RAPID MODELING AND SIMULATION METHODS FOR LARGE-SCALE AND CIRCUIT-INTUITIVE ELECTROMAGNETIC ANALYSIS OF INTEGRATED CIRCUITS AND SYSTEMS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Li Xue

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

Dec 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Dan Jiao, Chair

School of Electrical and Computer Engineering

Dr. Weng C. Chew

School of Electrical and Computer Engineering

Dr. Kevin Webb

School of Electrical and Computer Engineering

Dr. Alexander V. Kildishev

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

Head of Electrical and Computer Engineering

To my family.

ACKNOWLEDGMENTS

I would like to express my sincere thanks to my advisor Professor Dan Jiao to have a chance to do this work with her in Purdue University. She introduced me into the area of Computational Electromagnetics and imparts her knowledge to me step by step. Doing research sometimes feels like searching in the ocean and it is easy to get lost. Professor Jiao is like a lighthouse and a dose of encouragement, giving me the clear direction of my research and helping me get out of the mist. She is always there whenever I meet any difficulty or get stuck. Without her help, this work can never be complete. I feel lucky that Professor Jiao is my PhD advisor.

Also, I would like to thank to my other committee members: Professor Weng Cho Chew, Professor Kevin Webb and Professor Alexander V. Kildishev for their precious time and insight help to my research work.

Thanks to my labmates who worked with me in On-Chip Electromagnetics group: Dr. Jin Yan, Dr. Kaiyuan Zeng, Dr. Miaomiao Ma, Dr. Yuhang Dou, Michael R. Hayashi, Shuzhan Sun, Yifan Wang, Chang Yang, Daniel Wei for their professional discussion and friendly support.

Last but least, I would like to thank my family for their love and support during all these years.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

ABSTRACT

Xue, Li Ph.D., Purdue University, Dec 2020. Rapid Modeling and Simulation Methods for Large-Scale and Circuit-Intuitive Electromagnetic Analysis of Integrated Circuits and Systems. Major Professor: Professor Dan Jiao.

Accurate, fast, large-scale, and circuit-intuitive electromagnetic analysis is of critical importance to the design of integrated circuits (IC) and systems. Existing methods for the analysis of integrated circuits and systems have not satisfactorily achieved these performance goals. In this work, rapid modeling and simulation methods are developed for large-scale and circuit-intuitive electromagnetic analysis of integrated circuits and systems. The derived model is correct from zero to high frequencies where Maxwell's equations are valid. In addition, in the proposed model, we are able to analytically decompose the layout response into static and full-wave components with neither numerical computation nor approximation. This decomposed yet rigorous model greatly helps circuit diagnoses since now designers are able to analyze each component one by one, and identify which component is the root cause for the design failure. Such a decomposition also facilitates efficient layout modeling and simulation, since if an IC is dominated by RC effects, then we do not have to compute the full-wave component; and vice versa. Meanwhile, it makes parallelization straightforward. In addition, we develop fast algorithms to obtain each component of the inverse rapidly. These algorithms are also applicable for solving general partial differential equations for fast electromagnetic analysis.

The fast algorithms developed in this work are as follows. First, an analytical method is developed for finding the nullspace of the curl-curl operator in an arbitrary mesh for an arbitrary order of curl-conforming vector basis function. This method has been applied successfully to both a finite-difference and a finite-element based analysis of general 3-D structures. It can be used to obtain the static component of the inverse efficiently. An analytical method for finding the complementary space of the nullspace is also developed. Second, using the analytically found nullspace and its complementary space, a rigorous method is developed to overcome the low-frequency breakdown problem in the full-wave analysis of general lossy problems, where both dielectrics and conductors can be lossy and arbitrarily inhomogeneous. The method is equally valid at high frequencies without any need for changing the formulation. Third, with the static component part solved, the full-wave component is also ready to obtain. There are two ways. In the first way, the full-wave component is efficiently represented by a small number of high-frequency modes, and a fast method is created to find these modes. These modes constitute a significantly reduced order model of the complementary space of the nullspace. The second way is to utilize the relationship between the curl-curl matrix and the Laplacian matrix. An analytical method to decompose the curl-curl operator to a gradient-divergence operator and a Laplacian operator is developed. The derived Laplacian matrix is nothing but the curl-curl matrix's Laplacian counterpart. They share the same set of non-zero eigenvalues and eigenvectors. Therefore, this Laplacian matrix can be used to replace the original curl-curl matrix when operating on the full-wave component without any computational cost, and an iterative solution can converge this modified problem much faster irrespective of the matrix size. The proposed work has been applied to large-scale layout extraction and analysis. Its performance in accuracy, efficiency, and capacity has been demonstrated.

1. INTRODUCTION

1.1 Background and motivation

Accurate and large-scale layout models are of critical importance to the design of integrated circuits (IC), packages, and boards. A physical design tool built upon inaccurate or erroneous layout models, no matter how superior it is in machine learning and optimization algorithms, will fail to generate a working layout within feasible run time. In prevailing layout modeling, static or quasi-static field solvers are employed to extract C-, R-, and L-based circuit models of the physical layout of an IC, package, or board [1–10]. These models are then stitched together to perform a circuit simulation. There are many ways to stitch the C models with the RL-models, which are extracted independent of each other. Which one correctly captures the physics such as distributed effects and 3-D effects in the physical layout is unknown even at relatively low working frequencies, where static physics is dominant. The global effect of the substrate is not well captured in existing layout models. As far as full-wave layout modeling is concerned, such as the Finite Difference Method [11,12], the Finite Element Method [13–18], and the Integral Equation-based Method [19–25], although it is accurate at high frequencies, they are not capable of performing extraction and simulation at the full-chip scale in fast CPU run time because full-wave solvers need to capture more physics than their static and quasi-static-based counterparts. In addition, the direct field-based representation of the layout and the resulting field solution remain too abstract to be put into practical use. Moreover, a full-wave model is expected to reduce to the static- and quasi-static-based RLC models at low frequencies; however, this relationship is not established by existing full-wave solvers. In fact, full-wave solvers break down at low frequencies because of finite machine precision and the loss of frequency-dependent terms [26–29]. The problem is especially severe when dealing with VLSI circuit problems. The characteristic breakdown frequency is about tens of megahertz, which falls right into the range of circuit operating frequencies. Therefore, the low frequency breakdown problem is also critical and demands a good solution.

One common feature in existing layout modeling and simulations tools is that a forward model of the layout is being pursued. In the static- or quasi-static-based approaches, a forward model of the layout in a SPICE-compatible format is extracted. After extraction, circuit simulation is performed based on the circuit model to analyze the layout response. In a full-wave approach, no extraction is performed. The forward model is nothing but the discretized Maxwell's equations in the layout. Then it is directly solved in time or frequency domain and the solution is the field distribution in the whole structure. Both the static or quasi-static-based and the full-wave approaches focus on the forward model of the layout. Even though such a model is built efficiently in high accuracy, the actual layout response to circuit stimuli is unknown until a circuit simulation is performed on the extracted layout model. Such a modeling approach is not amenable for circuit design, because changing the C, R, L or other circuit parameters in the original layout does not directly reveal how the circuit performance will be changed. Therefore, it is essential to have an inverse model to represent the solution clearly and physically. Circuit designers can thus easily analyze circuit responses from the inverse model.

1.2 Contributions of This Work

In this work, different from prevailing approaches where the forward model of the layout is constructed, we analytically derive a closed-form model of the inverse of Maxwell's system of equations in the physical layout of an integrated circuit, package, and board, starting from full-wave Maxwell's equations where **E** and **H** are coupled. By simulating the full-wave Maxwell's equations, all the physical phenomena are accurately captured and we are able to bypass the inaccuracy issue arising from stitching circuit models independently extracted from decoupled **E** and **H** equations. The resultant model is correct from zero to high frequencies where Maxwell's equations are valid. More importantly, in the proposed inverse model, the layout response is analytically decomposed into R-, C-, L-, and full-wave components with neither numerical computation nor approximation and for an arbitrary physical layout. As a result, each component can be found in parallel and then summed up to obtain a total layout response of a layout to any circuit stimuli. This decomposition can also be viewed as an exact Helmholtz decomposition, where the electric field is decomposed into a solenoidal field and an irrotational field. In existing approaches of Helmholtz decomposition such as Loop-tree and Loop-star based methods [26, 30], the decomposition is not exact. In the proposed method, the RC- component is an irrotational field; while the L- and full-wave component is a solenoidal field, and hence constituting an exact Helmholtz decomposition. This decomposed yet rigorous model greatly helps circuit diagnoses since now designers are able to analyze each component one by one and identify which component is the root cause for the design failure. Such a decomposition also facilitates efficient layout modeling and simulation since if an IC is dominated by RC effects, then we do not have to compute the full-wave component, and vice versa. Meanwhile, it makes parallelization straightforward. Unlike existing full-wave models, the proposed inverse model naturally reduces to its RLC-based counterpart at low frequencies, and it also avoids the inaccuracy issues caused by stitching R, L, and C elements in an empirical way. Furthermore, it reveals the relationship between a full-wave model and a static model of the layout in a single model and in a closed form. From the view of the integral equation method [19–25], this inverse model can be viewed as a numerical green's function in inhomogeneous problems with arbitrarily shaped lossy conductors.

For each component in the inverse model, we develop fast algorithms to obtain it. The column spaces, such as the nullspace representing the RC effect used to decompose the model, are analytically derived for the curl-curl operator in an arbitrary mesh for an arbitrary order of vector basis function for both a finite-difference and a finite-element based analysis of a general 3-D structure. The computational cost to construct the nullspace is of linear complexity. This nullspace can be used to represent the R- and C- component of the inverse efficiently. The time marching and point-by-point frequency sweeping are avoided for the RC-component as its time and frequency dependence is analytically revealed in the proposed model, which can be done very fast. Moreover, an analytical way to figure out the complementary space of the nullspace is found. Based on the nullspace and its complementary space, we propose a method to get a rigorous full-wave finite-element-based solution of the general lossy problems from high frequencies all the way down to DC, where the dielectric and conductor loss can coexist and both are inhomogeneously distributed. On the other hand, for the full-wave component, we have developed two algorithms to obtain it fast. In the first algorithm, the full-wave component of the inverse model is efficiently represented by a small number of high-frequency modes of the curl-curl operator. A fast method is developed to generate the reduced number of high-frequency modes. The second algorithm is a new iterative method of guaranteed fast convergence. Existing techniques for expediting iterative solutions are mainly based on finding a good preconditioner for the full-wave system matrix. For example, in [31], a diagonal matrix is used as the preconditioner to improve the iterative performance of solving the Electric Field Integral Equation (EFIE); in [7], the symmetric successive overrelaxation (SSOR) is found to be a very effective preconditioner with Conjugate Gradient (CG) method to solve the full-wave matrix from the Finite-Element Method (FEM); [32] proposed a triangular matrix preconditioners to solve both static and harmonic problems resulted from the partial differential equations; [33] developed a preconditioner from an approximate system to solve the finite element-boundary integral system to achieve fast convergence; [34] applied the preconditioned generalized minimal residual method (GMRES) to fast solve the differential equations for the planar circuits. Among these existing preconditioners, the diagonal, block diagonal, SSOR and some other preconditioners are easy to construct and can have fast convergence for some problems. But, their performance is problem dependent and is not always reliable.

Another way is to use approximate inverse preconditioners or incomplete factorization preconditioners. This type of preconditioners is robust even for poorly conditioned system and can have a faster convergence. The shortcomings are additional computational cost and sometimes they can break down during the construction of the preconditioner. Other preconditioners suffer from a similar performance and cost trade-off. Instead of applying a preconditioner, we propose to replace the original matrix with its Laplacian counterpart when operating on the full-wave component. This Laplacian matrix can be built analytically without computational cost. Due to its good property, the iterative solution of this modified system converges fast irrespective of the matrix size, which achieves high performance and trivial cost.

Using the proposed inverse model, not only many accuracy issues related to the existing layout modeling can be addressed, but also we drastically speed up layout modeling and simulation, and provide circuit designers with an effective model for layout automation without low frequency breakdown. In addition, we develop fast and large-scale algorithms to find each component of the inverse in optimal (linear) complexity, where many steps are made analytical, thus further saving CPU run time. The proposed work has been applied to large-scale layout extraction and analysis. Superior performance has been demonstrated in accuracy, efficiency, and capacity

1.3 Dissertation Outline

The remainder of this dissertation is organized as follows.

In Chap. 2, we develop an analytical method to generate the nullspace of the stiffness matrix in the Finite Element Method, not only for the zeroth order, but also for any higher order vector basis function in arbitrary unstructured meshes. This also applies to the discretized curl-curl operator in other partial differential equation methods such as the finite difference method. Using the mesh information, we are able to analytically construct the nullspace without any difficulty, thus avoiding solving an eigenvalue problem for finding the nullspace. The proposed analytical method has been applied to a variety of 2- and 3-D irregular meshes for various orders of vector basis functions. Comparisons with the reference nullspace obtained from a brute-force eigenvalue solution have validated the proposed analytical method.

Based on the analytical nullspace, a rigorous and fast method is developed to overcome the low-frequency breakdown in the finite-element-based analysis of general lossy problems in Chap. 3, where both dielectrics and conductors can be lossy and arbitrarily inhomogeneous. The method is equally valid at high frequencies without any need for changing the formulation. In this method, a full-wave finite-element solution is first decomposed into two components: nullspace component and its complementary one. Each component is then found without breakdown from high down to any low frequency. Both nullspace and its complementary space are analytically generated in this work for an arbitrary mesh, thus incurring no computational cost. From the proposed method, the frequency dependence of a full-wave finite-element solution at low frequencies is also explicitly revealed for general inhomogeneous lossy problems. Numerical results have validated the accuracy and efficiency of the proposed method.

In Chapter 4, we develop a fast FDTD solver to perform the layout extraction and analysis of integrated circuits. In this solver, the time step is not restricted by the small space step encountered in the IC layout. Instead, it can be chosen to be arbitrarily large, thus making the full-wave FDTD simulation from DC to high frequencies become feasible. Meanwhile, the computational cost at each time step is also minimized via an analytical identification of the column space that determines the field solution. Numerical experiments have validated its accuracy and efficiency.

To reveal the physical meaning of the inverse of the Maxwell's equation, we derive a closed-form model of the inverse of the Maxwell's system of equations in the physical layout of an integrated circuit, package, and board in Chap. 5, starting from full-wave Maxwell's equations where **E** and **H** are coupled. In this model, we decompose the inverse rigorously into R-, C-, L- and full-wave components, with neither numerical computation nor approximation, and for an arbitrary physical layout. As a result, each component can be found independently, and then superposed to obtain the total response of a layout to any circuit stimuli. The time marching and point-by-point frequency sweeping are also avoided for the RC-component as its time and frequency dependence is analytically revealed in the proposed model. Moreover, the full-wave component is efficiently represented by a small number of high-frequency modes of the curl-curl operator. Using the proposed model, not only many accuracy issues related to existing layout modeling can be addressed, but also we drastically speed up layout modeling and simulation, and provide circuit designers with an effective model for layout automation. In addition, we develop fast and large-scale algorithms to find each component of the inverse in optimal (linear) complexity, where many steps are made analytical, thus further saving CPU run time. The proposed work has been applied to large-scale layout extraction and analysis. Superior performance has been demonstrated in accuracy, efficiency, and capacity.

To speed up the computation of the full-wave component of the solution, a fast method for accelerating the convergence in an iterative solution of partial differential equation methods is developed in Chap. 6 and 7 in time-domain and frequency-domain, respectively. The field solution in an integrated circuit layout is composed of both a gradient field and a full-wave component that has a nonzero curl. The full-wave component is difficult to be computed efficiently because the discretized curl-curl operator has both zero eigenvalues and large ones inversely proportional to the square of the smallest feature size. We find that the curl-curl operator can be decomposed into a discretized gradient divergence operator and a discretized Laplacian operator, both of which can be constructed analytically from mesh information without any need for computation. The gradient divergence operator vanishes when acting on the full-wave component of the field solution, whose curl is not zero. As a result, we can replace the highly ill-conditioned curl-curl operator by the Laplacian operator for computing the full-wave component. Since the Laplacian operator is full rank and well-conditioned, the new system matrix can converge quickly in a small number of steps. The idea has been applied to solving the system matrix resulting from the

FD-based analysis of realistic integrated circuit layouts, where static and full-wave components co-exist. Numerical results have validated its accuracy and efficiency.

There are also applications where conductors can be accurately approximated as perfect electric conductors (PECs). For this type of problems, if we discretize into the conductors, the number of unknowns to solve is unnecessarily increased. However, if we do not discretize into conductors, there are holes in a continuous computational domain. In Chap. 8, we extend the methods in previous two chapters to handle problems with perfect conductors. We show the curl-curl operator in such a setting can also be decomposed into a gradient divergence operator and a Laplacian. The former only involves nodes in the dielectric region and on the conductor surface, while the latter is in the dielectric region only. Both can be constructed analytically without incurring any computational cost. By replacing the original matrix with the Laplacian matrix, the resultant iterative solution again is significantly accelerated. Numerical experiments validated the accuracy and efficiency.

In Chap. 9, we summarize the work of this thesis and present potential future work.

2. METHOD FOR ANALYTICALLY FINDING THE NULLSPACE OF THE CURL-CURL OPERATOR IN UNSTRUCTURED MESHES

2.1 Introduction

In this chapter, we present an analytical method to generate the nullspace for the stiffness matrix in the Finite Element Method or the discretized curl-curl operator in other partial differential equation methods such as finite difference method. The nullspace size is linearly proportional to the unknown size, or matrix size of the problem being simulated. Hence, the size is nontrivial. The nullspace has many important applications. For example, it can be used to avoid the spurious modes [35]; it can also be used to solve the low-frequency breakdown problem of a full-wave solver [28] [36]. As shown in [28], the original full-wave system matrix can be projected onto the nullspace and its complementary space, yielding a new system free of the low-frequency breakdown. In addition, the nullspace can be employed to compute the DC solutions of circuits, accelerate the convergence of iterative solutions, develop fast solvers for layout extraction [37], etc

The conventional method to generate nullspace of the stiffness matrix is by solving an eigenvalue problem. Although this approach is general for any kind of basis functions and arbitrary meshes, the resultant computational cost is high especially when the matrix size is large. Another approach is to use the tree-cotree method [35]. In this scheme, the edges in a mesh are partitioned into two groups, with the tree edge bases replaced by gradient bases. Such a treatment is not feasible at high frequencies for the edges identified as tree edges. In theory, both gradient field and high-frequency modes co-exist on every edge, or at any point. At low frequencies, the former is dominant; whereas at high frequencies, the latter is more important;

and there also exist intermediate frequencies where both are important. In contrast, the nullspace vector of the stiffness matrix operates in a different way. Take a zeroth order edge element as an example, the length of a nullspace vector is the total number of edges. Each entry in this vector is associated with one original edge basis. Such a vector is also frequency independent, and it can be used to represent the field solution at any frequency together with the complementary eigenvectors of the stiffness matrix. The method in [38] is useful to find the nullspace of the zeroth order basis functions. However, finding the nullspace of any high order vector bases remains a challenge.

In this work, we develop an analytical method, which is also general, to obtain the nullspace of the stiffness matrix regardless of whether the basis function is a zeroth-order edge element or a higher-order basis. Meanwhile, this method is valid for unstructured meshes generated from arbitrarily shaped elements. As a result, the numerical computation of the nullspace can be completely bypassed, thus significantly accelerating the computation when a nullspace is required.

