A 149 Line Homogenization Code for Three-Dimensional Cellular Materials Written in matlab

Guoying Dong, Yunlong Tang, Yaoyao Fiona Zhao

A 149 Line Homogenization Code for 3D Cellular Materials Written in Matlab

Guoying Dong, Yunlong Tang, Yaoyao Fiona Zhao*

Department of Mechanical Engineering

McGill University

817 Sherbrooke West, Montreal, QC H3A 0C3, Canada

Abstract

Cellular architectures are promising in a variety of engineering applications due to attractive material properties. Additive Manufacturing (AM) has reduced the difficulty in the fabrication of three dimensional (3D) cellular materials. In this paper, the numerical homogenization method for 3D cellular materials is provided based on a short, self-contained Matlab code. It is an educational description that shows how the homogenized constitutive matrix is computed by a voxel model with one material to be void and another material to be solid. A voxel generation algorithm is proposed to generate the voxel model easily by the wireframe scripts of unit cell topologies. The format of the wireframe script is defined so that the topology can be customized. The homogenization code is then extended to multi-material cellular structures and thermal conductivity problems. The result of the numerical homogenization shows that different topologies exhibit anisotropic elastic properties to a different extent. It is also found that the anisotropy of cellular materials can be controlled by adjusting the combination of materials.

Keywords: Cellular material; Lattice structure; Matlab; Numerical homogenization

1. Introduction

Cellular materials are evolved from nature to achieve high stiffness and strength with low density. Humankind had manufactured cellular architectures such as foams and honeycombs to mimic natural cellular materials decades ago. However, they were less sophisticated than natural cellular materials due to the limitation of manufacturing techniques. Recently, with the development of Additive Manufacturing (AM), cellular materials have been fabricated with more complicated structures and novel architectures [1]. The lattice structure, which is a type of cellular material with interconnected structs and nodes in a three-dimensional (3D) space, has been successfully manufactured by several types of AM techniques. Due to controllable properties, periodic lattice structures with different topologies and relative densities are widely used in engineering applications for high stiffness-to-weight ratio [2, 3], energy absorption [4], thermal management [5], etc.

* Corresponding Author, Email: yaoyao.zhao@mcgill.ca

The complex structure of cellular materials not only makes it difficult for fabrications but also leads to high computational cost for the simulation of their property. Because of small features inside the material, Finite Element Analysis (FEA) requires fine mesh to model cellular materials. If the structure is constructed by lots of unit cells, the computational cost is extremely high. To reduce the complexity of simulation, homogenization methods can be used to compute the macroscopic cellular material properties. Homogenization refers to a method that can replace the composite with an equivalent material model in order to resolve the difficulty in the analysis of boundary value problem with high heterogeneities [6]. In this method, the local problem is solved to obtain the homogenized material property based on unit cell. Then the overall problem is computed by assigning a solid material with the homogenized property to substitute the periodic structure.

Asymptotic homogenization has been widely used to characterize the mechanical properties of cellular materials for several decades. It assumes that each field quantity depends on two different scales, macroscopic level, and microscopic level. The basic theory has been detailed covered in some literature [7, 8]. Bendsoe and Kikuchi [9] used this method to get the effective elastic modulus of a unit cell. It was also implemented in a design procedure to find the optimal topology of a unit cell under a certain boundary condition. Arabnejad and Pasini [10] investigated the mechanical properties of 6 different lattice topologies for a whole range of relative density by asymptotic homogenization. The homogenization equation is discretized and solved via FEA. This is often referred as numerical homogenization. Hassani and Hinton [6, 11] reviewed the methods of numerical homogenization to solve general boundary value problems with periodic boundary conditions. Andreassen and Andreasen [12] summarized the formula to compute the homogenized elasticity tensor E^H_{ijkl} and used numerical methods to solve the two-dimensional (2D) homogenization problem by Matlab. E^H_{ijkl} can be computed as:

$$E^H_{ijkl} = \frac{1}{|\Omega|}\int_\Omega E_{pqrs}\left(\varepsilon^{0(ij)}_{pq} - \varepsilon^{(ij)}_{pq}\right)\left(\varepsilon^{0(ij)}_{rs} - \varepsilon^{(ij)}_{rs}\right) d\Omega \tag{1}$$

where E_{pqrs} is the locally varying elasticity tensor, $|\Omega|$ is the volume of the unit cell, $\varepsilon^{0(ij)}_{pq}$ are prescribed macroscopic strain fields, while the locally varying strain fields $\varepsilon^{(ij)}_{pq}$ are defined as:

$$\varepsilon^{(ij)}_{pq} = \varepsilon_{pq}(\mathcal{X}^{ij}) = \frac{1}{2}(\mathcal{X}^{ij}_{p,q} + \mathcal{X}^{ij}_{q,p}) \tag{2}$$

where $\mathcal{X}^{kl}$ is the displacement fields which can be found by solving the following equation:

$$\int_\Omega E_{ijpq}\varepsilon_{ij}(v)\varepsilon_{pq}(\mathcal{X}^{kl})d\Omega = \int_\Omega E_{ijpq}\varepsilon_{ij}(v)\varepsilon^{0(kl)}_{pq}dV \;\; \forall v \in \Omega \tag{3}$$

where v is a virtual displacement field. The code provided in [12] can solve Eq.(3) and computes the homogenized elasticity tensor for a 2D composite materials. It can also be used to solve 2D cellular material by assigning an extremely soft second material.

However, as AM has greatly relieved the manufacturing constraints in the fabrication of three-dimensional (3D) cellular materials, the interest in cellular materials is shifting from 2D to 3D. 3D cellular materials have more design freedoms so that the mechanical performance can be further optimized than 2D cellular materials. The implementation of 2D problem detailed in [12] is not enough for the need of designers. Nevertheless, the difficulty in the analysis of cellular materials is increased by adding one dimension. The existing homogenization implementations for 3D problem mathematically complex, which hinder designers without strong mathematical backgrounds from utilizing those approaches. Therefore, it is necessary to have a simple, self-contained code that can analyze the mechanical properties of 3D cellular materials.

In this paper, an educational Matlab code is provided in Appendix A, which can compute the homogenized constitutive matrix for 3D cellular materials. It removes the barrier for designers who want to use the numerical homogenization. It also gives instructions for those who want to understand how the asymptotic homogenization method is implemented. The implementation of the homogenization code is restricted to analyzing the linear elastic property of the cellular material. The input of this code is a 3D voxel model with isotropic material properties for each element. The voxel model can be obtained by the voxelization of the geometrical model. Furthermore, to reduce the complex procedure of geometrical modeling and voxelization of 3D cellular materials, a simple and concise Matlab code provided in Appendix A can directly generate the voxel model of predefined topologies. In Section 2, the voxelization code is introduced. The format of the predefined topology will be explained, which is the input of the voxelization code. In Section 3, the implementation of numerical homogenization will be illustrated. Section 4 discusses the result obtained from the homogenization code. In Section 5, some extension of the Matlab code will be introduced to broaden the application. Finally, this paper is wrapped up in Section 6 with some future work.

2.Voxel Generation

Numerical homogenization requires a finite element model of the lattice structure. That means a geometrical model should be generated at first. Then, the geometrical model is discretized by 3D solid elements. These procedures are tedious and time-consuming. Another way is to use a voxel model to represent the lattice structure. This still requires the generation of the geometrical model. To facilitate the procedure of getting the homogenization model of lattice structures, a voxelization algorithm based on the lattice wireframe has been proposed. In this algorithm, the voxels are directly generated from the lattice wireframe, which skips the generation of the geometrical model. This voxel model can be used as an input to the numerical homogenization code.

