(Received 12 February 2025, Accepted 16 May 2025, Scheduled 15 July 2025)

The Finite Element Method for the Spatially-Variant Lattice Algorithm for Volumes and Doubly-Curved Surfaces

Edgar Bustamante[1, 2] and Raymond C. Rumpf[1, 3, *]

[1] *University of Texas at El Paso, USA*
[2] *Sandia National Laboratories, USA*
[3] *Kraetonics, LLC, USA*

ABSTRACT: A 2D flat, 2D curved, and 3D finite element method (FEM) implementation of a spatially-variant lattice (SVL) algorithm is presented. This powerful algorithm is used in electromagnetics to preserve the electromagnetic properties and geometry of periodic structures that are bent, twisted, conformed, or otherwise spatially varied. Applications of the SVL algorithm include photonic crystals, metamaterials, conformal frequency selective surfaces, cloaking devices, and volumetric circuits over complex geometries. The present work shows examples of SVLs over a planar surface lattice, a doubly-curved surface lattice, and a volumetric lattice.

1. INTRODUCTION

The finite element method (FEM) is a numerical method that has been widely used in electromagnetics, structural analysis, and other areas to solve partial differential equations (PDEs) [1]. Since its inception in 1941 by Alexander Hrennikoff [2], FEM gained rapid popularity because it was able to handle complex geometries and boundary conditions easily, unlike other numerical methods. In previous work, this allowed FEM to expand from structural analysis into broader fields such as heat transfer, fluid mechanics [3], and electromagnetics to now the spatially-variant lattice algorithm (SVLA).

This numerical method discretizes a domain into a mesh composed of many small finite elements such as triangles and quadrilaterals in 2D or tetrahedrons and hexahedrons in 3D. Inside each element, the PDE solution is approximated by a set of basis functions that ensure continuity across elements. A mesh can be tailored using multiple different elements which can be crucial for geometries, like cones, where tetrahedral elements are more appropriate, or cylindrical structures (extruded geometries) where hexahedron can effectively reduce the overall mesh size. In addition, FEM is very attractive for its accuracy and elegant use of meshes that model complex geometries and boundaries while being relatively easy to implement [1, 4–6].

In electromagnetics, a lattice is a collection of repeated unit cells that form a specific pattern. Lattices can provide extraordinary electromagnetic properties that are not achievable with homogenous materials. This is the case in frequency selective surfaces (FSSs) [7], photonic crystals [8], diffraction gratings [9], and metamaterials [10], to name a few. To maintain the electromagnetic properties of a lattice over a curved, bent, or twisted lattice, the uniformity of the unit cells must be preserved. In this regard, a uniform lattice can be modified using the spatially-variant lattice algorithm (SVLA) [11] to produce a new uniform lattice that curves, bends, twists, functionally grades, and even slowly transitions from one periodic symmetry to another. This is achieved by Fourier transforming the original uniform lattice to obtain a set of planar gratings. These are then spatially-varied individually using the SVLA and summed to create a lattice that has been spatially varied. The result is called a preservational spatially-variant lattice (PSVL). The SVLA enables the manipulation of lattice gratings by adjusting their orientation, density (fill factor), and inter-element spacing.

The SVLA was first introduced using the finite-difference method (FDM) for a self-collimating photonic crystal that had a 90° bend [11] that was later 3D printed [12] and microfabricated with multi-photon lithography [13]. The same authors have applied this algorithm to transmission line electromagnetic isolation via metamaterials [14] and multi-mode dielectric waveguides [15]. Others followed with applications in transformation optics devices [16], control of phase and power in photonic crystals [17], self-collimating photonic crystals with two 90° bends [18], conformal frequency selective surfaces (CFSSs) [19], and 3D volumetric circuits [20]. This algorithm has also been adapted to generate large SVLs iteratively via FDM [21].

The beauty in the SVLA is that it can easily be applied to virtually any periodic structure in electromagnetics. These periodic structures include photonic crystals, FSSs, metasurfaces, gratings, and even artistic patterns. In the area of FSSs, the SVLA has been used for a doubly-curved surface to enforce the size, shape, symmetry, and spacing of a Jerusalem Cross CFSS. The authors further improved the SVLA by incorporating deformation control (DC) [19]. Deformation control fine-tuned the placement of periodic elements along a doubly-curved surface to match the performance of the flat FSS. Compared to CF-

* Corresponding author: Raymond C. Rumpf (rcrumpf@utep.edu).

SSs created through projection techniques, the CFSS developed with deformation control exhibited superior performance [22–24]. The authors identified the lattice center points on the doubly-curved surface and used them as the placement points of the CFSS elements. Conforming a SVL to a doubly-curved surface is one of the aims of this paper due to the potential applications. For this reason, this paper aims at developing a general SVLA foundation in FEM that can develop PSVLs for complex geometries and volumes.

All of the work done in the SVLA space has been performed over rectangular structured grids due to the simplicity of applying the FDM to structured-orthogonal grids. These types of grids pose a problem for meshes with closed forms such as spheres where continuity becomes difficult to impose while keeping a healthy mesh. A solution is to use non-orthogonal unstructured rectangular grids [25] or even better, Delaunay triangulation [26]. Although finite-difference schemes have been developed for these types of grids, implementing them is not trivial [27, 28].

This work formulates the SVLA in the FEM framework for general geometries where its FDM counterpart can be difficult to implement (non-rectangular meshes) and be more restrictive. Results with the formulation presented here are showcased using a flat surface, a curved surface, and a volume that were meshed using different types of elements. As a bonus, a simple study on preconditioners and solvers is included that compares the FDM and FEM SVLAs solver times using the graphical processing unit (GPU) and central processing unit (CPU) within MATLAB.

2. FORMULATION

The core innovation of the SVLA lies in the construction of lattices by accumulating planar gratings with spatially modulated phases and orientations, rather than deforming the lattice itself. This approach ensures continuity and avoids defects, which is critical in applications such as optics and curved-surface conformations. This is done by decomposing a periodic structure into a complex Fourier series with M spatial harmonics in a coordinate system $\vec{r} = x\hat{a}_x + y\hat{a}_y + z\hat{a}_z$. Each spatial harmonic m is a 1D sinusoidal grating with a complex amplitude a_m and its period and direction dictated by the grating vector $\vec{K}_m$.

$$\varepsilon\left(\vec{r}\right) = \sum_{m=1}^{M} a_m \exp\left(j\vec{K}_m \cdot \vec{r}\right) \tag{1}$$

These complex amplitudes are obtained using the fast Fourier transform (FFT). The grating vectors of the expansion are integer multiples of primitive lattice vectors. The primitive lattice vectors are $2\pi/\Lambda_x\hat{a}_x$, $2\pi/\Lambda_y\hat{a}_y$, and $2\pi/\Lambda_z\hat{a}_z$ for some symmetries. Variables p, q, and r are integers and Λ_x, Λ_y, and Λ_z are the physical dimensions of the lattice unit cell in x, y, and z. In theory, an infinite sum of spatial harmonics rebuilds the original periodic structure. In practice, however, this is not possible and M is truncated to include only the lowest-order harmonics which produces an analog version of the lattice $\varepsilon(\vec{r})$.

The goal of the algorithm is to spatially vary the period and/or orientation of each grating, thus turning $\vec{K}$ into a function of position, $\vec{K}(\vec{r})$. When this happens, Eq. (1) fails to construct the gratings correctly. The solution is to relate $\vec{K}(\vec{r})$ to the grating phase, $\Phi(\vec{r})$, using Eq. (2), and solve for the grating phase. This is further explained in the seminal paper [9] that presented the SVLA.

$$\nabla\Phi\left(\vec{r}\right) = \vec{K}\left(\vec{r}\right) \tag{2}$$

This process works for any unit cell, but in this work, complex Fourier series are not used because the lattices presented in the Results section can be easily created using just two or three grating vectors through the reciprocal lattice vectors. Therefore, the complex amplitudes are assumed to be unity and M is a maximum of three. This FEM implementation however, also works with the complex Fourier series approach. These assumptions modify Eq. (1) to

$$\varepsilon_{\text{analog}}\left(\vec{r}\right) = \sum_{m=1}^{M} \cos\left(\Phi_m\left(\vec{r}\right)\right). \tag{3}$$

For 2D flat surfaces $\vec{r} = x\hat{a}_x + y\hat{a}_y$ or $\vec{r} = x\hat{a}_x + y\hat{a}_y + z\hat{a}_z$ for 2D curved surfaces and volumes. The results of each $\cos\left(\Phi_m(\vec{r})\right)$ in Eq. (3) produce an analog grating. The sum of these analog gratings produces an analog lattice. Very often, the analog gratings are converted into binary gratings by applying a threshold function, $\gamma(\vec{r})$, given a specific fill factor, $f(\vec{r})$. The threshold function is computed using $\gamma(\vec{r}) \simeq \cos\left(\pi f(\vec{r})\right)$. Finally, using $\varepsilon_{\text{binary}}(\vec{r}) = \varepsilon_{\text{analog}}(\vec{r}) > \gamma(\vec{r})$, the binary grating is computed from the analog grating. The binary grating can now be used to differentiate between two sets of permittivities, as shown by the 3D example in Fig. 10.