2.2 Background

A full-wave finite-element-method (FEM) based analysis [13] of a general electromagnetic problem results in the following matrix equation in frequency domain:

$$(\overline{\mathbf{S}} + j\omega\overline{\mathbf{R}} - \omega^2\overline{\mathbf{T}})\mathbf{x} = j\omega\mathbf{b}, \tag{2.1}$$

where ω is an angular frequency, $\overline{\mathbf{S}}$ is a stiffness matrix, $\overline{\mathbf{R}}$ is related to conductivity, $\overline{\mathbf{T}}$ is a mass matrix, and $\mathbf{b}$ denotes a current source vector. The element of the stiffness matrix can be written as

$$\overline{\mathbf{S}}_{ij} = \int \mu_r^{-1}(\nabla \times \mathbf{N}_i) \cdot (\nabla \times \mathbf{N}_j) d\Omega, \tag{2.2}$$

where μ_r denotes the relative permeability, and $\mathbf{N}_{i(j)}$ denotes the $i(j)$-th vector basis function used to expand the unknown field. The $\overline{\mathbf{S}}$ multiplied by a vector u represents a discretized $\int \mu_r^{-1}(\nabla \times \mathbf{N}_i) \cdot (\nabla \times \mathbf{E}) d\Omega$ operation, where $\mathbf{E} = \sum \mathbf{N}_j u_j$. Because a nullspace $\overline{\mathbf{V}}_0$ satisfies $\overline{\mathbf{S}}\overline{\mathbf{V}}_0 = 0$, a field that satisfies this property should have a zero curl. Hence, a nullspace vector of the stiffness matrix must be a gradient field.

Stiffness matrix denotes a discretized curl-curl operator of the Maxwell's equations. Such an operator also exists in other partial differential equation methods such as the finite difference method, the finite volume method, the Discontinuous Galerkin method, and the recently developed matrix-free time-domain method [39]. Therefore, the problem studied in this work and its solution can also be applied to other methods where an efficient approach for finding the nullspace is demanded.

2.3 Method for Finding the Nullspace for Zeroth-order Vector Bases

Zeroth-order edge basis functions are the most commonly used bases in the finite element method. In this section, we show how to find the nullspace of the stiffness matrix generated using the zeroth-order basis, for both triangular and tetrahedral meshes.

2.3.1 Nullspace in a triangular element

The zeroth-order edge basis functions in a triangular element have the following form

$$\begin{aligned}
\mathbf{N}_1^e &= l_{23}\mathbf{W}_{23} = l_{23}(\xi_2\nabla\xi_3 - \xi_3\nabla\xi_2) \\
\mathbf{N}_2^e &= l_{31}\mathbf{W}_{31} = l_{31}(\xi_3\nabla\xi_1 - \xi_1\nabla\xi_3) \\
\mathbf{N}_3^e &= l_{12}\mathbf{W}_{12} = l_{12}(\xi_1\nabla\xi_2 - \xi_2\nabla\xi_1)
\end{aligned} \tag{2.3}$$

where l_{ij} denotes the length of the edge connecting node i to node j, as shown in Fig. 2.1, and ξ_i is the area coordinate at node i.

We find the stiffness matrix in a triangular element using the zeroth-order bases has the following factorized form

$$\overline{\mathbf{S}}^e = \mu_r^{-1}\Omega^e(\overline{\mathbf{X}})_{3\times 1}(\overline{\mathbf{X}}^T)_{1\times 3}, \tag{2.4}$$

in which Ω^e is the area of element e, and $\overline{\mathbf{X}}$ is of dimension 3 by 1, which can be written as

$$\overline{\mathbf{X}} = \begin{bmatrix} (\nabla \times \mathbf{N}_1^e)_z \\ (\nabla \times \mathbf{N}_2^e)_z \\ (\nabla \times \mathbf{N}_3^e)_z \end{bmatrix}, \tag{2.5}$$

where subscript z represents the vector's z-component. From (2.4), it can be seen that the rank of $\overline{\mathbf{S}}^e$ is 1, and therefore $\overline{\mathbf{S}}^e$ should have two nullspace vectors since its dimension is 3.

Combining the edge bases entering/leaving each node in Fig. 2.1 with $\frac{1}{l}$ as coefficients, we find

$$\begin{aligned} \frac{1}{l_{31}}\mathbf{N}_2 - \frac{1}{l_{12}}\mathbf{N}_3 &= \nabla\xi_1 \\ \frac{1}{l_{12}}\mathbf{N}_3 - \frac{1}{l_{23}}\mathbf{N}_1 &= \nabla\xi_2 \\ \frac{1}{l_{23}}\mathbf{N}_1 - \frac{1}{l_{31}}\mathbf{N}_2 &= \nabla\xi_3, \end{aligned} \tag{2.6}$$

each of which corresponds to one gradient field. The first equation contains the two vector bases entering or leaving node 1, which is $\mathbf{N}_2$ and $\mathbf{N}_3$, and the coefficient in front of each basis is $\frac{1}{l_i}$, where l_i is the basis's corresponding edge length. The sign in front of each basis is related to the direction of the basis. The same observations can be made for the other two equations in (2.6). Taking a curl of (2.6), we obtain zero.

Therefore, we find the rule to generate a nullspace vector for one triangular element as follows:

- Find all the edges connected to one node.
- If the edge basis enters the node, $\frac{1}{l_i}$ appears on the row corresponding to this edge basis.
- If the edge basis leaves the node, $-\frac{1}{l_i}$ appears on the row corresponding to this edge basis.

An opposite sign convention can also be used, as long as it is used consistently for all nodes.

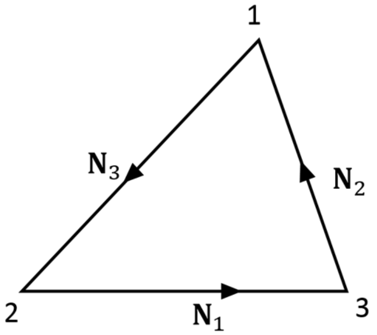


Fig. 2.1.: Zero-order basis functions in one triangular element.

Writing the coefficients of vector bases in each equation of (2.6) into a column vector, we can get the nullspace vectors of $\overline{\mathbf{S}}$. From (2.6), the nullspace vectors recognized from the first two equations can be written as

$$\overline{\mathbf{V}}_0 = \begin{bmatrix} 0 & -\frac{1}{l_{23}} \\ \frac{1}{l_{31}} & 0 \\ -\frac{1}{l_{12}} & \frac{1}{l_{12}} \end{bmatrix}, \tag{2.7}$$

where the row number of each coefficient corresponds to the index of the vector basis the coefficient is attached to. If we multiply $\overline{\mathbf{X}}^T$ by $\overline{\mathbf{V}}_0$, we obtain zero. Hence, $\overline{\mathbf{S}}\overline{\mathbf{V}}_0 = 0$ is satisfied. The column size of (2.7) is 2 instead of 3 because there is one equation linearly dependent on the other two in (2.6) due to $\nabla\xi_1 + \nabla\xi_2 + \nabla\xi_3 = 0$. We can remove any one of the three to obtain linearly independent $\overline{\mathbf{V}}_0$ column vectors shown in (2.7). As can be seen, the size of the nullspace also agrees with that determined from (2.4).

2.3.2 Nullspace in a tetrahedron element

Consider a tetrahedron element, as illustrated in Fig. 2.2. The zeroth-order edge basis functions can be written as

$$\begin{aligned}
\mathbf{N}_1^e &= l_{12}\mathbf{W}_{12} = l_{12}(\xi_1\nabla\xi_2 - \xi_2\nabla\xi_1) \\
\mathbf{N}_2^e &= l_{13}\mathbf{W}_{13} = l_{13}(\xi_1\nabla\xi_3 - \xi_3\nabla\xi_1) \\
\mathbf{N}_3^e &= l_{14}\mathbf{W}_{14} = l_{14}(\xi_1\nabla\xi_4 - \xi_4\nabla\xi_1) \\
\mathbf{N}_4^e &= l_{23}\mathbf{W}_{23} = l_{23}(\xi_2\nabla\xi_3 - \xi_3\nabla\xi_2) \\
\mathbf{N}_5^e &= l_{42}\mathbf{W}_{42} = l_{42}(\xi_4\nabla\xi_2 - \xi_2\nabla\xi_4) \\
\mathbf{N}_6^e &= l_{34}\mathbf{W}_{34} = l_{34}(\xi_3\nabla\xi_4 - \xi_4\nabla\xi_3),
\end{aligned} \tag{2.8}$$

where l_{ij} again denotes the length of the edge connecting node i to node j, and ξ_i is the volume coordinate at node i in this element.

The stiffness matrix in such a 3-D element also has a factorized form as the following

$$\overline{\mathbf{S}}^e = \mu_r^{-1} V^e (\overline{\mathbf{X}})_{6\times 3} (\overline{\mathbf{X}}^T)_{3\times 6}, \tag{2.9}$$

where V^e is the volume of element e, and $\overline{\mathbf{X}}$ is of dimension 6 by 3, which can be written as

$$\overline{\mathbf{X}} = \begin{bmatrix}
(\nabla\times\mathbf{N}_1^e)_x & (\nabla\times\mathbf{N}_1^e)_y & (\nabla\times\mathbf{N}_1^e)_z \\
(\nabla\times\mathbf{N}_2^e)_x & (\nabla\times\mathbf{N}_2^e)_y & (\nabla\times\mathbf{N}_2^e)_z \\
(\nabla\times\mathbf{N}_3^e)_x & (\nabla\times\mathbf{N}_3^e)_y & (\nabla\times\mathbf{N}_3^e)_z \\
(\nabla\times\mathbf{N}_4^e)_x & (\nabla\times\mathbf{N}_4^e)_y & (\nabla\times\mathbf{N}_4^e)_z \\
(\nabla\times\mathbf{N}_5^e)_x & (\nabla\times\mathbf{N}_5^e)_y & (\nabla\times\mathbf{N}_5^e)_z \\
(\nabla\times\mathbf{N}_6^e)_x & (\nabla\times\mathbf{N}_6^e)_y & (\nabla\times\mathbf{N}_6^e)_z
\end{bmatrix} \tag{2.10}$$

in which subscripts x, y, z represent the vector's x-, y-, and z-components, respectively.

Applying the same rule identified for the triangular bases, we find

$$\begin{aligned}
&-\frac{1}{l_{12}}\mathbf{N}_1-\frac{1}{l_{13}}\mathbf{N}_2-\frac{1}{l_{14}}\mathbf{N}_3=\nabla\xi_1\\
&\frac{1}{l_{12}}\mathbf{N}_1-\frac{1}{l_{23}}\mathbf{N}_4+\frac{1}{l_{42}}\mathbf{N}_5=\nabla\xi_2\\
&\frac{1}{l_{13}}\mathbf{N}_2+\frac{1}{l_{23}}\mathbf{N}_4-\frac{1}{l_{43}}\mathbf{N}_6=\nabla\xi_3\\
&\frac{1}{l_{14}}\mathbf{N}_3-\frac{1}{l_{42}}\mathbf{N}_5+\frac{1}{l_{34}}\mathbf{N}_6=\nabla\xi_4.
\end{aligned}\tag{2.11}$$

The coefficient vectors of these equations form the nullspace of $\overline{\mathbf{S}}$,

$$\overline{\mathbf{V}}_0=\begin{bmatrix}-\frac{1}{l_{12}} & \frac{1}{l_{12}} & 0\\ -\frac{1}{l_{13}} & 0 & \frac{1}{l_{13}}\\ -\frac{1}{l_{14}} & 0 & 0\\ 0 & -\frac{1}{l_{23}} & \frac{1}{l_{23}}\\ 0 & \frac{1}{l_{42}} & 0\\ 0 & 0 & -\frac{1}{l_{43}}\end{bmatrix},\tag{2.12}$$

since they satisfy $\overline{\mathbf{X}}^T\overline{\mathbf{V}}_0=0$, and hence $\overline{\mathbf{S}}\overline{\mathbf{V}}_0=0$ is satisfied.

Like the 2-D case, the row index in (2.12) also corresponds to the global index of the vector basis. Here we remove one of the vectors to obtain a linearly independent vector space, as the nullspace size determined by both (2.11) and (2.9) is 3. It is evident that the same method for finding the nullspace in a triangular element can be applied in a tetrahedral element. Take the first vector in (2.12) as an example, the bases connected to node 1 are $\mathbf{N}_1$, $\mathbf{N}_2$, and $\mathbf{N}_3$, whose corresponding edges are edge 12, 13, and 14, respectively. Since $\mathbf{N}_1$, $\mathbf{N}_2$, and $\mathbf{N}_3$ are all leaving node 1, we have $-\frac{1}{l_{12}}$, $-\frac{1}{l_{13}}$ and $-\frac{1}{l_{14}}$ on the first, second and third rows of the nullspace vector in (2.12).

2.3.3 Nullspace in a mesh

For a mesh with m elements, the stiffness matrix can be written as

$$\overline{\mathbf{S}}=\overline{\mathbf{S}}_1+\overline{\mathbf{S}}_2+\cdots+\overline{\mathbf{S}}_m,\tag{2.13}$$

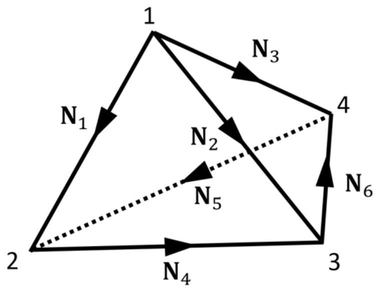


Fig. 2.2.: Zeroth-order basis functions in one tetrahedron element.

where $\overline{\mathbf{S}}_i$ is the stiffness matrix of element i, each of which is an augmented matrix of full unknown size. Since the nullspace vectors generated from the proposed method satisfy $\overline{\mathbf{S}}_i\overline{\mathbf{V}}_0 = 0$ $(i = 1, 2, \cdots, m)$ for each i, they are the nullspace vectors of the entire stiffness matrix as well because $\overline{\mathbf{S}\mathbf{V}}_0 = 0$. Since each vector in the nullspace corresponds to one node, the nullspace size should be $nd - 1$, where nd is the node number in the mesh.

Therefore, for a mesh regardless of it is two or three dimensional, the method to generate the nullspace for the zeroth-order basis is similar to that for one element, which is to go through all the nodes in the mesh (with one node removed, which can be an arbitrary node), and for each node

- Find all edges connected to the node.
- If the edge basis enters the node, $\frac{1}{l_i}$ appears on the row corresponding to this edge basis, where l_i is the length of the edge.
- If the edge basis leaves the node, $-\frac{1}{l_i}$ appears on the row corresponding to this edge basis.

For example, in Fig. 2.3 with triangular edge basis functions, the nullspace vector generated at node 6 is

$$\mathbf{v}_0 = \{0, 0, \frac{1}{l_{16}}, 0, \frac{1}{l_{26}}, 0, \frac{1}{l_{36}}, 0, \frac{1}{l_{46}}, \frac{1}{l_{56}}\}^T. \tag{2.14}$$

We can find $\frac{1}{l_{16}}\mathbf{N}_3+\frac{1}{l_{26}}\mathbf{N}_5+\frac{1}{l_{36}}\mathbf{N}_7+\frac{1}{l_{46}}\mathbf{N}_9+\frac{1}{l_{56}}\mathbf{N}_{10} = \nabla\xi_6$, which is actually a gradient field, and hence satisfying $\overline{\mathbf{S}}\mathbf{v}_0 = 0$. By going through all the nodes and writing a nullspace vector for each of them based on the proposed method, the nullspace $\overline{\mathbf{V}}_0$ can be generated. Each vector corresponds to a $\nabla\xi_i$ (i is the node index) field. Due to the fact that $\sum_{i=1}^{n}\nabla\xi_i = 0$, one vector in the space is redundant and we can remove any one to obtain a linearly independent vector space. This method is not only applicable to triangular and tetrahedral meshes, but also to other types of mesh with curl-conforming vector basis functions. When there is a PEC boundary, the nullspace vectors generated at the nodes on the PEC should be added up to obtain one vector, since the nodes have the same potential.

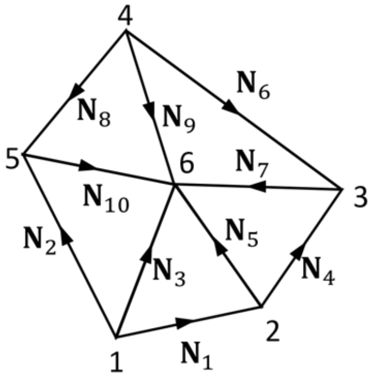


Fig. 2.3.: A triangular mesh.

2.4 Method for Finding the Nullspace for Higher-Order Vector Bases

In this section, we show how to analytically find the nullspace of the stiffness matrix constructed from higher-order bases. We will use the first-order triangular basis functions and tetrahedral basis functions as examples to elaborate the proposed method. But the method is equally applicable to other higher-order bases.

2.4.1 Triangular mesh

First-order homogeneous and interpolatory basis functions for one element

Interpolatory basis functions [40] are popular bases used for field expansion. Take a triangular mesh as an example, the first order interpolatory basis functions, illustrated in Fig. 2.4a, have the following expressions

$$\begin{aligned}
\mathbf{N}_1^e &= l_{23}(3\xi_2 - 1)\mathbf{W}_{23} \\
\mathbf{N}_2^e &= l_{23}(3\xi_3 - 1)\mathbf{W}_{23} \\
\mathbf{N}_3^e &= l_{31}(3\xi_3 - 1)\mathbf{W}_{31} \\
\mathbf{N}_4^e &= l_{31}(3\xi_1 - 1)\mathbf{W}_{31} \\
\mathbf{N}_5^e &= l_{12}(3\xi_1 - 1)\mathbf{W}_{12} \\
\mathbf{N}_6^e &= l_{12}(3\xi_2 - 1)\mathbf{W}_{12} \\
\mathbf{N}_7^e &= l_{23}\frac{9}{2}\xi_1\mathbf{W}_{23} \\
\mathbf{N}_8^e &= l_{31}\frac{9}{2}\xi_2\mathbf{W}_{31}
\end{aligned} \tag{2.15}$$

where $\mathbf{W}_{ij} = \xi_i\nabla\xi_j - \xi_j\nabla\xi_i$ is the edge vector basis associated with the edge connecting node i to node j, $\xi_{i(j)}$ denotes the area coordinate at node $i(j)$. We find that generating the nullspace vectors directly from (2.15) may not be a good idea since it does not share the same rule as we find for the zeroth-order basis. A better idea is to generate the nullspace first from the homogeneous basis functions, and then use them to obtain the nullspace of the interpolatory basis functions.

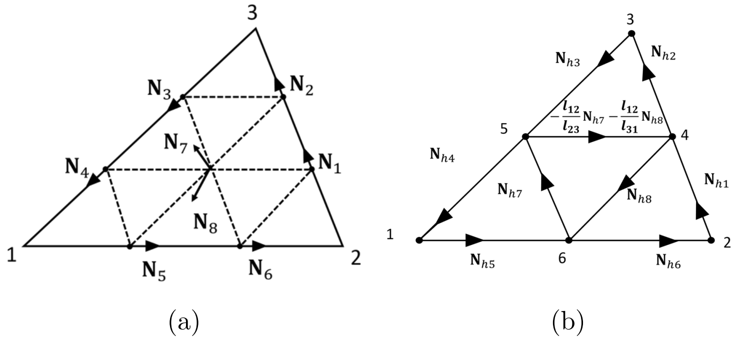

3
N₃
N₂
N₇
N₄
N₁
N₈
1
2
N₅
N₆
(a)
3
N_h3
N_h2
$-\frac{l_{12}}{l_{23}}\mathbf{N}_{h7}-\frac{l_{12}}{l_{31}}\mathbf{N}_{h8}$
5
4
N_h4
N_h1
N_h7
N_h8
1
2
N_h5
6
N_h6
(b)

Fig. 2.4.: (a) First order triangular interpolatory basis functions; (b) First order triangular homogeneous basis functions.

