Fast Method for Accelerating Convergence of Iterative Partial Differential Equation Solvers by Changing System Matrix to Laplacian Counterpart

Li Xue, *Student Member, IEEE*, and Dan Jiao, *Fellow, IEEE*

***Abstract*— In this work, we find that the matrix representing the curl–curl operator in a partial differential equation solver of Maxwell's equations can be analytically decomposed into a gradient divergence operator and a Laplacian, both of which can be constructed from the mesh information without any need for computation. The curl–curl operator can hence be replaced by the Laplacian to find the divergence-free component of the field solution. The Laplacian is positive definite and well-conditioned. As a result, the convergence of an iterative solution of Maxwell's equations can be guaranteed, and also significantly accelerated. Based on the finding, we represent the divergence-free component of the unknown field solution by deducting its curl-free component. The curl-free component resides in the nullspace of the curl–curl operator, which is also analytically known from the mesh information no matter it is a regular grid or an unstructured mesh. After the divergence-free component is rapidly solved from a Laplacian counterpart of the original system matrix, the curl-free component can also be solved from a Laplacian matrix, and hence having fast and guaranteed convergence. The total computational cost of the proposed method is simply a small number of sparse matrix-vector multiplications. The proposed method has been successfully applied to solve ill-conditioned on-chip, packaging, and antenna radiation problems at both low and high frequencies, involving both inhomogeneous dielectrics and lossy conductors. Numerical experiments have demonstrated its fast and guaranteed convergence, as well as trivial computational cost independent of problem size.**

***Index Terms*— Fast convergence, fast method, finite-difference method, finite-element method, frequency domain, Helmholtz decomposition, iterative solver, partial differential equation method.**

I. Introduction

THE system matrix resulting from a partial differential equation (PDE) based solution of Maxwell's equations in frequency domain is indefinite, involving both negative and positive eigenvalues. When loss especially conductor loss is involved, the system matrix is complex valued and unbalanced, which is even more difficult for an iterative solution to converge. It has also been found that iterative solutions are difficult to converge at low frequencies, in the presence of multiscaled geometries, and when problem size is large.

Existing techniques for expediting iterative PDE solutions are mainly based on finding a good preconditioner for solving the underlying system of equations [1]–[5]. To give a few examples, in [2], a preconditioner is developed from a good approximation of the original problem to solve the finite element-boundary integral system of equations; in [3], Jacob, symmetric successive over-relaxation (SSOR), and sparse approximate inverse (SAI) preconditioners are explored to accelerate generalized minimal residual method (GMRES) in an implicit time-domain finite-difference method; in [4], the SSOR preconditioner is used with a conjugate gradient (CG) iterative solver to expedite a finite-element method; and in [5], a preconditioner is constructed from a simplified version of the original finite-element system matrix to analyze layered structures. Among these methods, diagonal, block diagonal, Jacob, SSOR, and similar preconditioners are computationally efficient to construct. But their performance is problem dependent, and their convergence is not guaranteed when solving an ill-conditioned full-wave system of equations. Approximate inverse or incomplete factorization-based preconditioners are more robust and can exhibit a faster convergence. However, they are computationally expensive. Other preconditioners suffer from a similar performance and cost trade-off. In this work, instead of devising a good preconditioner, we propose and develop a different method to expedite the convergence of an iterative PDE-based solution of Maxwell's equations in frequency domain.

Our method is based on the following important finding. The curl–curl operator (stiffness matrix) resulting from the discretization of Maxwell's equations can be rigorously decomposed into a gradient divergence operator and a Laplacian, both of which can be constructed from the mesh information without any need for computation. Meanwhile, the field solution can be decomposed into a gradient field and a divergence-free component. And the column space of the gradient field can be analytically found from the mesh information. The gradient divergence operator of the curl–curl operator vanishes when operating on the divergence-free component of the field

Manuscript received November 11, 2020; revised June 10, 2021; accepted July 20, 2021. Date of publication September 15, 2021; date of current version February 3, 2022. This work was supported by the Defense Advanced Research Projects Agency (DARPA) through a grant under Award FA8650-18-2-7847. *(Corresponding author: Dan Jiao.)*

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: djiao@purdue.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAP.2021.3111509.

Digital Object Identifier 10.1109/TAP.2021.3111509

0018-926X © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Purdue University. Downloaded on May 12,2022 at 14:05:32 UTC from IEEE Xplore. Restrictions apply.

solution, and hence the singular curl–curl operator can be replaced by the Laplacian without any approximation. This Laplace operator shares the same nonzero eigenvalues and eigenvectors of the curl–curl operator. Since the Laplacian is positive definite and well-conditioned, the resultant iterative solution has guaranteed convergence. Iterative methods such as a multigrid method [1], [6] have been shown to converge the solution of a Laplacian matrix in a constant and small number of steps irrespective of matrix size. Based on the aforementioned idea, we develop a fast iterative solution to solve Maxwell's equations in frequency domain. We have also successfully applied it to solve a wide range of problems such as highly ill-conditioned on-chip integrated circuits where static- and full-wave components of the field solution coexist, and fields penetrate into lossy conductors, and integrated packaging and antenna radiation problems whose electrical sizes are much larger. The high-performance and trivial cost of the proposed method have been demonstrated.

This article is a significant expansion of our two-page conference paper [7]. In this article, we complete the development of the algorithm and the theory behind the algorithm, and also present them in detail. In addition, we conduct extensive numerical experiments to validate its accuracy and efficiency from structures in the deep sub-wavelength regime, to structures that are multiscaled where sub-wavelength and multiple-wavelength features coexist, to structures without sub-wavelength features and dominated by high-frequency field solutions. The remainder of this article is organized as follows. In Section II, the problem under study is introduced. In Section III, the essential idea of the proposed work is elaborated, where the discretized curl–curl operator is analytically decomposed into a Laplacian matrix and a gradient divergence matrix without computation. We also show the discretized curl–curl operator can be replaced by the Laplacian matrix rigorously when operating on the divergence-free component of the field solution, and explain how to use this fact to expedite an iterative solution. In Section IV, a fast method is developed based on the proposed idea to accelerate the convergence of an iterative PDE-based solution of Maxwell's equations in frequency domain. In Section V, we apply the proposed method to solve a wide range of electromagnetic problems from on-chip circuits to antenna radiation. Numerical experiments have demonstrated the accuracy, efficiency, fast, and guaranteed convergence of the proposed method. In Section VI, we draw our conclusions.

II. Problem Statement

A frequency-domain PDE-based solution of Maxwell's equations, in general, results in the following linear system of equations:

$$\left(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma + \overline{\mathbf{S}}\right)\mathbf{e} = -j\omega\mathbf{J} \tag{1}$$

where ω is the angular frequency, $\mathbf{e}$ denotes a vector of electric field unknowns, $\overline{\mathbf{D}}_\epsilon$ matrix is associated with permittivity, $\overline{\mathbf{D}}_\sigma$ matrix is with conductivity, and $\mathbf{J}$ denotes a current source vector. Regardless of the PDE method used to discretize Maxwell's equations, be it a finite-difference method (FDM), a finite-element method (FEM), a finite-volume method (FVM), or others, the final numerical system all can be cast in the form shown in (1). All matrices are sparse. In the FDM, the $\overline{\mathbf{D}}_\epsilon$ and $\overline{\mathbf{D}}_\sigma$ are also diagonal.

The $\overline{\mathbf{S}}$ in (1) represents a discretized $\nabla\times\mu^{-1}\nabla\times$ operator. Its smallest eigenvalue is zero since when $\overline{\mathbf{S}}$ operates on a gradient field, the result is zero. Its largest one is inversely proportional to the square of the smallest mesh size, since the matrix entry is obtained by taking the spatial derivatives twice. The ratio of $\omega^2\overline{\mathbf{D}}_\epsilon$ to $\overline{\mathbf{S}}$ is proportional to the square of the electrical size of the geometrical features [8]. This renders an iterative solution of (1) difficult to converge, especially when the frequency is low, electrical size is small, or the problem is multiscaled in geometry resulting in a high condition number of the numerical system. The presence of dielectric or conductor loss, characterized by the $\overline{\mathbf{D}}_\sigma$-term, makes an iterative solution of (1) even more difficult to converge. The iteration number, in general, grows with problem size as well. Larger problems typically require more iterations to achieve convergence. In next section, we present the essential idea of this work to address the aforementioned challenges.

III. Proposed Idea

In view of the following vector identity:

$$\nabla\times(\nabla\times\mathbf{A}) = \nabla(\nabla\cdot\mathbf{A}) - \nabla^2\mathbf{A} \tag{2}$$

we find that $\overline{\mathbf{S}}$ can be analytically decomposed into

$$\overline{\mathbf{S}} = -\overline{\mathbf{V}}_0\overline{\mathbf{V}}_{0a}^T/\mu + \overline{\mathbf{L}} \tag{3}$$

where $\overline{\mathbf{V}}_0$ represents a discretized form of the gradient operation, $\overline{\mathbf{V}}_{0a}^T$ denotes a discretized form of the divergence operation, $-\overline{\mathbf{V}}_0\overline{\mathbf{V}}_{0a}^T$ denotes a discretized form of $\nabla(\nabla\cdot)$, and $\overline{\mathbf{L}}$ corresponds to the discretized form of Laplacian $-\nabla^2$ divided by μ. If permeability is inhomogeneous, (3) can be modified to suit the scenario.