Near the end of Ref. [11], Eq. (2) is transformed into the Laplacian version of the grating equation due to a least-squares solution. This happens due to Eq. (2) forming a system of linear equations, $\mathbf{Ax} = \mathbf{b}$, that has more equations than unknowns (degrees of freedom). In Eq. (2), matrix $\mathbf{A}$ is the gradient operator ∇, $\mathbf{b}$ is the grating vector $\vec{K}(\vec{r})$, and $\mathbf{x}$ is the grating phase $\Phi(\vec{r})$ (unknown). A "best fit" solution is obtained via the least-squares method, $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$. This means that Eq. (2) in principle is also equivalent to

$$\nabla^2\Phi\left(\vec{r}\right) = \nabla \cdot \vec{K}\left(\vec{r}\right). \tag{4}$$

The presented FEM SVLA formulation starts with the Laplacian version of the grating equation, Eq. (4).

In this work, the FEM is implemented using the method of weighted residuals [29, 30] in conjunction with Galerkin's method [31]. Starting with the method of weighted residuals, Eq. (4) is tested with scalar basis function, $T(\vec{r})$ (see Appendix A), and integrated over the entire domain Ω (surface or volume). Doing this produces what is known as the strong form, Eq. (5), which simply means that it is the original PDE. This is done for each element that forms part of the finite element mesh in the volume.

$$\int_V T\left(\vec{r}\right) \cdot \left[\nabla^2\Phi\left(\vec{r}\right)\right] dV = \int_V T\left(\vec{r}\right) \cdot \left[\nabla \cdot \vec{K}\left(\vec{r}\right)\right] dV \tag{5}$$

Similar to the canonical Poisson's equation in FEM [1], scalar/nodal basis functions are used as the test functions (see

Appendix A). Next, the unknown function, $\Phi^e(\vec{r})$, of element e, is approximated as a linear combination of nodal basis functions $N_j^e(\vec{r})$, Eq. (6). This means that the grating phase anywhere inside element e can be interpolated by sampling the grating phase over a few points. These sampling points inside element e are defined by index j, n is the total number of sampling points, and $N_j^e(\vec{r})$ is the basis function associated with point j inside element e.

$$\Phi^e(\vec{r}) = \sum_{j=1}^{n} \varphi_j^e N_j^e(\vec{r}) \tag{6}$$

Substituting Eq. (6) into Eq. (5) shows that $N_j^e(\vec{r})$ must be doubly differentiable due to the Laplacian in Eq. (5). This is not possible when using linear basis functions (see Appendix A) because their second order partial derivatives are zero. Of course, using higher order basis functions such as quadratic or cubic basis functions is also an option but the complexity of the FEM implementation increases. To take advantage of the simplicity of linear basis functions, a weak form of Eq. (5) can be constructed. This weak form is mathematically equivalent to the strong form; however, it allows for the use of linear basis functions by having only first order partial derivatives in the PDE.

In order to formulate the weak form of Eq. (5) and obtain a system of equations, $\mathbf{Ax} = \mathbf{b}$, Green's first theorem,

$$\int_V \nabla\phi \cdot \nabla\psi dV + \int_V \phi\left[\nabla^2\psi\right] dV = \oint_{\partial V} \phi\left[\nabla\psi \cdot \hat{n}\right] dS, \tag{7}$$

is used to reformulate the left-hand side (LHS) and right-hand side (RHS) of Eq. (5). First, the LHS of Eq. (5) is expanded using Eq. (7) by setting $\psi = \Phi$ and $\phi = T$,

$$\begin{aligned}\int_V T \cdot \left[\nabla^2\Phi\right] dV &= -\int_V \left[\nabla T\right] \cdot \left[\nabla\Phi\right] dV \\ &\quad + \oint_{\partial V} T\left[\nabla\Phi \cdot \hat{n}\right] dS.\end{aligned} \tag{8}$$

Similarly, the RHS of Eq. (5) is expanded using Eq. (7) by setting $\nabla\psi = \vec{K}$ and $\phi = T$,

$$\begin{aligned}\int_V T \cdot \left[\nabla \cdot \vec{K}\right] dV &= -\int_V \vec{K} \cdot \left[\nabla T\right] dV \\ &\quad + \oint_{\partial V} T\left[\vec{K} \cdot \hat{n}\right] \cdot dS.\end{aligned} \tag{9}$$

This produces the weak form of Eq. (5),

$$\begin{aligned}&-\int_V \left[\nabla T\right] \cdot \left[\nabla\Phi\right] dV + \oint_{\partial V} T\left[\nabla\Phi \cdot \hat{n}\right] dS \\ &= -\int_V \vec{K} \cdot \left[\nabla T\right] dV + \oint_{\partial V} T\left[\vec{K} \cdot \hat{n}\right] \cdot dS.\end{aligned} \tag{10}$$

The two surface integrals in Eq. (10) cancel with the use of Eq. (2), and therefore, the equation becomes

$$\int_V \nabla T \cdot \nabla\Phi dV = \int_V \nabla T \cdot \vec{K} dV. \tag{11}$$

Finally, to create the system of linear equations $\mathbf{Ax} = \mathbf{b}$, substitute Eq. (6) into Eq. (11) and use the same basis functions, $N_j^e(\vec{r})$, for the testing function, $T(\vec{r})$. This is known as Galerkin's method

$$T = N_i^e(\vec{r}) \quad (i = 1, 2, \ldots, n). \tag{12}$$

Everything combined produces a system of linear equations for each element e,

$$\sum_{j=1}^{n} K_{ij}^e \varphi_j^e = b_i^e \quad (i = 1, 2, \ldots, n), \tag{13}$$

where

$$K_{ij}^e = \int_V \nabla N_i^e(\vec{r}) \cdot \nabla N_j^e(\vec{r})\, dV \tag{14}$$

$$b_i^e = \int_V \nabla N_i^e(\vec{r}) \cdot \vec{K}(\vec{r})\, dV. \tag{15}$$

There are two routes that Eqs. (14) and (15) can be solved, analytically or numerically using Gaussian quadrature [32–34]. The latter offers a more general solution because using higher order basis functions require minimal modification of a FEM implementation and is very accurate [32]. To conduct Gaussian quadrature, the integrands of an integral are evaluated at a set of points and multiplied by their weights (see Appendix A) and combined,

$$\int_V f(\vec{r})\, dV \approx \sum_{k=1}^{m} w_k f(\vec{r}_k). \tag{16}$$

An exact integration can be obtained if the total number of integration points, m, is $m \geq (p+1)/2$, where p is the polynomial degree of $f(\vec{r})$ [1]. The basis functions will need to be evaluated at the quadrature points k. However, this numerical integration is meant to be used over a perfect right triangle, right tetrahedron, square, or cube depending on the type of finite elements. Very likely the mesh used will be composed of elements that are none of these perfect geometric shapes. Therefore, the operations in $f(\vec{r})$ in Eq. (16) for all elements must be transformed onto a new coordinate system where Gaussian quadrature can be applied. This new coordinate system is $\vec{s} = \xi\hat{a}_\xi + \eta\hat{a}_\eta$ for flat and curved surfaces and $\vec{s} = \xi\hat{a}_\xi + \eta\hat{a}_\eta + \zeta\hat{a}_\zeta$ for volumes. The transform to the operations is done through the Jacobian, $\mathbf{J}$.