2.1 Read the information of lattice wireframe (lines 50-72)

To generate the voxel model, the information of the lattice topology wireframe needs to be pre-defined. The wireframe contains two parts, GRID, and STRUT. GRID defines the coordinates of all the node in the wireframe. Its format is shown in Table A1 in Appendix C, where ‘ID’ is the grid index; ‘x’ is the x-coordinate; ‘y’ is the y-coordinate and ‘z’ is

the z-coordinate. STRUT defines the start grid index and end grid index. Its format is shown in Table A2 in Appendix A, where 'ID' is the strut index; 'Start' is the grid index of the start node; 'End' is the grid index of the end node. These grid indices refer to the 'ID' in the GRID. Each field of GRID and STRUT take 8 spaces. The wireframe script can be downloaded from the link in Appendix A. Furthermore, users can define new wireframe script based on the format shown in Table A1 and Table A2 in Appendix A. The visualization of these topologies is shown in Figure 1.

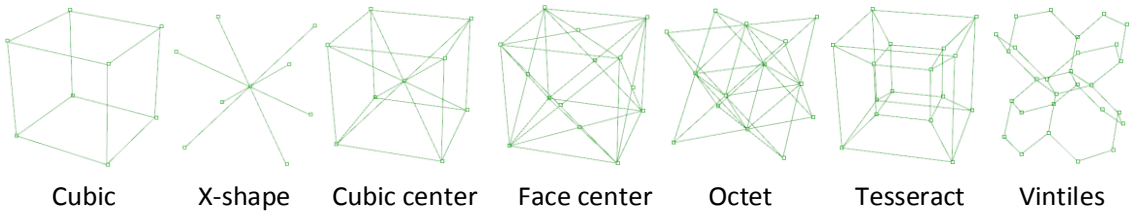

Figure 1 Wireframe of seven topologies.

The next step is to use the information of the wireframe to generate the voxel model. The Matlab code provided in Appendix A named 'GenerateVoxel.m' computes the voxel model. The first argument (n) is the resolution of the voxel, which means the number of voxels along x, y and z-axis. The second argument (address) is the address of the wireframe file. The third argument (radius) is the radius of the lattice strut. There are two outputs. The first output 'voxel' is a 3D logical matrix, in which '0' means the voxel in this place does not have material; '1' means this voxel has material. The second output measures the relative density of the lattice structure.

2.2 Generate the voxel in the design space (lines 7–21)

The design space in this research is dimensionless, which is inside a cube with a unit length. By dividing the unit length with the number of voxel on each axis, the size of each voxel can be obtained. To store the information of each voxel, the index and the center coordinates are calculated and saved. The index is used to identify the position of the voxel in the 3D logical matrix. As shown in Figure 2(a), the voxels are all generated in the first octant of the 3D Cartesian coordinate and they start from the origin. Figure 2(b) shows voxel in each layer along the z-axis. The gray voxel means it has material while the white color means there is no material in this voxel. The 3D logical matrix that represents this voxel model is shown in Figure 2(c). The coordinates measure the center position of each voxel which is used to calculate the distance between the voxel and the lattice strut. Once the number of voxels along each axis is determined, the size, index, and position of each voxel can be generated.

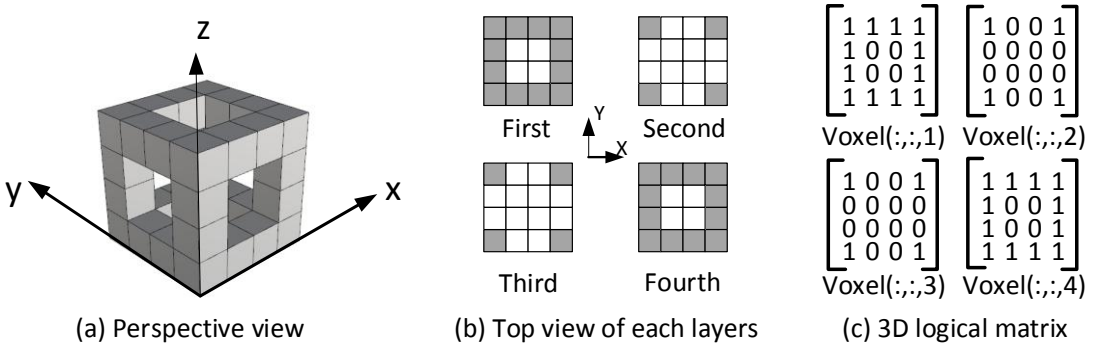

Figure 2 An example of the voxel model of a lattice structure with cubic topology.

2.3 Determine if the voxel has material (lines 23-46)

After the generation of voxels, the next step is to determine if the voxel has material. The criterion is to measure the minimum distance from the center of the voxel to each line segment of the wireframe. If the distance is less than the radius of the lattice strut, that means this voxel is inside the lattice. Therefore, the value of this voxel will be set to '1', which means it has material. Otherwise, the value of this voxel will be set to '0', which means it is empty. When calculating the minimum distance, two cases should be considered as shown in Figure 3. Two planes, Plane A and Plane B are defined to distinguish Case (1) and Case (2). Plane A and Plane B are located at the start point and the end point of the line segment, respectively. Both planes are perpendicular to the line segment. In Case (1), the center of the voxel is located inside the area between Plane A and Plane B. In this case, the minimum distance is from the center point to the line. However, if the center of the voxel is located outside the area between Plane A and Plane B, which is Case (2), the minimum distance is from the center point to the start point or the end point of the line segment. If the center point is exactly on Plane A or Plane B, it is included in Case (2).

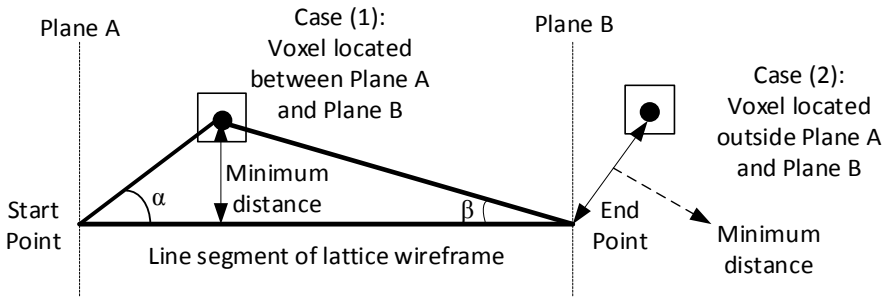

Figure 3 Two cases when calculating the minimum distance from voxel center to the line segment of lattice wireframe.

To decide whether a voxel is in Case (1) or Case (2), two angles, α and β are defined in Figure 3. If both α and β are acute angles, this voxel is in Case (1). Otherwise, it is in Case (2). Therefore, the first step to determine whether a voxel has material or not is to

calculate α and β. Then, the minimum distance between the voxel center and the ling segment is determined. The last step is to compare the minimum distance with the radius of lattice strut. If the minimum distance is smaller than the radius, the index of this voxel will be set to '1'. Otherwise, it is set to '0'. After calculating all the voxels, the 3D logical matrix represents the voxel model of lattice structures is obtained. Examples of voxel models for seven topologies of lattice structures are shown in Figure 4. These models can be directly used for the proposed homogenization method in Matlab.

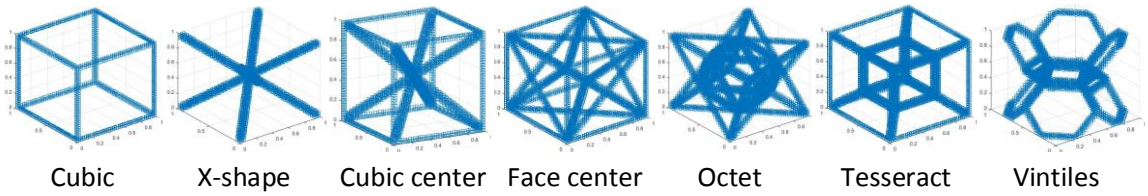

Figure 4 Voxel models of seven topologies.