The above description might be abstract. To explain it more clearly, consider a scalar field $a(\mathbf{r})$, which is a function of 3-D coordinate $\mathbf{r}$. Numerically, this scalar field is represented by a column vector after discretization, with each entry representing the a at a space point. Let this column vector be denoted by $\mathbf{a}$. It can be written as $\mathbf{a} = [a(\mathbf{r}_1), a(\mathbf{r}_2), \ldots, a(\mathbf{r}_n)]^T$. Then, $\overline{\mathbf{V}}_0\mathbf{a}$ represents discretized ∇a, thus a gradient operation. So mathematically, $\overline{\mathbf{V}}_0$ is a matrix, but when it is multiplied by a column vector, the resultant matrix-vector multiplication means to take the gradient of the scalar field represented by the column vector. The $\overline{\mathbf{V}}_0$ has multiple rows also. The row index denotes the point where the gradient field is evaluated. Similarly, $\overline{\mathbf{V}}_{0a}^T$ operating on (multiplied by) a column vector means to take the divergence of a vector field represented by the column vector. Each entry of the column vector represents the vector field at a certain point along a certain direction after discretization. So when we use matrix $\overline{\mathbf{S}}$ to multiply a column vector, it means taking a $\nabla\times(\nabla\times)/\mu$ operation (certainly in a discretized manner) on the vector field represented by that column vector. In all, (3) is a discretized manifestation of (2).

Authorized licensed use limited to: Purdue University. Downloaded on May 12,2022 at 14:05:32 UTC from IEEE Xplore. Restrictions apply.

The $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_{0a}$ in (3) can be found from the right and left nullspace of $\overline{\mathbf{S}}$, respectively, since they satisfy

$$\overline{\mathbf{S}}\,\overline{\mathbf{V}}_0 = 0 \tag{4}$$

and

$$\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{S}} = 0. \tag{5}$$

This is because $\overline{\mathbf{V}}_0$ denotes a gradient operation, $\overline{\mathbf{S}}$ is a double curl operation, and the curl of a gradient field is known to be zero. Similarly, $\overline{\mathbf{V}}_{0a}^T$ represents a divergence operation, and hence when it operates on a curl, the resultant should be zero. The property of (4) and (5) is what we use to analytically figure out $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_{0a}$ from mesh information without numerical computation. In addition, to clarify notations used in this work, the subscript $_0$ in $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_{0a}$ denotes nullspace, $\overline{\mathbf{V}}$ denotes a group of column *v*ectors, subscript a arises from the fact that the left nullspace is related to the averaged length in a nonuniform grid.

If $\overline{\mathbf{S}}$ is symmetric

$$\overline{\mathbf{V}}_{0a} = \overline{\mathbf{V}}_0 \tag{6}$$

otherwise

$$\overline{\mathbf{V}}_{0a} \neq \overline{\mathbf{V}}_0. \tag{7}$$

A symmetrical $\overline{\mathbf{S}}$ is observed in an FEM, and an FDM in a uniform grid. In a nonuniform grid, $\overline{\mathbf{S}}$ is not symmetrical in an FDM-based solution of Maxwell's equations, and hence its left and right nullspaces are different.

The $\overline{\mathbf{V}}_0$ can be analytically generated from the mesh information without any need for computation. This analytical method is given in detail in our previous work [9], [10]. Here, for the completeness of this work, we provide a brief summary of the method. The column dimension of $\mathbf{V}_0$ is the number of free nodes minus 1 in a mesh, no matter the mesh is a grid or an unstructured tetrahedron mesh. The ith column vector of $\mathbf{V}_0$ corresponds to the ith node in a mesh, which can be generated as follows:

1) find all edges connected to the node;
2) if the edge basis enters (leaves) the node, $(1/l_i)$ $(-(1/l_i))$ appears on the row of $\mathbf{V}_{0,i}$ corresponding to this edge basis, where l_i is the edge length.

The nodes residing on a perfect conductor are grouped as one node whose nullspace vector is the sum of all nullspace vectors of the nodes on the perfect conductor. The aforementioned approach for finding $\mathbf{V}_0$ is the same for the FDM and the FEM. In a symmetric FEM formulation, the left nullspace $\mathbf{V}_{0a}$ is the same as the right one, $\mathbf{V}_0$. However, in the FDM, when the grid is not uniform, $\mathbf{V}_{0a}$ is different from $\mathbf{V}_0$. But it can also be found analytically. We simply replace l_i in the above by an average length, the detail of which can be seen from [10, Sect. IV].

Based on (3), we further find that the $\overline{\mathbf{S}}$'s nonzero eigenvalues and their corresponding eigenvectors are also $\overline{\mathbf{L}}$'s eigenvalues and eigenvectors. To see this point clearly, let $\overline{\mathbf{V}}_h$ be the union of $\overline{\mathbf{S}}$'s eigenvectors corresponding to nonzero eigenvalues whose diagonal matrix is denoted by $\overline{\Lambda}_h$. Clearly, they satisfy

$$\overline{\mathbf{S}}\,\overline{\mathbf{V}}_h = \overline{\mathbf{V}}_h\overline{\Lambda}_h. \tag{8}$$

Therefore

$$\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{S}}\,\overline{\mathbf{V}}_h = \overline{\mathbf{V}}_{0a}^T\overline{\mathbf{V}}_h\overline{\Lambda}_h. \tag{9}$$

Because of (5), the left-hand side of (9) is zero. Hence, we obtain

$$\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{V}}_h = 0. \tag{10}$$

Since the left in the above represents a discretized $\nabla\cdot\overline{\mathbf{V}}_h$, we can see that each column vector of $\overline{\mathbf{V}}_h$ represents a divergence-free field. Multiplying both sides of (3) by $\overline{\mathbf{V}}_h$, and utilizing (10), we obtain

$$\overline{\mathbf{S}}\,\overline{\mathbf{V}}_h = \overline{\mathbf{L}}\,\overline{\mathbf{V}}_h. \tag{11}$$

This proves that the nonnullspace eigenvectors and eigenvalues of $\overline{\mathbf{S}}$ are also those of $\overline{\mathbf{L}}$. This further means when operating on $\overline{\mathbf{S}}$'s nonnullspace eigenmodes, i.e., full-wave modes, we can replace $\overline{\mathbf{S}}$ by $\overline{\mathbf{L}}$! Since these modes satisfy (10), they are also divergence-free modes. On the other hand, $\overline{\mathbf{V}}_0$ is curl-free because they satisfy (4).

From (3), the difference between $\overline{\mathbf{L}}$ and $\overline{\mathbf{S}}$ can also be seen. In addition to $\overline{\mathbf{V}}_h$ shared in common with $\overline{\mathbf{S}}$, $\overline{\mathbf{L}}$ has a subspace of $\overline{\mathbf{V}}_0$, but $\overline{\mathbf{L}}\overline{\mathbf{V}}_0 \neq 0$ as can be seen from (3). Instead, $\overline{\mathbf{L}}\overline{\mathbf{V}}_0 = \overline{\mathbf{V}}_0\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{V}}_0/\mu$, thus $\overline{\mathbf{V}}_0$ is not $\overline{\mathbf{L}}$'s nullspace. The $\overline{\mathbf{L}}$ resides in the space of $\overline{\mathbf{V}}_h$ and $\overline{\mathbf{V}}_0$, which is full-rank.

Based on the aforementioned findings, the essential idea of this work to accelerate convergence is to replace the original singular matrix $\overline{\mathbf{S}}$ by its Laplacian counterpart $\overline{\mathbf{L}}$ when operating on the divergence-free component of the field solution. By doing so, we can utilize the good property of $\overline{\mathbf{L}}$, which is positive definite and well-conditioned, to develop a fast iterative solution of guaranteed convergence to solve Maxwell's equations in frequency domain.

As far as the generation of $\overline{\mathbf{L}}$ is concerned, it does not involve any computational cost. Since $\overline{\mathbf{S}}$ is known, as well as $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_{0a}$, we can utilize them to generate $\overline{\mathbf{L}}$ as the following:

$$\overline{\mathbf{L}} = \overline{\mathbf{S}} + \overline{\mathbf{V}}_0\overline{\mathbf{V}}_{0a}^T/\mu. \tag{12}$$

In a finite-difference grid, the $\overline{\mathbf{L}}$ can also be readily constructed from scratch without using $\overline{\mathbf{S}}$.

IV. Proposed Method to Accelerate Convergence of Iterative Solutions

To take advantage of the property of (11), we separate the $\overline{\mathbf{V}}_0$-component of $\mathbf{e}$ from its $\overline{\mathbf{V}}_h$ component to solve. We hence expand $\mathbf{e}$ as

$$\mathbf{e} = \overline{\mathbf{V}}_0\mathbf{y}_0 + \mathbf{e}_h. \tag{13}$$

The above is equivalent to separating the curl-free component of $\mathbf{e}$ from its divergence-free component, since $\overline{\mathbf{V}}_0\mathbf{y}_0$ is in the right nullspace of $\overline{\mathbf{S}}$, and hence being curl-free; whereas the $\mathbf{e}_h$ is in the column space of $\overline{\mathbf{V}}_h$, and hence divergence-free.

In Fig. 1, we plot a flowchart to illustrate the overall procedure of the proposed method. The original full-wave system is divided into a zero-mode subsystem, whose solution is the curl-free component; and a high-order mode subsystem, whose solution is the divergence-free component. These two subsystems are both made as Laplacian matrices, and hence

Authorized licensed use limited to: Purdue University. Downloaded on May 12,2022 at 14:05:32 UTC from IEEE Xplore. Restrictions apply.