For 3D volumes, the Jacobian of element e is defined as

$$\mathbf{J}_{\mathrm{v}} = \begin{bmatrix} \frac{\partial x(\vec{s})}{\partial\xi} & \frac{\partial y(\vec{s})}{\partial\xi} & \frac{\partial z(\vec{s})}{\partial\xi} \\ \frac{\partial x(\vec{s})}{\partial\eta} & \frac{\partial y(\vec{s})}{\partial\eta} & \frac{\partial z(\vec{s})}{\partial\eta} \\ \frac{\partial x(\vec{s})}{\partial\zeta} & \frac{\partial y(\vec{s})}{\partial\zeta} & \frac{\partial z(\vec{s})}{\partial\zeta} \end{bmatrix} \tag{17}$$

$$\mathbf{J}_{\text{cs}} = \begin{bmatrix} \frac{\partial x(\vec{s})}{\partial \xi} & \frac{\partial y(\vec{s})}{\partial \xi} & \frac{\partial z(\vec{s})}{\partial \xi} \\ \frac{\partial x(\vec{s})}{\partial \eta} & \frac{\partial y(\vec{s})}{\partial \eta} & \frac{\partial z(\vec{s})}{\partial \eta} \\ p & q & r \end{bmatrix} \tag{18}$$

where p, q, and r in the last row of Eq. (18) are extracted from Eq. (21) after taking the cross product of Eqs. (19) and (20).

$$\vec{A} = \frac{\partial x(\vec{s})}{\partial \xi}\hat{a}_i + \frac{\partial y(\vec{s})}{\partial \xi}\hat{a}_j + \frac{\partial z(\vec{s})}{\partial \xi}\hat{a}_k \tag{19}$$

$$\vec{B} = \frac{\partial x(\vec{s})}{\partial \eta}\hat{a}_i + \frac{\partial y(\vec{s})}{\partial \eta}\hat{a}_j + \frac{\partial z(\vec{s})}{\partial \eta}\hat{a}_k \tag{20}$$

$$\vec{A} \times \vec{B} = p\hat{a}_i + q\hat{a}_j + r\hat{a}_k. \tag{21}$$

Finally, for 2D flat surfaces the Jacobian of element e is

$$\mathbf{J}_{\text{s}} = \begin{bmatrix} \frac{\partial x(\vec{s})}{\partial \xi} & \frac{\partial y(\vec{s})}{\partial \xi} \\ \frac{\partial x(\vec{s})}{\partial \eta} & \frac{\partial y(\vec{s})}{\partial \eta} \end{bmatrix}. \tag{22}$$

Changing the coordinate system of the linear operations using the Jacobian also means that the differential volumes and surfaces must be changed. These are the changes to the differentials in 3D volumes, 2D flat surfaces, and 2D curved surfaces respectively,

$$dV = |\mathbf{J}_{\text{v}}|\, d\xi d\eta d\zeta \tag{23}$$

$$dS = |\mathbf{J}_{\text{s}}|\, d\xi d\eta \tag{24}$$

$$dS = |\mathbf{J}_{\text{cs}}|\, d\xi d\eta \tag{25}$$

$|\mathbf{J}_{\text{v}}|$ and $|\mathbf{J}_{\text{s}}|$ are the determinants of their respective Jacobians, and for 2D curved surfaces the Jacobian determinant is the magnitude of Eq. (21), $|\mathbf{J}_{\text{cs}}| = |\vec{A} \times \vec{B}|$.

Computing the Jacobian requires the evaluation of $x(\vec{s})$, $y(\vec{s})$, and $z(\vec{s})$ by converting coordinate values from $\vec{r}$ to $\vec{s}$. Using Eq. (6), these terms are obtained by replacing $\Phi^e(\vec{r})$ with $x^e(\vec{s})$, $y^e(\vec{s})$, and $z^e(\vec{s})$ and φ_j^e with x_i^e, y_i^e, and z_i^e.

$$x^e(\vec{s}) = \sum_{i=1}^{n} x_i^e N_i^e(\vec{s}) \tag{26}$$

$$y^e(\vec{s}) = \sum_{i=1}^{n} y_i^e N_i^e(\vec{s}) \tag{27}$$

$$z^e(\vec{s}) = \sum_{i=1}^{n} z_i^e N_i^e(\vec{s}) \tag{28}$$

With the use of Eqs. (26)–(28), the Jacobian from Eq. (17) can be expanded to

$$\mathbf{J}_{\text{v}} = \begin{bmatrix} \sum_{i=1}^{n} x_i(\vec{r})\frac{\partial N_i(\vec{s})}{\partial \xi} & \sum_{i=1}^{n} y_i(\vec{r})\frac{\partial N_i(\vec{s})}{\partial \xi} & \sum_{i=1}^{n} z_i(\vec{r})\frac{\partial N_i(\vec{s})}{\partial \xi} \\ \sum_{i=1}^{n} x_i(\vec{r})\frac{\partial N_i(\vec{s})}{\partial \eta} & \sum_{i=1}^{n} y_i(\vec{r})\frac{\partial N_i(\vec{s})}{\partial \eta} & \sum_{i=1}^{n} z_i(\vec{r})\frac{\partial N_i(\vec{s})}{\partial \eta} \\ \sum_{i=1}^{n} x_i(\vec{r})\frac{\partial N_i(\vec{s})}{\partial \zeta} & \sum_{i=1}^{n} y_i(\vec{r})\frac{\partial N_i(\vec{s})}{\partial \zeta} & \sum_{i=1}^{n} z_i(\vec{r})\frac{\partial N_i(\vec{s})}{\partial \zeta} \end{bmatrix} \tag{29}$$

This can be simplified to a matrix multiplication as shown in Eq. (30) where the partial derivatives of the basis functions are computed analytically. Similarly, $\mathbf{J}_{\text{s}}$ and $\mathbf{J}_{\text{cs}}$ can be expanded.

$$\mathbf{J}_{\text{v}} = \begin{bmatrix} \frac{\partial N_1(\vec{s})}{\partial \xi} & \cdots & \frac{\partial N_n(\vec{s})}{\partial \xi} \\ \frac{\partial N_1(\vec{s})}{\partial \eta} & \cdots & \frac{\partial N_n(\vec{s})}{\partial \eta} \\ \frac{\partial N_1(\vec{s})}{\partial \zeta} & \cdots & \frac{\partial N_n(\vec{s})}{\partial \zeta} \end{bmatrix} \begin{bmatrix} x_1(\vec{r}) & y_1(\vec{r}) & z_1(\vec{r}) \\ \vdots & \vdots & \vdots \\ x_n(\vec{r}) & y_n(\vec{r}) & z_n(\vec{r}) \end{bmatrix} \tag{30}$$

The gradients of the basis functions in Eqs. (14) and (15) are transformed using Eq. (31) which leads to Eq. (32) for $\mathbf{J}_{\text{v}}$, but $\mathbf{J}_{\text{s}}$ or $\mathbf{J}_{\text{cs}}$ can also be used in Eq. (31) for flat surfaces or curved surfaces. This process essentially takes a linear operation from the $\vec{r}$ coordinates and applies it in $\vec{s}$ coordinates.

$$\nabla N_i(\vec{r}) = \mathbf{J}^{-1} \nabla N_i(\vec{s}) \tag{31}$$

$$\begin{bmatrix} \frac{\partial N_j(\vec{r})}{\partial x} \\ \frac{\partial N_j(\vec{r})}{\partial y} \\ \frac{\partial N_j(\vec{r})}{\partial z} \end{bmatrix} = \mathbf{J}_{\text{v}}^{-1} \begin{bmatrix} \frac{\partial N_j(\vec{s})}{\partial \xi} \\ \frac{\partial N_j(\vec{s})}{\partial \eta} \\ \frac{\partial N_j(\vec{s})}{\partial \zeta} \end{bmatrix} \tag{32}$$

The changes to Eqs. (14) and (15) after adding Gaussian quadrature to them are as follows

$$K_{ij}^e = \sum_{k=1}^{m} w_k |\mathbf{J}_k| \nabla N_i^e(\vec{s}_k) \cdot \nabla N_i^e(\vec{s}_k) \tag{33}$$

$$b_i^e = \sum_{k=1}^{m} w_k |\mathbf{J}_k| \nabla N_i^e(\vec{s}_k) \cdot \vec{K}(\vec{s}_k). \tag{34}$$

Note that the Jacobians used to compute the basis function gradients and Jacobian determinants are different for each integration point k, and they are computed from either $\mathbf{J}_{\text{v}}$, $\mathbf{J}_{\text{s}}$, or $\mathbf{J}_{\text{cs}}$. Finally, the grating vector in Eq. (34) must be evaluated at the quadrature points using Eqs. (26)–(28).