3.Matlab Implementation for 3D Homogenization

In the implementation of numerical homogenization by Matlab, there are three inputs (lx), (ly) and (lz) indicate the length of the unit cell along x, y, and z-axis, respectively. The input arguments (lambda) and (mu) are the Lame's first and second parameter for the solid material which the lattice structure is made of. The last input argument (voxel) is the material indicator matrix, which specifies whether a finite element contains materials ($\text{voxel}_e = 1$) or not ($\text{voxel}_e = 0$). This matrix can be obtained by the method proposed in Section 2. It requires the finite elements with materials to be connected. If they are not connected, a two materials model with a very soft second material should be used, which will be introduced in Section 5.

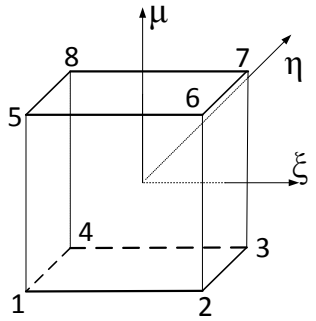

Figure 5 Local node numbers and the natural coordinate of hexahedron element.

3.1 Stiffness matrix and load vectors of hexahedron (lines 99-149)

The type of element used in this homogenization is an 8-node hexahedron element [13]. The local node number of hexahedron element is shown in Figure 5. The natural coordinates for it are called ξ, η, and μ. These coordinates are also shown in Figure 5. The shape function can be summarized as:

$$N_i^e = \frac{1}{8}(1 + \xi\xi_i)\,(1 + \eta\eta_i)(1 + \mu\mu_i) \tag{4}$$

where ξ_i, η_i and μ_i are the natural coordinates of the i^{th} node. The element stiffness matrix is given by the standard formula:

$$\boldsymbol{K}^{(e)} = \int_{\Omega^{(e)}} \boldsymbol{B}^T \boldsymbol{C}^{(e)} \boldsymbol{B} d\Omega^{(e)} \tag{5}$$

where **B** is the strain-displacement matrix, **C** is the constitutive matrix for the element, which is constant over the element. The material is assumed to be isotropic over each element. The constitutive matrix can be expressed by Lamé's parameters as:

$$\boldsymbol{C}^{(e)} = \lambda^{(e)} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \mu^{(e)} \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where $\lambda^{(e)}$ and $\mu^{(e)}$ are the first parameter and second parameter of Lamé's constants. They can be computed by the following formula [14]:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)},\ \mu = \frac{E}{2(1+\nu)} \tag{7}$$

where E is Young's modulus and ν is the Poisson's ratio. By using Lamé's parameters, the element stiffness matrix can be split into two parts as

$$\boldsymbol{K}^{(e)} = \lambda^{(e)} \boldsymbol{k}_\lambda + \mu^{(e)} \boldsymbol{k}_\mu \tag{8}$$

which will simplify the extension to two or more materials. However, it should be noted that Eq. (8) is restricted to isotropic materials. The load vector is calculated by the macroscopic volumetric straining

$$\boldsymbol{f}^i_{(e)} = \int_{\Omega^{(e)}} \boldsymbol{B}^T \boldsymbol{C}^{(e)} \boldsymbol{\varepsilon}^i d\Omega^{(e)} \tag{9}$$

where the macroscopic strains are

$$\begin{aligned} &\boldsymbol{\varepsilon}^1 = (1,0,0,0,0,0)^T, \boldsymbol{\varepsilon}^2 = (0,1,0,0,0,0)^T, \boldsymbol{\varepsilon}^3 = (0,0,1,0,0,0)^T \\ &\boldsymbol{\varepsilon}^4 = (0,0,0,1,0,0)^T, \boldsymbol{\varepsilon}^5 = (0,0,0,0,1,0)^T, \boldsymbol{\varepsilon}^6 = (0,0,0,0,0,1)^T \end{aligned} \tag{10}$$

Similarly, the load vector $\boldsymbol{f}^i_{(e)}$ can also be split into two parts as

$$\boldsymbol{f}^i_{(e)} = \lambda^{(e)} \boldsymbol{f}^i_\lambda + \mu^{(e)} \boldsymbol{f}^i_\mu \tag{11}$$

The split parts of element stiffness matrix and load vector are calculated by calling the function 'hexahedron', which will output $\boldsymbol{k}_\lambda$, $\boldsymbol{k}_\mu$, $\boldsymbol{f}^i_\lambda$ and $\boldsymbol{f}^i_\mu$.

3.2 Periodic boundary conditions (lines 22-39)

The structure of the mesh to model the unit cell is illustrated in Figure 6. There are 8 elements in this example. The ID of each element is shown in Figure 6(a). The nodes can be divided into three layers. Firstly, the node number without the periodic boundary

condition for each node are shown in Figure 6(b). As for each element, the 24 degrees of freedom are saved in 'edof'. Each row of 'edof' is for one element and the bold number represent the element Id. The sequences of degrees of freedoms in each row of 'edof' are arranged according to the local degrees of freedoms. To demonstrate it, the comparison between global node numbers and local node numbers for the first element is shown in Figure 6(c). Secondly, lines 24-32 generate the periodic boundary condition by mirroring the back, left and top border in sequence and the periodic degrees of freedom are shown in Figure 6(d). The relationship between the non-periodic and periodic degrees of freedom is obtained in lines 34-37 and stored in 'dofVector'. Finally, line 38 updates the degrees of freedom of each element from non-periodic to the periodic condition.

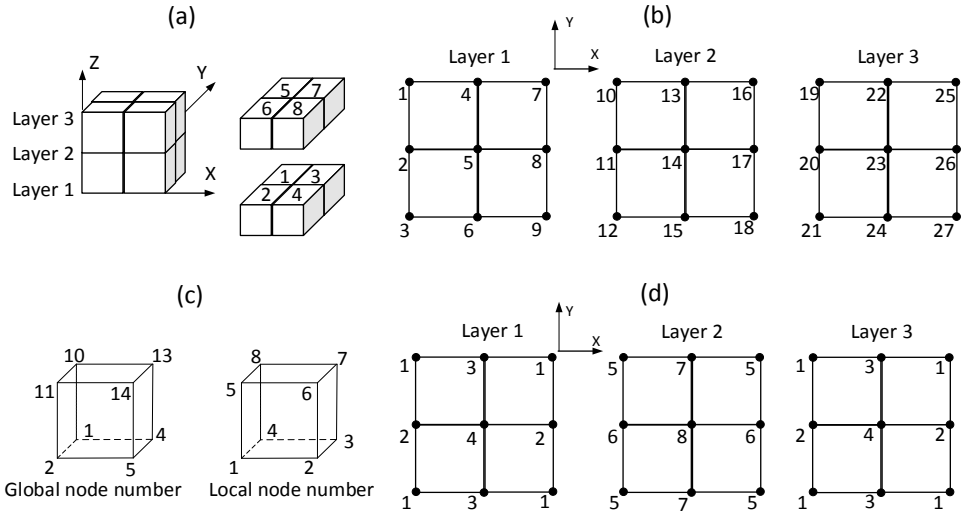

Figure 6 An example of hexahedron mesh used to model the unit cell, (a) the ID of each element, (b) the non-periodic node number, (c) global node number of element 1 compared to local node number, (d) periodic node numbers.

3.3 Global stiffness matrix and load vector (lines 42-55)

Before assembly of the global stiffness matrix, the indices of the non-zero entries in the global stiffness matrix are created in two vectors by line 42 and 43. These two vectors are used to denote the position of the entry in the global stiffness matrix. For single material lattice structures, there are two types of elements. One type is the solid part that will be assigned to material properties. The other type is void in the lattice structure, where the Lamé's parameters are zero. Line 45 creates the matrices with material properties $\lambda^{(e)}$ and $\mu^{(e)}$ for each element by the indicator matrix 'voxel'. These matrices are then used to compute the stiffness matrix for each element by multiplying the material properties $\lambda^{(e)}$ and $\mu^{(e)}$ with $\boldsymbol{k}_{\lambda}$ and $\boldsymbol{k}_{\mu}$ in line 47. After obtaining the stiffness matrix for each element, the sparse global stiffness matrix is assembled by indices vectors and element stiffness matrix in line 48. Due to rounding errors, the global stiffness matrix may not be exactly symmetric. Line 49 can eliminate the error to make the matrix symmetric so that the speed of the solver can be accelerated. For 3D problems, the computational cost could be

a problem. Therefore, the symmetry of global stiffness matrix is important in solving the large system of linear equations.