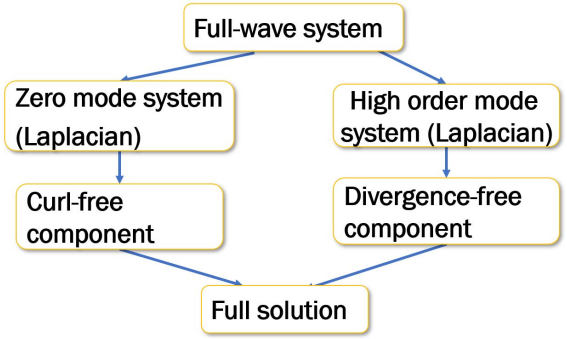


Fig. 1. Overall procedure of the proposed method.

can be solved fast iteratively with guaranteed convergence. After the two components are obtained, they are added up to obtain the final full solution.

The proposed method works well because we separate the nullspace from the high-frequency modes (whose curl is not zero) to solve, and further transform the solution of each to a Laplacian to solve. Since Laplacian has a guaranteed fast convergence, the proposed iteration solution is also fast. In contrast, in a conventional solution, the two are mixed together to solve, and one has to deal with a large eigenvalue spectrum from zero to largest eigenvalue to solve the original numerical system. In the scenario where only the curl-free component is dominant, or only the divergence-free component is dominant, only one branch of the flowchart shown in Fig. 1 is needed. In this case, the conventional solution does not suffer from the difficulty of mixing the two to solve, but can still suffer from slow convergence. In contrast, the proposed method expedites convergence since the singular $\mathbf{S}$ is replaced by a positive-definite Laplacian $\mathbf{L}$. Next, we elaborate the details of the proposed method.

A. Combined System of Equations With S Replaced by L

Substituting (13) into (1), and testing the resultant by $[\overline{\mathbf{V}}_{0a} \quad \overline{\mathbf{I}}]^T$, we transform (1) to the following to solve $\mathbf{y}_0$ and $\mathbf{e}_h$

$$\begin{bmatrix} \overline{\mathbf{V}}_{0a}^T(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_0 & \overline{\mathbf{V}}_{0a}^T(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma) \\ (-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_0 & (-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma + \overline{\mathbf{S}}) \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{e}_h \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{V}}_{0a}^T(-j\omega\mathbf{J}) \\ -j\omega\mathbf{J} \end{bmatrix}. \quad (14)$$

The $\overline{\mathbf{S}}$ vanishes in the first row of the above equation because of (5); and it vanishes in the first term of the second row because of (4).

Utilizing the property of (11), we can replace the $(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma + \overline{\mathbf{S}})$ in (14) by $(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma + \overline{\mathbf{L}})$ since that matrix operates on $\mathbf{e}_h$. Hence, (14) can be rigorously transformed to the following to solve:

$$\begin{bmatrix} \overline{\mathbf{V}}_{0a}^T(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_0 & \overline{\mathbf{V}}_{0a}^T(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma) \\ (-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_0 & (-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma + \overline{\mathbf{L}}) \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{e}_h \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{V}}_{0a}^T(-j\omega\mathbf{J}) \\ -j\omega\mathbf{J} \end{bmatrix}. \quad (15)$$

Denoting the above in short by

$$\begin{bmatrix} \overline{\mathbf{A}}_{00} & \overline{\mathbf{A}}_{0h} \\ \overline{\mathbf{A}}_{h0} & \overline{\mathbf{A}}_{hh} \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{e}_h \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{V}}_{0a}^T(-j\omega\mathbf{J}) \\ -j\omega\mathbf{J} \end{bmatrix} \quad (16)$$

where

$$\overline{\mathbf{A}}_{00} = \overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}}\overline{\mathbf{V}}_0, \quad \overline{\mathbf{A}}_{0h} = \overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}} \quad (17)$$

$$\overline{\mathbf{A}}_{h0} = \overline{\mathbf{D}}\overline{\mathbf{V}}_0, \quad \overline{\mathbf{A}}_{hh} = \overline{\mathbf{D}} + \overline{\mathbf{L}} \quad (18)$$

with

$$\overline{\mathbf{D}} = \left(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma\right). \quad (19)$$

Here, we use subscript $_h$ to denote the part corresponding to $\overline{\mathbf{V}}_h$, and $_0$ for the part corresponding to $\overline{\mathbf{V}}_0$. The matrix $\overline{\mathbf{A}}_{hh}$ now has good properties since the original ill-conditioned matrix $\overline{\mathbf{S}}$ is changed to $\overline{\mathbf{L}}$, which is a Laplacian matrix. To fully take advantage of the good property of $\overline{\mathbf{A}}_{hh}$, we solve (16) in the following way.

B. Solution of Divergence-Free Component: $\mathbf{e}_h$

First, we compute $\mathbf{e}_h$ from the following:

$$\widetilde{\mathbf{A}}_{hh}\mathbf{e}_h = (-j\omega\mathbf{J}) - \overline{\mathbf{A}}_{h0}\overline{\mathbf{A}}_{00}^{-1}\overline{\mathbf{V}}_{0a}^T(-j\omega\mathbf{J}) \quad (20)$$

where the $\widetilde{\mathbf{A}}_{hh}$ is the Schur complement of $\overline{\mathbf{A}}_{hh}$, that is,

$$\widetilde{\mathbf{A}}_{hh} = \overline{\mathbf{A}}_{hh} - \overline{\mathbf{A}}_{h0}\overline{\mathbf{A}}_{00}^{-1}\overline{\mathbf{A}}_{0h}. \quad (21)$$

This matrix appears to be complicated, and difficult to solve. In fact, it corresponds to the Laplacian counterpart of the original system matrix shown in (1) in an inhomogeneous material, and its iterative solution has a fast convergence. To see this point clearly, we can write the $\widetilde{\mathbf{A}}_{hh}$ in full, obtaining

$$\widetilde{\mathbf{A}}_{hh} = (\overline{\mathbf{D}} + \overline{\mathbf{L}}) - \overline{\mathbf{D}}\,\overline{\mathbf{V}}_0\left(\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}}\overline{\mathbf{V}}_0\right)^{-1}\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}} \quad (22)$$

which can be further rewritten in short as

$$\widetilde{\mathbf{A}}_{hh} = \overline{\mathbf{D}}_h + \overline{\mathbf{L}} \quad (23)$$

in which

$$\overline{\mathbf{D}}_h = \overline{\mathbf{D}} - \overline{\mathbf{D}}\,\overline{\mathbf{V}}_0\left(\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}}\overline{\mathbf{V}}_0\right)^{-1}\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}}. \quad (24)$$

The $\overline{\mathbf{D}}_h$ is nothing but the component of $\overline{\mathbf{D}}$ in $\overline{\mathbf{V}}_h$ space. To prove, we can expand the matrix $\overline{\mathbf{D}}$ in the space of $\overline{\mathbf{D}}\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_h$ as follows:

$$\overline{\mathbf{D}} = \overline{\mathbf{D}}\overline{\mathbf{V}}_0\overline{\mathbf{X}}_0 + \overline{\mathbf{V}}_h\overline{\mathbf{X}}_h \quad (25)$$

where $\overline{\mathbf{X}}_0$ and $\overline{\mathbf{X}}_h$ are the coefficient matrices to be found. Multiplying $\overline{\mathbf{V}}_{0a}^T$ to both sides of the above equation, using the property of $\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{V}}_h = 0$, we obtain

$$\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}} = \overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}}\overline{\mathbf{V}}_0\overline{\mathbf{X}}_0 \quad (26)$$

and hence

$$\overline{\mathbf{X}}_0 = \left(\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}}\overline{\mathbf{V}}_0\right)^{-1}\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}}. \quad (27)$$

Substituting the above into (25) and moving the $\overline{\mathbf{X}}_0$-term to the left-hand side, we have

$$\overline{\mathbf{V}}_h\overline{\mathbf{X}}_h = \overline{\mathbf{D}} - \overline{\mathbf{D}}\,\overline{\mathbf{V}}_0\left(\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}}\overline{\mathbf{V}}_0\right)^{-1}\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}} = \overline{\mathbf{D}}_h. \quad (28)$$

Authorized licensed use limited to: Purdue University. Downloaded on May 12,2022 at 14:05:32 UTC from IEEE Xplore. Restrictions apply.

Thus, (24) is nothing but the component of $\overline{\mathbf{D}}$ in $\overline{\mathbf{V}}_h$ space. Following a similar proof, we can see the right-hand side of (20) is the original right-hand side $(-j\omega\mathbf{J})$'s component in $\overline{\mathbf{V}}_h$ space. We hence denote it by

$$\mathbf{b}_h = (-j\omega\mathbf{J}) - \overline{\mathbf{A}}_{h0}\overline{\mathbf{A}}_{00}^{-1}\overline{\mathbf{V}}_{0a}^T(-j\omega\mathbf{J}). \tag{29}$$

From the above, $\widetilde{\mathbf{A}}_{hh}$ is the component of the original system matrix

$$\overline{\mathbf{A}} = -\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma + \overline{\mathbf{S}} = \overline{\mathbf{D}} + \overline{\mathbf{S}} \tag{30}$$

in $\overline{\mathbf{V}}_h$ space, where $\overline{\mathbf{S}}$ is reduced to Laplacian matrix $\overline{\mathbf{L}}$ as well. Highlighting this fact, we denote (22) by

$$\overline{\mathbf{A}}_L = \widetilde{\mathbf{A}}_{hh} = \overline{\mathbf{D}}_h + \overline{\mathbf{L}}. \tag{31}$$

We find when both curl-free and divergence-free components are important, the iterative solution of $\overline{\mathbf{A}}_L$ converges fast in a small number of steps. In contrast, the original matrix shown in (1) is difficult to converge.