The matrix equations in Eqs. (33) and (34) are assembled according to Eq. (13) which operates on the individual elements in a mesh. These matrices can be easily assembled using MATLAB's "page" functions which operate on multidimensional arrays. Each "page" would contain the local elemental matrices as seen in Fig. 1.

The sum of the local interactions is assembled into a sparse and symmetric global matrix $\mathbf{A}$ along with the global forcing vector $\mathbf{b}$ as shown in Fig. 2. The indices for the elements of $\mathbf{A}$ and $\mathbf{b}$ are obtained from an elements-to-nodes connectivity array as shown in the top right of Fig. 2. This connectivity array defines how the elements are connected in a mesh through the global nodes or unique nodes in the mesh. These global nodes are shared by multiple elements and the grating phase (unknowns) is solved for at the global nodes.

Finally, with the $\mathbf{Ax} = \mathbf{b}$ system of linear equations assembled, the grating phase, $\Phi(\vec{r})$, is solved for as $\mathbf{x}$. This

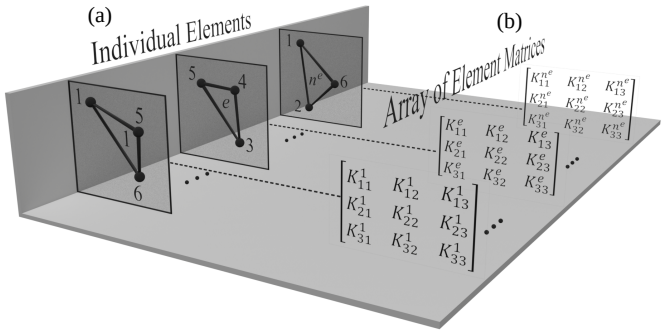

FIGURE 1. (a) Local elements numbered with their global nodes. (b) Local elemental matrices assembled as multidimensional arrays.

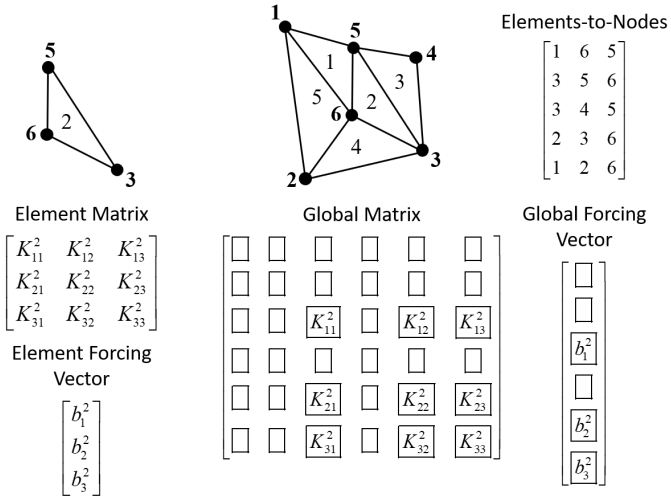

FIGURE 2. Example of the matrix of mesh element #2 and its global matrix assembly.

can be done with a direct or iterative solver [35]. Next, using Eq. (3), the analog version of the spatially varied grating is calculated and summed with the rest of the spatially varied gratings. This sum of analog spatially varied gratings produces an analog SVL.

3. RESULTS

In the following examples, the FEM meshes were replicated using different meshing schemes to verify and validate the results. The 2D FEM surfaces were solved using triangular meshes and quadrilateral meshes. The results of the 2D FEM surfaces employ a quadrilateral mesh with four nodes per element and a four-point Gaussian quadrature rule (see Appendix A). The 2D FDM surfaces used a structured orthogonal grid of quadrilaterals. In addition, all 2D surfaces used 1,000 grid points in the x and y directions. This was purposely done to compare the SVLs produced by the FDM and FEM using the exact same number of grid points. For the 3D FEM example, a tetrahedral mesh and a hexahedral mesh with the same number of grid points were used to solve and verify the results. The hexahedral mesh used eight-noded hexahedral elements and an eight-point Gaussian quadrature rule (see Appendix A). All variations of the SVL algorithm presented in the following examples used MATLAB's (R2024a) version of the preconditioned conjugate gradient (PCG) method to solve the system of linear equations [35].

3.1. 2D Flat Surface

For the first example, a lattice with square symmetry over a flat surface was produced. This symmetry is described by two reciprocal lattice vectors, $\vec{T}_1 = (2\pi/\Lambda)\hat{a}_x$ and $\vec{T}_2 = (2\pi/\Lambda)\hat{a}_y$ which were defined uniformly across the entire grid, with a period of $\Lambda = 0.1$ m. The size of the lattice was set to ten periods for both length and width. Feeding $\vec{T}_1$ and $\vec{T}_2$ as grating vectors, $\vec{K}$, one at a time into the SVLA without applying any modifications to the grating vectors such as orientation, fill factor, or spacing produces the grating phase of each grating vector. The cosine of the grating phase forms the analog grating and the sum of the analog gratings produces the analog lattice as shown in Fig. 3.

A binary lattice can be easily created by applying a scalar threshold to the analog lattice. A "less than or equal to" threshold of 0.519 over the analog lattice in Fig. 3 produced a binary lattice, Fig. 4, with a fill factor of 70% (white region) of the total surface.

The example presented in Fig. 3 will now be presented with a 90° bend using the FDM and FEM SVLAs. This design was achieved by first computing the orientation map using

$$\theta_{\text{map}} = \tan^{-1}\left(\frac{y}{x}\right) \tag{35}$$

where x and y are defined in $\vec{r}$. Note that in FEM, the orientation map must be computed at the quadrature points using Eq. (26) to use it in Eq. (34). The grating vectors $\vec{T}_1$ and $\vec{T}_2$ are then transformed from their original cartesian coordinate system to a cylindrical coordinate system $\vec{t} = r\hat{a}_r + \theta\hat{a}_\theta + z\hat{a}_z$. In the cylindrical coordinate system, the orientation map is added to the theta component as follows $\theta_{\text{new}} = \theta_{\text{old}} + \theta_{\text{map}}$. Then, the grating vectors are transformed back to their original Cartesian coordinate system. These new grating vectors are then used to create the RHS of Eq. (34) and solve for the grating phase.

Following these steps, the FDM and FEM SVLAs produced very similar results as shown visually in Figs. 5(a) and (b). Table 2 shows the numerical error between the FDM and FEM SVLA results by means of the relative error of the grating phase. This numerical error comes from difference in the location between of the finite-difference points and the quadrature points. The location of the FEM quadrature points is dictated by the quadrature rule which differs from the location of the FDM points. In order to compare numerically the FDM and FEM SLVs, the FEM results had to be interpolated to the same location where the FDM points exist, and this is the source of the slight numerical differences in both solutions. For the second set of results in Figs. 5(c) and (d), ten iterations of DC that enforced the orientation defined by Eq. (35) were applied to the results of Figs. 5(a) and (b). The two numerical methods also produced very similar results when DC was applied; Table 1 has the relative error computed between these two sets of results.

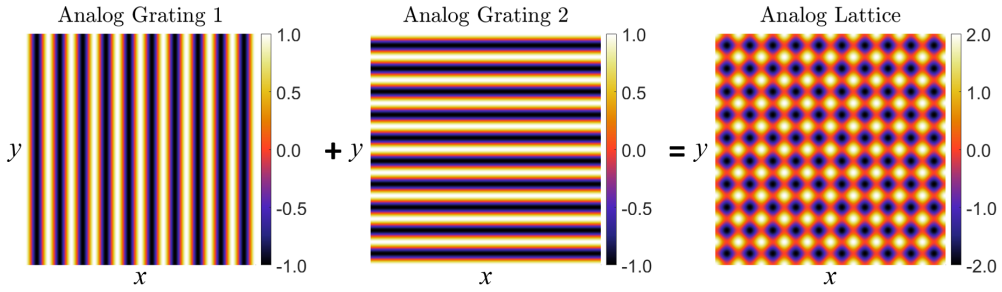

FIGURE 3. Two analog gratings produced using $\vec{T}_1 = (2\pi/\Lambda)\hat{a}_x$ and $\vec{T}_2 = (2\pi/\Lambda)\hat{a}_y$ over a flat surface, creating an analog lattice with square symmetry when combined.

