \( \Omega_1 \) and \( \Omega_2 \), we apply the Laplace equation solver CBEM_LAP with the iteration strategy based on the transmission conditions (5) and (6). We first give a rough description of the scheme used in CBEM_LAP. The iteration steps are described in section 3.2.

3.1 Laplace solver CEBM_LAP

To solve the Laplace equation

\begin{align*}
\nabla^2 u = 0
\end{align*} (8)

in a bounded domain \( \Omega \subset \mathbb{R}^3 \) with boundary \( \Gamma \), the Boundary Integral Equation (BIE) method uses the Green’s representation formula expressed as

\begin{align*}
\int_{\Gamma} G(x, y) t(y) \, d\Gamma_y - \int_{\Gamma} H(x, y) \cdot \mathbf{n}(y) u(y) \, d\Gamma_y = \begin{cases} u(x), & x \in \Omega \\
0, & x \in \mathbb{R}^3 \setminus \Omega \end{cases},
\end{align*} (9)

where \( \Omega = \Omega \cup \Gamma \), and the kernels \( G \) and \( H \) are given respectively by

\begin{align*}
G(x, y) = \frac{1}{4\pi \|x - y\|}, \quad H(x, y) = \frac{1}{4\pi \|x - y\|^3}, \quad x, y \in \mathbb{R}^3, \quad x \neq y.
\end{align*} (10)
In (9), \( n \) is the unit normal to \( \Gamma \) directed towards the exterior of \( \Omega \) and \( t = \partial u / \partial n \) is the flux associated with the potential \( u \). One can see from (9) that solving for \( u \) in \( \Omega \) reduces to finding \( u \) and \( t \) on \( \Gamma \). To this end, let \( x_\varepsilon \in \mathbb{R}^3 \setminus \overline{\Omega} \). The potential \( u \) and flux \( t \) on \( \Gamma \) can be determined by solving the singular BIE
\[
\lim_{x_\varepsilon \to x \in \Gamma} \left( \int_{\Gamma} G(x_\varepsilon, y) t(y) \, d\Gamma_y - \int_{\Gamma} H(x_\varepsilon, y) \cdot n(y) u(y) \, d\Gamma_y \right) = 0.
\] (11)

Note that in (11), \( x_\varepsilon \) approaches the boundary \( \Gamma \) from outside the domain \( \Omega \). The integral statement expressed in (11) corresponds to the so-called limit to the boundary approach.

### 3.1.1 Numerical approximation by constant element

To deal with (11), assume that (i) \( \Gamma = \bigcup \Gamma_j \) can be triangulated into closed and non-overlapping surface elements such that \( \Gamma_j \) is an open flat triangle, and (ii) on each element (triangle) \( u \) and \( t \) are constants. Let \( N \) be the total number of boundary elements on \( \Gamma \).

With these assumptions, a collocation approach for resolving (11) requires that the singular BIE be satisfied exactly at a set of collocation points \( \{x^i\}_{i=1}^N \) resting on \( \Gamma \). This requirement leads to a dense linear system of algebraic equations for boundary potential \( u \) and flux \( t \) as
\[
G \{ t \} = H \{ u \},
\] (12)
where \( \{ u \} \) and \( \{ t \} \) are respectively vectors containing potentials \( u^j \) and fluxes \( t^j \) on each boundary element \( \Gamma_j \) \((j = 1, 2, \ldots, N)\); components of influence matrices \( G \) and \( H \) can be written as
\[
G_{ij} = \lim_{x_\varepsilon \to x^j} \, g_j(x_\varepsilon), \quad H_{ij} = \lim_{x_\varepsilon \to x^j} \, h_j(x_\varepsilon), \quad x^j \in \Gamma,
\] (13)
with the single-layer potential \( g_j \) and double-layer potential \( h_j \) expressed as
\[
g_j(x) = \int_{\Gamma_j} G(x, y) \, d\Gamma_y, \quad h_j(x) = \int_{\Gamma_j} H(x, y) \cdot n(y) \, d\Gamma_y, \quad x \in \mathbb{R}^3.
\] (14)