It is worth mentioning that the Schur complement $\widetilde{\mathbf{A}}_{hh}$ never needs to be explicitly formed in an iterative solution. What needs to be computed is $\widetilde{\mathbf{A}}_{hh}$ multiplying a vector at each step of an iterative solution. As can be seen from (21), multiplying $\widetilde{\mathbf{A}}_{hh}$ by a vector is simply to perform a few sparse matrix multiplications, and also solving $\overline{\mathbf{A}}_{00}$ for a right-hand side vector. The former has a linear complexity, while the latter is the solution of a Laplacian system, and hence can be obtained fast. The fast solution of $\overline{\mathbf{A}}_{00}$ is given in the next section.

The aforementioned algorithm can further be simplified for electrically small problems like on-chip-integrated circuits, where the coupling is weak between the $\overline{\mathbf{V}}_0$- and $\overline{\mathbf{V}}_h$-component since $\overline{\mathbf{S}}$ dominates. In this case, the Schur complement can be accurately approximated as $\overline{\mathbf{A}}_{hh}$, hence we have

$$\overline{\mathbf{A}}_L = \widetilde{\mathbf{A}}_{hh} \approx \overline{\mathbf{A}}_{hh} = \overline{\mathbf{D}} + \overline{\mathbf{L}}. \tag{32}$$

Since $\overline{\mathbf{A}}_{hh}$ is the sum of a Laplacian and $\overline{\mathbf{D}}$ matrix, and the Laplacian is dominant in electrically small problems, iterative solvers such as GMRES or multigrid can be used to solve it fast also.

For problems where only the divergence-free component is important, we only need to solve (20) because the resultant $\mathbf{e}_h$ makes a good total solution of (1). This can happen when electrical size of the problem under study is relatively large since the curl of electric field is not zero any more, and/or under certain excitations where only divergence-free modes are excited, or under other conditions. In this case, since $\overline{\mathbf{V}}_0$-component is negligible, the Schur complement can be replaced by the $\overline{\mathbf{A}}_{hh}$ itself, hence (20) is simplified to the following to solve:

$$\left(\overline{\mathbf{D}} + \overline{\mathbf{L}}\right)\mathbf{e}_h = \mathbf{b}_h \tag{33}$$

where $\mathbf{b}_h$ is the right-hand side of (20). Since the nullspace of $\overline{\mathbf{S}}$ is not important anymore, it appears that (33) has a similar convergence performance as (1), since both rely on the $\overline{\mathbf{V}}_h$ to converge. However, because $\overline{\mathbf{L}}$ is a Laplacian and its solution is computationally trivial, it can be used as the preconditioner to solve (33) fast, i.e., solving the following:

$$\left(\overline{\mathbf{L}}^{-1}\overline{\mathbf{D}} + \overline{\mathbf{I}}\right)\mathbf{e}_h = \overline{\mathbf{L}}^{-1}\mathbf{b}_h. \tag{34}$$

In contrast, when solving the original (1), it is not feasible to use $\overline{\mathbf{S}}$ as the preconditioner to capture $\overline{\mathbf{V}}_h$ modes fast since $\overline{\mathbf{S}}$ is singular.

C. Solution of Curl-Free Component: $\overline{\mathbf{V}_0}\mathbf{y}_0$

After obtaining $\mathbf{e}_h$, we substitute it into the first equation of (16), and solve $\mathbf{y}_0$ from

$$\overline{\mathbf{A}}_{00}\mathbf{y}_0 = \overline{\mathbf{V}}_{0a}^T(-j\omega\mathbf{J}) - \overline{\mathbf{A}}_{0h}\mathbf{e}_h. \tag{35}$$

When there are no lossy conductors, $\overline{\mathbf{A}}_{00}$ is a Laplacian matrix in dielectric materials, which can be solved fast. When there are lossy conductors, to solve $\overline{\mathbf{A}}_{00}$ efficiently, we transform it to the solutions of two Laplacians as follows. We expand the curl-free component $\overline{\mathbf{V}}_0\mathbf{y}_0$ into

$$\overline{\mathbf{V}}_0\mathbf{y}_0 = \overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0c}\mathbf{y}_{0c} \tag{36}$$

where $\overline{\mathbf{V}}_{0d}$ are the $\overline{\mathbf{V}}_0$ at dielectric nodes, and $\overline{\mathbf{V}}_{0c}$ are at the conductor nodes, with $\overline{\mathbf{V}}_{0da}$ and $\overline{\mathbf{V}}_{0ca}$ being their left counterparts. In an FEM, the latter are the same as the former, whereas in an FDM, they are different. The analytical generation of $\overline{\mathbf{V}}_{0da}$, $\overline{\mathbf{V}}_{0ca}$ can be found in [10]. Write $\overline{\mathbf{V}}_0 = [\overline{\mathbf{V}}_{0d}, \overline{\mathbf{V}}_{0c}]$ and $\overline{\mathbf{V}}_{0a} = [\overline{\mathbf{V}}_{0da}, \overline{\mathbf{V}}_{0ca}]$, $\overline{\mathbf{A}}_{00}$ can be rewritten as

$$\begin{aligned}
&\overline{\mathbf{A}}_{00} \\
&= \begin{bmatrix} \overline{\mathbf{V}}_{0da}^T \\ \overline{\mathbf{V}}_{0ca}^T \end{bmatrix}\left(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma\right)\begin{bmatrix} \overline{\mathbf{V}}_{0d} & \overline{\mathbf{V}}_{0c} \end{bmatrix} \\
&= \begin{bmatrix} \overline{\mathbf{V}}_{0da}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon\right)\overline{\mathbf{V}}_{0d} & \overline{\mathbf{V}}_{0da}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon\right)\overline{\mathbf{V}}_{0c} \\ \overline{\mathbf{V}}_{0ca}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon\right)\overline{\mathbf{V}}_{0d} & \overline{\mathbf{V}}_{0ca}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma\right)\overline{\mathbf{V}}_{0c} \end{bmatrix}.
\end{aligned} \tag{37}$$

Therefore, (35) can be solved as

$$\begin{bmatrix} \overline{\mathbf{V}}_{0da}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon\right)\overline{\mathbf{V}}_{0d} & \overline{\mathbf{V}}_{0da}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon\right)\overline{\mathbf{V}}_{0c} \\ \overline{\mathbf{V}}_{0ca}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon\right)\overline{\mathbf{V}}_{0d} & \overline{\mathbf{V}}_{0ca}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma\right)\overline{\mathbf{V}}_{0c} \end{bmatrix}\begin{bmatrix} \mathbf{y}_{0d} \\ \mathbf{y}_{0c} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{0d} \\ \mathbf{b}_{0c} \end{bmatrix} \tag{38}$$

where $\overline{\mathbf{V}}_{0d}$, $\overline{\mathbf{V}}_{0da}$, $\overline{\mathbf{V}}_{0c}$, and $\overline{\mathbf{V}}_{0ca}$ are all normalized, where each column vector's L^2 norm is 1, and

$$\mathbf{b}_{0d} = \overline{\mathbf{V}}_{0da}^T(-j\omega\mathbf{J}) - \overline{\mathbf{V}}_{0da}^T\overline{\mathbf{D}}\mathbf{e}_h \tag{39}$$

$$\mathbf{b}_{0c} = \overline{\mathbf{V}}_{0ca}^T(-j\omega\mathbf{J}) - \overline{\mathbf{V}}_{0ca}^T\overline{\mathbf{D}}\mathbf{e}_h. \tag{40}$$

Equation (38) can be written in short as

$$\begin{bmatrix} \overline{\mathbf{M}}_{dd} & \overline{\mathbf{M}}_{dc} \\ \overline{\mathbf{M}}_{cd} & \overline{\mathbf{M}}_{cc} \end{bmatrix}\begin{bmatrix} \mathbf{y}_{0d} \\ \mathbf{y}_{0c} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{0d} \\ \mathbf{b}_{0c} \end{bmatrix} \tag{41}$$

in which

$$\begin{aligned}
\overline{\mathbf{M}}_{dd} &= \overline{\mathbf{V}}_{0da}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon\right)\overline{\mathbf{V}}_{0d} \\
\overline{\mathbf{M}}_{dc} &= \overline{\mathbf{V}}_{0da}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon\right)\overline{\mathbf{V}}_{0c} \\
\overline{\mathbf{M}}_{cd} &= \overline{\mathbf{V}}_{0ca}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon\right)\overline{\mathbf{V}}_{0d} \\
\overline{\mathbf{M}}_{cc} &= \overline{\mathbf{V}}_{0ca}^T\left(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma\right)\overline{\mathbf{V}}_{0c}.
\end{aligned} \tag{42}$$

Authorized licensed use limited to: Purdue University. Downloaded on May 12,2022 at 14:05:32 UTC from IEEE Xplore. Restrictions apply.