The first-order homogeneous basis functions for a triangular element can be written as

$$\begin{aligned}
\mathbf{N}_{\mathbf{h}_1}^e &= l_{23}\xi_2\mathbf{W}_{23} \\
\mathbf{N}_{\mathbf{h}_2}^e &= l_{23}\xi_3\mathbf{W}_{23} \\
\mathbf{N}_{\mathbf{h}_3}^e &= l_{31}\xi_3\mathbf{W}_{31} \\
\mathbf{N}_{\mathbf{h}_4}^e &= l_{31}\xi_1\mathbf{W}_{31} \\
\mathbf{N}_{\mathbf{h}_5}^e &= l_{12}\xi_1\mathbf{W}_{12} \\
\mathbf{N}_{\mathbf{h}_6}^e &= l_{12}\xi_2\mathbf{W}_{12} \\
\mathbf{N}_{\mathbf{h}_7}^e &= l_{23}\xi_1\mathbf{W}_{23} \\
\mathbf{N}_{\mathbf{h}_8}^e &= l_{31}\xi_2\mathbf{W}_{31}
\end{aligned} \tag{2.16}$$

Here, *homogeneous* [40] means that the polynomials present in the basis are of the homogeneous form

$$\xi_i^r\xi_j^s\xi_k^t, \quad r+s+t=p \tag{2.17}$$

where p is the order of the basis function and i, j, k are the three vertices of the triangular element.

Nullspace of the homogeneous vector bases in a single element

Take the triangular element as an example, Fig. 2.4b shows the distribution of the first order homogeneous basis functions of (2.16). Each vector basis is placed at the point where the dot product of the basis and the unit vector associated with the basis is the largest. Here, the unit vector of each basis is defined as the unit vector tangential to the edge specified by the subscripts of $\mathbf{W}$. Hence, the unit vectors $\widehat{t_i}$ of the eight vector bases are respectively $\widehat{e}_{23}$, $\widehat{e}_{23}$, $\widehat{e}_{31}$, $\widehat{e}_{31}$, $\widehat{e}_{12}$, $\widehat{e}_{12}$, $\widehat{e}_{23}$, and $\widehat{e}_{31}$, with $\widehat{e}_{ij}$ denoting a unit vector pointing from node i to j.

To draw Fig. 2.4, we find the point where $\mathbf{N}_{hi} \cdot \widehat{t_i}$ is the largest. We place $\mathbf{N}_{hi}$ at this point, and also mark the direction of the basis along the unit vector $\widehat{t_i}$. For example, $\mathbf{N}_{h1} \cdot \widehat{t_1}$ is the largest at node 2, and its $\widehat{t_1}$ is along the edge connecting node 2 to 3. Hence, $\mathbf{N}_{h1}$ is drawn leaving node 2 along edge 23. $\mathbf{N}_{h6} \cdot \widehat{t_6}$ is also the largest at node 2, with a direction pointing from node 6 to 2. Then $\mathbf{N}_{h6}$ is marked on the edge 62 pointing from node 6 to node 2. $\mathbf{N_{h7}} \cdot \widehat{t_7}$ is the largest along the edge 65, so $\mathbf{N_{h7}}$ is marked on the edge 65, pointing from node 6 to node 5. There are edges that are not directly labeled with basis functions, such as edge 45. This is because the basis functions on this edge can be represented by the sum of other basis functions as $-\frac{l_{12}}{l_{23}}\mathbf{N_{h7}} - \frac{l_{12}}{l_{31}}\mathbf{N_{h8}}$.

The method for generating the nullspace vectors for the homogeneous basis functions is similar to that for the zeroth-order bases shown in Fig. 2.4b, which is summarized below:

- Find all the homogeneous vector bases connected to a node.
- If the basis enters the node, $\frac{1}{l_i}$ appears on the row corresponding to the global index of this basis function, where l_i is the length of the edge this basis function corresponds to.
- If the basis leaves the node, $-\frac{1}{l_i}$ appears on the row corresponding to the global number of this basis function.

For example, based on the above method, considering node 1, its nullspace vector for the homogeneous basis is

$$\mathbf{v}_1 = \{0, 0, 0, \frac{1}{l_{13}}, -\frac{1}{l_{12}}, 0, 0, 0\}^T \tag{2.18}$$

where the non-zero entries correspond to the bases connected to node 1.

It is interesting to point out that the combination of the basis functions connected to each node in Fig. 2.4b actually corresponds to one gradient term. For example, $\mathbf{v}_1$ in (2.18) is the nullspace vector written for node 1, because $\frac{1}{l_{13}}\mathbf{N}_{h4} - \frac{1}{l_{12}}\mathbf{N}_{h5} = \frac{1}{2}\nabla\xi_1^2$. Another example is the nullspace vector of node 6, which yields $\frac{1}{l_{12}}\mathbf{N}_{h5} - \frac{1}{l_{12}}\mathbf{N}_{h6} - \frac{1}{l_{23}}\mathbf{N}_{h7} + \frac{1}{l_{13}}\mathbf{N}_{h8} = \nabla\xi_1\xi_2$. Following the same principle, after going through all the 6 nodes shown in Fig. 2.4b, we can obtain the space for the gradient of the second order ξ terms: $\nabla\xi_1^2, \nabla\xi_2^2, \nabla\xi_3^2, \nabla\xi_1\xi_2, \nabla\xi_2\xi_3, \nabla\xi_1\xi_3$ (only five of them are linearly independent). These terms are complete to denote the gradient space in one element based on the area coordinates (ξ_1, ξ_2, ξ_3) corresponding to the node $1, 2, 3$.

If we view $\nabla\xi_1^2, \nabla\xi_2^2, \nabla\xi_3^2$ and $\nabla\xi_1\xi_2, \nabla\xi_2\xi_3, \nabla\xi_1\xi_3$ as the nullspace vectors associated with nodes $1, 2, 3$ and edges $12, 23, 13$, respectively, the nullspace size of the first order basis function for one element can be written as $nd - 1 + ed = 3 - 1 + 3 = 5$, where nd denotes the node number, corresponding to node 1, 2, 3, and ed represents the edge number, corresponding to intermediate node 4, 5, 6 in Fig. 2.4b.

Nullspace of the interpolatory vector bases in a single element

After getting the nullspace of the homogeneous basis functions, we can use the relationship between the interpolatory basis functions and the homogeneous basis functions to find the nullspace of the interpolatory basis functions. To do so, we expand the field as

$$\mathbf{E} = \sum_{i=1}^{8} \mathbf{N}_i u_i = \sum_{i=1}^{8} \mathbf{N}_{\mathbf{h}i} u_{hi}, \tag{2.19}$$

where $\mathbf{N}_i$ and $\mathbf{N}_{\mathbf{h}i}$ are the i-th interpolatory and homogeneous basis function, respectively, and u_i and u_{hi} are their corresponding coefficients. The relationship between

$\{\mathbf{u}\}$ and $\{\mathbf{u}_h\}$ can be readily obtained as follows by taking a dot product of the field with the unit vector of the basis, $\widehat{t_i}$, at the point $\vec{r_i}$ where the degree of freedom of the interpolatory basis is assigned at,

$$\begin{aligned}
\begin{bmatrix} \mathbf{E}(\vec{r_1})\cdot\widehat{t_1} \\ \vdots \\ \mathbf{E}(\vec{r_8})\cdot\widehat{t_8} \end{bmatrix} &= \begin{bmatrix} \mathbf{N}_1(\vec{r_1})\cdot\widehat{t_1} & \cdots & \mathbf{N}_8(\vec{r_1})\cdot\widehat{t_1} \\ \vdots & \ddots & \vdots \\ \mathbf{N}_1(\vec{r_8})\cdot\widehat{t_8} & \cdots & \mathbf{N}_8(\vec{r_8})\cdot\widehat{t_8} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_8 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{N_h}_1(\vec{r_1})\cdot\widehat{t_1} & \cdots & \mathbf{N_h}_8(\vec{r_1})\cdot\widehat{t_1} \\ \vdots & \ddots & \vdots \\ \mathbf{N_h}_1(\vec{r_8})\cdot\widehat{t_8} & \cdots & \mathbf{N_h}_8(\vec{r_8})\cdot\widehat{t_8} \end{bmatrix} \begin{bmatrix} u_{h1} \\ \vdots \\ u_{h8} \end{bmatrix}
\end{aligned} \tag{2.20}$$

Let

$$\begin{aligned}
\overline{\mathbf{A}}_{ij} &= [\mathbf{N_h}_j(\vec{r_i})\cdot\widehat{t_i}] \\
\overline{\mathbf{B}}_{ij} &= [\mathbf{N}_j(\vec{r_i})\cdot\widehat{t_i}].
\end{aligned} \tag{2.21}$$

They can be evaluated analytically as the following

$$\begin{aligned}
&\overline{\mathbf{A(B)}}_{ij} \\
&= \begin{cases} P_{\overline{\mathbf{A(B)}}}(\xi_1,\xi_2,\xi_3)(1-\xi_{\widehat{t_j}})\frac{l_{\mathbf{N}_i}}{l_{\widehat{t_j}}}, \text{if } edge(\mathbf{N}_i) = edge(\widehat{t_j}) \\ P_{\overline{\mathbf{A(B)}}}(\xi_1,\xi_2,\xi_3)(-\xi_{\widehat{t_j}})\frac{l_{\mathbf{N}_i}}{l_{\widehat{t_j}}}, \text{if } edge(\mathbf{N}_i) \neq edge(\widehat{t_j}) \end{cases}
\end{aligned} \tag{2.22}$$

where $P_{\overline{\mathbf{A(B)}}}(\xi_1,\xi_2,\xi_3)$ is the polynomial of ξ in the interpolatory or homogeneous basis function, $\xi_{\widehat{t_j}}$ is the area coordinate of the node that is opposite to the edge along $\widehat{t_j}$, and $edge(\mathbf{N}_i)$ is the corresponding edge of the basis function. For example, for one triangular element, $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ are:

$$\overline{\mathbf{A}} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{2}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{2}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{2}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{2}{3} & 0 & 0 \\ \frac{2}{9} & \frac{2}{9} & \frac{-l_{31}}{9l_{23}} & \frac{-l_{31}}{9l_{23}} & \frac{-l_{12}}{9l_{23}} & \frac{-l_{12}}{9l_{23}} & \frac{2}{9} & \frac{-l_{31}}{9l_{23}} \\ \frac{-l_{23}}{9l_{31}} & \frac{-l_{23}}{9l_{31}} & \frac{2}{9} & \frac{2}{9} & \frac{-l_{12}}{9l_{31}} & \frac{-l_{12}}{9l_{31}} & \frac{-l_{23}}{9l_{31}} & \frac{2}{9} \end{bmatrix} \tag{2.23}$$

$$\overline{\mathbf{B}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{-l_{31}}{2l_{23}} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-l_{23}}{2l_{31}} & 1 \end{bmatrix} \tag{2.24}$$

From (2.20), we have

$$\overline{\mathbf{B}}\{\mathbf{u}\} = \overline{\mathbf{A}}\{\mathbf{u}_h\}. \tag{2.25}$$

Hence, the coefficients of the interpolatory basis functions can be obtained from

$$\{\mathbf{u}\} = \overline{\mathbf{B}}^{-1}\overline{\mathbf{A}}\{\mathbf{u}_h\}. \tag{2.26}$$

Notice that $\overline{\mathbf{B}}$ is block diagonal, whose block size is either 1 or 2. Therefore, the inverse of the matrix $\overline{\mathbf{B}}$ can be analytically evaluated. Since the nullspace vector $\{\mathbf{u}_h\}$ has already been obtained, the nullspace vector $\{\mathbf{u}\}$ for the interpolatory basis functions can be found from (2.26).

The nullspace in an arbitrary mesh

The single-element nullspace vectors derived in the above can be readily extended to the nullspace of an arbitrary mesh composed of many elements. We generate a nullspace vector for each node by placing the nullspace vector coefficients in each element in a global vector based on the global index of each vector basis. This is done for the homogeneous basis functions. Then we generate the matrices for $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ based on (2.22), and $\overline{\mathbf{B}}$ remains to be block diagonal of block size either 1 or 2. We then apply $\{\mathbf{u}\} = \overline{\mathbf{B}}^{-1}\overline{\mathbf{A}}\{\mathbf{u}_h\}$ to obtain the final nullspace of the interpolatory bases. The number of linearly independent nullspace vectors is $nd - 1 + ed$ for the first order basis functions, where nd is the node number, and ed denotes the edge number.

2.4.2 Tetrahedral mesh

First-order homogeneous and interpolatory basis functions for one element

The first-order interpolatory basis functions [40] for one tetrahedral element, shown in Fig. 2.5, can be expressed as

$$\begin{aligned}
&\mathbf{N}_1^e = l_{12}(3\xi_1 - 1)\mathbf{W}_{12} \quad \mathbf{N}_2^e = l_{12}(3\xi_2 - 1)\mathbf{W}_{12} \\
&\mathbf{N}_3^e = l_{13}(3\xi_1 - 1)\mathbf{W}_{13} \quad \mathbf{N}_4^e = l_{13}(3\xi_3 - 1)\mathbf{W}_{13} \\
&\mathbf{N}_5^e = l_{14}(3\xi_1 - 1)\mathbf{W}_{14} \quad \mathbf{N}_6^e = l_{14}(3\xi_4 - 1)\mathbf{W}_{14} \\
&\mathbf{N}_7^e = l_{23}(3\xi_2 - 1)\mathbf{W}_{23} \quad \mathbf{N}_8^e = l_{23}(3\xi_3 - 1)\mathbf{W}_{23} \\
&\mathbf{N}_9^e = l_{24}(3\xi_2 - 1)\mathbf{W}_{24} \quad \mathbf{N}_{10}^e = l_{24}(3\xi_4 - 1)\mathbf{W}_{24} \\
&\mathbf{N}_{11}^e = l_{34}(3\xi_3 - 1)\mathbf{W}_{34} \quad \mathbf{N}_{12}^e = l_{34}(3\xi_4 - 1)\mathbf{W}_{34} \\
&\mathbf{N}_{13}^e = l_{13}3\xi_2\mathbf{W}_{13} \quad \mathbf{N}_{14}^e = l_{23}3\xi_1\mathbf{W}_{23} \\
&\mathbf{N}_{15}^e = l_{14}3\xi_2\mathbf{W}_{14} \quad \mathbf{N}_{16}^e = l_{24}3\xi_1\mathbf{W}_{24} \\
&\mathbf{N}_{17}^e = l_{14}3\xi_3\mathbf{W}_{14} \quad \mathbf{N}_{18}^e = l_{34}3\xi_1\mathbf{W}_{34} \\
&\mathbf{N}_{19}^e = l_{24}3\xi_3\mathbf{W}_{24} \quad \mathbf{N}_{20}^e = l_{34}3\xi_2\mathbf{W}_{34}
\end{aligned} \tag{2.27}$$

where $\mathbf{W}_{ij} = \xi_i \nabla \xi_j - \xi_j \nabla \xi_i$ is the edge basis functions for the edge ij, $\xi_{i(j)}$ denotes the volume coordinate at node $i(j)$. The homogeneous basis functions, which are shown in Fig. 2.6, for one tetrahedral element, are

$$\begin{aligned}
&\mathbf{N}_{h1}^e = l_{12}\xi_1\mathbf{W}_{12} \quad \mathbf{N}_{h2}^e = l_{12}\xi_2\mathbf{W}_{12}\\
&\mathbf{N}_{h3}^e = l_{13}\xi_1\mathbf{W}_{13} \quad \mathbf{N}_{h4}^e = l_{13}\xi_3\mathbf{W}_{13}\\
&\mathbf{N}_{h5}^e = l_{14}\xi_1\mathbf{W}_{14} \quad \mathbf{N}_{h6}^e = l_{14}\xi_4\mathbf{W}_{14}\\
&\mathbf{N}_{h7}^e = l_{23}\xi_2\mathbf{W}_{23} \quad \mathbf{N}_{h8}^e = l_{23}\xi_3\mathbf{W}_{23}\\
&\mathbf{N}_{h9}^e = l_{24}\xi_2\mathbf{W}_{24} \quad \mathbf{N}_{h10}^e = l_{24}\xi_4\mathbf{W}_{24}\\
&\mathbf{N}_{h11}^e = l_{34}\xi_3\mathbf{W}_{34} \quad \mathbf{N}_{h12}^e = l_{34}\xi_4\mathbf{W}_{34}\\
&\mathbf{N}_{h13}^e = l_{13}\xi_2\mathbf{W}_{13} \quad \mathbf{N}_{h14}^e = l_{23}\xi_1\mathbf{W}_{23}\\
&\mathbf{N}_{h15}^e = l_{14}\xi_2\mathbf{W}_{14} \quad \mathbf{N}_{h16}^e = l_{24}\xi_1\mathbf{W}_{24}\\
&\mathbf{N}_{h17}^e = l_{14}\xi_3\mathbf{W}_{14} \quad \mathbf{N}_{h18}^e = l_{34}\xi_1\mathbf{W}_{34}\\
&\mathbf{N}_{h19}^e = l_{24}\xi_3\mathbf{W}_{24} \quad \mathbf{N}_{h20}^e = l_{34}\xi_2\mathbf{W}_{34}
\end{aligned} \tag{2.28}$$

These bases are marked at the points where $\mathbf{N}_{hi} \cdot \widehat{t_i}$ is the largest. Here, $\mathbf{N}_{hi}$ is the i-th homogeneous basis function and $\widehat{t_i}$ is the unit vector along $\mathbf{N}_{hi}$'s corresponding edge.

Nullspace of the homogeneous vector bases in a single element

Again, we first find the nullspace based on the homogeneous basis functions, and then find that of the interpolatory bases. We find the same method for the triangular bases is valid here. For example, around node 1, we have

$$-\frac{1}{l_{12}}\mathbf{N}_{h1} - \frac{1}{l_{13}}\mathbf{N}_{h3} - \frac{1}{l_{14}}\mathbf{N}_{h5} = \frac{1}{2}\nabla\xi_1^2. \tag{2.29}$$

Around node 5, we have

$$\begin{aligned}
&\frac{1}{l_{12}}\mathbf{N}_{h1} - \frac{1}{l_{12}}\mathbf{N}_{h2} - \frac{1}{l_{23}}\mathbf{N}_{h14} - \frac{1}{l_{13}}\mathbf{N}_{h13} -\\
&\frac{1}{l_{14}}\mathbf{N}_{h15} - \frac{1}{l_{14}}\mathbf{N}_{h16} = \nabla\xi_1\xi_2.
\end{aligned} \tag{2.30}$$

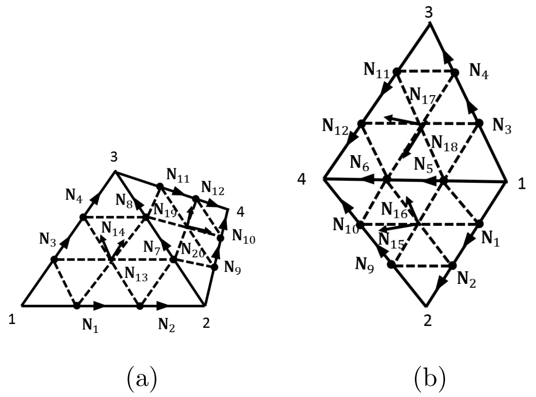
Based on this approach, we can find the complete gradient space of the second order ξ terms after going through all the nodes: $\nabla\xi_1^2$, $\nabla\xi_2^2$, $\nabla\xi_3^2$, $\nabla\xi_4^2$, $\nabla\xi_1\xi_2$, $\nabla\xi_1\xi_3$, $\nabla\xi_1\xi_4$, $\nabla\xi_2\xi_3$, $\nabla\xi_2\xi_4$, $\nabla\xi_3\xi_4$ (one of them is redundant).