For mathematical consistency, \( u^j \) and \( t^j \) \((j = 1, 2, \ldots, N)\) are assumed to be potentials and fluxes at the centroid of the boundary element \( \Gamma_j \) respectively. Upon prescribing the boundary conditions for the specific boundary-value problem associated with (8), the linear system (12) can be rearranged as
\[
A \{ z \} = \{ b \},
\] (15)
where \( \{ z \} \in \mathbb{R}^N \) is a vector containing unknown potentials or fluxes on \( \Gamma \), and \( \{ b \} \in \mathbb{R}^N \) is a vector whose entries are obtained from known boundary data. For an arbitrary source point \( x \in \mathbb{R}^3 \), surface potentials \( g_j(x) \) and \( h_j(x) \) are calculated exactly via recursive expressions provided in [Fat09].

The solution of the linear system (15) together with the specified boundary conditions complete the task of finding \( u \) and \( t \) on the entire boundary \( \Gamma \). Finally, the remaining exercise of computing
the potential \( u(x) \) at an arbitrary interior point \( x \) in \( \Omega \) can be accomplished by use of (9) and (14) as

\[
   u(x) = \sum_{j=1}^{N} t^j g_j(x) - \sum_{j=1}^{N} u^j h_j(x), \quad x \in \Omega.
\] (16)

Note that the numerical analysis presented herein is applicable to the Dirichlet, Neumann and Mixed boundary conditions.

3.2 Iteration Steps

Our strategy for solving (BVP1) is to solve the Laplace equation in \( \Omega_1 \) (inner problem) and in \( \Omega_2 \) (outer problem) using CBEM_LAP. The solutions communicate to each other through the transmission conditions. The iteration details are described below.

**Step 1** Make initial guess of \( \frac{\partial \phi_2}{\partial n_2} \) on \( I \),

\[
   \frac{\partial \phi_2}{\partial n_2} \bigg|_I = g_0
\] (17)

In terms of terminology in CBEM_LAP, the fluxes \( t \) is prescribed on \( I \) by \( g_0 \). In all the cases presented in this paper, we let \( g_0 \equiv 0 \).

**Step 2** Solve the mixed outer problem for \( \phi \) in \( \Omega_2 \).

\[
   \begin{align*}
   \triangle \phi_2 &= 0 \\
   \frac{\partial \phi_2}{\partial n_2} \bigg|_I &= g_0 \\
   \phi_2\big|_{\partial \Omega_2 \setminus I} &= E_0 \cdot x
   \end{align*}
\] (18) (19) (20)

The solution of this problem contains the *potential* on \( I \), i.e. \( \phi_2|_I \).

**Step 3** Use transmission condition (5) to find the Dirichlet data on \( I \). Solve the inner problem

\[
   \begin{align*}
   \triangle \phi_1 &= 0 \\
   \phi_1|_I &= \phi_2|_I \\
   \phi_1\big|_{\partial \Omega_1 \setminus I} &= E_0 \cdot x
   \end{align*}
\] (21) (22) (23)

The solution of this problem contains the *fluxes* on \( I \), i.e. \( \frac{\partial \phi_1}{\partial n_1} |_I \).
Step 4 Use transmission condition (6) to find the Neumann data on \( \Gamma \). Solve the mixed outer problem

\[
\begin{align*}
\phi_2 & = 0 \\
\frac{\partial \phi_2}{\partial n}_|_{\Gamma} & = \frac{\epsilon_1}{\epsilon_2} \frac{\partial \phi_1}{\partial n}_|_{\Gamma} \\
\phi_2|_{\partial \Omega_2 \setminus \Gamma} & = E^0 \cdot \mathbf{x}
\end{align*}
\]

(BVP-outer) \( (24) \) \( (25) \) \( (26) \)

The solution of this problem contains the potential on \( \Gamma \), i.e. \( \phi_2|_{\Gamma} \).

Step 5 Go to Step 3 and repeat until convergence criteria is satisfied.