Notice that $\overline{\mathbf{M}}_{cc}$ is *inside* conductors, and hence the displacement current therein is much smaller than the conduction current, thus

$$\overline{\mathbf{M}}_{cc} \approx \overline{\mathbf{V}}_{0ca}^{T}\left(j\omega\overline{\mathbf{D}}_{\sigma}\right)\overline{\mathbf{V}}_{0c}. \tag{43}$$

To solve (41), first, we solve

$$\left(\overline{\mathbf{M}}_{cc} - \overline{\mathbf{M}}_{cd}\overline{\mathbf{M}}_{dd}^{-1}\overline{\mathbf{M}}_{dc}\right)\mathbf{y}_{0c} = \mathbf{b}_{0c} - \overline{\mathbf{M}}_{cd}\overline{\mathbf{M}}_{dd}^{-1}\mathbf{b}_{0d}. \tag{44}$$

Because $\|\overline{\mathbf{M}}_{cc}\| \gg \|\overline{\mathbf{M}}_{cd}\overline{\mathbf{M}}_{dd}^{-1}\overline{\mathbf{M}}_{dc}\|$ for good conductors, (44) can be accurately approximated as

$$\overline{\mathbf{M}}_{cc}\mathbf{y}_{0c} = \mathbf{b}_{0c} - \overline{\mathbf{M}}_{cd}\overline{\mathbf{M}}_{dd}^{-1}\mathbf{b}_{0d}. \tag{45}$$

After $\mathbf{y}_{0c}$ is obtained, $\mathbf{y}_{0d}$ can be computed from

$$\overline{\mathbf{M}}_{dd}\mathbf{y}_{0d} = \mathbf{b}_{0d} - \overline{\mathbf{M}}_{dc}\mathbf{y}_{0c}. \tag{46}$$

Notice that the two matrices to be solved, $\overline{\mathbf{M}}_{dd}$ and $\overline{\mathbf{M}}_{cc}$, are nothing but discretized $\nabla\cdot\epsilon\nabla$ and $\nabla\cdot\sigma\nabla$, and hence Laplacian matrices. Therefore, the above (45) and (46) can be rapidly solved by an iterative method such as a multigrid method. The aforementioned solution of $\overline{\mathbf{A}}_{00}$ is also used in the iterative solution of (20). After all of the aforementioned computation, the final solution is obtained by adding the curl-free component with the divergence-free one as follows:

$$\mathbf{e} = \overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0c}\mathbf{y}_{0c} + \mathbf{e}_h. \tag{47}$$

D. Complexity Analysis

The computational complexity of the proposed algorithm is $O(N)$ in both time and memory, i.e., linearly scaling with the number of unknowns N. This can be analyzed as follows.

First, we solve (20) to find the divergence-free component. Because the system has been transformed to a Laplacian counterpart, iterative solvers such as GMRES and multigrid can converge the solution in a very small number of steps. At each step of such an iterative solver, the computational cost is to multiply $\widetilde{\mathbf{A}}_{hh}$ by a vector (say ν). Based on the expression of $\widetilde{\mathbf{A}}_{hh}$ shown in (21), this can be carried out by first multiplying $\overline{\mathbf{A}}_{0h}$ by ν, which has a linear complexity since $\overline{\mathbf{A}}_{0h}$ is sparse. Let the resultant vector be x_1. Then we compute $(\overline{\mathbf{A}}_{00})^{-1}x_1$. This is equivalent to solve $(\overline{\mathbf{A}}_{00})x_2 = x_1$. The x_2 can be found by the fast solution of $\overline{\mathbf{A}}_{00}$ described in Section IV-C, which has a linear complexity since it is the solution of two Laplace matrices. After x_2 is computed, we multiply $\overline{\mathbf{A}}_{h0}$ with it, which again has a linear cost since $\overline{\mathbf{A}}_{h0}$ is sparse. Let the resultant vector be x_3. We then do a sparse matrix vector multiplication of $\overline{\mathbf{A}}_{hh}$ with ν, and subtract x_3 from the resultant vector. The end result is $\widetilde{\mathbf{A}}_{hh}\nu$. Such a matrix-vector multiplication needs to be performed for k steps, where k is the number of iterations. Since k is a small constant here due to the proposed Laplacian treatment, the total cost of finding $\mathbf{e}_h$ is linear.

As for the computational cost of finding the curl-free component, as shown in Section IV-C, we only need to solve (45) and (46). Both matrices are Laplacian, and hence they can be converged in a small and constant number of iterations. As a result, the time cost of finding the curl-free component is also linear.

Regarding the memory, the main memory cost is incurred during the iterative solution, which is of $kO(N)$, where k is the number of iterations. Since the iteration number k is made small, the memory usage of the proposed method is modest, and also scales linearly with N. The aforementioned complexity analysis will become more clear in the section of Numerical Results, where we compare the time and memory consumption of the proposed method with conventional cost.

E. Remarks

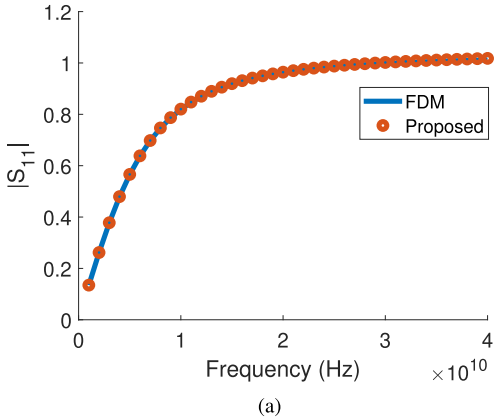
Before finishing this section, we provide another understanding of the proposed work in the context of Helmholtz Decomposition. For example, in [11]–[13], loop-tree bases are introduced in integral equation solvers to decompose an unknown current into a divergence-free component and a nondivergence-free one. In [14]–[16], tree–cotree bases are used in FEM solvers to decompose an unknown field into a curl-free component and a noncurl-free one. In both methods, an inexact Helmholtz decomposition is performed. In the former, the loop basis is divergence-free, but the tree basis is not curl-free; in the latter, the tree basis is curl-free, but the cotree basis is not divergence-free. In comparison, in the proposed method, we have an exact Helmholtz decomposition, where the unknown field is split into a curl-free and a divergence-free component. Moreover, we change the system matrix operating on the divergence-free component to its Laplacian, which is exact since the curl–curl operator is naturally reduced to a Laplacian when operating on a divergence-free field. Such a change of the original system matrix is further performed without any computational cost in the proposed method. Meanwhile, the resultant Laplacian-based system has a guaranteed fast convergence.

In addition, existing loop-tree and tree-cotree-based methods are accurate for use at low frequencies, but do not perform equally well at high frequencies. The reason can be understood as follows. In a tree-cotree-based method, after some edges are identified as tree edges, their original vector edge bases are replaced by gradient field bases, while the rest of the edges keep their original vector bases. This would yield inaccuracy at high frequencies, since the field solution at tree edges should not be just a gradient field. Different from tree-cotree splitting, in the proposed method, we keep the original vector edge bases as they are. Instead of decomposing discretized edges or edge bases into two sets, we decompose the solution space governing the field solution into two subspaces. One is the nullspace of the curl–curl operator, and the other is orthogonal to it. Hence, each edge in the computational domain, i.e., each unknown field has both a gradient field component, and a divergence-free one. As a result, the proposed scheme is equally effective at low and high frequencies. In addition, it is free of low-frequency breakdown.

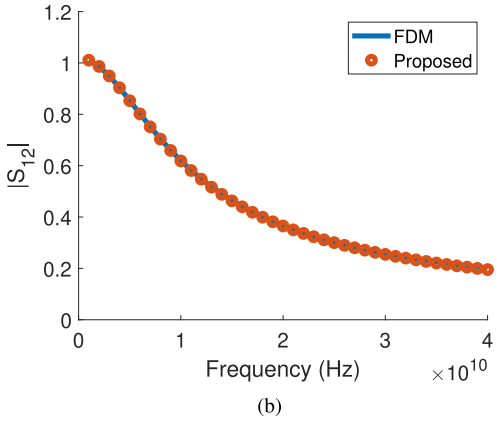
V. Numerical Results

In this section, we simulate a variety of problems from on-chip circuits to antenna radiation problems to validate the proposed method, and examine its performance. The FDM is used to perform these simulations, but the formulations

Authorized licensed use limited to: Purdue University. Downloaded on May 12,2022 at 14:05:32 UTC from IEEE Xplore. Restrictions apply.



(a)



(b)

Fig. 2. Simulation of a test-chip interconnect. (a) $|S_{11}|$. (b) $|S_{12}|$.

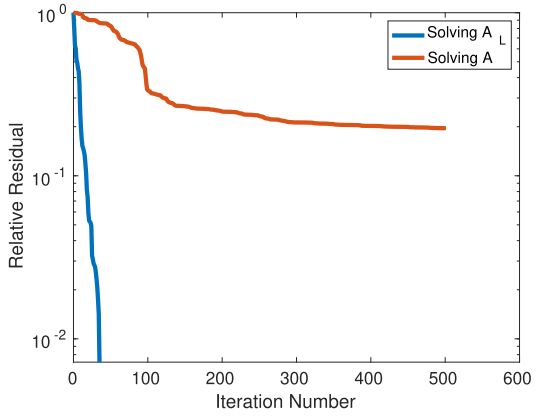
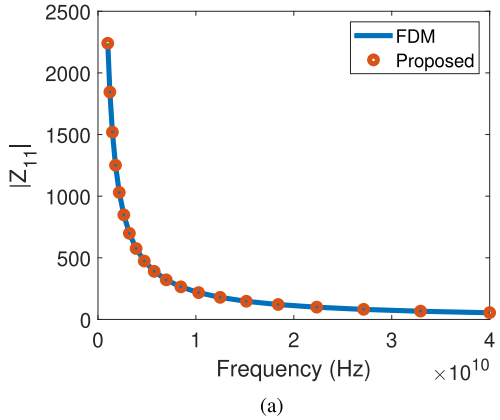
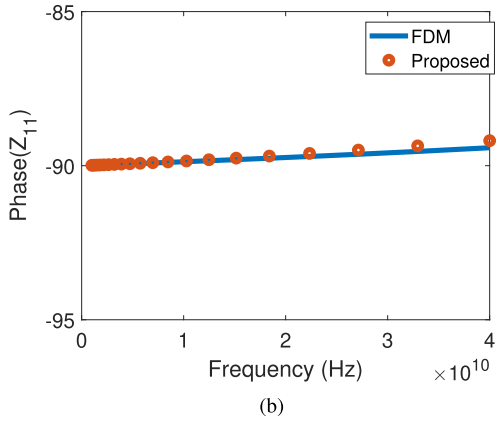


Fig. 3. Convergence comparison between solving $\overline{\mathbf{A}}$ and solving $\overline{\mathbf{A}}_L$ in simulating the test-chip interconnect.