In the assembly of load vectors, the indices vectors 'iF' and 'jF' are generated in lines 51-52 to denote the row and column of non-zero entries of the global load vector. For the element with solid materials, the local load vector is calculated by multiplying the material properties $\lambda^{(e)}$ and $\mu^{(e)}$ with $\boldsymbol{f}_\lambda$ and $\boldsymbol{f}_\mu$ in line 54. For the elements that are void, the load vectors are zero. Finally, the global load vector is assembled in line 55.

3.4 Solution by PCG method (lines 59-66)

Compared with two-dimensional problems, the 3D problem considerably increases the computational cost. Using $\boldsymbol{\chi}^i = \mathbf{K}\backslash\boldsymbol{f}^i$ in Matlab to solve global equation cost a lot of memory if the number of elements is large. Preconditioned conjugate gradients (PCG) method is used to solve the system of linear equations. Even though it not as fast as using $\boldsymbol{\chi}^i = \mathbf{K}\backslash\boldsymbol{f}^i$ in some cases, it can save lots of memory. The comparison between these two methods are shown in Table 1. It is found that both methods have advantages in different cases. In this code, PCG is the default method. However, $\boldsymbol{\chi}^i = \mathbf{K}\backslash\boldsymbol{f}^i$ is also an optional method which is deactivated in lines 65-66. If users need to use the direct method, they can simply activate lines 65-66 and deactivated lines 60-64. It should be noted that when using PCG method, the matrix should be symmetric and positive definite. The global stiffness matrix satisfies these requirements so that it can be efficiently solved by PCG method. To accelerate the speed of computation, only the degrees of freedom of element with solid materials are activated in line 58. The void element without material properties will not be considered in the computation. But this procedure requires these active elements to be connected.

Table 1 The comparison between the PCG method and the direct method

			PCG method		$\boldsymbol{\chi}^i = \mathbf{K}\backslash\boldsymbol{f}^i$	
Resolution	Topology	Relative Density	Memory (MB)	Time (s)	Memory (MB)	Time (s)
50	Grid	30%	3250	46	6300	15
		50%	3690	77	19600	127

To accelerate the convergence of PCG method, a proper preconditioner should be obtained at first. The function 'ichol' with only one input argument is used to construct an incomplete Cholesky factorization with zero fill. The output can be used as a preconditioner in PCG method. Furthermore, the PCG method should specify the tolerance and the maximum number of iteration. If the relative residual is less than the tolerance or if the number of iteration exceeds the maximum value, the PCG method will stop and return a result with the minimum relative residual. In this paper, the default tolerance is 10^{-10} and the maximum number of iteration is 300. These two parameters can be adjusted by users. For instance, a lower tolerance and a larger number of iteration will return a more accurate result but it takes more time to run the PCG method. In lines 61-64, PCG is run six times to get the displacement field under six loading conditions.

3.5 Homogenization (lines 69-96)

The homogenization procedure is similar to the procedure for two-dimensional problems proposed by Andreassen and Andreasen [12]. Firstly, the element displacements corresponding to unit strain cases are found in lines 69-82. Elements' nodal displacement are solved under the uniform strains in Eq. (3). The difference between the 2D and 3D problem is the number of constrained degrees of freedom. The 2D problem only needs to constrain 3 degrees of freedom, while the 3D problem needs to constrain 6 degrees of freedom so that the element stiffness matrix is not singular. Line 72 explains which degrees of freedom are constrained to calculate the elementary displacements. Because all the elements are equivalent, the resulting displacements are the same for each element.

The next step is to use the element displacements and the global displacement field to calculate the homogenized constitutive matrix C^H. The entries in C^H can be calculated by the following equation:

$$C_{ij}^H = \frac{1}{|\Omega|}\sum_{(e)} \int_{\Omega^{(e)}} (\chi_{(e)}^{0(i)} - \chi_{(e)}^{(i)})^T k_e (\chi_{(e)}^{0(j)} - \chi_{(e)}^{(j)}) d\Omega^{(e)} \tag{12}$$

where $\chi_{(e)}^{0(i)}$ is the element displacements corresponding to the ith unit strain in Eq. (10). $\chi_{(e)}^{(i)}$ is the displacement field obtained from the global stiffness equation. Ω is the total volume of the unit cell of the lattice structure. The summation is computed in line 85-96. The λ part and μ part are computed separately and these two parts are multiplied with $\lambda^{(e)}$ and $\mu^{(e)}$. Then, the result for all the element are summed together to get C_{ij}^H. After the iteration for all six unit strains, the 6×6 homogenized constitutive matrix C^H is obtained.

4. Results and Discussion

4.1 Homogenized constitutive matrix

In this sub-section, an example is used to illustrate how to use the proposed method to get the homogenized property of lattice structures. To run the homogenization code, the voxel model has to be generated first. The lattice structure with the cubic topology is used as an example. The strut radius of the lattice structure is set to be 0.1; the path of the strut file is saved in variable 'address'; the resolution is set to be 50. Run the file 'GenerateVoxel.m' that can be downloaded in Appendix A as:

```
[voxel,Density] = GenerateVoxel(50,address,0.1);
```

where 'voxel' is a logical matrix that indicates the element in the lattice structure has materials or not. 'Density' estimates the relative density of this lattice structure, which is 8.44% in this case. The next step is to determine material property. It is assumed that the Young's modulus of the material is 200GPa and the Poisson's is 0.3. From Eq.(7), the Lame's first and second parameters are calculated as 115.4 and 76.9, respectively. The size of the unit cell is 1 in x, y, and z-direction. Run the file 'homo3D.m' that can be downloaded in Appendix A as:

```
CH = homo3D(1,1,1,115.4,79.6,voxel);
```

The homogenized constitutive matrix is

$$C^H = \begin{bmatrix} 7.06 & 0.37 & 0.37 & 0 & 0 & 0 \\ 0.37 & 7.06 & 0.37 & 0 & 0 & 0 \\ 0.37 & 0.37 & 7.06 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.14 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.14 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.14 \end{bmatrix} \quad (13)$$

From the homogenized constitutive matrix, it can be concluded that this lattice structure has orthotropic material properties. Because of symmetry, the property along x, y and z directions are the same. To get the Young's modulus along the axial direction, the inverse of the constitutive matrix should be calculated, which is the homogenized compliance matrix S^H. In the compliance matrix, it has:

$$S_{11}^H = \frac{1}{E_x},\ S_{22}^H = \frac{1}{E_y},\ S_{33}^H = \frac{1}{E_z} \quad (14)$$

where E_x, E_y and E_z are the Young's modulus along x, y, and z-direction. From Eq. (14), it can be computed that $E_x = E_y = E_z = 7.0225GPa$. Because the material property is anisotropic, the Young's modulus is different along different orientations. To get the Young's modulus in other directions, the constitutive matrix has to be transformed. Firstly, the 6×6 constitutive matrix is rewritten as a fourth-order stiffness tensor. Then the tensor is transformed with a rotation matrix. The transformed tensor can be used to get elastic modulus along the new direction. Through this procedure, directional properties of a cellular can be obtained.