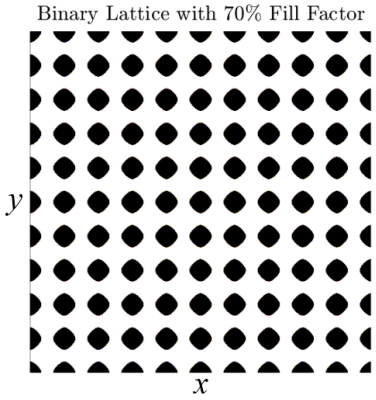

FIGURE 4. Flat surface binary lattice produced by applying a 70% fill factor to the analog lattice.

3.2. 2D Doubly-Curved Surface

In the case of a 2D doubly-curved surface, the SVLA modifications presented in Ref. [19] were implemented for the FDM SVLA to compare results against the FEM SVLA. These modifications projected a doubly-curved surface onto a flat surface where the 2D FDM SVLA could be easily applied. However, the FEM version of the SVLA results presented in this section do not project the doubly-curved surface onto a flat surface. Instead, the full curvature of the surface is taken into account by the FEM SVLA.

Similar to the 2D flat surface example, this symmetry used two reciprocal lattice vectors that started as $\vec{T}_1 = (2\pi/\Lambda)\hat{a}_x$ and $\vec{T}_2 = (2\pi/\Lambda)\hat{a}_y$ over the surface $z = 0.2\,[\cos(2\pi x) + \sin(2\pi y)]$. The grating vectors were defined uniformly across the entire grid, with a lattice period of $\Lambda = 0.1\,\text{m}$. However, due to the curvature of the surface, the reciprocal lattice vectors were rotated to conform to the mesh and point along the same direction as the surface. This modified them to $\vec{T}_1 = (2\pi/\Lambda)\hat{a}_i$ and $\vec{T}_2 = (2\pi/\Lambda)\hat{a}_j$ where $\hat{a}_i$ and $\hat{a}_j$ are the directions pointing along the principle axes of the lattice over the doubly-curved surface. The lattice was produced using ten periods along $\hat{a}_i$ and $\hat{a}_j$. This produced grating vectors that were conformed to the doubly-curved surface as shown in Fig. 6.

The same modifications presented in Ref. [19] were also implemented in the FEM SVLA initially where the curvature of the surface was projected onto a flat surface. This was done for verification purposes but those results were not included here since this FEM SVLA formulation does not require such modifications. This makes the FEM SVLA a more general and useful implementation. The FEM SVLA results presented from this point forward include the full 3D coordinates of the doubly-curved surfaces without any simplifications.

Results of the doubly-curved surface SVL using the grating vectors from Fig. 6 are presented in Fig. 7 which show that they are in good agreement. A side and top view of the doubly-curved surface with their respective analog grating vectors and SVLs can be compared visually. In addition, a frequency selective surface composed of rings, Fig. 8, was designed using the analog SVL from Fig. 7. The coordinate placement points for the FSS elements were obtained from the positions of the maxima of the analog SVL. This demonstrates the elegant periodic designs that can be achieved using the SVLA in electromagnetic structures. This has been shown by previous authors who devised the SVLA using the FDM [19].

The FDM and FEM SVLAs were solved in MATLAB using the preconditioned conjugate gradient (PCG) iterative solver using a CPU and GPU with the same mesh (quadrilaterals) and grating vectors, equal total number of degrees of freedom, and a minimal relative residual tolerance of $1e^{-5}$. The total number of iterations in the solvers and computation time were recorded and are shown in Table 1. Different preconditioners were applied to lower the total number of iterations. These were the incomplete LU [36] and incomplete Cholesky factorizations [37] which were applied to the FDM and FEM SVLAs. It is clear from the results in Table 1 that the GPU PCG solver is by far the fastest, even with a high number of iterations. However, if a CPU is only available for computation, then the CPU ILU PCG is the next best solver. This is true for the FDM and FEM SVLAs.

The analog PSVL results presented for the surfaces using FDM and FEM SVLAs were conducted using the exactly same meshes for a direct numerical comparison. The FDM flat and curved SVLs were compared against their FEM counterpart results. The relative residual was computed with the FDM results being the target of the FEM results. The analog SVL relative error was found to be low as Table 2 shows.

3.3. 3D Lattice

The 3D FDM SVLA has been used in the past to design self-collimating photonic crystals in grids discretized with rectangular bricks. For the 3D example, a cylindrical photonic crystal

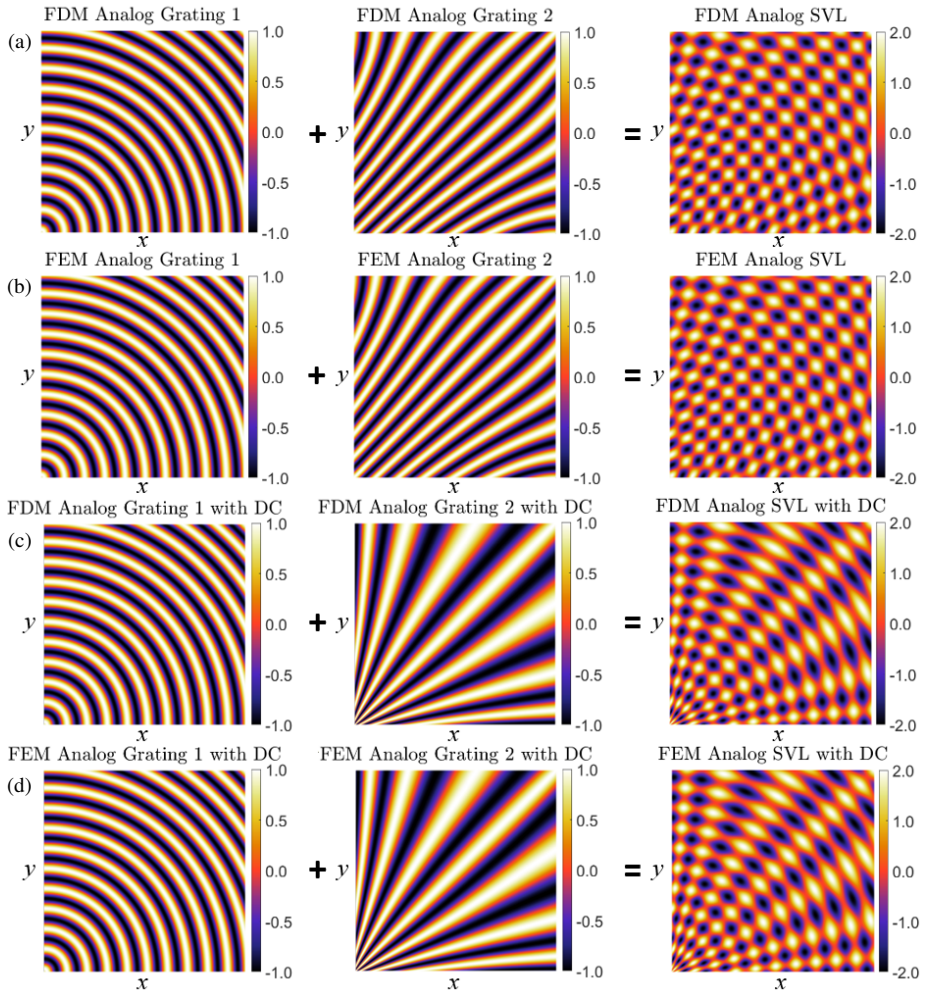

FIGURE 5. (a) Flat surface analog gratings and analog SVL with a 90° bend generated using the FDM SVLA. (b) Flat surface analog gratings and analog SVL with a 90° bend generated using the FEM SVLA. (c) Flat surface analog gratings and analog SVL with a 90° bend generated using the FDM SVLA with ten orientation DC iterations. (d) Flat surface analog gratings and analog SVL with a 90° bend generated using the FEM SVLA with ten orientation DC iterations.