Nullspace of the interpolatory vector bases in a single element and in a mesh

The relationship between the interpolatory basis coefficients and the homogeneous basis coefficients again can be written as (2.26). $\overline{\mathbf{B}}$ remains to be a block diagonal matrix whose block size 1 or 2, which means that $\overline{\mathbf{B}}^{-1}$ can be found analytically. Eq. (2.31) shows the matrix $\overline{\mathbf{B}}$ in one tetrahedral element, where $\overline{\mathbf{I}}_{12}$ denotes an identity matrix of size 12,

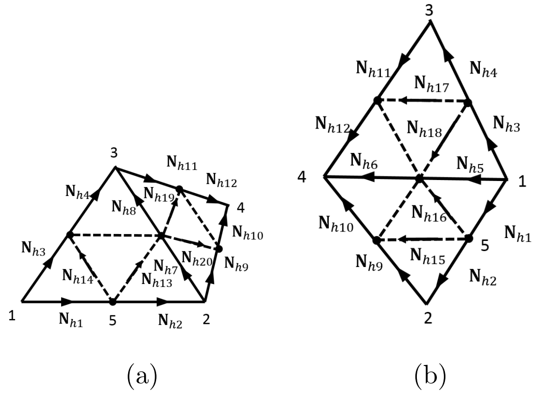
$$\overline{\mathbf{B}} = \begin{bmatrix} \overline{\mathbf{I}}_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{2}{3} & \frac{l_{23}}{3l_{13}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{l_{13}}{3l_{23}} & \frac{2}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{3} & \frac{l_{24}}{3l_{14}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{l_{14}}{3l_{24}} & \frac{2}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{3} & \frac{l_{34}}{3l_{14}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{l_{14}}{3l_{34}} & \frac{2}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2}{3} & \frac{l_{34}}{3l_{24}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{l_{24}}{3l_{34}} & \frac{2}{3} \end{bmatrix} \tag{2.31}$$

When generating the nullspace in an arbitrary mesh, first, we generate the nullspace vectors by finding all the bases connected to one node from the distribution of the homogeneous basis functions. We then place the nullspace vector coefficients in a global nullspace vector based on the global index of the basis and, thus, we obtain $\{\mathbf{u}_h\}$. We then generate $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ based on (2.22), from which the nullspace of interpolatory bases is obtained from $\{\mathbf{u}\} = \overline{\mathbf{B}}^{-1}\overline{\mathbf{A}}\{\mathbf{u}_h\}$.



(a) (b)

Fig. 2.5.: First order tetrahedral interpolatory basis functions.



(a) (b)

Fig. 2.6.: First order tetrahedral homogeneous basis functions.

2.4.3 Higher-order bases

For even higher-order bases, the same method developed for the first-order bases can be used to generate the nullspace. In Table. 2.1 and 2.2, we list the nullspace size of the stiffness matrix for the triangular mesh and the tetrahedral mesh for different orders of vector bases. In these tables, nd, ed, em and $face$ denote node number, edge number, element number and face number in the mesh. The variable $m = \sum_{i=2}^{n-1} \frac{i*(i-1)}{2}$.

Table 2.1.: The Nullspace Size for Triangular Basis Functions of Different Orders

Triangular mesh	Nullspace size
Zeroth order	$nd - 1$
First order	$nd - 1 + ed$
Second order	$nd - 1 + 2 * ed + em$
nth order	$nd - 1 + n * ed + \frac{(n-1)*n}{2} * em$

Table 2.2.: The Nullspace Size for Tetrahedral Basis Functions of Different Orders

Tetrahedral mesh	Nullspace size
Zeroth order	$nd - 1$
First order	$nd - 1 + ed$
Second order	$nd - 1 + 2 * ed + face$
nth order	$nd - 1 + n * ed + \frac{(n-1)*n}{2} * face + m * em$

2.5 Numerical Validation

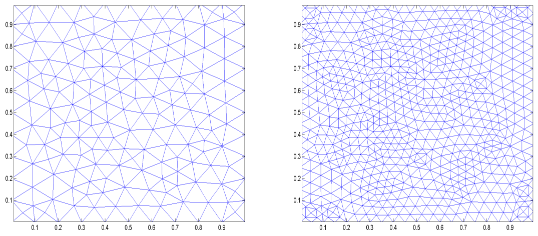
In this section, we validate the proposed analytical method for finding the nullspace. We also apply it to solve the low frequency breakdown problem of full-wave solvers, and to perform fast layout parasitics extraction.

2.5.1 Validation in unstructured triangular and tetrahedral meshes

To validate the proposed analytical method, we generate the nullspace from a number of irregular triangular and tetrahedral meshes, and for different orders of vector basis functions. Two criteria are designed to assess the accuracy of the proposed method. One is to use $\frac{||\overline{\mathbf{S}}\overline{\mathbf{V}}||}{||\overline{\mathbf{S}}||||\overline{\mathbf{V}}||}$, where $\overline{\mathbf{S}}$ is the stiffness matrix and $\overline{\mathbf{V}}$ is the nullspace vectors generated from the proposed method. If the generated nullspace vectors are accurate, the value of $\frac{||\overline{\mathbf{S}}\overline{\mathbf{V}}||}{||\overline{\mathbf{S}}||||\overline{\mathbf{V}}||}$ should approach to zero. The other criterion is to compare the nullspace $\overline{\mathbf{V}}$ generated from the proposed method with the nullspace $\overline{\mathbf{V}_0}$ obtained from a brute-force eigenvalue solution. This brute-force solution is to solve $\overline{\mathbf{S}}\mathbf{x} = \lambda\mathbf{x}$, and choose the eigenvectors corresponding to zero eigenvalues. If $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}$ are the same column space, their transpose, $\overline{\mathbf{V}}^T$ and $\overline{\mathbf{V}}_0^T$, should have the same reduced row echelon form. Notice that the nullspace $\overline{\mathbf{V}}$ multiplied by any full rank matrix $\overline{\mathbf{Z}}$ remains to be a nullspace of $\overline{\mathbf{S}}$ because $\overline{\mathbf{SVZ}} = 0$, in other words, a liner superposition of the nullspace vectors remains to be a nullspace vector. Hence, we cannot directly check the difference between $\overline{\mathbf{V}}$ and $\overline{\mathbf{V}}_0$. However, they should have the same reduced row echelon form.

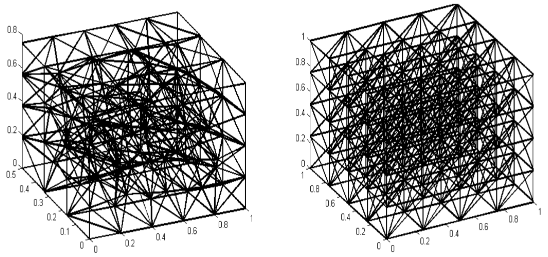
The results obtained from the proposed method for the triangular and the tetrahedral mesh are shown in Tables. 2.3 and 2.4, for the zeroth-, the first, and the second order basis functions respectively. The meshes are shown in Fig. 2.7 and Fig. 2.8. In the mesh of Fig. 2.7a, the node number is 185, edge number is 512 and element number is 328. In the mesh shown in Fig. 2.7b, the node number is 697, the edge number is 2,008 and the element number is 1,312. From Table 2.3, we can find that the nullspace size shown in the third column matches that of Table 2.1 very well. In addition, the value of $\frac{\|\overline{\mathbf{SV}}\|}{\|\overline{\mathbf{S}}\|\|\overline{\mathbf{V}}\|}$ is shown to be at the machine precision, which is even smaller than that of the reference $\overline{\mathbf{V}}_0$ found from a brute-force eigenvalue solution, and hence $\overline{\mathbf{V}}$ is more accurate. The reduced row echelon form is also shown to agree well with that of $\overline{\mathbf{V}}_0$.

For the tetrahedral mesh shown in Fig. 2.8a, the node number is 113, the edge number is 544, the element number is 350, and the face number is 782. In Fig. 2.8b, the node number is 125, the edge number is 604, the element number is 384, and the face number is 864. From Table 2.4, it can be seen that the nullspace size shown in the third column agrees very well with that predicted in Table 2.2. From the fourth column, we can see that the value of $\frac{\|\overline{\mathbf{SV}}\|}{\|\overline{\mathbf{S}}\|\|\overline{\mathbf{V}}\|}$ is very small, which verifies that $\overline{\mathbf{V}}$ is the nullspace of $\overline{\mathbf{S}}$. The sixth column shows that the value of $\frac{\|rref(\overline{\mathbf{V}}_0^T)-rref(\overline{\mathbf{V}}^T)\|}{\|rref(\overline{\mathbf{V}}_0^T)\|}$ is small, confirming that the analytical nullspace found in this work is accurate.



(a) (b)

Fig. 2.7.: Examples of triangular meshes.



(a) (b)

Fig. 2.8.: The tetrahedral meshes for testing the proposed method.

2.6 Conclusion

In this paper, an analytical method is developed to generate the nullspace of the stiffness matrix for both the zeroth order, and higher order vector basis functions in arbitrary unstructured meshes. In this method, we utilize the properties of the vector basis functions, and combine the bases in such a way to build a gradient field, which is the nullspace of the stiffness matrix. Using the mesh information, we are able to analytically construct the nullspace analytically, thus avoiding solving an eigenvalue problem for finding the nullspace. In addition, a geometrical interpretation of the curl-conforming vector basis function of an arbitrary order is also developed, which facilitates the generation of the nullspace. Instead of directly obtaining the nullspace

Table 2.3.: Results for Nullspace in Triangular Meshes

Basis function		Nullspace size	$\frac{\|\overline{\mathbf{S}}\overline{\mathbf{V}}\|}{\|\overline{\mathbf{S}}\|\|\overline{\mathbf{V}}\|}$	$\frac{\|\overline{\mathbf{S}}\overline{\mathbf{V}}_0\|}{\|\overline{\mathbf{S}}\|\|\overline{\mathbf{V}}_0\|}$	$\frac{\|rref(\overline{\mathbf{V}}_0^T)-rref(\overline{\mathbf{V}}^T)\|}{\|rref(\overline{\mathbf{V}}_0^T)\|}$
Zeroth order	Mesh1	184	8.6214×10^{-17}	1.0523×10^{-15}	1.5809×10^{-13}
	Mesh2	696	6.3897×10^{-17}	1.15488×10^{-15}	7.3595×10^{-13}
First order	Mesh1	696	5.4415×10^{-17}	8.5516×10^{-16}	1.2275×10^{-12}
	Mesh2	2704	6.1292×10^{-17}	1.4678×10^{-15}	1.0197×10^{-11}
Second order	Mesh1	1536	4.5698×10^{-16}	1.4242×10^{-15}	2.3434×10^{-12}
	Mesh2	6024	3.4928×10^{-16}	1.9527×10^{-15}	5.1×10^{-12}

of the interpolatory bases, we first obtain the nullspace of the homogeneous bases so that we can unify the method for finding the nullspace of the zeroth-order bases with that of the higher-order bases. The proposed analytical method has been validated by extensive numerical experiments. It has also been applied to solve the low-frequency breakdown problem of full-wave solvers, and perform fast layout parasitic extraction.

Table 2.4.: Results for Nullspace in Tetrahedral Meshes

Basis function		Nullspace size	$\frac{\|\overline{\mathbf{S}}\overline{\mathbf{V}}\|}{\|\overline{\mathbf{S}}\|\|\overline{\mathbf{V}}\|}$	$\frac{\|\overline{\mathbf{S}}\overline{\mathbf{V}}_0\|}{\|\overline{\mathbf{S}}\|\|\overline{\mathbf{V}}_0\|}$	$\frac{\|rref(\overline{\mathbf{V}}_0^T)-rref(\overline{\mathbf{V}}^T)\|}{\|rref(\overline{\mathbf{V}}_0^T)\|}$
Zeroth order	Mesh1	112	4.9122 $\times 10^{-17}$	5.0249 $\times 10^{-16}$	2.7778 $\times 10^{-14}$
	Mesh2	124	5.1242 $\times 10^{-17}$	5.4182 $\times 10^{-16}$	3.9393 $\times 10^{-14}$
First order	Mesh1	656	3.3241 $\times 10^{-17}$	1.0318 $\times 10^{-15}$	4.9559 $\times 10^{-13}$
	Mesh2	728	4.0596 $\times 10^{-17}$	1.1519 $\times 10^{-15}$	2.7543 $\times 10^{-13}$
Second order	Mesh1	1982	3.4606 $\times 10^{-17}$	1.8994 $\times 10^{-15}$	2.8017 $\times 10^{-12}$
	Mesh2	2196	4.2985 $\times 10^{-17}$	1.7961 $\times 10^{-15}$	2.3354 $\times 10^{-12}$

3. A RIGOROUS METHOD TO LOW-FREQUENCY BREAKDOWN IN FULL-WAVE FINITE-ELEMENT-BASED ANALYSIS OF GENERAL LOSSY PROBLEMS

3.1 Introduction

A full-wave finite-element-based solution of Maxwell's equations breaks down at low frequencies. This problem is especially severe when dealing with VLSI circuit problems, because the breakdown frequency falls right in the range of circuit operating frequencies [28]. Therefore, it is important to solve the low-frequency breakdown problem. Among existing methods for solving the problem, one class of methods stitches a static- or quasistatic-based electromagnetic solver with a full-wave-based electromagnetic solver. However, this kind of method is inaccurate due to the fundamental approximation of decoupled **E** and **H**, which is only true at DC. Also, this approach requires deciding at which frequency to switch between different solvers. It also fails to handle the scenario where the solution of Maxwell's equations is not a static solution, while a full-wave solver still breaks down. Another class is to extend the validity of full-wave solvers to low frequencies. Existing approaches that belong to this category utilize low-frequency approximations. For example, the tree-cotree splitting [27] was used to provide an approximate Helmholtz decomposition for edge elements in finite-element-based methods. However, numerical errors exists in these methods at low frequencies. In addition, while lossless problems are extensively studied, the lossy problems are seldom examined for overcoming the low-frequency breakdown. In [29], a method without using any low-frequency approximation is proposed to solve the breakdown problem involving lossy conductors embedded in inhomogeneous dielectrics. For cases where conductivity is space dependent, whose value can also be small like that in lossy dielectrics, the formulations of [29] are not applicable. In addition, eigenvalue solutions are involved in [29], which can be expensive for simulating large-scale problems.

In this work, we propose a method to obtain a rigorous full-wave finite-element-based solution of general lossy problems from high frequencies all the way down to DC, where the dielectric and conductor loss can co-exist and both can be inhomogeneously distributed in space. In addition, the column space we use to decompose a full-wave solution is generated analytically without solving an eigenvalue problem, thus significantly speeding up computation.

3.2 Proposed method

A full-wave FEM-based analysis of such a problem results in the following matrix equation in the frequency domain:

$$(\overline{\mathbf{S}} + j\omega\overline{\mathbf{R}} - \omega^2\overline{\mathbf{T}})\mathbf{x} = j\omega\mathbf{b} \tag{3.1}$$

where $\overline{\mathbf{S}}$ is the stiffness matrix, $\overline{\mathbf{R}}$ is related with conductivity and $\overline{\mathbf{T}}$ is the mass matrix. When frequency is low, the frequency dependent terms will be lost in (4.1) due to finite machine precision. Since $\overline{\mathbf{S}}$ is singular, the resultant numerical solution of (4.1) would break down.

The solution $\mathbf{x}$ can be written as the expansion of the space $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_h$, where $\overline{\mathbf{V}}_0$ is the nullspace of the stiffness matrix $\overline{\mathbf{S}}$ and $\overline{\mathbf{V}}_h$ is the complementary space of $\overline{\mathbf{V}}_0$. This can be written as

$$\mathbf{x} = \overline{\mathbf{V}}_0\mathbf{x}_0 + \overline{\mathbf{V}}_h\mathbf{x}_h \tag{3.2}$$

From (3.2), we can see that the solution consists of $\mathbf{x}_0$ and $\mathbf{x}_h$, which are the coefficients of nullspace $\overline{\mathbf{V}}_0$ and its complementary space $\overline{\mathbf{V}}_h$, where $\overline{\mathbf{V}}_0$ can be got analytically from the previous chapter and $\overline{\mathbf{V}}_h$ is also got analytically shown later. The $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_h$ can be generated with little computational cost.

Multiply $\overline{\mathbf{V}}_0^T$ to both sides of (4.1), and we can get

$$\overline{\mathbf{V}}_0^T(j\omega\overline{\mathbf{R}} - \omega^2\overline{\mathbf{T}})\overline{\mathbf{V}}_0\mathbf{x}_0 = j\omega\overline{\mathbf{V}}_0^T\mathbf{b} - \overline{\mathbf{V}}_0^T(j\omega\overline{\mathbf{R}} - \omega^2\overline{\mathbf{T}})\overline{\mathbf{V}}_h\mathbf{x}_h \tag{3.3}$$

However, the matrix $\overline{\mathbf{V}}_0^T(j\omega\overline{\mathbf{R}} - \omega^2\overline{\mathbf{T}})\overline{\mathbf{V}}_0$ may still break down at low frequencies because the term $\omega^2\overline{\mathbf{V}}_0^T\overline{\mathbf{T}}\overline{\mathbf{V}}_0$ will be neglected and the matrix $\overline{\mathbf{V}}_0^T(j\omega\overline{\mathbf{R}} - \omega^2\overline{\mathbf{T}})\overline{\mathbf{V}}_0$ becomes singular. In order to solve this problem, we further decompose $\mathbf{x}_0$ into the space $\overline{\mathbf{X}}_0$ and $\overline{\mathbf{X}}_h$, where $\overline{\mathbf{X}}_0$ is the nullspace of $\overline{\mathbf{V}}_0^T\overline{\mathbf{R}}\overline{\mathbf{V}}_0$ and $\overline{\mathbf{X}}_h$ is the orthogonal space of $\overline{\mathbf{X}}_0$. $\overline{\mathbf{X}}_0$ and $\overline{\mathbf{X}}_h$ can be generated analytically, which is shown later. The solution $\mathbf{x}_0$ can be written as $\mathbf{x}_0 = \overline{\mathbf{X}}_0\mathbf{u}_0 + \overline{\mathbf{X}}_h\mathbf{u}_h$.

The space $\overline{\mathbf{X}}_0$ and $\overline{\mathbf{X}}_h$ are orthogonal to each other, that is $\overline{\mathbf{X}}_0^T\overline{\mathbf{X}}_h = 0$. In order to find $\overline{\tilde{\mathbf{X}}}_h$, which satisfies $\overline{\mathbf{X}}_0^T\overline{\mathbf{T}}_0\overline{\tilde{\mathbf{X}}}_h = 0$, we can set $\overline{\tilde{\mathbf{X}}}_h = \overline{\mathbf{T}}_0^{-1}\overline{\mathbf{X}}_h$. The newly found $\overline{\tilde{\mathbf{X}}}_h$ can decouple the system when solving $\mathbf{u}_0$ and $\mathbf{u}_h$. Multiply $\overline{\mathbf{X}}_0^T$ and $\overline{\tilde{\mathbf{X}}}_h^T$ to both sides of (3.3). Another thing is to multiply $\overline{\mathbf{V}}_h^T$ to both sides of (4.1) and, finally combining all the equations, we can get

$$\begin{bmatrix} j\omega\overline{\mathbf{X}}_0^T\overline{\mathbf{T}}_0\overline{\mathbf{X}}_0 & 0 & \overline{\mathbf{A}}_{00,h} \\ 0 & \overline{\tilde{\mathbf{X}}}_h^T(\overline{\mathbf{R}}_0+j\omega\overline{\mathbf{T}}_0)\overline{\tilde{\mathbf{X}}}_h & \overline{\mathbf{A}}_{0h,h} \\ \overline{\mathbf{A}}_{00,h}^T & \overline{\mathbf{A}}_{0h,h}^T & \overline{\mathbf{V}}_h^T(\overline{\mathbf{S}}/(j\omega)+\overline{\mathbf{R}}+j\omega\overline{\mathbf{T}})\overline{\mathbf{V}}_h \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_h \\ \mathbf{x}_h \end{bmatrix}$$

$$= \begin{bmatrix} \overline{\mathbf{X}}_0^T\overline{\mathbf{V}}_0^T b \\ \overline{\tilde{\mathbf{X}}}_h^T\overline{\mathbf{V}}_0^T b \\ \overline{\mathbf{V}}_h^T b \end{bmatrix}, \tag{3.4}$$

where $\overline{\mathbf{T}}_0 = \overline{\mathbf{V}}_0^T\overline{\mathbf{T}}\overline{\mathbf{V}}_0$, $\overline{\mathbf{R}}_0 = \overline{\mathbf{V}}_0^T\overline{\mathbf{R}}\overline{\mathbf{V}}_0$, $\overline{\mathbf{A}}_{00,h} = \overline{\mathbf{X}}_0^T\overline{\mathbf{V}}_0^T(\overline{\mathbf{R}} + j\omega\overline{\mathbf{T}})\overline{\mathbf{V}}_h$, and $\overline{\mathbf{A}}_{0h,h} = \overline{\tilde{\mathbf{X}}}_h^T\overline{\mathbf{V}}_0^T(\overline{\mathbf{R}} + j\omega\overline{\mathbf{T}})\overline{\mathbf{V}}_h$.