(a)



(b)

Fig. 4. Simulated **Z**-parameters of a NAND gate. (a) $\mathbf{Z}_{11}$ magnitude. (b) $\mathbf{Z}_{11}$ phase.

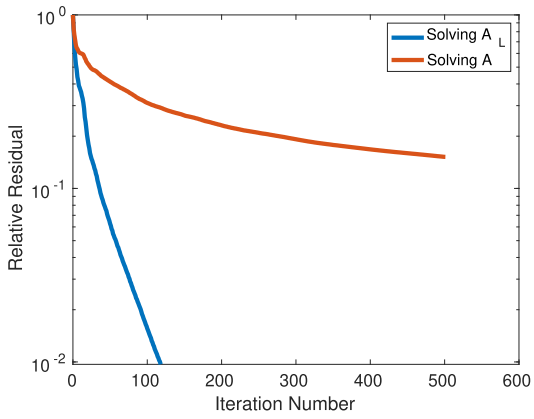


Fig. 5. Convergence comparison between solving $\overline{\mathbf{A}}$ and solving $\overline{\mathbf{A}}_L$ for a NAND structure.

described in previous section are equally applicable to accelerating the convergence of the FEM and other PDE solvers. The computer used has 256 GB memory, and an Intel(R) Xeon(R) CPU E5-2660 v3 running at 2.60 GHz.

A. Test-Chip Interconnect

The first example is a test-chip interconnect, which is the same as that shown in [10, Fig. 4]. The conductors have a conductivity of 5.8×10^7 S/m. The dimensions along the x, y and z-direction are 300, 100, and 3.19 μm respectively. We evaluate $((\|\overline{\mathbf{LV}}_h - \overline{\mathbf{SV}}_h\|)/(\|\overline{\mathbf{SV}}_h\|))$, and find it to be 1.3373×10^{-11}, thus verifying our finding. The S-parameters extracted from the proposed method are compared with those from a brute-force solution of (1) in Fig. 2. They are shown to agree well with each other. In this example, we employ GMRES to solve (32), which takes only 34 steps to reach a relative residual of 0.0087 at frequency 10 GHz. In contrast, if we solve (1) by GMRES directly, after 4943 steps, the convergence has not been achieved. Fig. 3 shows the relative residual vs. the iteration number of the proposed method that

Authorized licensed use limited to: Purdue University. Downloaded on May 12,2022 at 14:05:32 UTC from IEEE Xplore. Restrictions apply.

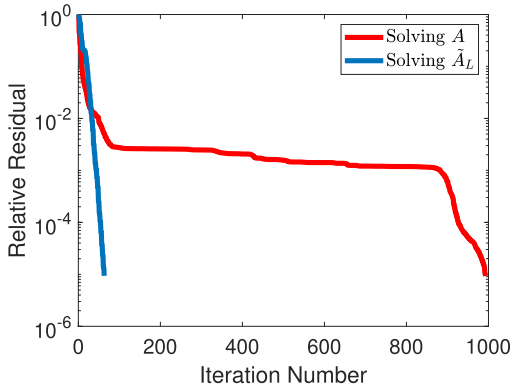


Fig. 6. Convergence comparison between solving $\overline{\mathbf{A}}$ and solving $\overline{\mathbf{A}}_L$ for a package structure.

solves $\overline{\mathbf{A}}_L$ shown in (32), as compared to conventional one that solves $\overline{\mathbf{A}}$ shown in (1). As we can see, solving $\overline{\mathbf{A}}_L$ has a much more rapid convergence.

After the divergence-free component $\mathbf{e}_h$ is rapidly solved from (20), we solve (45) and (46) to find the curl-free component. In this step, it only takes a multigrid solver about 4 to 8 steps to reach a relative residual of 10^{-5} when solving $\overline{\mathbf{M}}_{dd}$ and $\overline{\mathbf{M}}_{cc}$, since both are Laplacian. The efficiency of the proposed method can also be seen from the condition number. For example, at 1 GHz, the condition number of the original matrix from (1) is 2.15×10^{11}. In contrast, the condition numbers of $\widetilde{\mathbf{A}}_{hh}$, $\overline{\mathbf{M}}_{dd}$ and $\overline{\mathbf{M}}_{cc}$ are 6×10^6, 1.5×10^3 and 5.3×10^6, respectively, which are much smaller. A well-conditioned matrix may not converge in a small number of iterations. However, the three matrices to be solved in the proposed method are all made Laplacian, and hence they can be converged fast in a small number of iterations.

The elapsed time for the conventional solver to solve this example is 173.49 s and the peak memory usage is 355.796 MB. In contrast, the entire CPU time of the proposed method is only 0.22 s including both divergence- and curl-free solutions, and the peak memory usage is 20.47 MB. This verifies the efficiency of the proposed method.

B. NAND *Gate*

The second example is a NAND gate, the structure of which is shown in [10]. It has 17 layers. We discretize the structure with a nonuniform grid yielding 690 142 unknowns. The extracted **Z**-parameters are shown to agree very well with those from the brute-force solution of (1) in Fig. 4. The $\overline{\mathbf{A}}_L$ shown in (32) is solved in this example, for which GMRES takes 94 steps to reach a relative residual of 0.0098 at 10 GHz. The relationship between the relative residual and the iteration number for solving $\overline{\mathbf{A}}$ and that for solving $\overline{\mathbf{A}}_L$ is shown in Fig. 5, from which it can be seen that solving $\overline{\mathbf{A}}_L$ is much more efficient. The solution of $\overline{\mathbf{M}}_{dd}$ and $\overline{\mathbf{M}}_{cc}$ takes at most 43 steps to achieve a relative residual of 10^{-5}.

The total CPU time of the proposed method is 35.89 s and the peak memory usage is 1.88 GB. If we solve the original matrix equation using a conventional method, it takes 2000 steps to reach a relative residual of 0.09499 without converging, costing 8243.25 s in CPU time and 22.62 GB peak memory.

C. A Package Interconnect

The third example is a package interconnect, the structure of which is shown in [9]. It contains three layers. The bottom conductor is a ground plane of copper conductivity. There are two ground-signal-ground launchers at the two ends of the interconnect line. The length of the line is 14.854 mm. The **S**-parameters are extracted at the far and near ends of the interconnect. Due to a larger electrical size of this problem, we apply GMRES to solve (31) directly instead of the approximated (32) used in previous two on-chip examples. At 10 GHz, it only takes the proposed method 43 steps to obtain a relative residual of 0.000999, while a conventional solution costs many more steps, as can be seen from the convergence comparison shown in Fig. 6. We compare the results from the proposed method and the FDM in Fig. 7. Excellent agreement is observed. The solution of $\overline{\mathbf{M}}_{dd}$ and $\overline{\mathbf{M}}_{cc}$ takes at most nine steps to achieve a relative residual of 10^{-5} in the proposed method. The total CPU time and peak memory usage of the proposed method are 4.24 s, and 27.98 MB, respectively, whereas the CPU time of the conventional method is 46.41 s and peak memory usage is 219.6 MB.

D. Cavity-Backed Microstrip Patch Antenna

The fourth example is a cavity-backed path antenna example shown in [17]. The patch size is $W = 3.4$ cm, $L = 5$ cm. There is a 50-Ω load at the point $x_L = -2.2$ cm and $y_L = -1.5$ cm. The current is injected at the $x_f = 1.22$ cm and $y_f = 0.85$ cm point. The antenna patch is modeled with a finite conductivity of 5.8×10^7 S/m. We extract the **Z**-parameters and compare them with those obtained from the FDM. They show good agreement in Fig. 8. We also record the iteration number the proposed method takes to solve (31). At 1 GHz, only 38 steps are taken to reach a relative residual of 0.0005434, whereas the original system takes many more steps to converge as can be seen from Fig. 9. The conventional method performs better in this example than previous examples because this example is less ill-conditioned. Solving $\overline{\mathbf{M}}_{dd}$ and $\overline{\mathbf{M}}_{cc}$ takes at most seven steps to reach a relative residual of 10^{-5}. The total CPU time taken by the proposed method is 2.04 s and the memory usage is 23.56 MB, whereas the conventional method costs a CPU time of 12.72 s and 125.23 MB memory.

E. Large-Scale IBM Plasma Package Structure

In this subsection, we simulate a large-scale IBM plasma package structure shown in [18]. We use a nonuniform grid to discretize the structure, resulting in 15 407 710 unknowns. The **S**-parameters are extracted at two selected ports and compared with the reference results generated from [18]. As can be seen in Fig. 10, good agreement with reference solution is observed. The performance of the proposed iterative solver

Authorized licensed use limited to: Purdue University. Downloaded on May 12,2022 at 14:05:32 UTC from IEEE Xplore. Restrictions apply.