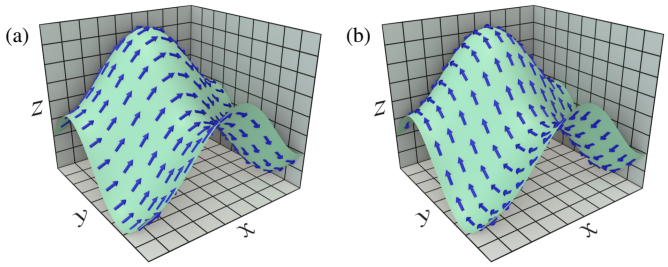

FIGURE 6. Grating vectors of the 2D doubly-curved surface. (a) First grating vector. (b) Second grating vector.

was designed to show the capabilities of the FEM SVLA. The reciprocal lattice vectors describing this 3D PSVL were as follows, $\vec{T}_1 = (2\pi/\Lambda)\hat{a}_x$, $\vec{T}_2 = (2\pi/\Lambda)\hat{a}_y$, and $\vec{T}_3 = (2\pi/\Lambda)\hat{a}_z$, where the lattice period was $\Lambda = 0.1\,\text{m}$. The x and y coordinates were constrained by radius $R = 0.25\,\text{m}$ with the z coordinate by a height of $h = 0.5\,\text{m}$. The grating vectors were rotated along the z axis as a function of height without any DC iterations. This was done using a rotation matrix

$$R_z = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{36}$$

The rotation was applied from 0 to 90° as a function of height to the grating vectors where at $z = 0$ there was no rotation, and

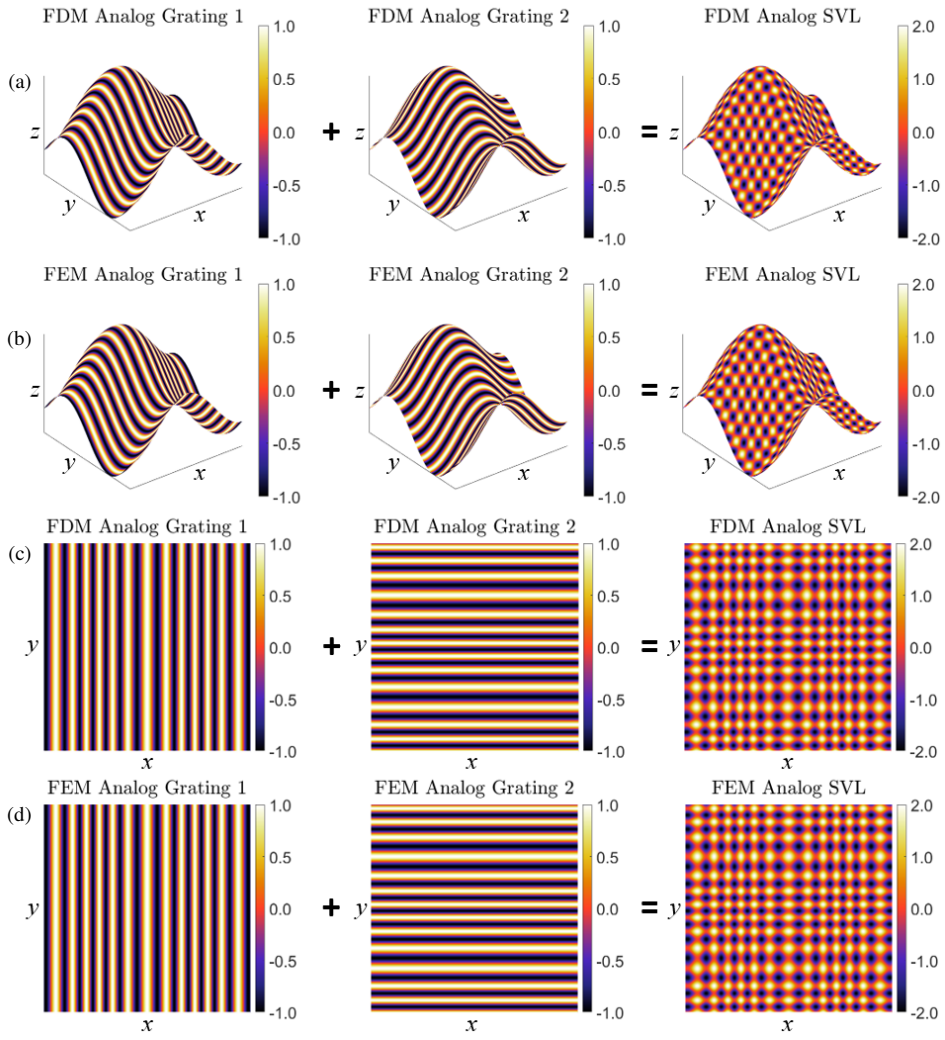

FIGURE 7. (a) Doubly-curved surface analog gratings and analog SVL using the FDM SVLA. (b) Doubly-curved surface analog gratings and analog SVL using the FEM SVLA. (c) Top view of the doubly-curved surface analog gratings and analog SVL using the FDM SVLA. (d) Top view of the doubly-curved surface analog gratings and analog SVL using the FEM SVLA.

TABLE 1. FDM and FEM SVLA iterative solver comparison.

Solver	Computation Time (s)	Total Iterations	Relative Residual
FDM CPU PCG	21.31	1730	$9.8e^{-6}$
FDM GPU PCG	1.46	1728	$9.8e^{-6}$
FDM CPU ILU PCG	14.41	425	$9.9e^{-6}$
FDM CPU ICHOL PCG	14.6	425	$9.9e^{-6}$
FEM CPU PCG	31.62	1942	$1.0e^{-5}$
FEM GPU PCG	1.81	1942	$1.0e^{-5}$
FEM CPU ILU PCG	29.48	646	$9.8e^{-6}$
FEM CPU ICHOL PCG	29.66	646	$9.8e^{-6}$

at $z = 0.5\,\text{m}$ the rotation matrix was $z = 90°$. This results in the binary PSVL shown in Fig. 9 where the analog PSVL was applied with a "greater than or equal to" threshold of 0.5 to produce a fill factor of 33.65%.

There are cases where the direction of the grating vectors produces a configuration that does not have a solution. For example, any time the grating vector function forms a closed loop, there is not direct way to obtain a solution for grating phase.

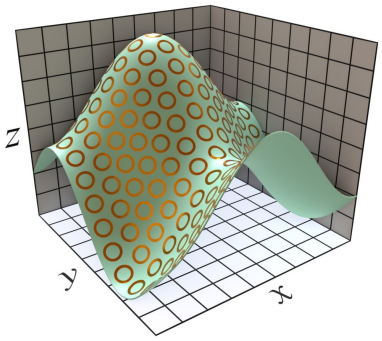

FIGURE 8. Frequency selective surface example created using the results from the doubly-curved surface SVL.

TABLE 2. 2D FDM and FEM SVL surfaces relative error.

Surface Type	$\frac{\|\Phi_{FDM}-\Phi_{FEM}\|}{\|\Phi_{FDM}\|}$
Flat Surface SVL	$3.0e^{-3}$
Flat Surface SVL with ten DC Iterations	$4.5e^{-4}$
Curved Surface SVL	$1.5e^{-4}$
Curved Surface SVL with ten DC Iterations	$1.4e^{-4}$

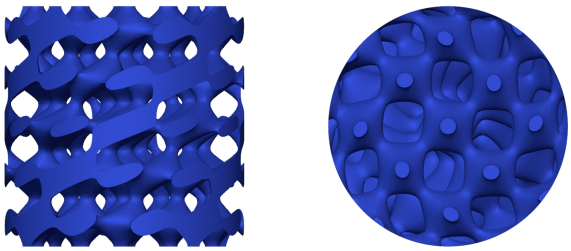

FIGURE 9. 3D SVL twisted over its height.

This happens because the RHS of Eq. (4) is divergence-free (zero) under grating vectors that form loops. To remedy this, a seam is introduced into the mesh to breaks the loops.

In the example shown in Fig. 10, a negative uniaxial metamaterial (NUM) [38] is spatially varied to create what is known as a spatially-variant anisotropic metamaterial (SVAM) [39]. This metamaterial is formed by alternating two layers with different permittivity values, effectively stretching space electromagnetically along the direction of the layers. These alternating layers result in an effective medium that exhibits uniaxial anisotropy as a function of position. This concept was previously used to achieve electromagnetic near-field sculpting to isolate a metal ball from a microstrip transmission line [40]. In Ref. [20], the Voronoi algorithm was used to generate two alternating layers that surround the electronic components of a volumetric circuit to reduce its electromagnetic coupling and interference between adjacent components. This concept is another application for the FEM SVLA to aid in volumetric circuit design.