The convergence criteria is based on convergence of the energy norm

\[
E^{(n)} \overset{\text{def}}{=} \int_{\Omega} \phi^{(n)} \nabla \cdot [\phi^{(n)} (\epsilon_1 \chi_1 + \epsilon_2 \chi_2) \nabla \phi^{(n)}] \, d\mathbf{x}
\]

\[
= \int_{\partial \Omega_1} \epsilon_1 \phi^{(n)} \frac{\partial \phi^{(n)}}{\partial n}_1 \, dS + \int_{\partial \Omega_2} \epsilon_2 \phi^{(n)} \frac{\partial \phi^{(n)}}{\partial n}_2 \, dS
\]

\( (27) \)

Let \( \epsilon^{*}_{n} \) denote the approximation of a column vector of \( \epsilon^{*} \) at the \( n \)-th iteration. Note the definition of \( \epsilon^{*} \) in (7) implies

\[
\epsilon^{*}_{n} \overset{\text{def}}{=} \frac{1}{|\Omega|} \int_{\Omega} (\epsilon_1 \chi_1 + \epsilon_2 \chi_2) \nabla \phi^{(n)} \, d\mathbf{x}
\]

\[
= \frac{1}{|\Omega|} \left( \epsilon_1 \int_{\partial \Omega_1} \phi^{(n)}_1 \mathbf{n}_1 \, dS + \epsilon_2 \int_{\partial \Omega_2} \phi^{(n)}_2 \mathbf{n}_2 \, dS \right)
\]

\( (28) \)

In each iteration of \( n = 1, 2, \cdots \), \( E^{(n)} \) and \( \epsilon^{*}_{n} \) are evaluated and recorded in the file \( \text{enorm.dat} \). The iteration process terminates when \( \Delta E^{(n)} \overset{\text{def}}{=} E^{(n+1)} - E^{(n)} \) is less than a preset tolerance. We’d like to remark that calculation of the energy norm and the effective property only requires information of potential and fluxes on the boundary elements, no value at interior points need to be evaluated for this purpose.

4 Comparison with analytic solution

We use analytic solution of the sphere-in-sphere problem for validating our scheme. For this problem, the computational domain is made of two concentric spheres with radius 1 and 2, centered at the origin, i.e.

\[
\Omega_1 := \{ \mathbf{x} : |\mathbf{x}| < 1 \} \quad \Omega_2 := \{ \mathbf{x} : 1 < |\mathbf{x}| < 2 \} \quad \Gamma := \{ \mathbf{x} : |\mathbf{x}| = 1 \}
\]

\( (29) \)

with transmission conditions imposed on \( \Gamma \) and Dirichlet boundary condition on \( \partial \Omega \):

\[
\phi|_{\partial \Omega} = \cos \theta
\]
where $0 \leq \theta \leq \pi$ is the azimuthal angle in spherical coordinates. By the separation of variable method, noting that $\cos \theta = P_1(\cos \theta)$ with $P_1$ being first order Legendre polynomial, it can be easily checked that

$$\phi_1 = (A + B)r \cos \theta$$

$$\phi_2 = (Ar + Br^{-2}) \cos \theta$$

with $A$ and $B$ being solution of the linear system in terms of material property $\epsilon_1$ and $\epsilon_2$

$$\begin{align*}
2A + \frac{B}{4} &= 1 \\
(\epsilon_1 - \epsilon_2)A + (\epsilon_1 + 2\epsilon_2)B &= 0
\end{align*}$$

The exact energy norm can be computed analytically. With $\epsilon_1 = 0.2$ for $\Omega_1$ and $\epsilon_2 = 1.96$ for $\Omega_2$, the exact energy norm is 0.41548387. With 768 elements on the inner sphere $r = 1$ and 1952 elements on the outer sphere, the numerical energy norm $E$ converges to 0.41437429 in 3 iterations.