The final solution can be got by solving the equations in first two rows to express $\mathbf{u}_0$ and $\mathbf{u}_h$ in terms of $\mathbf{x}_h$. Then we can substitute $\mathbf{u}_0$ and $\mathbf{u}_h$ into the equation on the third row to get the $\mathbf{x}_h$, the equation is shown as

$$\overline{\mathbf{A}}\mathbf{x}_h = \tilde{\mathbf{b}} \tag{3.5}$$

where

$$\begin{aligned} \overline{\mathbf{A}} = {} & \overline{\mathbf{V}}_h^T(\overline{\mathbf{S}} + j\omega\overline{\mathbf{R}} - \omega^2\overline{\mathbf{T}})\overline{\mathbf{V}}_h \\ & - \overline{\mathbf{V}}_h^T(\overline{\mathbf{R}} + j\omega\overline{\mathbf{T}})\overline{\mathbf{V}}_0\overline{\mathbf{X}}_0(\overline{\mathbf{X}}_0^T\overline{\mathbf{T}}_0\overline{\mathbf{X}}_0)^{-1}\overline{\mathbf{X}}_0^T\overline{\mathbf{V}}_0^T(\overline{\mathbf{R}} + j\omega\overline{\mathbf{T}})\overline{\mathbf{V}}_h \\ & - \overline{\mathbf{V}}_h^T(j\omega\overline{\mathbf{R}} - \omega^2\overline{\mathbf{T}})\overline{\mathbf{V}}_0\overline{\tilde{\mathbf{X}}}_h(\overline{\tilde{\mathbf{X}}}_h^T\overline{\mathbf{R}}_0 + j\omega\overline{\mathbf{T}}_0)\overline{\tilde{\mathbf{X}}}_h)^{-1}\overline{\tilde{\mathbf{X}}}_h^T\overline{\mathbf{V}}_0^T \\ & (\overline{\mathbf{R}} + j\omega\overline{\mathbf{T}})\overline{\mathbf{V}}_h \end{aligned} \tag{3.6}$$

$\mathbf{u}_0$ and $\mathbf{u}_h$ can also be solved because they are expressed in terms of $\mathbf{x}_h$. With $\mathbf{u}_0$, $\mathbf{u}_h$ and $\mathbf{x}_h$, the final solution can be got by the expression $\mathbf{x} = \overline{\mathbf{V}}_0\overline{\mathbf{X}}_0\mathbf{u}_0 + \overline{\mathbf{V}}_0\overline{\tilde{\mathbf{X}}}_h\mathbf{u}_h + \overline{\mathbf{V}}_h\mathbf{x}_h$.

The whole process doesn't break down at low frequencies. It involves 3 inverse operations, as $(\overline{\mathbf{X}}_0^T\overline{\mathbf{T}}_0\overline{\mathbf{X}}_0)^{-1}$, $(\overline{\tilde{\mathbf{X}}}_h^T(j\omega\overline{\mathbf{R}}_0 - \omega^2\overline{\mathbf{T}}_0)\overline{\tilde{\mathbf{X}}}_h)^{-1}$ and $\overline{\mathbf{A}}^{-1}$. The low frequency approximation of the first matrix is still $(\overline{\mathbf{X}}_0^T\overline{\mathbf{T}}_0\overline{\mathbf{X}}_0)^{-1}$, which doesn't depend on the frequency and never breaks down at any frequency. For the second matrix, the matrix becomes $(\overline{\mathbf{X}}_h^T(j\omega\overline{\mathbf{R}}_0)\overline{\mathbf{X}}_h)^{-1}$ for approximation at very low frequencies, which is certainly not singular because $\overline{\mathbf{X}}_h$ is the complementary space for $\overline{\mathbf{R}}_0$'s nullspace. The most complicated one is the matrix $\overline{\mathbf{A}}$. At low frequencies, we can write this matrix as $\overline{\mathbf{V}}_h^T\overline{\mathbf{SV}}_h + \overline{\mathbf{V}}_h^T\overline{\mathbf{RV}}_0\overline{\mathbf{X}}_0(\overline{\mathbf{X}}_0^T\overline{\mathbf{T}}_0\overline{\mathbf{X}}_0)^{-1}\overline{\mathbf{X}}_0^T\overline{\mathbf{V}}_0^T\overline{\mathbf{RV}}_h$. This matrix actually equals to $\overline{\mathbf{V}}_h^T\overline{\mathbf{SV}}_h$ because we can show that $\overline{\mathbf{RV}}_0\overline{\mathbf{X}}_0 = 0$. Therefore the matrix $\overline{\mathbf{A}} = \overline{\mathbf{V}}_h^T\overline{\mathbf{SV}}_h$. It doesn't break down at low frequencies because $\overline{\mathbf{V}}_h$ is the complementary space of the nullspace $\overline{\mathbf{V}}_0$ for $\overline{\mathbf{S}}$. To conclude, all of the inverses involved in this process don't break down at low frequencies and the solution should be valid at low frequencies.

For $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_h$, we can get them analytically instead of solving the generalized eigenvalue problem. The nullspace of the stiffness matrix $\overline{\mathbf{S}}$, $\overline{\mathbf{V}}_0$, is

- if the edge base goes into the node, $\frac{1}{l_i}$ appears on the row corresponding to the global indices of this edge.

- if the edge bases goes out of the node, $-\frac{1}{l_i}$ appears on the row corresponding to the global indices of this edge.

The generation of $\overline{\mathbf{V}}_h$ is also very simple. From the graph theory, the vectors corresponding to the cycles should be orthogonal to the space with vectors corresponding to bonds, that is $\overline{\mathbf{V}}_0$. Therefore, we can generate the vectors of $\overline{\mathbf{V}}_h$ by finding the independent cycle bases in the mesh. For each $\overline{\mathbf{V}}_h$, l_i appears on the row corresponding to the global indices of this edge around this cycle.

For $\overline{\mathbf{R}}_0$'s nullspace $\overline{\mathbf{X}}_0$ and its orthogonal space $\overline{\mathbf{X}}_h$, there are also analytical rules to follow. For $\overline{\mathbf{X}}_0$, the rule is to firstly find out the set of nodes outside the conductors, n_1, and the other set inside the conductors, n_2, with nodes in each conductor considered as a group. For each node outside the conductor, there is one $\overline{\mathbf{X}}_0$ mode, whose entry is 1 at the row corresponding to the node, and zero elsewhere. For each disconnected conductor, there is one $\overline{\mathbf{X}}_0$ mode, whose entry is 1 at the rows corresponding to all of the nodes inside the conductor. The $\overline{\mathbf{X}}_h$ can also be generated easily based on the simple structure of $\overline{\mathbf{X}}_0$. For each disconnected conductor whose node number is nc_i, there are $nc_i - 1$ $\overline{\mathbf{X}}_h$ modes, in each of which there are two nonzero entries: 1 and -1. The entry of 1 appears at one node inside the conductor, and -1 appears at another node inside the conductor. By this way, we can make sure $\overline{\mathbf{X}}_0^T\overline{\mathbf{X}}_h = 0$.

The generation of $\overline{\mathbf{V}}_0$, $\overline{\mathbf{V}}_h$, $\overline{\mathbf{X}}_0$ and $\overline{\mathbf{X}}_h$ is shown in Fig. 3.1, where the colored part is the conductor. For node 1, the nullspace vector has non-zero entries only on the first, second and third entries, with values $-\frac{1}{l_1}$, $-\frac{1}{l_2}$, $-\frac{1}{l_3}$, respectively. For $\overline{\mathbf{V}}_h$ corresponding to the triangle on the bottom, it has non-zero entries on the first, second and fourth entries, with values l_1, $-l_2$ and l_4. The generation of $\overline{\mathbf{X}}_0$ and $\overline{\mathbf{X}}_h$ is also shown in this figure. The node 5 doesn't need to be considered if we consider node 5 as the ground and it is removed in the $\overline{\mathbf{V}}_0$, so the row size of the vectors $\overline{\mathbf{X}}_0$ and $\overline{\mathbf{X}}_h$ is 4. For node 2,3,4, they are located in a conductor, and its corresponding $\overline{\mathbf{X}}_0$ is shown in the figure. While, node 1 is outside the conductor and its corresponding $\overline{\mathbf{X}}_0$ only has one non-zero entry on the first element. It also shows how to generated their $\overline{\mathbf{X}}_h$ according to the $\overline{\mathbf{X}}_0$ we got. This figure shows how the rules are applied to generate the spaces we want.

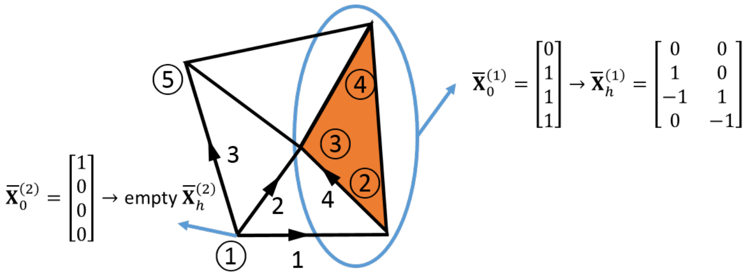

$\overline{\mathbf{X}}_0^{(1)} = \begin{bmatrix}0\\1\\1\\1\end{bmatrix} \rightarrow \overline{\mathbf{X}}_h^{(1)} = \begin{bmatrix}0 & 0\\1 & 0\\-1 & 1\\0 & -1\end{bmatrix}$
$\overline{\mathbf{X}}_0^{(2)} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \rightarrow$ empty $\overline{\mathbf{X}}_h^{(2)}$
5
4
3
2
1
3
2
4
1

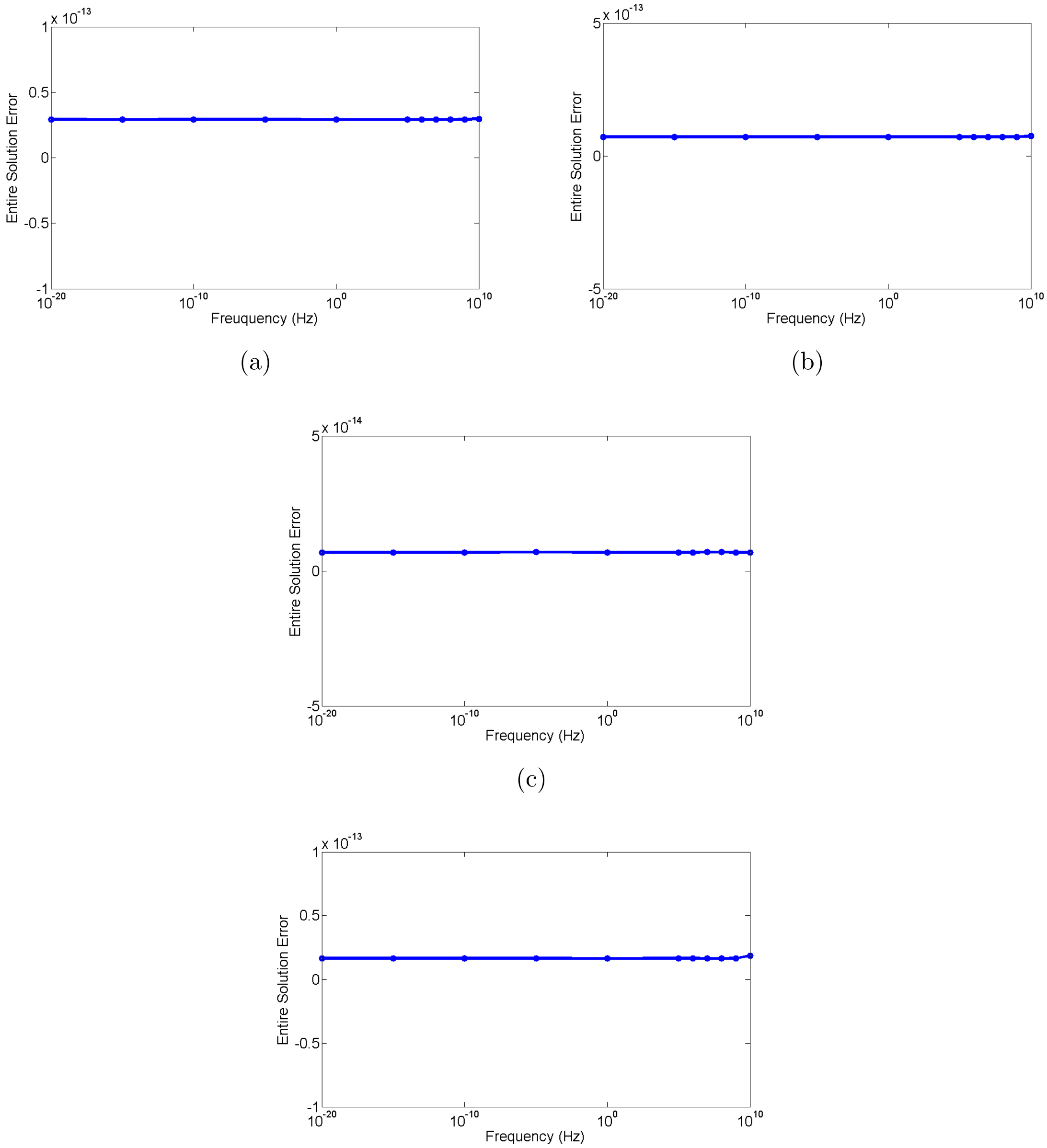
Fig. 3.1.: The generation of $\overline{\mathbf{V}}_0$, $\overline{\mathbf{V}}_h$, $\overline{\mathbf{X}}_0$ and $\overline{\mathbf{X}}_h$.

Practically speaking, as for the problems with the Dirichlet boundary condition, the idea is similar that we can regard all the nodes on the boundary as one single node. Except for this operation, other generation procedure is the same as that with no Dirichlet boundaries.

3.3 Numerical result

The nullspace can be used to solve low-frequency breakdown problem of full-wave solvers [28]. We test this on two types of meshes, the triangular mesh and the tetrahedral mesh, shown in Fig. 2.7a and 2.8a, respectively. The solution error is calculated as $\frac{\|\mathbf{x}-\mathbf{x}_r\|}{\|\mathbf{x}_r\|}$, where $\mathbf{x}$ is the solution from our method and $\mathbf{x}_r$ is the reference solution generated based on [28]. The results are shown in Fig. 3.2, where Fig. 3.2a and 3.2c are the results from the meshes without the PEC boundary at the top and the bottom, while Fig. 3.2b and 3.2d are the results from the meshes with the PEC boundary. They clearly validate the proposed method.

For lossy problems, the nullspace can also be used to solve low-frequency breakdown problem as shown in [36] [29]. However, in [36] [29], the nullspace is found numerically, whereas in this work, we find it analytically. We apply our method to a lossy 3-D on-chip interconnect structure, which is shown in Fig. 3.3. The colored part is metal, and the metal conductivity is 5.8×10^7 S/m. The solution error is shown in Fig. 3.4, which is calculated as $\frac{\|\mathbf{x}-\mathbf{x}_r\|}{\|\mathbf{x}_r\|}$, where $\mathbf{x}$ is the solution from the proposed method and $\mathbf{x}_r$ is the solution from [29]. As can be seen, the proposed method is accurate from very low to high frequencies, which verifies the correctness of this method.



(a)

(b)

(c)

(d)

Fig. 3.2.: (a) The entire solution error for a lossless problem for the triangular mesh shown in Fig. 2.7a; (b) The entire solution error for a lossless problem for the triangular mesh with PEC in Fig. 2.7a; (c) The entire solution error for a lossless problem for the tetrahedral mesh shown in Fig. 2.8a; (d) The entire solution error for a lossless problem for the tetrahedral mesh with PEC in Fig. 2.8a.

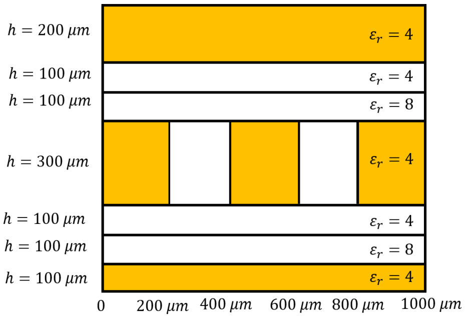


Fig. 3.3.: Illustration of an on-chip 3-D interconnect

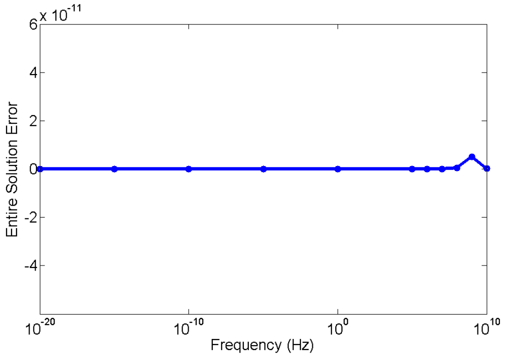


Fig. 3.4.: The entire solution error.

4. FAST FDTD METHOD FOR LARGE-SCALE LAYOUT EXTRACTION AND ANALYSIS OF INTEGRATED CIRCUITS

4.1 Introduction

Accurate and rapid layout extraction and simulation of an integrated circuit (IC) is of critical importance to the design of ICs. The problem is challenging because the numerical system resulting from the discretization of an IC is of an ultra large scale. Meanwhile, the on-chip materials are inhomogeneous, and the conductors are lossy. Furthermore, a broad band of frequencies from DC to microwave frequencies are encountered. Most of the on-chip structures are Manhatton type, and hence an FDTD method is natural for use. However, using a traditional FDTD method to simulate an IC layout, the time step restricted by the space step is orders of magnitude smaller than that determined by accuracy. As a result, it is not feasible to perform a full-chip layout extraction and analysis in feasible run time. In this work, we propose a fast FDTD method to perform layout analysis. The time step of this method is not restricted by the space step. One can choose the time step solely based on accuracy. Moreover, the computational cost at each time step scales linearly with the number of unknowns. Hence, the overall computation is made efficient. Numerical results have validated the proposed new method.