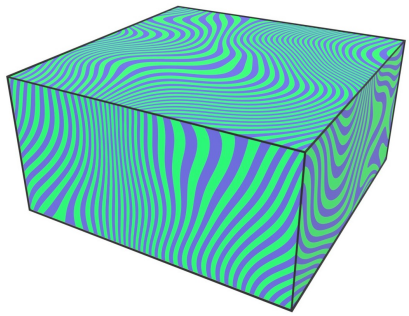

FIGURE 10. 3D example of a spatially-variant anisotropic metamaterial (SVAM).

4. CONCLUSIONS

The SVLA that was previously formulated in the FDM framework is now presented in the context of FEM. Examples showing comparisons between FDM and FEM producing SVLs over a flat surface and a doubly-curved surface were shown to be in good agreement. In addition, an example of how the SVLA can be used to design FSSs was provided. The presented volumetric PSVL example shows that this formulation also works well in 3D. Furthermore, a 3D example was provided where the SVLA produced a negative uniaxial metamaterial that was spatially varied. A simple study on different preconditioners for PCG was conducted in MATLAB and compared in overall computation time and total number of iterations. Out of all solvers used in Table 1, the GPU PCG solver was the best option to use for the FEM SVLA but if a GPU is not available, then the CPU ILU PCG is the next best solver.

The main strengths in using a FEM based SVLA is in the ability to easily refine or coarsen a mesh based on location and in using different types of meshing elements such as triangles, quadrilaterals, tetrahedrons, and hexahedrons to name a few. Complex geometries pose a challenge for the FDM because it performs best with structured orthogonal grids composed of quadrilaterals or hexahedra. Implementing such grids for intricate shapes can be problematic. In FEM, however, this is not a limitation due to the many meshing discretization options. For example, an intricate geometry can be partitioned into different regions, each meshed with different element shapes that may be more appropriate for those specific sections. This meshing freedom facilitates generating meshes with complex geometries that would otherwise be difficult to mesh with quadrilaterals or hexahedra for the FDM.

ACKNOWLEDGEMENT

Edgar Bustamante would like to thank Dr. Ryan Chilton and Dr. Denis Ridzal for their FEM conversations that helped inspire this work.

APPENDIX A. ELEMENT QUADRATURE RULES AND BASIS FUNCTIONS

The basis functions and Gaussian quadrature rules used in this work for triangles, quadrilaterals, tetrahedrons, and hexahedrons are shown in this section. The basis functions $N_j(\vec{s})$ are

TABLE A1. Triangle Gaussian quadrature rule.

Quadrature Points (i)	ξ_i	η_i	Weights (w_i)
1	$1/2$	$1/2$	$1/6$
2	$1/2$	0	$1/6$
3	0	$1/2$	$1/6$

TABLE A2. Tetrahedron Gaussian quadrature rule.

Quadrature Points (i)	ξ_i	η_i	ζ_i	Weights (w_i)
1	$\frac{(5-\sqrt{5})}{20}$	$\frac{(5-\sqrt{5})}{20}$	$\frac{(5-\sqrt{5})}{20}$	$1/24$
2	$\frac{(5+3\sqrt{5})}{20}$	$\frac{(5-\sqrt{5})}{20}$	$\frac{(5-\sqrt{5})}{20}$	$1/24$
3	$\frac{(5-\sqrt{5})}{20}$	$\frac{(5+3\sqrt{5})}{20}$	$\frac{(5-\sqrt{5})}{20}$	$1/24$
4	$\frac{(5-\sqrt{5})}{20}$	$\frac{(5-\sqrt{5})}{20}$	$\frac{(5+3\sqrt{5})}{20}$	$1/24$

defined at the quadrature points (i) to approximate the grating phase inside each finite element. The partial derivatives of the basis functions are obtained analytically for the Jacobian matrices as shown in the triangle element section. The rest of the analytical basis function partial derivatives belonging to the other element shapes can be easily calculated. Tables A1–A4 provide the Gaussian quadrature rules for each element. These include the quadrature points and their respective weights.

A.1. Triangle Element

Linear basis functions and their partial derivatives:

$$N_1(\vec{s}) = 1-\xi-\eta \tag{A1}$$
$$N_2(\vec{s}) = \xi \tag{A2}$$
$$N_3(\vec{s}) = \eta \tag{A3}$$

$$\begin{bmatrix} \frac{\partial N_1(\vec{s})}{\partial \xi} & \frac{\partial N_2(\vec{s})}{\partial \xi} & \frac{\partial N_3(\vec{s})}{\partial \xi} \\ \frac{\partial N_1(\vec{s})}{\partial \eta} & \frac{\partial N_2(\vec{s})}{\partial \eta} & \frac{\partial N_3(\vec{s})}{\partial \eta} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{A4}$$

A.2. Tetrahedron Element

Linear basis functions:

$$N_1(\vec{s}) = 1-\xi-\eta-\zeta \tag{A5}$$
$$N_2(\vec{s}) = \xi \tag{A6}$$
$$N_3(\vec{s}) = \eta \tag{A7}$$
$$N_4(\vec{s}) = \zeta \tag{A8}$$

A.3. Quadrilateral Element

Linear basis functions:

$$N_1(\vec{s}) = \frac{1}{4}(1-\xi)(1-\eta) \tag{A9}$$
$$N_2(\vec{s}) = \frac{1}{4}(1+\xi)(1-\eta) \tag{A10}$$
$$N_3(\vec{s}) = \frac{1}{4}(1+\xi)(1+\eta) \tag{A11}$$

TABLE A3. Quadrilateral Gaussian quadrature rule.

Quadrature Points (i)	ξ_i	η_i	Weights (w_i)
1	$-1/\sqrt{3}$	$-1/\sqrt{3}$	1
2	$1/\sqrt{3}$	$-1/\sqrt{3}$	1
3	$1/\sqrt{3}$	$1/\sqrt{3}$	1
4	$-1/\sqrt{3}$	$1/\sqrt{3}$	1

TABLE A4. Hexahedron Gaussian quadrature rule.

Quadrature Points (i)	ξ_i	η_i	ζ_i	Weights (w_i)
1	$-1/\sqrt{3}$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	1
2	$1/\sqrt{3}$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	1
3	$1/\sqrt{3}$	$1/\sqrt{3}$	$-1/\sqrt{3}$	1
4	$-1/\sqrt{3}$	$1/\sqrt{3}$	$-1/\sqrt{3}$	1
5	$-1/\sqrt{3}$	$-1/\sqrt{3}$	$1/\sqrt{3}$	1
6	$1/\sqrt{3}$	$-1/\sqrt{3}$	$1/\sqrt{3}$	1
7	$1/\sqrt{3}$	$1/\sqrt{3}$	$1/\sqrt{3}$	1
8	$-1/\sqrt{3}$	$1/\sqrt{3}$	$1/\sqrt{3}$	1

$$N_4(\vec{s}) = \frac{1}{4}(1-\xi)(1+\eta) \tag{A12}$$

A.4. Hexahedron Element

Linear basis functions:

$$N_1(\vec{s}) = \frac{1}{4}(1-\xi)(1-\eta)(1-\zeta) \tag{A13}$$
$$N_2(\vec{s}) = \frac{1}{4}(1+\xi)(1-\eta)(1-\zeta) \tag{A14}$$
$$N_3(\vec{s}) = \frac{1}{4}(1+\xi)(1+\eta)(1-\zeta) \tag{A15}$$
$$N_4(\vec{s}) = \frac{1}{4}(1-\xi)(1+\eta)(1-\zeta) \tag{A16}$$
$$N_5(\vec{s}) = \frac{1}{4}(1-\xi)(1-\eta)(1+\zeta) \tag{A17}$$
$$N_6(\vec{s}) = \frac{1}{4}(1+\xi)(1-\eta)(1+\zeta) \tag{A18}$$
$$N_7(\vec{s}) = \frac{1}{4}(1+\xi)(1+\eta)(1+\zeta) \tag{A19}$$
$$N_8(\vec{s}) = \frac{1}{4}(1-\xi)(1+\eta)(1+\zeta) \tag{A20}$$

REFERENCES

[1] Jin, J.-M., *The Finite Element Method in Electromagnetics*, 3rd ed., John Wiley & Sons, Hoboken, New Jersey, 2015.