<table>
<thead>
<tr>
<th>Number of Iteration</th>
<th>Energy Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.41488856</td>
</tr>
<tr>
<td>2</td>
<td>0.41437526</td>
</tr>
<tr>
<td>3</td>
<td>0.41437429</td>
</tr>
<tr>
<td>4</td>
<td>0.41437429</td>
</tr>
</tbody>
</table>

5 Mesh generation by MATLAB and CUBIT

The BEM mesh on the 3D structure of cancellous bone is constructed from digitized $\mu$-CT scans by a combination of MATLAB and CUBIT. Each slice is a logic file. As can be seen from the 3D structure in Figure 1(a), the outer boundary contains both trabeculae and marrow. In order to implement the iteration scheme proposed in previous sections, the meshing algorithm needs to be able to extract the boundary surface for each of the materials and group them separately from the elements lying on the interface. Two Matlab M-files are written for creating a 3D structure and generate BEM mesh for CUBIT. We describe the two files here.

imag.proc.prog.m This file reads the digitized $\mu$-CT slices (‘1’ for trabeculae and ‘0’ for marrow) by command `imread`. The purpose of this M-file is for visualizing the 3D structure constructed from $\mu$-CT scans and its mesh. It then constructs the 3D structure by using command `smooth3` to interpolate data. The interface between trabeculi and marrow is defined by the level set of value 0.5.

facet.m In this file, the triangular mesh for the interface between trabeculi and marrow is generated by command `isosurface` and that for the surface where marrow intersects the outer boundary is created by `isocap`. This mesh is stored as a facet file (.fac format, i.e. list of all vertices and list of all oriented triangles) and imported to CUBIT for mesh improvement and for constructing the mesh for the outer problem.
The meshing package CUBIT developed by the Sandia National Laboratories is used for improving the mesh is for constructing the outer mesh such as Figure 3(b). For CBEM LAP, the mesh has to be conformal in the sense that the mesh on the caps of the $\Omega_1$ has to be conformal with both the mesh on the interface $\mathcal{I}$ and the mesh on $\partial\Omega_2$. The function *imprint* of CUBIT was implemented to ensure this. After this, CUBIT meshing tool was applied to generate the outer mesh. We’d like to remark on the restriction of this approach. For smaller piece as shown in Figure 3, CUBIT is able to reconstruct the object from the facet file generated by Matlab and build a better mesh on the inner piece such as Figure 1(b) and Figure 3(a). The imprint command works only with geometry based format such as acis-sat format. This prompts the conversion of file from .fac to .stl and can be carried out in CUBIT. Then a MATLAB m-file CONVERT_stl_to_sat.m (can be downloaded from Mathwork website) is used to obtain the .sat files of the inner mesh to be imprinted on the outer bounding box.

However, a larger piece such as in Figure 5(a), CUBIT was not able to mesh the surface even though the input file was water-tight. In order to improve the mesh generated by *isosurface/isocap* in MATLAB, the Laplacian smoothing algorithm in CUBIT is used. See Figure 6 for comparison.

![Digitized µ scan](a) Digitized µ scan ![3D structure](b) 3D structure

Figure 2: From scans to 3D structure

### 6 Numerical Results

Here we present the numerical result for the sample in Figure 3 with $\epsilon_1 = 0.3$, $\epsilon_2 = 1.96$, 773 elements on interface $\mathcal{I}$, 261 elements on the trabeculi caps and 1542 elements on the marrow region of the outer box. The red elements are those with Dirichlet boundary condition prescribed on them. With $E^0 = e^1 = <1, 0, 0>$, the solution $\phi$ on the interface is shown in Figure 4 the first column of $e^*$ and energy norm $E$ converges as follows *(enorm.dat)*

<table>
<thead>
<tr>
<th>Dielectric Property and Energy Norm</th>
<th>Number of Iteration</th>
<th>$e_x$</th>
<th>$e_y$</th>
<th>$e_z$</th>
<th>Energy Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.66470733</td>
<td>-0.00703687</td>
<td>-0.00848460</td>
<td>1.74869252</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.67873116</td>
<td>-0.00325823</td>
<td>-0.00338472</td>
<td>1.74237449</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.67617441</td>
<td>-0.00435536</td>
<td>-0.00525362</td>
<td>1.74236949</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.67677172</td>
<td>-0.00403942</td>
<td>-0.00463539</td>
<td>1.74231926</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.67661096</td>
<td>-0.00412805</td>
<td>-0.00491999</td>
<td>1.74232197</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.67655807</td>
<td>-0.00410403</td>
<td>-0.00477119</td>
<td>1.74232102</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1.67664348</td>
<td>-0.00411023</td>
<td>-0.00479605</td>
<td>1.74232115</td>
<td></td>
</tr>
</tbody>
</table>
Figure 3: BEM mesh