The proposed method can also be used to homogenize the periodic cellular materials with the randomized unit cell as shown in Figure 7. The length of the randomized unit cell is 2 in x-axis and 1 in y, z-axis. Therefore, the input of the unit cell length in the code should correspond to designed length. The dimension of the voxel model of the unit cell is $120 \times 60 \times 60$ as shown in Figure 7(c). Run the file 'homo3D.m' that can be downloaded in Appendix A as:

```
CH = homo3D(2,1,1,115.4,79.6,voxel);
```

The result of this type of cellular material is $E_x = 2.2\ Gpa, E_y = 3.2\ Gpa, E_z = 3.1\ Gpa$.

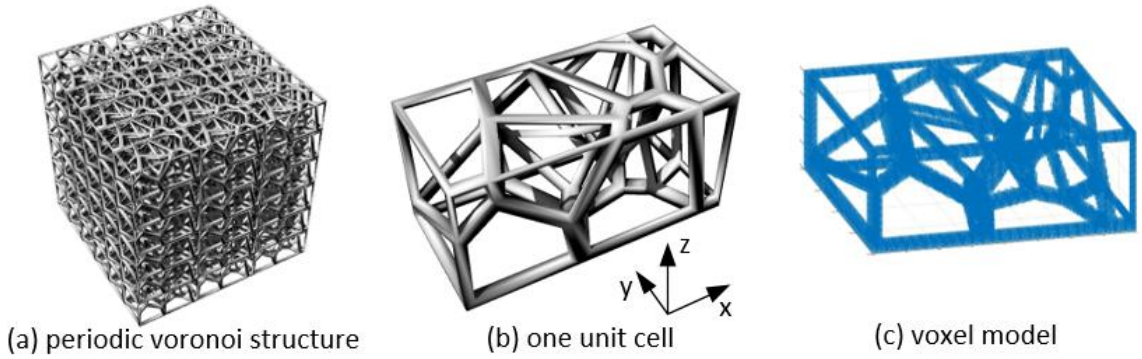

Figure 7 An example of periodic cellular materials with randomized unit cell

4.2 Effective Young's modulus surface

To further illustrate how the homogenized constitutive matrix C^H can be used to predict the elastic properties of lattice structure in different directions. Seven types of topologies listed in Figure 1 have been analyzed by the proposed Matlab code. The bulk material properties are the same as those used in Section 4.1. For each topology, the voxel model of lattice structure is generated with three different relative density: 10%, 30% and 50%. Because the total strut lengths in different topologies are different, the strut radius need to be determined case by case. Table 2 shows the strut radius for each topology with different relative densities. The strut radius is dimensionless and the length of the unit cell is regarded as 1.

Table 2 Strut radius of lattice structures with different topologies

	Cubic	X-shape	Cubic center	Face center	Octet	Tesseract	Vintiles
10%	0.113	0.074	0.063	0.058	0.045	0.054	0.065
30%	0.208	0.134	0.114	0.109	0.087	0.100	0.128
50%	0.287	0.187	0.157	0.150	0.120	0.138	0.177

Because the elastic properties of lattice structures exhibit high anisotropy, the homogenized Young's modulus varies in different orientations [15]. To illustrate the anisotropy of different types of lattice structures, the effective Young's modulus surfaces are used to display the Young's modulus in all the directions in a 3D space as shown in Figure 8-10.

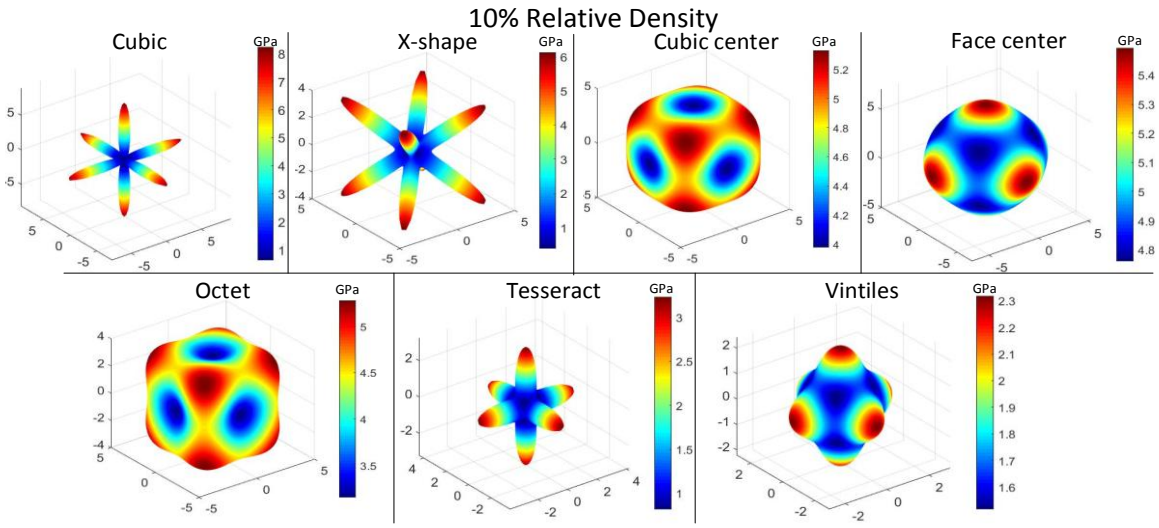

Figure 8 Effective Young's modulus of lattice structures with 10% relative density.

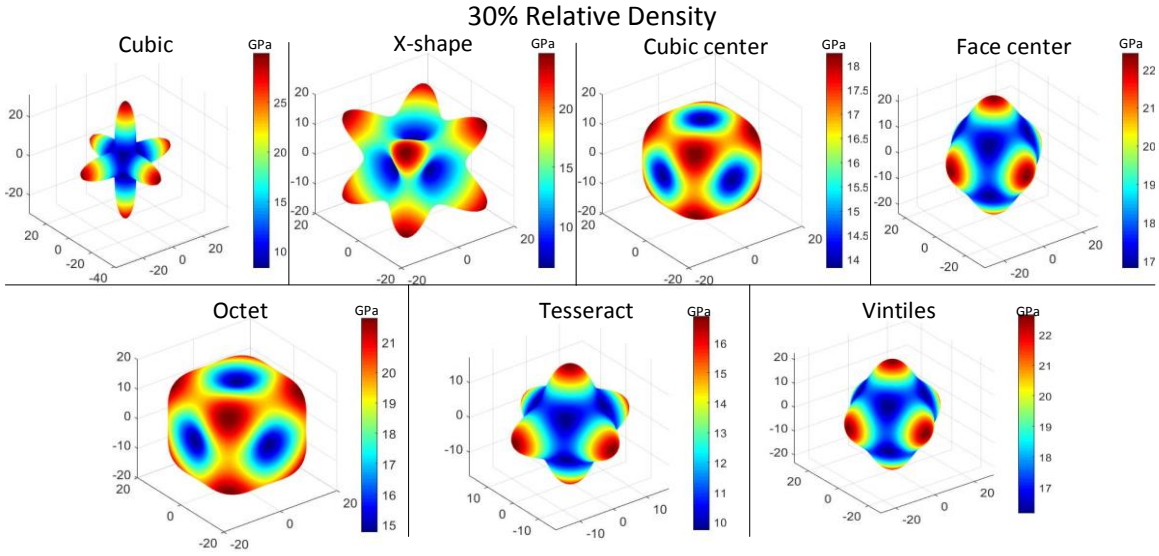

Figure 9 Effective Young's modulus of lattice structures with 30% relative density.