[2] Hrennikoff, A., "Solution of problems of elasticity by the framework method," *J. Appl. Mech.*, Vol. 8, No. 4, A169–A175, 1941.

[3] Anderson, D., J. C. Tannehill, R. H. Pletcher, R. Munipalli, and V. Shankar, *Computational Fluid Mechanics and Heat Transfer*, 4th ed., 974, CRC Press, Boca Raton, 2020.

[4] Rahman, T. and J. Valdman, "Fast MATLAB assembly of FEM matrices in 2D and 3D: Nodal elements," *Applied Mathematics and Computation*, Vol. 219, No. 13, 7151–7158, 2013.

[5] Anjam, I. and J. Valdman, "Fast MATLAB assembly of FEM matrices in 2D and 3D: Edge elements," *Applied Mathematics and Computation*, Vol. 267, 252–263, 2015.

[6] Özgün, O. and M. Kuzuoğlu, *MATLAB-based Finite Element Programming in Electromagnetic Modeling*, 427, CRC Press, Boca Raton, 2018.

[7] Munk, B. A., *Frequency Selective Surfaces: Theory and Design*, John Wiley & Sons, 2005.

[8] Joannopoulos, J., S. Johnson, J. Winn, and R. Meade, "Photonic crystals molding the flow of light second edition introduction," in *Photonic Crystals: Molding the Flow of Light*, 1–5, Princeton University Press, 2008.

[9] Ichikawa, H., "Electromagnetic analysis of diffraction gratings by the finite-difference time-domain method," *Journal of the Optical Society of America A*, Vol. 15, No. 1, 152–157, 1998.

[10] Schurig, D., J. J. Mock, B. J. Justice, S. A. Cummer, J. B. Pendry, A. F. Starr, and D. R. Smith, "Metamaterial electromagnetic cloak at microwave frequencies," *Science*, Vol. 314, No. 5801, 977–980, 2006.

[11] Rumpf, R. C. and J. Pazos, "Synthesis of spatially variant lattices," *Optics Express*, Vol. 20, No. 14, 15 263–15 274, 2012.

[12] Rumpf, R. C., J. Pazos, C. R. Garcia, L. Ochoa, and R. Wicker, "3D printed lattices with spatially variant self-collimation," *Progress In Electromagnetics Research*, Vol. 139, 1–14, 2013.

[13] Digaum, J. L., J. J. Pazos, J. Chiles, J. D'Archangel, G. Padilla, A. Tatulian, R. C. Rumpf, S. Fathpour, G. D. Boreman, and S. M. Kuebler, "Tight control of light beams in photonic crystals with spatially-variant lattice orientation," *Optics Express*, Vol. 22, No. 21, 25 788–25 804, 2014.

[14] Rumpf, R. C., C. R. Garcia, H. H. Tsang, J. E. Padilla, and M. D. Irwin, "Electromagnetic isolation of a microstrip by embedding in a spatially variant anisotropic metamaterial," *Progress In Electromagnetics Research*, Vol. 142, 243–260, 2013.

[15] Pazos, J. J., "Digitally manufactured spatially variant photonic crystals," Ph.D. dissertation, The University of Texas at El Paso, TX, USA, 2014.

[16] Berry, E. A. and R. C. Rumpf, "Generating spatially-variant metamaterial lattices designed from spatial transforms," *Progress In Electromagnetics Research M*, Vol. 92, 103–113, 2020.

[17] Gutierrez, J. J., N. P. Martinez, and R. C. Rumpf, "Independent control of phase and power in spatially variant self-collimating photonic crystals," *Journal of the Optical Society of America A*, Vol. 36, No. 9, 1534–1539, 2019.

[18] Gutierrez, J. J., "Independent and simultaneous control of electromagnetic wave properties in self-collimating photonic crystals using spatial variance," Ph.D. dissertation, The University of Texas at El Paso, TX, USA, 2020.

[19] Valle, C. L., G. T. Carranza, and R. C. Rumpf, "Conformal frequency selective surfaces for arbitrary curvature," *IEEE Transactions on Antennas and Propagation*, Vol. 71, No. 1, 612–620, 2022.

[20] Gulib, A. U. H., J. Dumas, C. L. Valle, E. Bustamante, D. Panozzo, and R. C. Rumpf, "Generation of spatially-variant anisotropic metamaterials in 3D volumetric circuits," *Progress In Electromagnetics Research C*, Vol. 134, 93–102, 2023.

[21] Martinez, M. F., J. J. Gutierrez, J. E. Touma, and R. C. Rumpf, "Formulation of iterative finite-difference method for generating large spatially variant lattices," *Applied Computational Electromagnetics Society Journal*, Vol. 37, No. 2, 141–148, 2022.

[22] Begam, N., S. Saha, S. Sarkar, D. C. Sarkar, and P. Sarkar, "Design of compact patch type curved frequency selective surface," *International Journal of RF and Microwave Computer-Aided Engineering*, Vol. 29, No. 9, e21803, 2019.

[23] Begam, N., P. Samaddar, S. Saha, S. Sarkar, D. Chanda, and P. P. Sarkar, "Design of curved frequency selective surface with high roll off," *Microwave and Optical Technology Letters*, Vol. 59, No. 10, 2660–2664, 2017.

[24] Ehrenberg, I. M., S. E. Sarma, and B.-I. Wu, "Fully conformal FSS via rapid 3D prototyping," in *Proceedings of the 2012 IEEE International Symposium on Antennas and Propagation*, 1–2, Chicago, IL, USA, Jul. 2012.

[25] Calhoun, D. A., C. Helzel, and R. J. LeVeque, "Logically rectangular grids and finite volume methods for PDEs in circular and spherical domains," *SIAM Review*, Vol. 50, No. 4, 723–752, 2008.

[26] Chen, H. and J. Bishop, "Delaunay triangulation for curved surfaces," *Meshing Roundtable*, 115–127, 1997.

[27] LeVeque, R. J., "Finite difference methods for differential equations," *Draft Version for Use in AMath*, Vol. 585, No. 6, 112, 1998.

[28] Shashkov, M., *Conservative Finite-Difference Methods on General Grids*, CRC Press, 2018.

[29] Crandall, S. H., *Engineering Analysis: A Survey of Numerical Procedures*, 417, McGraw-Hill, 1956.

[30] Finlayson, B. A. and L. E. Scriven, "The method of weighted residuals — A review," *Appl. Mech. Rev.*, Vol. 19, No. 9, 735–748, 1966.

[31] Galerkin, B. G., "Series solution of some problems of elastic equilibrium of rods and plates," *Vestnik Inzhenerov I Tekhnikov*, Vol. 19, No. 7, 897–908, 1915.

[32] Dunavant, D. A., "High degree efficient symmetrical Gaussian quadrature rules for the triangle," *International Journal for Numerical Methods in Engineering*, Vol. 21, No. 6, 1129–1148, 1985.

[33] Keast, P., "Moderate-degree tetrahedral quadrature formulas," *Computer Methods in Applied Mechanics and Engineering*, Vol. 55, No. 3, 339–348, 1986.

[34] Zhang, L., T. Cui, and H. Liu, "A set of symmetric quadrature rules on triangles and tetrahedra," *Journal of Computational Mathematics*, Vol. 27, No. 1, 89–96, 2009.

[35] Barrett, R., M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, 1994.

[36] Saad, Y., *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, 2003.

[37] Manteuffel, T. A., "An incomplete factorization technique for positive definite linear systems," *Mathematics of Computation*, Vol. 34, No. 150, 473–497, 1980.

[38] Avila, J., C. L. Valle, E. Bustamante, and R. C. Rumpf, "Optimization and characterization of negative uniaxial metamaterials," *Progress In Electromagnetics Research C*, Vol. 74, 111–121, 2017.

[39] Gulib, A. U. H., J. Dumas, C. L. Valle, E. Bustamante, D. Panozzo, and R. C. Rumpf, "Generation of spatially-variant anisotropic metamaterials in 3D volumetric circuits," *Progress In Electromagnetics Research C*, Vol. 134, 93–102, 2023.

[40] Rumpf, R. C., J. J. Pazos, J. L. Digaum, and S. M. Kuebler, "Spatially variant periodic structures in electromagnetics," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 373, No. 2049, 20140359, 2015.