7 Discussion and future work

In this paper, we propose to evaluate the effective property of bone composite from \(\mu\)-CT scans by using BEM. The advantage of this method is that only the interface and boundary needs to be meshed, rather than the entire volume. This seems to be suitable for the thin-plate like structure of trabeculi. Also, to solve a two-material problem, the transmission conditions imposed on the material interface require good approximation of normal vectors to the interface, which can be easily computed from the BEM mesh. The most expensive part in terms of memory use is in assembling matrices \(H_{ij}\) and \(G_{ij}\) in (13) for the inner problem and the outer problem; they are dense matrices with dimensions of the square of total number of elements in each problem. These matrices only need to constructed once because they do not depend on the solution. After they are assembled, the
iteration process involves solving one large linear system and pass the solution to the other linear system, utilizing the map between Dirichlet data and Neumann data. Currently, we solve these systems by using serial versions of LAPACK/BLAS routines for LU decomposition and matrix-vector multiplication, obviously not an optimal choice. For the problem of dehomogenization of composites, the cases with complex valued \( \epsilon_1 \) and \( \epsilon_2 \), hence complex-valued solutions need to be implemented; this means the memory use will become two-fold. To generalize this method to elasticity problem, the size of \( G_{ij} \) and \( H_{ij} \) will grow 9-fold. There are several ways to address the issues of memory use and speed-up of solution process of large linear systems. We are currently working on parallelizing the codes by using Intel MKL in order to speed up the process. Another focus of our current research is on decreasing the number of elements. As can be seen from Figure 5(a), the mesh generated by MATLAB isosurface/isocap is unnecessarily fine and CUBIT is not able to handle re-meshing of sample of this size. On the other hand, MATLAB isosurface/isocap is able to mesh pieces larger than this sample even though the mesh quality is not very good. One way to go around this problem is to use smoothing algorithm to improve the mesh quality and apply mesh coarsening algorithms to the smoothed mesh. We’d like to remark that there is no apparent way of tuning the mesh size generated by MATLAB isosurface/isocap commands and that CUBIT crashed whenever we tried to use the mesh coarsening function on the sample in Figure 5(a). With the current mesh shown in Figure 5(a) and double precision arithmetics, the memory use for solving the two linear systems is about 57G. The direction we’d go is to replace the constant elements with linear elements after we successfully reduce the number of triangles used in the BEM mesh; it is known that constant element BEM has sublinear convergence rate while linear-element
BEM has quadratic rate in terms of mesh size. Techniques for dense matrices such as multigrid methods and iterative solvers will also be implemented and incorporated into our BIBLAP package. The convergence rate of the Dirichlet-Neumann maps iteration process is also currently under investigation.

**Acknowledgement**

This work of MYO and CK is partially sponsored by NSF-DMS Math. Bio. Grant 0920852 and NIH CBER PILOT Grant 322 159. SNF’s work was supported by the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under contract No. DE-AC05-00OR22725 with UT-Battelle, LLC. The work of TKM and MX was sponsored by NSF-DMS Math. Bio. Grant 0920852 and the summer undergraduate research program between Joint Institute of Computational Sciences (JICS) of U. Tennessee at Knoxville, Oak Ridge National Laboratory (ORNL)
and Chinese University of Hong Kong. We gratefully acknowledge the generous help from Drs. Richard Achibald (ORNL) and Kwai Wong (JICS). The μ-CT scans of human cancellous bone are provided by Prof. Luis Cardoso of Department of Biomedical Engineering, The City College of The City University of New York.

A Implementation

BIBLAP is written in Fortran 90. The main program is lapEx.f90. To implement the BIBLAP code, just use the name of the desired bndata file in the field 'InputFileName' in lapEx.f90, followed by commands 'make' and './lapEx'.

A.1 Format of Input Data: bndata.dat

The input file "bndata.dat" has the following data structure

Line 1: Number of components (box) This is the number of connected domains of the given 3d structure. For example, this number equals 2 for a composite of material A with a sphere filled with material B embedding inside.