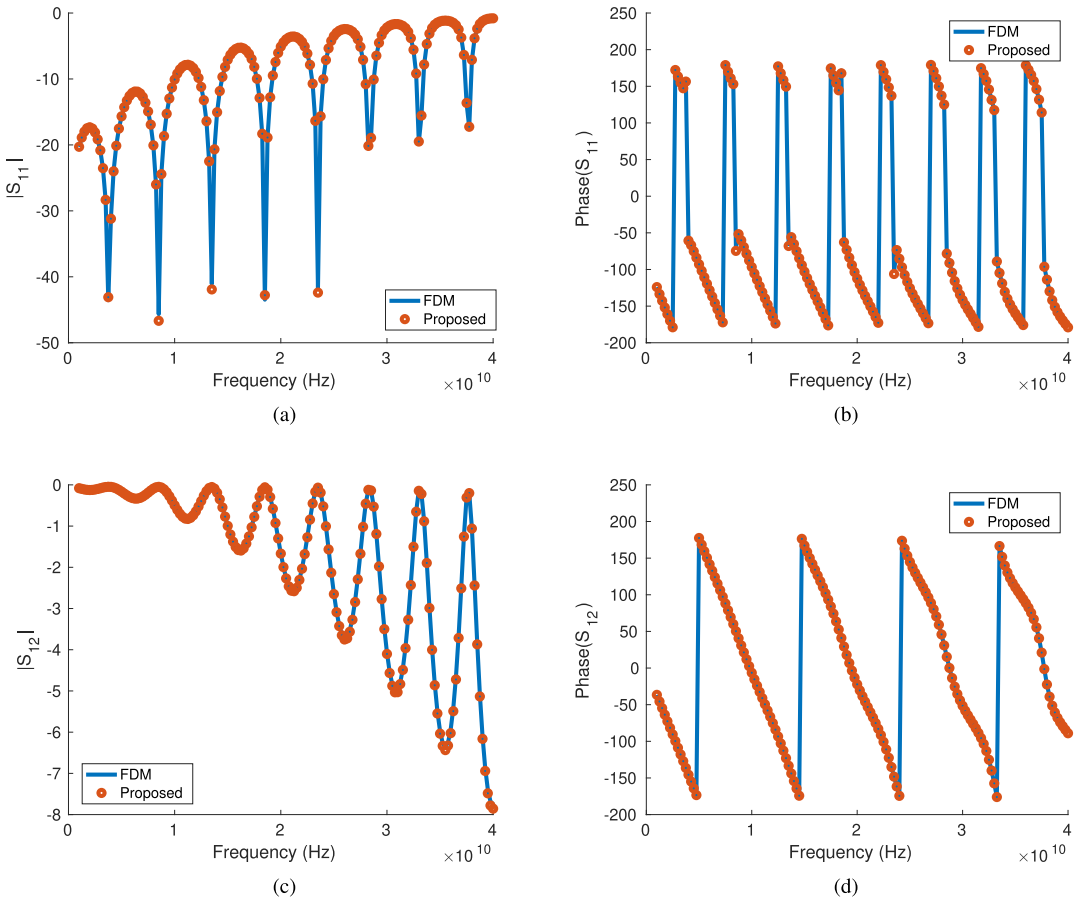


Fig. 7. Simulation of a package interconnect structure. (a) $|S_{11}|$. (b) S_{11}'s phase. (c) $|S_{12}|$. (d) S_{12}'s phase.

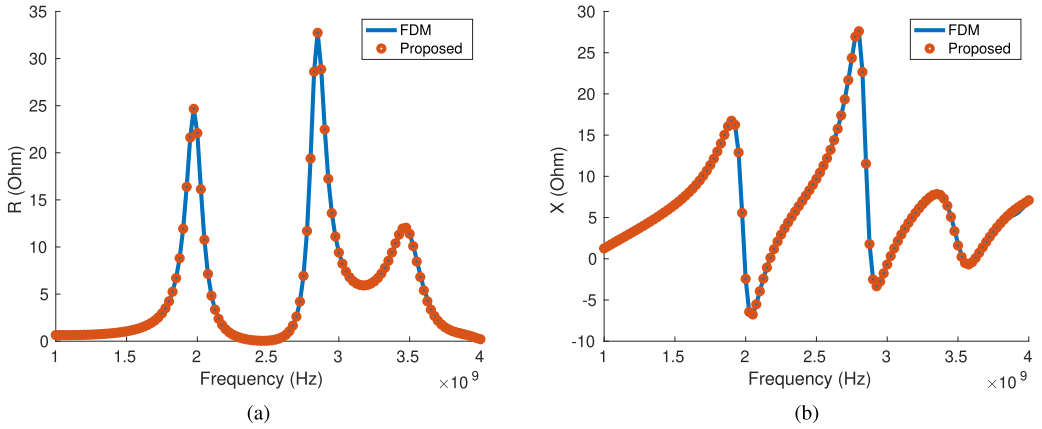


Fig. 8. Simulation of a cavity-backed patch antenna. (a) Input resistance (Ohms). (b) Input reactance (Ohms).

is also checked. At 10 GHz, solving $\overline{\mathbf{A}}_L$ takes 274 steps to achieve a relative residual of 0.01. In contrast, solving $\overline{\mathbf{A}}$ fails to converge after a few thousand of steps. Solving $\overline{\mathbf{M}}_{dd}$ takes about 1 or 2 steps to reach a relative residual of 10^{-5}, and solving $\overline{\mathbf{M}}_{cc}$ takes at most 43 steps to achieve the same accuracy. The conventional method fails to solve this example. To make a comparison with the conventional method, we simulate a smaller plasma structure with 1 796 632 unknowns. The proposed method costs 834.07 s only, whereas the conventional method takes 2919.77 s.

F. Dielectric Cavity With Divergence-Free Solution

All examples given in the above have both curl-free and divergence-free components in the field solution. In this subsection, we simulate an example whose solution is dominated by the divergence-free component. It is a dielectric cube of side length 1 m, filled with a material of dielectric constant 4. The frequency simulated is 300 MHz, and hence the structure is of two wavelengths considering the dielectric fill-in. The structure is discretized with a mesh size of wavelength over 10, and thereby 0.05 m. A current source is launched from the

Authorized licensed use limited to: Purdue University. Downloaded on May 12,2022 at 14:05:32 UTC from IEEE Xplore. Restrictions apply.

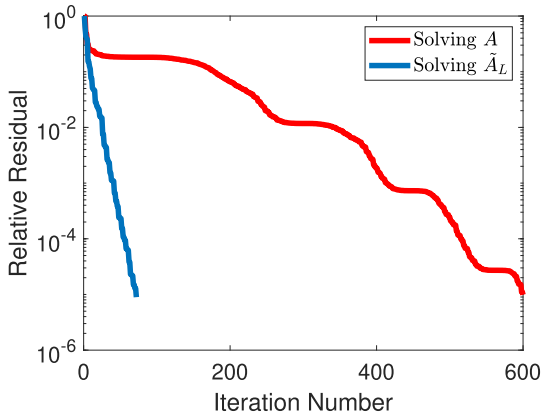


Fig. 9. Convergence comparison between solving $\overline{\mathbf{A}}$ and solving $\overline{\mathbf{A}}_L$ for an antenna example.

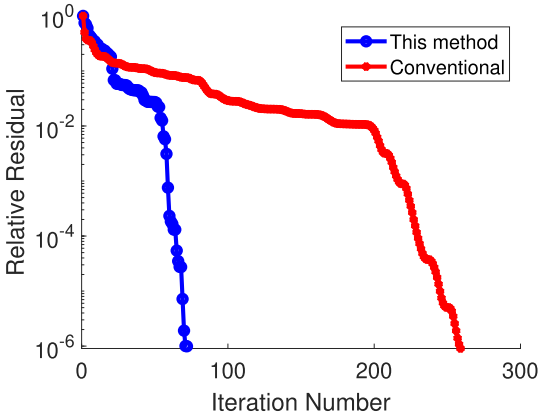


Fig. 11. Convergence comparison for solving a dielectric cavity dominated by a divergence-free solution.

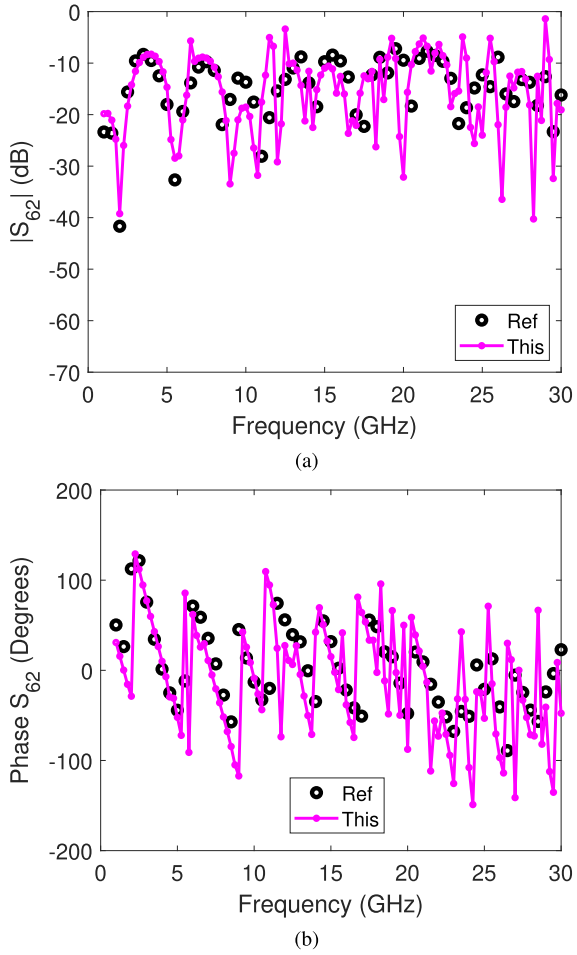


Fig. 10. **S**-parameters simulated from a large-scale package structure having over 15 million unknowns. (a) $\mathbf{S}_{6,2}$ magnitude. (b) $\mathbf{S}_{6,2}$ phase.

bottom to the top of the cavity along z-direction and located at the center of the cavity. The cavity is truncated by a Neumann boundary condition. The $\overline{\mathbf{V}}_h$ component accounts for 99.97% of the field solution in this example. This is found by solving (1) as it is to obtain $\mathbf{e}$, and then deducting the $\overline{\mathbf{V}}_0$-component from $\mathbf{e}$ to get its $\mathbf{e}_h$ component.