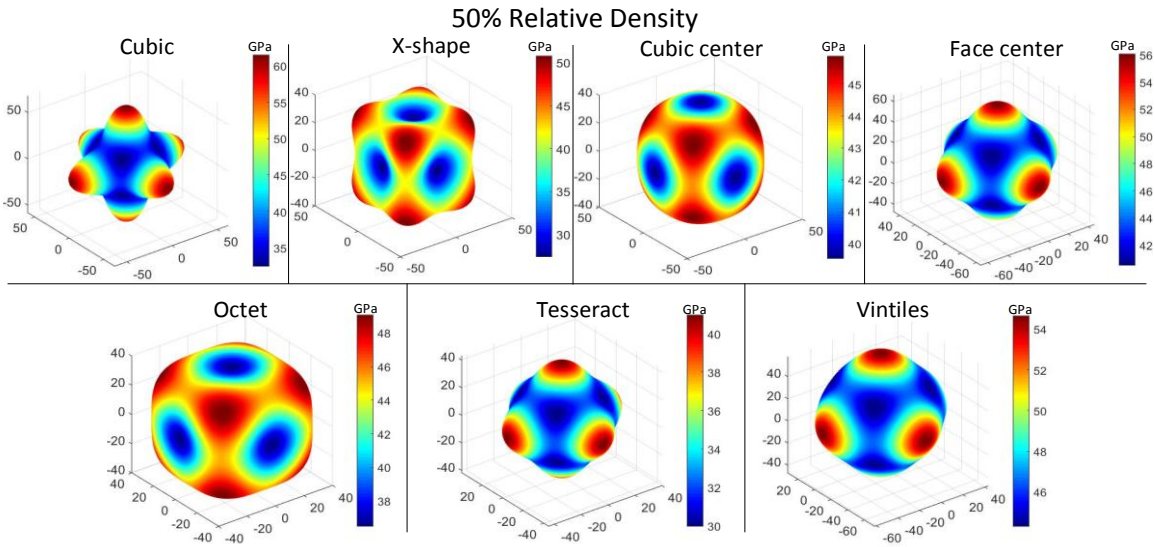

Figure 10 Effective Young's modulus of lattice structures with 50% relative density.

It can be found that the Cubic and X-shape topologies exhibit the higher anisotropy than other topologies. Because for isotropic materials, the Young's modulus surface should be a spherical surface that means the Young's modulus is identical in all directions. However, the Cubic topology has much higher Young's modulus in axial directions while the X-shape topology has higher Young's in diagonal directions. When the relative density is 10%, the strength of these two topologies in weak direction is close to zero, which means they are very soft in those directions (diagonal direction for Cubic and axial direction for X-shape). However, with the increase of relative density, the anisotropy in these two topologies is decreasing. For instance, the maximum Young's modulus is more than 10 times larger than the minimum Young's modulus of the Cubic topology when relative density is 10%. But the ratio is reduced to around 2:1 when the relative density increases to 50%. The X-shape topology follows the similar trend. Even though they exhibit high anisotropy, the maximum Young's modulus in these two topologies is higher than other topologies. Therefore, if the principal stress is in a certain direction, these two topologies might be preferred. But if the loading case is varying and the stress field is complex, these two topologies with a low relative density should be avoided.

The material property of Tesseract topology also exhibits high anisotropy when the relative density is low. But when the relative density reaches to 50%, it is close to isotropic materials. As for Cubic-center, Face-center, Octet and Vintiles topologies, they are less anisotropic compared to other topologies. For example, when the relative density is 10%, the Face-center topology has a maximum Young's modulus around 5.5 GPa and a minimum Young's modulus around 4.8 GPa. It is suitable for complex loading cases because the stiffness in all the directions is higher than other topologies. Furthermore, the anisotropy of these four types of topologies does not reduce much with the increase of relative density. It is also found that these topologies have a lower maximum Young's modulus than Cubic and X-shape topologies. However, the Tesseract topology has a

lowest maximum Young's modulus in all topologies. Therefore, if the design objective is a high stiffness, it should be avoided to use Tesseract topology. Another interesting finding in Figure 10 is that these topologies can be classified into two groups according to the direction of their maximum Young's modulus. The Cubic, Face-center, Tesseract, and Vintiles topologies have a higher stiffness along axial directions. On the contrary, the X-shape, Cubic-center and Octet topologies have a higher stiffness along diagonal directions. As a result, designers can choose the topology based on the direction of the principal stress if a higher stiffness is required. The result from the homogenization code can be used as a guide for designers to make decisions.

4.3 Convergence of the result

It is known that a higher resolution will result in more computational costs. Appropriate voxel size must be chosen to ensure accurate results with allowable computational costs. In this subsection, the relationship between the resolution of the voxel model and the convergence of the result will be discussed. The Cubic-center topology is used in this investigation. The resolution is set from 20 to 100 with 5 as an increment. The resolution means the number of voxels in the voxel model along x, y, and z-direction. The axial Young's modulus is selected to represent the result. The bulk material property is also the same as in Section 4.1. The homogenized constitutive matrix is computed for two relative densities: 10% and 30%. The result is shown in Figure 11. It can be found that when the number of voxels reaches to 100, the result has a trend to converge. However, because the computational cost is quite large when the resolution is 100, there is no need to set such a high resolution if high accuracy is not the primary goal of simulation. A resolution around 50 can already give an indicative result. But if a high accuracy is required, 100 resolution is recommended.

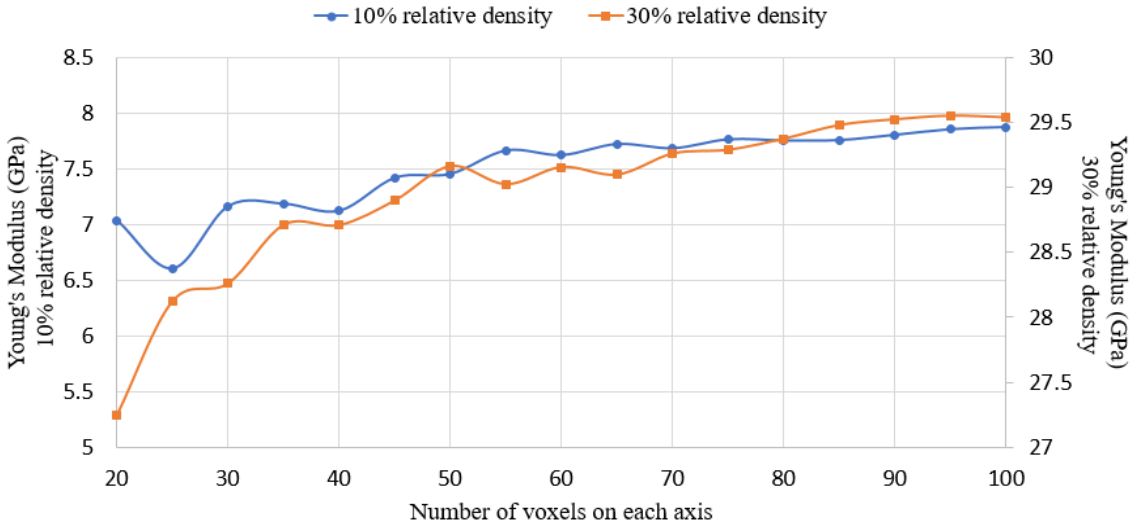

Figure 11 The relationship between the resolution of the voxel and the axial Young's modulus