4.2 Proposed method

A full-wave FDTD-based discretizaion of Maxwell's equations results in the following matrix equation in time domain:

$$\overline{\mathbf{D}}_{\epsilon}\frac{\partial^2\{\mathbf{e}\}}{\partial t^2}+\overline{\mathbf{D}}_{\sigma}\frac{\partial\{\mathbf{e}\}}{\partial t}+\overline{\mathbf{S}}\{\mathbf{e}\}=-\frac{\partial \mathbf{J}}{\partial t} \tag{4.1}$$

where $\overline{\mathbf{D}}_\epsilon$ is the diagonal matrix of permittivity, $\overline{\mathbf{D}}_\sigma$ is the diagonal matrix of conductivity, $\overline{\mathbf{S}}$ is the matrix denoting the curl-curl operator, $\{\mathbf{e}\}$ is the vector containing all electric field unknowns in the computational domain, and $\mathbf{J}$ denotes a vector of current source. The solution $\{\mathbf{e}\}$ can be decomposed into $\mathbf{e} = \overline{\mathbf{V}}_0\mathbf{u}_0 + \overline{\mathbf{V}}_h\mathbf{u}_h$, where $\overline{\mathbf{V}}_0$ is the nullspace of $\overline{\mathbf{S}}$, and $\overline{\mathbf{V}}_h$ is its complementary space. As shown in [41], $\overline{\mathbf{V}}_0$ can be found analytically, which is from Chap. 2. In an FDTD grid, there are $(nd-1)$ $\overline{\mathbf{V}}_0$ modes, in which nd denotes the node number. Each $\overline{\mathbf{V}}_0$ mode has m entries, where m is the number of edges connected to a node, and hence being 4 in a 2-D grid, and 6 in a 3-D grid. Along an edge connected to a node, if the direction of electric field unknown is into the node, $1/l_i$ appears at the corresponding row entry of $\overline{\mathbf{V}}_0$, otherwise, $-1/l_i$ appears, where l_i is the edge length.

For the $\overline{\mathbf{V}}_0$-component of the field solution, from (4.1), we obtain

$$\overline{\mathbf{A}}_\epsilon \frac{\partial\{\mathbf{u}_0\}}{\partial t} + \overline{\mathbf{A}}_\sigma\{\mathbf{u}_0\} = -\overline{\mathbf{V}}_0^T\mathbf{J}, \tag{4.2}$$

where $\overline{\mathbf{A}}_\epsilon = \overline{\mathbf{V}}_0^T\overline{\mathbf{D}}_\epsilon\overline{\mathbf{V}}_0$, and $\overline{\mathbf{A}}_\sigma = \overline{\mathbf{V}}_0^T\overline{\mathbf{D}}_\sigma\overline{\mathbf{V}}_0$. It is evident that the $\overline{\mathbf{V}}_0$-component of the field solution captures the RC effects of an IC.

A brute-force explicit simulation of (4.2) would require a time step as small as 10^{-18} s in on-chip circuits whose physical dimensions are at the μm level. In order to make the explicit time marching unconditionally stable, we further decompose $\mathbf{u}_0$ into the following two components

$$\mathbf{u}_0 = \overline{\mathbf{X}}_0\mathbf{x}_0 + \overline{\mathbf{X}}_h\mathbf{x}_h \tag{4.3}$$

where $\overline{\mathbf{X}}_0$ is the nullspace of $\overline{\mathbf{A}}_\sigma$ and $\overline{\mathbf{X}}_h$ is $\overline{\mathbf{X}}_0$'s orthogonal space. These two spaces again can be obtained analytically. The rule to generate $\overline{\mathbf{X}}_0$ is to firstly find out the set of nodes outside the conductors, and the other set inside the conductors, with nodes in each conductor considered as a group. For each node outside the conductors, there is one $\overline{\mathbf{X}}_0$ mode, whose entry is 1 at the row corresponding to the node, and zero elsewhere. For each disconnected conductor, there is one $\overline{\mathbf{X}}_0$ mode, whose entry is 1 at the rows corresponding to all of the nodes inside the conductor. Hence, the number of $\overline{\mathbf{X}}_0$ modes is the node number outside conductors plus the conductor number minus 1. The $\overline{\mathbf{X}}_h$ can also be generated easily based on the simple structure of $\overline{\mathbf{X}}_0$. For each disconnected conductor whose node number is nc_i, there are $nc_i - 1$ $\overline{\mathbf{X}}_h$ modes, in each of which there are two nonzero entries: 1 and -1. The entry of 1 appears at one node inside the conductor, and -1 appears at another node inside the conductor. In this way, $\overline{\mathbf{X}}_h$ is clearly orthogonal to $\overline{\mathbf{X}}_0$. From the aforementioned description of $\overline{\mathbf{X}}_0$ and $\overline{\mathbf{X}}_h$, it is evident that $\overline{\mathbf{X}}_0\mathbf{x}_0$ denotes the fields outside the conductors with each conductor represented by a single mode; and $\overline{\mathbf{X}}_h\mathbf{x}_h$ represents the fields inside the conductors.

Multiplying $\overline{\mathbf{X}}_0^T\overline{\mathbf{A}}_\epsilon^{-1}$ and $\overline{\mathbf{X}}_h^T\overline{\mathbf{A}}_\epsilon^{-1}$ to both sides of (4.2), we obtain

$$\begin{aligned} \overline{\mathbf{X}}_h^T\overline{\mathbf{X}}_h\frac{\partial \mathbf{u}_h}{\partial t} + \overline{\mathbf{X}}_h^T\overline{\mathbf{A}}_\epsilon^{-1}\overline{\mathbf{A}}_\sigma\overline{\mathbf{X}}_h\mathbf{u}_h &= -\overline{\mathbf{X}}_h^T\overline{\mathbf{A}}_\epsilon^{-1}\overline{\mathbf{V}}_0^T\mathbf{J} \\ \overline{\mathbf{X}}_0^T\overline{\mathbf{X}}_0\frac{\partial \mathbf{u}_0}{\partial t} + \overline{\mathbf{X}}_0^T\overline{\mathbf{A}}_\epsilon^{-1}\overline{\mathbf{A}}_\sigma\overline{\mathbf{X}}_h\mathbf{u}_h &= -\overline{\mathbf{X}}_0^T\overline{\mathbf{A}}_\epsilon^{-1}\overline{\mathbf{V}}_0^T\mathbf{J} \end{aligned} \tag{4.4}$$

In this way, the first equation in (4.4) used for solving u_h is totally decoupled from u_0's equation. Moreover, since (4.4) is an equation inside conductors, and the conduction current is much larger than displacement current, the first term in (4.4) can be ignored without affecting accuracy. Therefore, (4.4) becomes

$$\overline{\mathbf{X}}_h^T\overline{\mathbf{A}}_\epsilon^{-1}\overline{\mathbf{A}}_\sigma\overline{\mathbf{X}}_h\mathbf{u}_h = -\overline{\mathbf{X}}_h^T\overline{\mathbf{A}}_\epsilon^{-1}\overline{\mathbf{V}}_0^T\mathbf{J}. \tag{4.5}$$

We only need to calculate the above once because $\mathbf{u}_h$'s time dependence is the same as that of $\mathbf{J}$. In addition, (4.5) in one conductor is fully decoupled from that in another conductor. Hence, the maximum number of unknowns to be solved in (4.5) is the number of nodes inside the largest conductor present in the layout. For example, if there are one million wires in an on-chip layout, the $\mathbf{u}_h$ subsystem of each wire is separated from that of the others. Furthermore, the lefthand matrix has a constant condition number which is also less than 10, its conjugate gradient solution can be converged in a less than 10 steps of iteration.

After solving $\mathbf{u}_h$ from (4.5), it can be substituted into the second equation in (4.4). The time step of the second equation can be chosen arbitrarily large without affecting stability, because it is simply the integration of the current source and the $\mathbf{u}_h$ component over the time. Thus, $\mathbf{u}_0$ can be obtained as

$$\begin{aligned}\mathbf{u}_0(t) = &-\overline{\mathbf{D}}\overline{\mathbf{X}}_0^T\overline{\mathbf{A}}_\epsilon^{-1}\overline{\mathbf{V}}_0^T\int_0^t \mathbf{J}(t)dt \\ &-\overline{\mathbf{D}}\overline{\mathbf{X}}_0^T\overline{\mathbf{A}}_\epsilon^{-1}\overline{\mathbf{A}}_\sigma\mathbf{X}_h\int_0^t \mathbf{u}_h(t)dt\end{aligned} \tag{4.6}$$

where $\overline{\mathbf{D}} = (\overline{\mathbf{X}}_0^T\overline{\mathbf{X}}_0)^{-1}$ is diagonal because of the structure of $\overline{\mathbf{X}}_0$. From (4.6), we can see that we need to solve $\overline{\mathbf{A}}_\epsilon$, which is $(\overline{\mathbf{V}}_0^T\overline{\mathbf{D}}_\epsilon\overline{\mathbf{V}}_0)x = b$. This is a very sparse matrix, whose solution can be quickly found in a few steps of sparse matrix-vector multiplications if we use $(\overline{\mathbf{V}}_0^T\overline{\mathbf{V}}_0)^{-\frac{1}{2}}$ as the preconditioner or multigrid iterative method. Meanwhile, $(\overline{\mathbf{V}}_0^T\overline{\mathbf{V}}_0)$ is nothing but a Laplacian operator in a single material and in a Cartisian grid. Its solution is analytically known. As for the $\overline{\mathbf{V}}_h$-component of the field solution in the layout analysis, fast techniques of a similar performance can also be developed.

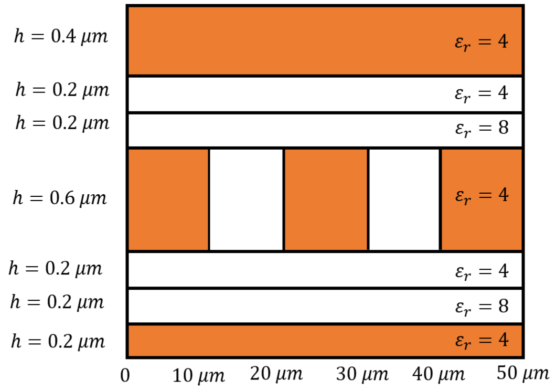


Fig. 4.1.: Illustration of an on-chip 3-D interconnect.

4.3 Numerical Results

To validate the proposed fast method, a 3-D on-chip interconnect shown in Fig. 4.1 is simulated. The colored parts are metals, whose conductivity is 5.7×10^7 S/m. The length of the structure is $50\mu m$. The current source is launched between the

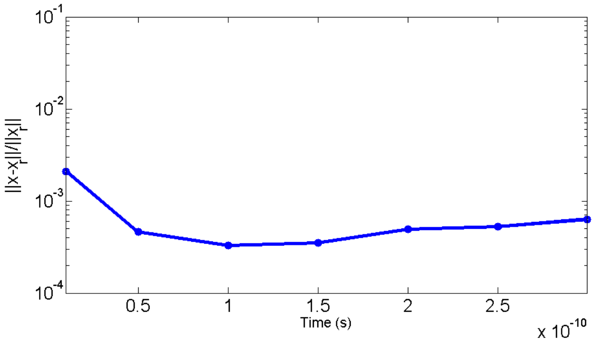


Fig. 4.2.: Solution error compared with the reference solution.

ground plane and the metal in the middle, and it is a Gaussian source with $\tau = 10^{-10}$ s. The error of the proposed fast solution is shown in Fig. 4.2, as compared with the reference solution from the traditional FDTD marching of (4.1). The error is measured by $\frac{\|\mathbf{x}-\mathbf{x}_r\|}{\|\mathbf{x}_r\|}$, where $\mathbf{x}$ is the entire field solution of the fast method; while $\mathbf{x}_r$ is the reference FDTD solution. As can be seen, the error is small across the entire time window simulated, validating the accuracy of the proposed fast method. We also compare the CPU time between the new method with the reference method for different sizes of the mesh, which is shown in Table. 4.1. Owing to the small space step, the time step for the conventional FDTD is 10^{-15} s, which cannot be increased due to the stability issue. In contrast, the proposed method allows for the use of an arbitrarily large time step. Hence, the choice of time step is solely determined by accuracy. Using a time step of 10^{-12} s, we compare the run time of the new method and the FDTD in Table 4.1. It is evident that the proposed method is much more efficient even for a small number of unknowns.

Table 4.1.: Comparison of CPU run time.

Unknown Size (E Number)	Simulation time by this method (s)	Simulation time by FDTD (s)
224	1.98	30.522309
424	4.46	360.81
1020	8.60	2197.67

5. RAPID INVERSE MODELING OF INTEGRATED CIRCUIT LAYOUT IN BOTH FREQUENCY AND TIME DOMAIN

5.1 Introduction

Accurate and large-scale layout models are of critical importance to the design of integrated circuits (IC), packages, and boards. There are two types of existing methods. The first one is to employ static or quasi-static field solvers to extract C-, R-, and L-based circuit models of the physical layout of an IC, package, or board. R, C, and L models are independently extracted. These models are then stitched together to perform a circuit simulation. The shortcomings of this method are that there are many ways to stitch the C models with the RL-models. Which one correctly captures the physics such as distributed effects and 3-D effects in the physical layout is unknown even at relatively low working frequencies, where static physics is dominant. As another example, the substrate is considered to be one of the main sources for interference and crosstalk. However, the global effect of the substrate is not well captured in existing layout models. And this method is inaccurate at high frequencies. The second type of method is full-wave layout modeling. This method performs a full-wave analysis, such as using Finite Difference Time Domain Method, Finite Element Method or Integral Equation Method and then the network parameters, such as S-, Z-, and Y-parameters are obtained. Although this type of method is accurate at high frequencies, it has a limited capability. It is difficult to handle full-chip and complete package within feasible computational resources. In addition, the direct field-based representation of the layout and the resulting field solution remain too abstract to be put into practical use by circuit designers for circuit diagnosis and performance optimization.

One common feature in existing layout modeling and simulations tools [2,9,18] is that they focus on a forward model of the layout in a SPICE-compatible format. Even though such a model is built efficiently in high accuracy, the actual layout response to circuit stimuli is unknown until a circuit simulation is performed on the extracted layout model. In other words, what has been modeled is the layout instead of layout response, i.e., the inverse of the system of equations governing the layout. There is no direct relationship between the layout response and the change of the layout model. Such a modeling approach is not amenable for circuit design, because changing the C, R, L or other circuit parameters in the original layout does not directly reveal how the circuit performance will be changed.

In this work, different from prevailing approaches where the forward model of the layout is constructed, we analytically derive a closed-form model of the inverse of Maxwell's system of equations in the physical layout. We also derive such an inverse from *full-wave* Maxwell's equations where **E** and **H** are coupled, so that we bypass the inaccuracy issue arising from stitching circuit models independently extracted from decoupled **E** and **H** equations. Moreover, the model is correct from zero to high frequencies where Maxwell's equations are valid. More importantly, in the proposed inverse model, we are able to analytically decompose the layout response into R-, C-, L-, and full-wave components with neither numerical computation nor approximation. As a result, each component can be obtained in parallel and then summed up to obtain a total layout response. This decomposed yet rigorous model greatly helps circuit diagnoses since now designers are able to analyze each component one by one, and identify which component is the root cause for the design failure. Such a decomposition also facilitates efficient layout modeling and simulation, since if an IC is dominated by RC effects, then we do not have to compute the full-wave component; and vice versa. Meanwhile, it makes parallelization straightforward. In addition, we develop fast algorithms to obtain each component of the inverse rapidly. The time marching and point-by-point frequency sweeping are also avoided for the R- and C-component of the layout response as their time and frequency dependency is analytically known in the proposed inverse model. The proposed work has been applied to large-scale layout extraction and analysis. Its performance in accuracy, efficiency, and capacity has been demonstrated.

In this section, we present a detailed derivation of the inverse of the Maxwell's system of equations in the layout, and show how it can be decomposed into R-, C-, L-, and full-wave components. The analytical methods for finding the nullspace governing the RC-component are presented in both uniform and non-uniform grids. A fast algorithm for finding the full-wave components is developed. Many new and realistic examples are simulated to examine the performance of the proposed work.

5.2 Proposed Closed-Form Model of the Inverse in Arbitrary Layouts and Its Decomposition into R-, C-, L- and Full-wave Components

Consider an arbitrary layout of analog and mixed-signal ICs, packages, and boards, which consists of interconnects, RF/analog components, substrates, materials, etc. The physical phenomena in such a layout from DC to high frequencies are governed by Maxwell's Equations as the following

$$\nabla \times [\mu_r^{-1} \nabla \times \mathbf{E}(\mathbf{r}, t)] + \mu_0 \varepsilon \partial_t^2 \mathbf{E}(\mathbf{r}, t) + \mu_0 \sigma \partial_t \mathbf{E}(\mathbf{r}, t) = -\mu_0 \partial_t \mathbf{J}(\mathbf{r}, t) \text{ in } V, \tag{5.1}$$

where $\mathbf{E}$ is electric field, μ_0 is free-space permeability, μ_r is relative permeability, ε is permittivity, σ is conductivity, $\mathbf{J}$ is current density, and $\mathbf{r}$ denotes a point in a 3-D space.

We discretize the entire physical layout into a grid to capture the geometry and inhomogeneous materials. A Cartesian grid is used instead of irregular meshes because it is natural for discretizing a majority of the layout structures, and also it removes the step of 3-D meshing which can be computationally expensive. Let e be a vector consisting of the tangential electric field along each edge of the grid, the length of which is N_e. We employ a recently developed patch-based FDTD formulation to discretize Maxwell's equations in such a grid [42], which yields the following linear system of equations:

$$diag\{\epsilon\}\frac{d^2\mathbf{e}}{dt^2} + diag\{\sigma\}\frac{d\mathbf{e}}{dt} + \overline{\mathbf{S}}\mathbf{e} = -\frac{d\mathbf{J}}{dt}, \tag{5.2}$$

in which $\mathbf{J}$ is a vector of current density, $diag\{\epsilon\}$ and $diag\{\sigma\}$ are diagonal matrices of permittivity, and conductivity respectively, whose i-th entry represents the permittivity and conductivity at the center of the i-th edge, where the i-th $\mathbf{E}$-unknown is located.

The $\overline{\mathbf{S}}$ in (6.1) is a sparse matrix, and $\overline{\mathbf{S}}\mathbf{e}$ represents a discretized $\nabla \times [\mu^{-1}\nabla \times \mathbf{E}]$ operation. To generate $\overline{\mathbf{S}}$, we only need to loop over all the patches in the grid, regardless of whether the grid is a 2-D grid or a 3-D grid. For each patch, we generate one row vector $\overline{\mathbf{S}}_e^{(i)}$ shown as the following

$$\overline{\mathbf{S}}_e^{(i)} = \begin{bmatrix} -\frac{1}{L_i} & \frac{1}{L_i} & \frac{1}{W_i} & -\frac{1}{W_i} \end{bmatrix} \oplus zeros(1, N_e), \tag{5.3}$$

where $\oplus$ denotes an extended addition based on the global indexes of the four local $\mathbf{E}$ unknowns of patch i, and L_i and W_i are, respectively, the two side lengths of patch i. We also generate one column vector $\overline{\mathbf{S}}_h^{(i)}$ in each patch. In a uniform grid, the following is true

$$\overline{\mathbf{S}}_h^{(i)} = \left(\overline{\mathbf{S}}_e^{(i)}\right)^T. \quad \text{(Uniform grid)} \tag{5.4}$$

In a non-uniform grid, for better accuracy, we should replace the length in $\overline{\mathbf{S}}_h^{(i)}$ by an average length across the two patches sharing the $\mathbf{E}$ edge. Hence,

$$\overline{\mathbf{S}}_h^{(i)} = \begin{bmatrix} -\frac{1}{L_{j1}^{ave}} & \frac{1}{L_{j2}^{ave}} & \frac{1}{L_{j3}^{ave}} & -\frac{1}{L_{j4}^{ave}} \end{bmatrix}^T \oplus zeros(N_e, 1), \quad \text{(Non-uniform grid)} \tag{5.5}$$

where the subscripts $j1, ..., j4$ denote the four edge indexes in the i-th patch, and the superscript ave denotes an average length.