We then solve (34) only, and find the resultant solution agrees very well with that solved from (1), without the need for adding the curl-free component. The resultant convergence curve is shown in Fig. 11 in comparison with the conventional iterative solution of (1). As can be seen, even for this example in which the conventional solution is not difficult to converge, the proposed method converges much faster. The time cost of the proposed method is 3.02 s, while that of the conventional one is 13.73 s.

VI. Conclusion

A new method is developed to accelerate the convergence of an iterative PDE-based solution of Maxwell's equations. Different from prevailing methods that rely on the development of a good preconditioner, the proposed method directly replaces the original system matrix by its Laplacian counterpart when solving the divergence-free component of the field solution. Such a change of the original system matrix is exact without making approximations. Furthermore, the Laplacian counterpart is built analytically from the original system matrix and the mesh information without any computational cost. And the iterative solution of the resultant Laplacian-based system matrix converges fast irrespective of the matrix size. After the divergence-free component is found, the curl-free component is also solved from Laplacian matrices, and hence having a fast convergence. In addition, the proposed method is rigorous and effective at both high and low frequencies. Applications to a variety of problems including on-chip circuits, antenna problems, and large-scale packaging structures have demonstrated the superior performance of the proposed new method. These problems include those dominated by curl-free solutions, divergence-free solutions, and the combination of both, for which the proposed method is shown to have advantages.

References

[1] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Philadelphia, PA, USA: SIAM, 2003.

[2] J. Liu and J.-M. Jin, "A highly effective preconditioner for solving the finite element-boundary integral matrix equation of 3-D scattering," *IEEE Trans. Antennas Propag.*, vol. 50, no. 9, pp. 1212–1221, Sep. 2002.

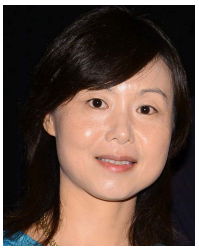
Authorized licensed use limited to: Purdue University. Downloaded on May 12,2022 at 14:05:32 UTC from IEEE Xplore. Restrictions apply.

[3] Y. Yang, Z. H. Fan, D. Z. Ding, and S. B. Liu, "Application of the preconditioned GMRES to the crank-nicolson finite-difference time-domain algorithm for 3D full-wave analysis of planar circuits," *Microw. Opt. Technol. Lett.*, vol. 50, no. 6, pp. 1458–1463, 2008.
[4] R.-S. Chen, E. K.-N. Yung, C. H. Chan, D. X. Wang, and D. G. Fang, "Application of the SSOR preconditioned CG algorithm to the vector FEM for 3D full-wave analysis of electromagnetic-field boundary-value problems," *IEEE Trans. Microw. Theory Techn.*, vol. 50, no. 4, pp. 1165–1172, Apr. 2002.
[5] F. Sheng, H. Gan, and D. Jiao, "Fast iterative solution algorithms in the frequency-domain layered finite element method for analyzing integrated circuits," *IEEE Trans. Adv. Packag.*, vol. 33, no. 2, pp. 524–533, May 2010.
[6] A. Brandt, S. McCormick, and J. Ruge, *Algebraic Multigrid (AMG) for Sparse Matrix Equations, in Sparsity and its Applications*. Cambridge, U.K.: Cambridge Univ. Press, 1985.
[7] L. Xue and D. Jiao, "Fast method for accelerating convergence in iterative solution of frequency-domain partial differential equation methods," in *Proc. IEEE Int. Symp. Antennas Propag. North Amer. Radio Sci. Meeting*, Jul. 2020, pp. 1003–1004.
[8] J. Zhu, S. Omar, and D. Jiao, "Solution of the electric field integral equation when it breaks down," *IEEE Trans. Antennas Propag.*, vol. 62, no. 8, pp. 4122–4134, Aug. 2014.
[9] L. Xue and D. Jiao, "Method for analytically finding the nullspace of stiffness matrix for both zeroth-order and higher order curl-conforming vector bases in unstructured meshes," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 2, pp. 456–468, Feb. 2020.
[10] L. Xue and D. Jiao, "Rapid modeling and simulation of integrated circuit layout in both frequency and time domains from the perspective of inverse," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 4, pp. 1270–1283, Apr. 2020.
[11] D. R. Wilton and A. W. Glisson, "On improving the electric field integral equation at low frequencies," in *URSI Radio Sci. Meet. Dig.*, Jun. 1981, p. 24.
[12] G. Vecchi, "Loop-star decomposition of basis functions in the discretization of the EFIE," *IEEE Trans. Antennas Propag.*, vol. 47, no. 2, pp. 339–346, Feb. 1999.
[13] J.-S. Zhao and W. C. Chew, "Integral equation solution of Maxwell's equations from zero frequency to microwave frequencies," *IEEE Trans. Antennas Propag.*, vol. 48, no. 10, pp. 1635–1645, Oct. 2000.
[14] R. Albanese and G. Rubinacci, "Solution of three dimensional eddy current problems by integral and differential methods," *IEEE Trans. Magn.*, vol. 24, no. 1, pp. 98–101, Jan. 1988.
[15] S.-C. Lee, J.-F. Lee, and R. Lee, "Hierarchical vector finite elements for analyzing waveguiding structures," *IEEE Trans. Microw. Theory Techn.*, vol. 51, no. 8, pp. 1897–1905, Aug. 2003.
[16] S. Lee and J. Jin, "Application of the tree-cotree splitting for improving matrix conditioning in the fullwave finite element analysis of highspeed circuits," *Microw. Opt. Technol. Lett.*, vol. 50, no. 6, pp. 1476–1481, 2008.
[17] J. Jin, *The Finite Element Method in Electromagnetics*. Hoboken, NJ, USA: Wiley, 2014.
[18] B. Zhou and D. Jiao, "Direct finite element solver of linear complexity for large-scale 3-D electromagnetic analysis and circuit extraction," *IEEE Trans. MTT*, vol. 63, no. 10, pp. 3066–3080, Oct. 2015.

Li Xue (Student Member, IEEE) received the B.S. degree in information engineering from Zhejiang University, Hangzhou, China, in 2015, and the Ph.D. degree in electrical and computer engineering with the On-Chip Electromagnetics Group, Purdue University, West Lafayette, IN, USA, in December 2020.

She has been a Research and Development engineer with Prime Shield Team in Synopsys, Sunnyvale, CA, USA, since 2020. Her research interests include computational electromagnetics, fast and high-performance algorithm, and high-capacity numerical methods.

Dr. Xue was a recipient of the Best Student Finalist Award from the IEEE International Microwave Symposium in 2019, the Best Student Finalist Award from the Antennas and Propagation (APSURSI) IEEE International Symposium, and the Bilsland Dissertation Fellowship in 2020.



Dan Jiao (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, in 2001.

She worked with the Technology Computer-Aided Design (CAD) Division, Intel Corporation, until September 2005, as a Senior CAD Engineer, Staff Engineer, and Senior Staff Engineer. In September 2005, she joined Purdue University, West Lafayette, IN, as an Assistant Professor with the School of Electrical and Computer Engineering, where she is currently a Professor. She has authored three book chapters and over 320 articles in refereed journals and international conferences. Her current research interests include computational electromagnetics, high-frequency digital, analog, mixed-signal, and RF-integrated circuit (IC) design and analysis, high-performance VLSI CAD, modeling of microscale and nanoscale circuits, applied electromagnetics, fast and high-capacity numerical methods, fast time-domain analysis, scattering and antenna analysis, RF, microwave, and millimeter-wave circuits, wireless communication, and bio-electromagnetics.

Dr. Jiao has served as the reviewer for many IEEE journals and conferences. She is an Associate Editor of the IEEE TRANSACTIONS ON COMPONENTS, PACKAGING, AND MANUFACTURING TECHNOLOGY and an Associate Editor of the IEEE JOURNAL ON MULTISCALE AND MULTIPHYSICS COMPUTATIONAL TECHNIQUES. She served as the General Chair for the 2019 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO), Boston, USA. She was selected as an IEEE MTT-Society Distinguished Microwave Lecturer in 2020. She was a recipient of Intel's 2019 Outstanding Researcher Award. She received the 2013 S. A. Schelkunoff Prize Paper Award of the IEEE Antennas and Propagation Society, which recognizes the Best Paper published in the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION during the previous year. She was among the 21 women faculty selected across the country as the 2014–2015 Fellow of Executive Leadership in Academic Technology and Engineering (ELATE) at Drexel, a national leadership program for women in the academic STEM fields. She has been named a University Faculty Scholar by Purdue University since 2013. She was among the 85 engineers selected throughout the nation for the National Academy of Engineering's 2011 US Frontiers of Engineering Symposium. She was the recipient of the 2010 Ruth and Joel Spira Outstanding Teaching Award, the 2008 National Science Foundation (NSF) CAREER Award, the 2006 Jack and Cathie Kozik Faculty Startup Award (which recognizes an outstanding new faculty member of the School of Electrical and Computer Engineering, Purdue University), a 2006 Office of Naval Research (ONR) Award under the Young Investigator Program, the 2004 Best Paper Award presented at the Intel Corporation's annual corporate-wide technology conference (Design and Test Technology Conference) for her work on generic broadband model of high-speed circuits, the 2003 Intel Corporation's Logic Technology Development (LTD) Divisional Achievement Award, the Intel Corporation's Technology CAD Divisional Achievement Award, the 2002 Intel Corporation's Components Research the Intel Hero Award (Intel-wide she was the tenth recipient), the Intel Corporation's LTD Team Quality Award, and the 2000 Raj Mittra Outstanding Research Award presented by the University of Illinois at Urbana-Champaign.

Authorized licensed use limited to: Purdue University. Downloaded on May 12,2022 at 14:05:32 UTC from IEEE Xplore. Restrictions apply.