5.Extension

5.1 Shifted incomplete Cholesky factorization

To further accelerate the convergence of the PCG method, the shifted incomplete Cholesky factorization with threshold dropping can be used to modify the preconditioner. The default preconditioner in this code is an incomplete Cholesky factorization with zero-fill. Nevertheless, the incomplete Cholesky with threshold dropping (ICT) can perform better convergence than zero-fill. A nonnegative scalar is used as a drop tolerance when performing ICT. Elements which are smaller in magnitude than a local drop tolerance are dropped from the resulting factor except for the diagonal element which is never dropped. However, Incomplete Cholesky factorizations of positive definite matrices do not always exist. Sometimes non-positive will be encountered when constructing an incomplete Cholesky preconditioner. To solve this problem, the 'diagcomp' option is used to construct a shifted incomplete Cholesky factorization. That is, instead of constructing L such that L*L' approximates A, incomplete Cholesky factorization with diagonal compensation constructs L such that $\mathrm{L} \times \mathrm{L}'$ approximates $\mathrm{M} = \mathrm{A} + \mathrm{alpha} \times \mathrm{diag}(\mathrm{diag}(\mathrm{A}))$ without explicitly forming M. As incomplete factorizations always exist for diagonally dominant matrices, alpha can be found to make M diagonally dominant. These can be done by substituting line 61 in Appendix A with the following lines:

```
opts.type = 'ict';
opts.droptol = 1e-3;
alpha = 0.01;
opts.diagcomp = alpha;
L = ichol(K(activedofs(4:end),activedofs(4:end)),opts);
```

where 'opts' saves the options for 'ichol', the drop tolerance is 0.001 and the diagonal compensation is 0.01. To show the improvement of the shifted incomplete Cholesky factorization, a PCG method is performed on the Octet topology with a 10% relative density. The result is shown in Figure 12. It can be found that the result didn't converge to 10^{-10} in 300 iterations when incomplete Cholesky preconditioner had zero-fill. When the threshold dropping is active and the diagonal compensation is 0.1, the result converged to 10^{-10} in 250 iterations. Furthermore, when the diagonal compensation is 0.01, the result converged in 150 steps. But it is also found that if there is no diagonal compensation or the diagonal compensation is less than 0.001, the non-positive pivot error will be encountered. Therefore, a proper diagonal compensation should be selected in the shifted incomplete Cholesky factorization.

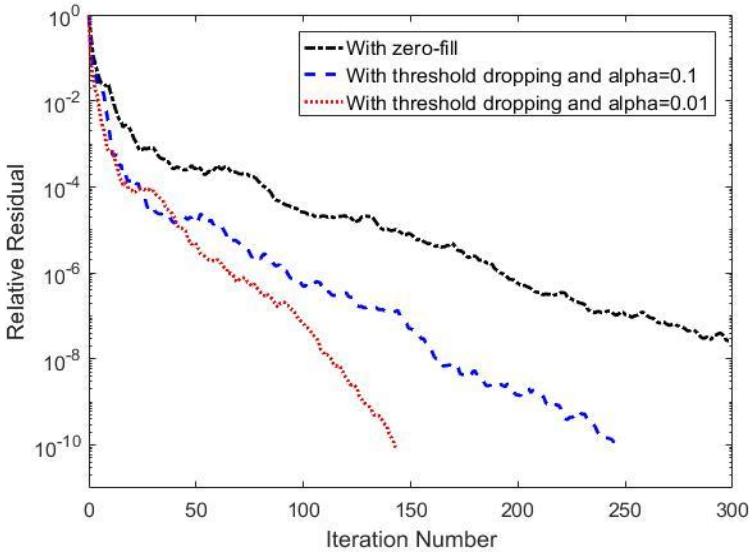

Figure 12 The convergence plot for PCG method with different preconditioners

5.2 Two or more materials

Recently, Multi-material structures have received more and more attention from researchers, and have been successfully fabricated by some AM technologies [16]. Cellular architectures fabricated with multi-materials can realize specific functions such as controllable thermal expansion [17] and self-deformation [18]. To simulate the material properties of multi-material cellular architectures, the homogenization code can be extended to assign more than one material properties. In this section, the extension to three materials is described. Firstly, the indicator matrix will not only have '0' and '1' elements to represent the void and the first material, but also have '2' and '3' elements to represent the second and the third material, respectively. Because the bulk material property is divided into two parts, λ and μ. If there is only one material, λ and μ are scalars. But if there are more than one materials, λ and μ will be vectors. Line 45 in Appendix A needs to be replaced by

```
lambda = lambda(1)*(voxel==1) + lambda(2)*(voxel==2)...
    + lambda(3)*(voxel==3);
mu = mu(1)*(voxel==1) + mu(2)*(voxel==2)...
    + mu(3)*(voxel==3);
```

The vector λ and μ should have 3 entries, which are the Lame's first and second parameter for three materials, respectively. Also, the active degrees of freedom should also be changed to consider the elements with the second material and the third material. The line 62 should be substituted by

```
activedofs = edof((voxel==1 | voxel==2 | voxel==3),:);
```

which means if the element has material 1, 2 or 3, its degrees of freedom will be activated in the global stiffness equation. Figure 13 shows an example of a three-material lattice structure model with 10% relative density. The Cubic-center topology can be disassembled to Cubic topology and X-shape topology. The voxels in the Cubic topology is assigned with material 1; the voxels in the X-shape topology is assigned with material

2; the connection between these two topologies is assigned with material 3. These three types of voxels are assembled together to get the multi-material Cubic-center topology.

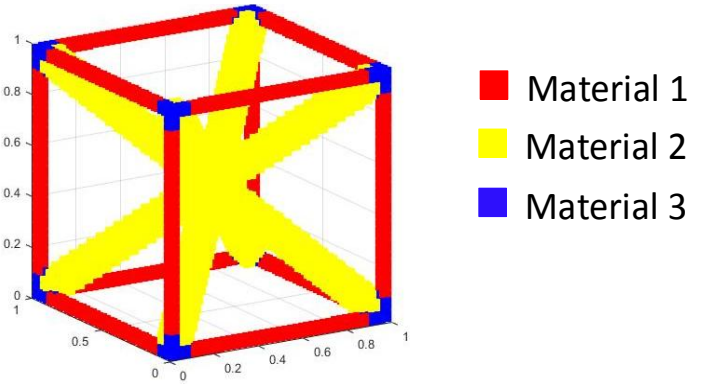

Figure 13 A three-material lattice cellular structure with Cubic-center topologies

To get the homogenized constitutive matrix of this three-material structure, the material properties is assigned with the data in Table 3. The anisotropy can be controlled by adjusting the material properties. Calling the modified function as:

```
CH = homo3D(1,1,1,[170.8,80.8,115.4],[117.8,55.7,79.6],voxel);
```

The homogenized constitutive matrix for the multi-material lattice structure is obtained. To visualize the elastic properties along different orientations, the Young's modulus surface of the multi-materials lattice is shown in Figure 14(a). Figure 14(b) shows the Young's modulus surface of the single material lattice with $\lambda = 115.4$ and $\mu = 79.6$. The multi-materials lattice has an isotropic elastic property because the Young's modulus is the same along all the directions. Therefore, this homogenization code can help designers to design lattice structure with isotropic material properties.

Table 3 Material properties of three materials in the multi-material structure

Material	λ	μ
1	170.8	117.8
2	80.8	55.7
3	115.4	79.6

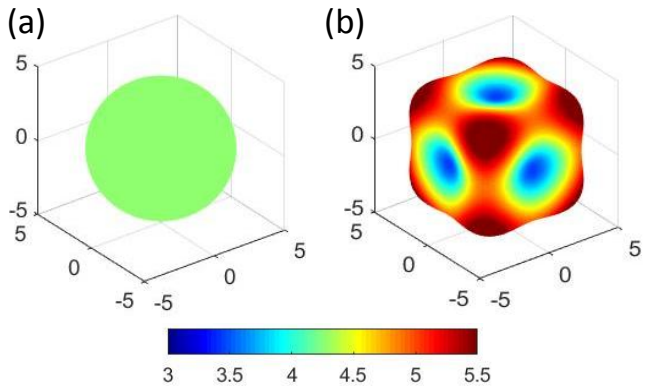

Figure 14 The Young's modulus surface of Cubic-center lattice with (a) multi-materials, (b) single material