Multiplying $\overline{\mathbf{S}}_h^{(i)}$ by $\overline{\mathbf{S}}_e^{(i)}$ and adding the resultant rank-1 matrix of each patch, we obtain a global $\overline{\mathbf{S}}$ as the following

$$\overline{\mathbf{S}} = \sum_{i=1}^{N_h} {\mu_i}^{-1}\left(\overline{\mathbf{S}}_h^{(i)}\right)\left(\overline{\mathbf{S}}_e^{(i)}\right), \tag{5.6}$$

where N_h is the patch number, which is also the number of magnetic field unknowns. The above can also be rewritten as

$$\overline{\mathbf{S}} = \overline{\mathbf{S}}_h \overline{\mathbf{D}}_{1/\mu} \overline{\mathbf{S}}_e, \tag{5.7}$$

in which $\overline{\mathbf{S}}_h$'s i-th column is $\overline{\mathbf{S}}_h^{(i)}$, whereas $\overline{\mathbf{S}}_e$'s i-th row is $\overline{\mathbf{S}}_e^{(i)}$, and $\overline{\mathbf{D}}_{1/\mu}$ is a diagonal matrix of μ^{-1}.

Eqn. (6.1) has an obvious frequency-domain counterpart as the following

$$-\omega^2 diag\{\epsilon\}\mathbf{e}(\omega) + j\omega diag\{\sigma\}\mathbf{e}(\omega) + \overline{\mathbf{S}}\mathbf{e}(\omega) = -j\omega\mathbf{J}(\omega), \tag{5.8}$$

which we denote in short by

$$\overline{\mathbf{Y}}(\omega)\mathbf{e}(\omega) = \overline{\mathbf{I}}(\omega), \tag{5.9}$$

where

$$\overline{\mathbf{Y}}(\omega) = \overline{\mathbf{D}} + \overline{\mathbf{S}}, \tag{5.10}$$

and

$$\overline{\mathbf{D}} = -\omega^2 diag\{\epsilon\} + j\omega diag\{\sigma\} \tag{5.11}$$

is diagonal. Next, we derive a closed-form model of the inverse of $\overline{\mathbf{Y}}$.

$\overline{\mathbf{S}}$ has a nullspace, which is evident from (6.4) as $\overline{\mathbf{S}}_h$'s column number N_h is less than row number N_e. Since $\overline{\mathbf{S}}$ represents a discretized $\nabla \times \mu^{-1}\nabla\times$ operation, the nullspace represents a gradient field in the grid. Let it be $\overline{\mathbf{V}}_0$, which is from Chap. 2. It satisfies

$$\overline{\mathbf{S}}\overline{\mathbf{V}}_0 = 0. \tag{5.12}$$

Let $\overline{\mathbf{V}}_h$ be its complementary space. The number of vectors in $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_h$ is equal to the matrix size of $\overline{\mathbf{S}}$. Therefore, the solution of (5.8) can be rigorously expanded in the space of $\left[\overline{\mathbf{V}}_0 \;\; \overline{\mathbf{D}}^{-1}\overline{\mathbf{V}}_h\right]$ as the following

$$\mathbf{e} = \underbrace{\overline{\mathbf{V}}_0\mathbf{y}_0}_{\text{RC-effects}} + \underbrace{\overline{\mathbf{D}}^{-1}\overline{\mathbf{V}}_h\mathbf{y}_h}_{\text{RL and full-wave effects}} . \tag{5.13}$$

Here, we use $\left[\overline{\mathbf{V}}_0 \;\; \overline{\mathbf{D}}^{-1}\overline{\mathbf{V}}_h\right]$ instead of $\left[\overline{\mathbf{V}}_0 \;\; \overline{\mathbf{V}}_h\right]$ because in this way, after testing (5.8) by $\overline{\mathbf{V}}_0^T$ and $\left(\overline{\mathbf{D}}^{-1}\overline{\mathbf{V}}_h\right)^T$, we can decouple the solution of $\mathbf{y}_0$ from that of $\mathbf{y}_h$, which will become clear in the sequel. Otherwise, the two are coupled, and it becomes difficult to develop an explicit inverse model. In addition, since $\overline{\mathbf{D}}$ is diagonal, computing $\overline{\mathbf{D}}^{-1}$ is trivial. The $\overline{\mathbf{V}}_0$-component of $\mathbf{e}$ has a zero curl, hence it represents the RC-component of the layout response, as noted in (5.13). In contrast, the $\overline{\mathbf{V}}_h$-component characterizes the inductance and full-wave effects.

The $\overline{\mathbf{V}}_0$ can be found by solving the eigenvectors of $\overline{\mathbf{S}}$ corresponding to zero eigenvalues, however, this is computationally expensive. In this work, we develop an analytical method for finding $\overline{\mathbf{V}}_0$ solely based on the mesh information, thus removing the cost of numerically computing $\overline{\mathbf{V}}_0$. The details are given in Chap. 2 and also in the following Section 5.3. As far as the $\overline{\mathbf{V}}_h$ is concerned, if we use the eigenvectors of $\overline{\mathbf{S}}$ corresponding to the nonzero eigenvalues, we can also find an analytical way to obtain them. However, the number of such $\overline{\mathbf{V}}_h$ can be too many to use. For example, a static field distribution can also be decomposed into many of such $\overline{\mathbf{V}}_h$ modes since the static field distribution can also have a rapid space variation. We find if using the eigenvectors of a quadratic eigenvalue problem governing (6.1), the number of $\overline{\mathbf{V}}_h$ required is very small. Nevertheless, solving the quadratic eigenvalue problem can also be computationally expensive. In this work, we develop an efficient solution for finding $\overline{\mathbf{V}}_h$, the details of which are given in Section 5.4.

Now, assuming the $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_h$ have been obtained, we show how to derive a closed form model of the inverse. To find the solution of $\mathbf{y}_0$, we can substitute (5.13) into (5.8), and multiply (5.8) from left by $\overline{\mathbf{V}}_0^T$, obtaining

$$\overline{\mathbf{V}}_0^T(\overline{\mathbf{D}}+\overline{\mathbf{S}})(\overline{\mathbf{V}}_0\mathbf{y}_0+\overline{\mathbf{D}}^{-1}\overline{\mathbf{V}}_h\mathbf{y}_h) = -j\omega\overline{\mathbf{V}}_0^T\mathbf{J}. \tag{5.14}$$

If $\overline{\mathbf{V}}_0^T\overline{\mathbf{S}} = 0$, then the above can be readily simplified to

$$\overline{\mathbf{V}}_0^T\overline{\mathbf{D}}\overline{\mathbf{V}}_0\mathbf{y}_0 = -j\omega\overline{\mathbf{V}}_0^T\mathbf{J}. \tag{5.15}$$

However, we find the above is only true in a uniform grid. This is because in a uniform grid, $\overline{\mathbf{S}}$ is symmetric, hence from $\overline{\mathbf{S}}\overline{\mathbf{V}}_0 = 0$, taking a transpose we obtain $\overline{\mathbf{V}}_0^T\overline{\mathbf{S}} = 0$. This is not the case in a non-uniform grid, as can be seen from (5.5). A non-uniform grid is unavoidable in discretizing a physical layout. If using a uniform grid, then the number of discretization cells could be too many to be computed efficiently due to the presence of various fine features in the layout. This problem is solved in this work by finding a *left* nullspace of $\overline{\mathbf{S}}$, and also analytically. This left nullspace is denoted by $\overline{\mathbf{V}}_{0a}$, which satisfies

$$\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{S}} = 0, \tag{5.16}$$

while preserving the property of

$$\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{V}}_h = 0. \tag{5.17}$$

The analytical approach for finding $\overline{\mathbf{V}}_{0a}$ is detailed in Section 5.3.

Multiplying $\overline{\mathbf{V}}_{0a}^T$ to both sides of (5.8) and utilizing (5.16) and (5.17), we obtain

$$\overline{\mathbf{V}}_{0a}^T(-\omega^2 diag\{\epsilon\} + j\omega diag\{\sigma\})\overline{\mathbf{V}}_0\mathbf{y}_0 = -j\omega\overline{\mathbf{V}}_{0a}^T\mathbf{J}, \tag{5.18}$$

which is a system of equations for $\mathbf{y}_0$ only, and hence we can solve the above without concerning about the $\mathbf{y}_h$ component. We further decompose the solution of the above into

$$\overline{\mathbf{V}}_0\mathbf{y}_0 = \overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0c}\mathbf{y}_{0c}, \tag{5.19}$$

where $\overline{\mathbf{V}}_{0d}$ is in the nullspace of $diag\{\sigma\}$, denoting the field outside the conductors, and $\overline{\mathbf{V}}_{0c}$ is $\overline{\mathbf{V}}_{0d}$'s complementary space in $\overline{\mathbf{V}}_0$. These two column spaces can again be analytically obtained without any computation, the details of which are given in Section 5.3. The left nullspace $\overline{\mathbf{V}}_{0a}$ can also be decomposed into $\overline{\mathbf{V}}_{0da}$ and $\overline{\mathbf{V}}_{0ca}$, in the same way as $\overline{\mathbf{V}}_0$ is decomposed into $\overline{\mathbf{V}}_{0d}$ and $\overline{\mathbf{V}}_{0c}$. Substituting (5.19) into (5.18), the resulting rows of equations corresponding to $\overline{\mathbf{V}}_{0da}$ can be written as

$$\begin{aligned}&\overline{\mathbf{V}}_{0da}^T(-\omega^2 diag\{\epsilon\} + j\omega diag\{\sigma\})\left(\overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0c}\mathbf{y}_{0c}\right)\\ &= -j\omega\overline{\mathbf{V}}_{0da}^T\mathbf{J};\end{aligned} \tag{5.20}$$

and the rest corresponding to $\overline{\mathbf{V}}_{0ca}$ can be written as

$$\begin{aligned}&\overline{\mathbf{V}}_{0ca}^T(-\omega^2 diag\{\epsilon\} + j\omega diag\{\sigma\})\left(\overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0c}\mathbf{y}_{0c}\right)\\ &= -j\omega\overline{\mathbf{V}}_{0ca}^T\mathbf{J}.\end{aligned} \tag{5.21}$$

Since $diag\{\sigma\}\overline{\mathbf{V}}_{0d} = 0$ and $\overline{\mathbf{V}}_{0da}^T diag\{\sigma\} = 0$, (5.20) can be rewritten as

$$\begin{aligned}&\overline{\mathbf{V}}_{0da}^T(j\omega diag\{\epsilon\})\overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0da}^T(j\omega diag\{\epsilon\})\overline{\mathbf{V}}_{0c}\mathbf{y}_{0c} \\ &= -\overline{\mathbf{V}}_{0da}^T\mathbf{J}.\end{aligned} \tag{5.22}$$

Also because $diag\{\sigma\}\overline{\mathbf{V}}_{0d} = 0$, (5.21) becomes

$$\begin{aligned}&\overline{\mathbf{V}}_{0ca}^T(j\omega diag\{\epsilon\})\overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0ca}^T(j\omega diag\{\epsilon\} + diag\{\sigma\})\overline{\mathbf{V}}_{0c}\mathbf{y}_{0c} \\ &= -\overline{\mathbf{V}}_{0ca}^T\mathbf{J}.\end{aligned} \tag{5.23}$$

Because in the conductor, the displacement current is much smaller than the conduction current, (5.23) can be accurately approximated as

$$\overline{\mathbf{V}}_{0ca}^T(j\omega diag\{\epsilon\})\overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0ca}^T diag\{\sigma\}\overline{\mathbf{V}}_{0c}\mathbf{y}_{0c} = -\overline{\mathbf{V}}_{0ca}^T\mathbf{J}. \tag{5.24}$$

In order to solve $\mathbf{y}_0$, we first solve the imaginary part of $\mathbf{y}_{0d}$, Im$[\mathbf{y}_{0d}]$, from (5.22) as

$$\text{Im}[\mathbf{y}_{0d}] = \frac{(\overline{\mathbf{V}}_{0da}^T diag\{\epsilon\}\overline{\mathbf{V}}_{0d})^{-1}(\overline{\mathbf{V}}_{0da}^T\mathbf{J})}{\omega}. \tag{5.25}$$

Denote it in short by $\mathbf{y}_{0d,i}/\omega$. We then substitute Im$[\mathbf{y}_{0d}]$ into (5.24) to solve $\mathbf{y}_{0c}$, which is

$$\begin{aligned}\mathbf{y}_{0c} = (\overline{\mathbf{V}}_{0ca}^T diag\{\sigma\}\overline{\mathbf{V}}_{0c})^{-1}(-\overline{\mathbf{V}}_{0ca}^T\mathbf{J} + \\ \overline{\mathbf{V}}_{0ca}^T diag\{\epsilon\}\overline{\mathbf{V}}_{0d}\mathbf{y}_{0d,i}).\end{aligned} \tag{5.26}$$

After getting $\mathbf{y}_{0c}$, we substitute it back to (5.22) to obtain the real part of $\mathbf{y}_{0d}$, which is

$$\text{Re}[\mathbf{y}_{0d}] = -(\overline{\mathbf{V}}_{0da}^T diag\{\epsilon\}\overline{\mathbf{V}}_{0d})^{-1}(\overline{\mathbf{V}}_{0da}^T diag\{\epsilon\}\overline{\mathbf{V}}_{0c}\mathbf{y}_{0c}). \tag{5.27}$$

The above (5.25), (7.17), and (5.27) make the complete solution of $\mathbf{y}_0$ in (5.19).

Multiplying $\left(\overline{\mathbf{D}}^{-1}\overline{\mathbf{V}}_{h1}\right)^T$ to both sides of (5.8) (where $\overline{\mathbf{V}}_{h1}^T\overline{\mathbf{V}}_0 = 0$), we obtain

$$\left(\overline{\mathbf{V}}_{h1}^T\overline{\mathbf{D}}^{-1}\overline{\mathbf{YD}}^{-1}\overline{\mathbf{V}}_h\right)\mathbf{y}_h = -j\omega\overline{\mathbf{V}}_{h1}^T\overline{\mathbf{D}}^{-1}\mathbf{J}, \tag{5.28}$$

from which $\mathbf{y}_h$ can be solved.

Summarizing the results in (5.25), (7.17), (5.27), and (5.28), we obtain a final model of the inverse of $\overline{\mathbf{Y}}$ as

$$\begin{aligned}
-j\omega\overline{\mathbf{Y}}^{-1} = {} & \underbrace{\overline{\mathbf{V}}_{0d}\overline{\mathbf{D}}_{\epsilon,0}^{-1}\overline{\mathbf{V}}_{0da}^{T}}_{\text{C component}}/(-j\omega)+ \\
& \underbrace{\overline{\mathbf{V}}_{0d}\overline{\mathbf{D}}_{\epsilon,0}^{-1}\overline{\mathbf{V}}_{0da}^{T}diag\{\epsilon\}\overline{\mathbf{V}}_{0c}\overline{\mathbf{D}}_{\sigma,0}^{-1}\overline{\mathbf{V}}_{0ca}^{T}(\overline{\mathbf{I}}–diag\{\epsilon\}\overline{\mathbf{V}}_{0d}\overline{\mathbf{D}}_{\epsilon,0}^{-1}\overline{\mathbf{V}}_{0da}^{T})}_{\text{R component}} \\
& \underbrace{+\overline{\mathbf{V}}_{0c}\overline{\mathbf{D}}_{\sigma,0}^{-1}\overline{\mathbf{V}}_{0ca}^{T}(diag\{\epsilon\}\overline{\mathbf{V}}_{0d}\overline{\mathbf{D}}_{\epsilon,0}^{-1}\overline{\mathbf{V}}_{0da}^{T} - \overline{\mathbf{I}})}_{\text{R component}} \\
& -j\omega\underbrace{\overline{\mathbf{D}}^{-1}\overline{\mathbf{V}}_{h}\left(\overline{\mathbf{V}}_{h1}^{T}\overline{\mathbf{D}}^{-1}\overline{\mathbf{YD}}^{-1}\overline{\mathbf{V}}_{h}\right)^{-1}\overline{\mathbf{V}}_{h1}^{T}\overline{\mathbf{D}}^{-1}}_{\text{L and full-wave component}}, \qquad (5.29)
\end{aligned}$$

in which

$$\overline{\mathbf{D}}_{\epsilon,0} = \overline{\mathbf{V}}_{0da}^{T}diag\{\epsilon\}\overline{\mathbf{V}}_{0d} \qquad (5.30)$$

$$\overline{\mathbf{D}}_{\sigma,0} = \overline{\mathbf{V}}_{0ca}^{T}diag\{\sigma\}\overline{\mathbf{V}}_{0c}, \qquad (5.31)$$

are both frequency independent. From (5.29), we can clearly identify the C-, R-, L-, and full-wave component of the layout response. For the RC-component, their frequency, and hence time dependence, are also analytically revealed. In next sections, we present fast algorithms for computing each component.

It is also worth mentioning although (6.1) is free of matrix solution in its explicit time marching, a direct simulation of (6.1) is computationally prohibitive for large layouts because a tremendous number of time steps must be simulated due to the extremely small space step. Meanwhile, even though the simulation can be carried out, the field based solution is not circuit intuitive, and is difficult to be used to guide the design. In addition, one cannot separately obtain each component of the layout response like in the proposed method.

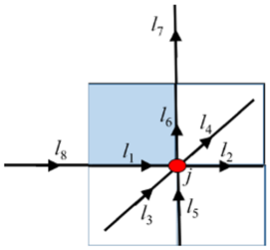


Fig. 5.1.: Illustration of the nullspace vector at a node.

5.3 Analytical Method for Finding $\overline{\mathbf{V}}_0$ and Efficient Computation of the RC-component of Layout Response

We find an analytical method to generate the $\overline{\mathbf{V}}_0$ from the mesh information, without the need for solving an eigenvalue problem of $\overline{\mathbf{S}}$. The number of $\overline{\mathbf{V}}_0$ modes is the total number of nodes in the grid, n, minus 1

$$\#\overline{\mathbf{V}}_0 = n - 1. \tag{5.32}$$

Each node has a nullspace vector, whose number of nonzero entries is the number of edges connected to the node. Such a vector can be generated in the following way:

- If the electric field reference direction along an edge enters the node, $\frac{1}{l_i}$ appears on the row corresponding to the global e index of this edge.
- If the electric field reference direction along the edge leaves the node, $-\frac{1}{l_i}$ appears on the row corresponding to the global index of this e edge,

where l_i is the length of the i-th edge, at which the i-th electric field unknown is located. In the above, we use a positive sign for edges whose directions enter the node, and a minus sign for edges leaving the node. Certainly, an opposite sign convention can also be used. An edge's direction here is referred to as the electric field reference direction defined on the edge. The aforementioned description could be abstract, take the node j shown in Fig. 5.1 as an example, its associated nullspace vector can be written as

$$\overline{\mathbf{V}}_{0,j} = \left[\frac{1}{l_1}, -\frac{1}{l_2}, \frac{1}{l_3}, -\frac{1}{l_4}, \frac{1}{l_5}, -\frac{1}{l_6}\right]^T \oplus zeros(Ne, 1), \tag{5.33}$$

in which $\oplus zeros(Ne, 1)$ again denotes adding the preceding nonzero entries associated with each edge at the rows corresponding to the global index of the edge, in a vector of length N_e. Because $\overline{\mathbf{S}}$ has a format shown in (5.6), when multiplying $\overline{\mathbf{S}}$ by the $\overline{\mathbf{V}}_{0,j}$ above, only those patches that contain the six edges associated with the node are involved in the product of $\overline{\mathbf{S}}\overline{\mathbf{V}}_{0,j}$. The number of such patches is 12 in a 3-D grid, and 4 in a 2-D grid. On each of these patches, the $\overline{\mathbf{S}}_e^{(i)}\overline{\mathbf{V}}_{0,j} = 0$, and hence $\overline{\mathbf{S}}\overline{\mathbf{V}}_{0,j} = 0$ is satisfied. To see this point clearly, take the blue patch shown in Fig. 5.1 as an example, its $\overline{\mathbf{S}}_e^{(i)}$ can be written as

$$\overline{\mathbf{S}}_e^{(i)} = \left[\frac{1}{l_6}, -\frac{1}{l_6}, \frac{1}{l_1}, -\frac{1}{l_1}\right] \oplus zeros(1, N_e). \tag{5.34}$$

When multiplying the above by $\overline{\mathbf{V}}_{0,j}$, we obtain

$$\overline{\mathbf{S}}_e^{(i)}\overline{\mathbf{V}}_{0,j} = \frac{1}{l_6 l_1} - \frac{1}{l_6 l_1} = 0. \tag{5.35}$$

The same is true for all other patches that own node j.