Line 2: $\epsilon_1 (e(1))$ Dielectric property of material in $\Omega_1$

Line 3: $\epsilon_2 (e(2))$ Dielectric property of material in $\Omega_2$

Line 4: Number of Nodes (nnodes) Number of vertices of boundary element. We denote it by $n$.

Line 5: Number of elements on the interface (nelem(1)) Interface refers to material boundary that is not on the outer surface. If there are more than two connected regions, the number of interface elements will be listed for each region. We denote the number by $i$

Line 6: Number of elements on the outer boundary $\partial\Omega$ and $\partial\Omega_1 (n_{out,1})$ (nelem(1))

Line 7: Number of elements on the outer boundary $\partial\Omega$ and $\partial\Omega_2 (n_{out,2})$ (nelem(2))

List of all vertices This block contains $n$ lines with format $x$ $y$ $z$, the Cartesian coordinates of each vertex. The vertex defined in the first line in the list has label 1. All other vertices are labelled in the same fashion.

List of elements on interface This block contains all elements that do not lie on the outer boundary. The format is $v_1$ $v_2$ $v_3$ in which the labels $v_i$, $i = 1, 2, 3$, is arranged in such a way that the normal to the element, determined by the right-hand-rule, points inward.

List of elements on boundary This block contains all elements that lie on the outer boundary. The format is $v_1$ $v_2$ $v_3$ $g$ $type$ in which the labels $v_i$, $i = 1, 2, 3$ is arranged in such a way that the normal to the element, determined by the right-hand-rule, points outward, $a$ the given
boundary value. For \( type = 0 \), \( g \) is the Dirichlet boundary value. For \( type = 1 \), \( g \) is the Neumann boundary value.

Once the iteration converges, the values of interior points inside each region can be evaluated by Green’s Identity. The input file `indata.dat` specifies at which interior points we would like the valued of solution to be evaluated. The data structure is as followed

**Line 1: Number of interior points in** \( \Omega_1 \) **Denote this by** \( Nip_1 \).
**Line 2: Number of interior points in** \( \Omega_2 \) **Denote this by** \( Nip_2 \).

**List of vertices of points in** \( \Omega_1 \) **There are** \( Nip_1 \) **lines of data** \( x \ y \ z \) **for Cartesian coordinates.**
**List of vertices of points in** \( \Omega_2 \) **There are** \( Nip_2 \) **lines of data** \( x \ y \ z \) **for Cartesian coordinates.**

## B Code structure

The main program is lapEx.f90. It calls CBEM_BIBLAP (biblap.f90) and writes the potential \( \phi \) and fluxes \( \frac{\partial \phi}{\partial n} \) for elements on the interface to the output files `potl.dat` and `flux.dat`.

CBEM_BIBLAP calls three subroutines: CBEM_INPUT, CBEM_INIT and CBEM_ASSEMHG.

The first one reads into the mesh data and the boundary conditions from the input file `bndata.dat`.

The second one computes the centroid of each triangle. The third routine assembles matrices \( H_{ij} \) and \( G_{ij} \) for the inner problem and the outer problem.

Then CBEMITER is called for the iterations between Dirichlet data and Neumann data on the interface \( I \). CBEM_LRHS rearranges the linear systems so all the unknown are on the left-hand-side of the linear system. CBEMITER calls CBEMITER_OTI to do the “outer-to-inner” iteration until the user specified tolerance on energy norm is achieved. It also write the effective properties and energy for each iteration into the file `enorm.dat`.

## C variable names used in the programs

<table>
<thead>
<tr>
<th>variable</th>
<th>definition</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>belem</td>
<td>nbelem(1)+nbelem(2)</td>
<td>number of elements on ( \partial \Omega )</td>
</tr>
<tr>
<td>telem</td>
<td>belem+telem</td>
<td>number of elements on ( \partial \Omega \cup I )</td>
</tr>
<tr>
<td>tielem</td>
<td>telem-nbelem(2)</td>
<td>number of elements on material 1</td>
</tr>
<tr>
<td>toelem</td>
<td>telem-nbelem(1)</td>
<td>number of elements on material 2</td>
</tr>
</tbody>
</table>

## References