5.3 Thermal conductivity

In thermal conductivity problems, the temperature is a scalar field. The homogenization equations are analogous to those of the elastic problem, which can be written as [12]:

$$\int_\Omega \upsilon_{,i}\mu_{ij}T^{(k)}_{,j}d\Omega = \int_\Omega \upsilon_{,i}\mu_{ij}T^{0(k)}_{,j}d\Omega \quad \forall \upsilon \in \Omega \tag{15}$$

$$\mu^H_{ij} = \frac{1}{|\Omega|}\int_\Omega \left(T^{0(i)}_{,l} - T^{(i)}_{,l}\right)\mu_{lm}\left(T^{0(i)}_{,m} - T^{(i)}_{,m}\right)d\Omega \tag{16}$$

where T is the temperature field, υ is a virtual temperature field and μ_{ij} is the conductivity tensor and. Unlike the elastic problem, only one material parameter is used to determine the conductivity of an isotropic material. Therefore, the input argument 'lambda' is a zero vector. The conductivity of different materials is given by the input argument 'mu'. To change it in the code, line 101 is substituted by:

```
CMu = diag([1 1 1 0 0 0]); CLambda = zeros(6);
```

The element matrix 'keMu' contains the contributions in x, y and z directions in different rows and columns. Because only a scalar field is necessary, the contributions in x, y and z directions can be summed together to form an 8×8 element matrix. The summation is done by adding the following line after line 15:

```
keMu(1:3:end,1:3:end) = keMu(1:3:end,1:3:end)...
    + keMu(2:3:end,2:3:end) + keMu(3:3:end,3:3:end);
```

To consider the conductivity of both materials in the cellular architecture, line 45 is changed to:

```
lambda = lambda(1)*(voxel==0) + lambda(2)*(voxel==1);
mu = mu(1)*(voxel==0) + mu(2)*(voxel==1);
```

Even though 'lambda' has no meaning in the thermal conductivity problem. 'lambda' is kept here so that the code can be modified as less as possible. Because every three rows and columns are saved in one row and column, the solution of the global stiffness equation should be modified. Line 59-66 is replaced by the following lines:

```
L = ichol(K(4:3:end,4:3:end));
X = zeros(ndof,3);
for i = 1:3
    X(4:3:end,i) = pcg(K(4:3:end,4:3:end),F(3+i:3:end,i),1e-
10,300,L,L');
end
```

Also, element strain cases in line 75 should be modified as:

```
X0_e(4:3:end,1:3) = keMu(4:3:end,4:3:end)...
                    \[feMu(4:3:end,1),feMu(5:3:end,2),feMu(6:3:end,3)];
```

The loops start from line 85 and line 85 should only run 3 times instead of 6 times. By modifying the code like this, the 3D conductivity problem with two materials can be solved. If three or more materials exist in the unit cell, the code can be further extended by the procedure similar to that in Section 5.2.

6.Conclusion

In this paper, a simple and efficient 3D homogenization code written in Matlab is provided to reduce the barrier of the implementation of numerical homogenization. Designers can use this code to obtain the homogenized constitutive matrix of 3D cellular materials or multi-material composites even without strong mathematical background. To further facilitate the implementation of this code, an algorithm is proposed to generate the voxel model of lattice structures. This model can be directly imported into the homogenization code. A wireframe script is needed in the generation of voxel models. Its format is defined so that users can customize the topology of the lattice structure.

By using the proposed method, the homogenized constitutive matrix and the Young's modulus surface of lattice structures with seven topologies are calculated. The results have shown that the lattice structures exhibit high anisotropy. The Cubic, Face-center, Tesseract, and Vintiles topologies have a higher stiffness along axial directions. On the contrary, the X-shape, Cubic-center and Octet topologies have a higher stiffness along diagonal directions.

The convergence of the PCG method in the homogenization code is accelerated by using Shifted incomplete Cholesky factorization. The code is then extended to multi-materials and thermal conductivity problems. It is found that the anisotropy of multi-material lattice structures can be controlled. With the proper combination of material properties, the Cubic center lattice structure is able to have isotropic elastic properties. Finally, future work can focus on the further improvement of the efficiency. For instance, the assembly-free method can be implemented to reduce the computational cost.

Acknowledgment

This research work is supported by National Sciences and Engineering Research Council of Canada Discovery Grant RGPIN436055-2013.

Appendix A. Matlab files and the wireframe definition

All the Matlab files including the 'homo3D.m' and the 'GenerateVoxel.m' and other supporting files such as the wireframe files can be downloaded by the link:

https://github.com/GuoyingDong/homogenization

or

https://www.mathworks.com/matlabcentral/fileexchange/67457-3d-homogenization-of-cellular-materials

Table A1 Definition and description of GRID.

1	2	3	4	5
GRID	ID	x	y	z

Table A2 Definition and description of STRUT.

1	2	3	4
STRUT	ID	Start	End

Reference

[1] T. A. Schaedler and W. B. Carter, "Architected Cellular Materials," in *Annual Review of Materials Research* vol. 46, ed, 2016, pp. 187-210.

[2] D. T. Queheillalt and H. N. Wadley, "Cellular metal lattices with hollow trusses," *Acta Materialia,* vol. 53, pp. 303-313, 2005.

[3] E. C. Clough, J. Ensberg, Z. C. Eckel, C. J. Ro, and T. A. Schaedler, "Mechanical performance of hollow tetrahedral truss cores," *International Journal of Solids and Structures,* vol. 91, pp. 115-126, 2016.

[4] T. A. Schaedler, C. J. Ro, A. E. Sorensen, Z. Eckel, S. S. Yang, W. B. Carter, *et al.*, "Designing metallic microlattices for energy absorber applications," *Advanced Engineering Materials,* vol. 16, pp. 276-283, 2014.

[5] H. N. Wadley and D. T. Queheillalt, "Thermal applications of cellular lattice structures," in *Materials science forum*, 2007, pp. 242-247.

[6] B. Hassani and E. Hinton, "A review of homogenization and topology optimization I—homogenization theory for media with periodic structure," *Computers & Structures,* vol. 69, pp. 707-717, 12// 1998.

[7] A. Bensoussan, J.-L. Lions, and G. Papanicolaou, *Asymptotic analysis for periodic structures* vol. 374: American Mathematical Soc., 1978.

[8] S. Torquato, *Random heterogeneous materials: microstructure and macroscopic properties* vol. 16: Springer Science & Business Media, 2013.

[9] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering,* vol. 71, pp. 197-224, 1988/11/01 1988.

[10] S. Arabnejad and D. Pasini, "Mechanical properties of lattice materials via asymptotic homogenization and comparison with alternative homogenization methods," *International Journal of Mechanical Sciences,* vol. 77, pp. 249-262, 12// 2013.

[11] B. Hassani and E. Hinton, "A review of homogenization and topology opimization II - Analytical and numerical solution of homogenization equations," *Computers and Structures,* vol. 69, pp. 719-738, 1998.

[12] E. Andreassen and C. S. Andreasen, "How to determine composite material properties using numerical homogenization," *Computational Materials Science,* vol. 83, pp. 488-495, 2014.

[13] T. R. Chandrupatla, A. D. Belegundu, T. Ramesh, and C. Ray, *Introduction to finite elements in engineering* vol. 2: Prentice Hall Upper Saddle River, NJ, 2002.

[14] B. Lautrup, "Physics of continuous matter," *Exotic and Everyday Phenomena in the Macroscopic World, IOP,* 2005.

[15] S. Xu, J. Shen, S. Zhou, X. Huang, and Y. M. Xie, "Design of lattice structures with controlled anisotropy," *Materials and Design,* vol. 93, pp. 443-447, 2016.

[16] M. Vaezi, S. Chianrabutra, B. Mellor, and S. Yang, "Multiple material additive manufacturing – Part 1: a review," *Virtual and Physical Prototyping,* vol. 8, pp. 19-50, 2013/03/01 2013.

[17] H. Xu and D. Pasini, "Structurally efficient three-dimensional metamaterials with controllable thermal expansion," *Scientific reports,* vol. 6, p. 34924, 2016.

[18] C.-H. Yu, K. Haller, D. Ingber, and R. Nagpal, "Morpho: A self-deformable modular robot inspired by cellular structure," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 3571-3578.

List of Figures

List of Tables