For a non-uniform grid, since $\overline{\mathbf{S}}$ is not symmetric anymore, the aforementioned $\overline{\mathbf{V}}_0$ satisfies $\overline{\mathbf{S}}\overline{\mathbf{V}}_0 = 0$, but it does not make $\overline{\mathbf{V}}_0^T\overline{\mathbf{S}}$ vanish. So we also find a way to analytically generate the left nullspace of $\overline{\mathbf{S}}$, $\overline{\mathbf{V}}_{0a}$. For each node, there is also one $\overline{\mathbf{V}}_{0a}$ vector. The nonzero entries in this vector are at the same entries as those in $\overline{\mathbf{V}}_0$. However, different from $\overline{\mathbf{V}}_0$, we need to use an average length to build $\overline{\mathbf{V}}_{0a}$ instead of the original length of the edge. The rule to generate a $\overline{\mathbf{V}}_{0a}$ vector is as the following:

- If the electric field reference direction along an edge enters the node, $\frac{1}{l_i^{ave}}$ appears on the row corresponding to the global e index of this edge, where l_i^{ave} denotes the averaged length of edge i and its adjacent edge connected to the node along the same direction.

- If the electric field reference direction along an edge leaves the node, $-\frac{1}{l_i^{ave}}$ appears on the row corresponding to the global e index of this edge, where l_i^{ave}

denotes the averaged length of edge i and its adjacent edge connected to the node along the same direction.

Using the node j shown in Fig. 5.1 as an example, its $\overline{\mathbf{V}}_{0a}$ vector can be written as

$$\overline{\mathbf{V}}_{0a,j} = \left[\frac{2}{l_1+l_2}, -\frac{2}{l_1+l_2}, \frac{2}{l_3+l_4}, -\frac{2}{l_3+l_4}, \right. \\ \left. \frac{2}{l_5+l_6}, -\frac{2}{l_5+l_6}\right]^T \oplus zeros(Ne, 1). \tag{5.36}$$

When computing $\overline{\mathbf{V}}^T_{0a,j}\overline{\mathbf{S}}$, $\overline{\mathbf{V}}^T_{0a,j}$ is multiplied by $\overline{\mathbf{S}}_h$. In a non-uniform grid, $\overline{\mathbf{S}}^{(i)}_h$ has a form shown in (5.5) where the average length is used. Take the blue patch shown in Fig. 5.1 as an example, its $\overline{\mathbf{S}}^{(i)}_h$ can be written as

$$\overline{\mathbf{S}}^{(i)}_h = \left[\frac{2}{l_5+l_6}, -\frac{2}{l_6+l_7}, \frac{2}{l_1+l_2}, -\frac{1}{l_1+l_8}\right]^T \oplus zeros(N_e, 1). \tag{5.37}$$

When multiplying $\overline{\mathbf{V}}^T_{0a,j}$ by the above, we obtain

$$\overline{\mathbf{V}}^T_{0a,j}\overline{\mathbf{S}}^{(i)}_h = \frac{2}{l_1+l_2}\frac{2}{l_5+l_6} - \frac{2}{l_1+l_2}\frac{2}{l_5+l_6} = 0. \tag{5.38}$$

As can be seen, the average length is used in $\overline{\mathbf{V}}_{0a,j}$ to vanish $\overline{\mathbf{V}}^T_{0a,j}\overline{\mathbf{S}}^{(i)}_h$.

In (5.19), we further decompose the $\overline{\mathbf{V}}_0$ into two sets: $\overline{\mathbf{V}}_{0d}$ and $\overline{\mathbf{V}}_{0c}$. The $\overline{\mathbf{V}}_{0d}$ is composed of all the nullspace vectors obtained at the nodes outside conductors, i.e., whose node conductivity is zero, and additional $\#c$ vectors, where $\#c$ is the number of conductors in the layout. In these $\#c$ vectors, each vector corresponds to one conductor, which is the sum of the nullspace vectors generated at the nodes inside and on the surface of the conductor. The $\overline{\mathbf{V}}_{0d}$ can be written as

$$\overline{\mathbf{V}}_{0d} = \\ \left[\underbrace{\{\overline{\mathbf{V}}_{0,i}(\sigma_i = 0)\}}_{nd-1 \text{ vectors}} \mid \underbrace{\left(\sum_{\mathbf{r}_j \in \Omega_{c,1}} \overline{\mathbf{V}}_{0,j}\right) |...| \left(\sum_{\mathbf{r}_j \in \Omega_{c,\#c}} \overline{\mathbf{V}}_{0,j}\right)}_{\#c \text{ vectors}} \right], \tag{5.39}$$

where in the first set, nd denotes the number of dielectric nodes, σ_i denotes the conductivity at node i, and $\mathbf{r}_j \in \Omega_{c,j}$ $(j = 1, 2, ...\#c)$ represents the nodes in the j-th conductor, including those falling onto the j-th conductor's surface. Each summation in (5.39) results in a vector containing all the edges exterior to the conductor, and perpendicular to the conductor surface, which have one and only one node falling onto the conductor. All the entries corresponding to the edges, thus electric field unknowns, inside the conductors are canceled in the summation because of opposite signs. From (5.39), it can be seen that $\overline{\mathbf{V}}_{0d}$ satisfies

$$diag\{\sigma\}\overline{\mathbf{V}}_{0d} = 0. \tag{5.40}$$

Each vector in $\overline{\mathbf{V}}_{0c}$ is the nullspace vector corresponding to one node inside or on the surface of the conductors. The number of $\overline{\mathbf{V}}_{0c}$ for each conductor is $nc_i - 1$, where nc_i denotes the number of nodes in the i-th conductor. Hence,

$$\overline{\mathbf{V}}_{0c,j} = \overline{\mathbf{V}}_{0,j}(\sigma_j \neq 0), \tag{5.41}$$

and for each conductor, one node is excluded for generating the above, since there is one vector considered in (5.39).

The left nullspace $\overline{\mathbf{V}}_{0a}$ can also be decomposed into $\overline{\mathbf{V}}_{0da}$ and $\overline{\mathbf{V}}_{0ca}$, in the same way as $\overline{\mathbf{V}}_0$ is decomposed into $\overline{\mathbf{V}}_{0d}$ and $\overline{\mathbf{V}}_{0c}$. The $\overline{\mathbf{V}}_{0da}$ contains all the nullspace vectors obtained at the nodes outside conductors plus $\#c$ vectors. Each vector in $\#c$ vectors corresponds to one conductor, which is the weighted sum of the $\overline{\mathbf{V}}_{0a}$ vectors from the nodes inside and on the surface of the conductor. The weights are chosen such that the summation eliminates the edges inside and on the surface of each conductor, which makes $\overline{\mathbf{V}}_{0da}$ have only non-zeros entries in the dielectric part, and thus,

$$diag\{\sigma\}\overline{\mathbf{V}}_{0da} = \overline{\mathbf{V}}_{0da}^T diag\{\sigma\} = 0. \tag{5.42}$$

$\overline{\mathbf{V}}_{0ca}$ is the subset of $\overline{\mathbf{V}}_{0a}$ generated at the node inside or on the surface of the conductors, which is similar to $\overline{\mathbf{V}}_{0c}$.

As shown in the above, $\overline{\mathbf{V}}_{0d}$, $\overline{\mathbf{V}}_{0c}$, $\overline{\mathbf{V}}_{0da}$ and $\overline{\mathbf{V}}_{0ca}$ in (5.29) are all found analytically in this work from mesh information; and hence greatly saving the computational cost. Furthermore, the number of nonzero entries in each $\overline{\mathbf{V}}_{0d}(\overline{\mathbf{V}}_{0da})$ and $\overline{\mathbf{V}}_{0c}(\overline{\mathbf{V}}_{0ca})$ vector is bounded by the number of edges connected to a single node, thus very sparse; and $\overline{\mathbf{D}}$ is diagonal. As for the $\overline{\mathbf{D}}_{\epsilon,0}$ and $\overline{\mathbf{D}}_{\sigma,0}$ shown in (5.30), after substituting the analytical nullspace vectors into their expressions, we find the two matrices are nothing but the Laplacian operator formulated for the dielectric region, and the conductor region respectively, i.e., discretized $\nabla\left(\epsilon\cdot\nabla\right)$ and $\nabla\left(\sigma\cdot\nabla\right)$ using a finite difference method. Hence, their matrix solutions can be obtained efficiently using either an iterative solver like a multigrid method [43], or an advanced direct solver [6], both of which have been achieved in linear complexity. Hence, the $\overline{\mathbf{V}}_0$-component can be found rapidly in this work.

5.4 Efficient Method for Finding $\overline{\mathbf{V}}_h$ and Fast Computation of the L- and Full-wave-component of Layout Response

For many of the IC layouts, we find the $\overline{\mathbf{V}}_0$ solution is sufficient to obtain an accurate layout response at their current operating frequencies. However, when frequency increases, and/or the layout becomes larger such as the layout of packages and boards, the $\overline{\mathbf{V}}_h$ part becomes important in the layout solution, and we need to find an efficient way to obtain it. In this section, we show how to obtain the high order space $\tilde{\overline{\mathbf{V}}}_h = \overline{\mathbf{D}}^{-1}\overline{\mathbf{V}}_h$ fast in order to solve (5.28) with little computational cost.

The solution of (6.1) is governed by a quadratic eigenvalue problem

$$(\lambda^2\overline{\mathbf{D}}_\epsilon + \lambda\overline{\mathbf{D}}_\sigma + \overline{\mathbf{S}})\mathbf{v} = 0, \tag{5.43}$$

where $\overline{\mathbf{D}}_\epsilon$ and $\overline{\mathbf{D}}_\sigma$ are, respectively, diagonal matrices of permittivity and conductivity, λ is an eigenvalue, and v is the corresponding eigenvector. The eigenvectors corresponding to nonzero eigenvalues of (5.43) can be used as $\overline{\mathbf{V}}_h$. These eigenvectors are also frequency- and time-independent. Compared to using the non-nullspace eigenvectors of $\overline{\mathbf{S}}$, we find that using the eigenvectors of (5.43), the resulting number of $\overline{\mathbf{V}}_h$ modes to synthesize the layout solution is very small. This is because each eigenvector of (5.43) represents a source-free solution in the original physical problem satisfying all the material and boundary conditions. In contrast, the eigenvector of $\overline{\mathbf{S}}$

is a source-free solution in an empty computational domain, which does not represent the solution well in the actual problem. The eigenvalue of (5.43) has a clear physical meaning, which is the complex resonance frequency of the layout. For a prescribed frequency, the weight of an eigenmode of (5.43) in the field solution is inversely proportional to the difference between the eigenvalue and the solving frequency. In other words, the contribution of those eigenvectors that resonate at a higher frequency is little to the layout response at a lower working frequency. Hence, the number of $\overline{\mathbf{V}}_h$ computed in this way is small, thus making the whole solution efficient. However, solving (5.43) is known to be computationally expensive, especially when conductor loss is involved which is true in the problem studied in this work. Most of the eigenvalues and eigenmodes are complex valued, and due to the large discrepancy in the norm of the underlying matrices, the solution of (5.43) is also error prone. In this work, based on our prior work in [44, 45], we develop a fast algorithm to extract $\overline{\mathbf{V}}_h$ without solving (5.43).

In this fast algorithm, we solve (6.1) in a small time window using an explicit time marching. In this way, there is no matrix solution involved. The computational complexity is linear (optimal) at every time step. Although the time step is restricted by the smallest space step for stability, we do not need to perform the time marching for a long time since we can identify the $\overline{\mathbf{V}}_h$ from a short time simulation. We collect the solution of (6.1) every SG steps, which is termed sampling step. At the first sampling step, we record the solution e, normalize it and store it as a column vector in $\overline{\mathbf{X}}$; at the following sampling steps, we orthogonalize the newly obtained solution with existing columns in $\overline{\mathbf{X}}$, and store the resultant in $\overline{\mathbf{X}}$. Using such $\overline{\mathbf{X}}$, we transform (5.43) to a much smaller eigenvalue problem of

$$(\lambda^2 \overline{\mathbf{D}}_{\epsilon r} + \lambda \overline{\mathbf{D}}_{\sigma r} + \overline{\mathbf{S}}_r)\mathbf{v}_r = 0, \tag{5.44}$$

where

$$\overline{\mathbf{D}}_{\epsilon r} = \overline{\mathbf{X}}^T \overline{\mathbf{D}}_{\epsilon} \overline{\mathbf{X}} \tag{5.45}$$

$$\overline{\mathbf{D}}_{\sigma r} = \overline{\mathbf{X}}^T \overline{\mathbf{D}}_{\sigma} \overline{\mathbf{X}} \tag{5.46}$$

$$\overline{\mathbf{S}}_r = \overline{\mathbf{X}}^T \overline{\mathbf{S}} \overline{\mathbf{X}}. \tag{5.47}$$

The (5.44) can further be transformed to a generalized eigenvalue problem as the following

$$\overline{\mathbf{A}} \begin{bmatrix} \mathbf{v} \\ \lambda \mathbf{v} \end{bmatrix} = \lambda \overline{\mathbf{B}} \begin{bmatrix} \mathbf{v} \\ \lambda \mathbf{v} \end{bmatrix}, \tag{5.48}$$

in which

$$\overline{\mathbf{A}} = \begin{bmatrix} -\overline{\mathbf{S}}_r & 0 \\ 0 & \overline{\mathbf{D}}_{\epsilon r} \end{bmatrix}$$

$$\overline{\mathbf{B}} = \begin{bmatrix} \overline{\mathbf{D}}_{\sigma r} & \overline{\mathbf{D}}_{\epsilon r} \\ \overline{\mathbf{D}}_{\epsilon r} & 0 \end{bmatrix}. \tag{5.49}$$

The size of (5.48) is $2p$, where p is the column size of $\overline{\mathbf{X}}$, i.e., the number of time domain solutions that have been collected.

There exists a big difference in the norms of $\overline{\mathbf{S}}$, $\overline{\mathbf{D}}_{\epsilon}$, and $\overline{\mathbf{D}}_{\sigma}$, directly calculating (5.48) may not be accurate. In order to solve that, based on [46], we multiply scaling factors ρ to make $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ balanced in norm. Hence, we transform the matrices to

$$\overline{\tilde{\mathbf{A}}} = \begin{bmatrix} -\overline{\mathbf{S}}_r & 0 \\ 0 & \rho^2 \overline{\mathbf{D}}_{\epsilon r} \end{bmatrix}$$

$$\overline{\tilde{\mathbf{B}}} = \begin{bmatrix} \rho \overline{\mathbf{D}}_{\sigma r} & \rho^2 \overline{\mathbf{D}}_{\epsilon r} \\ \rho^2 \overline{\mathbf{D}}_{\epsilon r} & 0 \end{bmatrix}, \tag{5.50}$$

and (5.48) to

$$\overline{\tilde{\mathbf{A}}} \begin{bmatrix} \mathbf{v} \\ \frac{\lambda}{\rho} \mathbf{v} \end{bmatrix} = \frac{\lambda}{\rho} \tilde{\mathbf{B}} \begin{bmatrix} \mathbf{v} \\ \frac{\lambda}{\rho} \mathbf{v} \end{bmatrix}. \tag{5.51}$$

The above scaling does not change the upper part of the eigenvectors. The original eigenvalues can be obtained by multiplying the eigenvalues of (5.51) by ρ. From the expression of each matrix, we know that $\|\overline{\mathbf{S}}\| = \frac{1}{l^2\mu}$, $\|\overline{\mathbf{D}}_\sigma\| = \sigma$ and $\|\overline{\mathbf{D}}_\epsilon\| = \epsilon$, where l is feature size. Based on this information, we can determine the scaling factor ρ [46]. For example, for a micro-scale circuit, we choose $\rho = 10^{12}$.

Since we need to select eigenmodes corresponding to the nonzero eigenvalues to build $\overline{\mathbf{V}}_h$, there should be an estimation of the magnitude of the smallest nonzero eigenvalue. From (5.43), the magnitude of the eigenvalues can be analyzed from the norm of each matrix. Specifically, the eigenvalues can be estimated as $\frac{-\|\overline{\mathbf{D}}_\sigma\| \pm \sqrt{\|\overline{\mathbf{D}}_\sigma\|^2 - 4\|\overline{\mathbf{D}}_\epsilon\|\|\overline{\mathbf{S}}\|}}{2\|\overline{\mathbf{D}}_\epsilon\|}$. Using this method, for a micro-scale circuit whose feature size is at the level of μm, the magnitude of the eigenvalues can be found in the range of 10^{10} and 10^{18}. Therefore, for those eigenvalues smaller than 10^{10}, we can identify them as zero eigenvalues, and exclude their eigenmodes from $\overline{\tilde{\mathbf{V}}}_h$.

When we march on in time, we find eigenvalues repeatedly show up from the small eigenvalue problem (5.51). The reason for this can be found from [44]. Although a lossless problem is studied in [44], the same theoretical reason applies to the lossy problem studied in this work. There are two criteria we use to terminate the time-domain solution collection process. Firstly, we need to make sure the eigenmodes corresponding to the repeating eigenvalues become dominant in the field solution. If the weight of the modes corresponding to the repeating eigenvalues is larger than that of the other modes based on an accuracy parameter ϵ_1, they can be collected as $\overline{\mathbf{V}}_h$. To calculate the weight of the modes, we denote the upper half of the eigenvectors of (5.51) corresponding to the repeating eigenvalues by $\overline{\mathbf{V}}_{re}$, and other eigenvectors by $\overline{\mathbf{V}}_{nre}$. Let $\Phi = [\overline{\mathbf{V}}_{re}, \overline{\mathbf{V}}_{nre}]$. Firstly, we orthogonalize $\overline{\mathbf{V}}_{re}$ to be unitary $\overline{\tilde{\mathbf{V}}}_{re}$. Next, we remove $\overline{\mathbf{V}}_{nre}$'s $\overline{\tilde{\mathbf{V}}}_{re}$ component, as $\overline{\tilde{\mathbf{V}}}_{nre} = \overline{\mathbf{V}}_{nre} - \overline{\tilde{\mathbf{V}}}_{re}\overline{\tilde{\mathbf{V}}}_{re}^H\overline{\mathbf{V}}_{nre}$. Then from $\Phi_{new} = [\overline{\tilde{\mathbf{V}}}_{re}, \overline{\tilde{\mathbf{V}}}_{nre}]$, the coefficient w is calculated as

$$w = (\Phi_{new}^H \Phi_{new})^{-1}(\Phi_{new}^H e), \tag{5.52}$$

where $w = [w_{re}, w_{nre}]$. If the weight ratio

$$(w_{re}^H w_{re})/(w_{nre}^H w_{nre}) > \epsilon_1, \tag{5.53}$$

$\overline{\mathbf{V}}_{re}$ would be counted as $\overline{\mathbf{V}}_h$.

The second criterion is used to ensure the accuracy of $\overline{\mathbf{V}}_h$. At every SG steps, we compare the eigenvalues from two adjacent steps, q and $q+1$. If the difference between the eigenvalues is less than a prescribed error tolerance ϵ_2, which is

$$\frac{|\lambda^q - \lambda^{q+1}|}{|\lambda^q|} < \epsilon_2, \tag{5.54}$$

the corresponding eigenmode can be identified as an accurate $\overline{\mathbf{V}}_h$ mode. After (5.53) and (5.54) are satisfied, the solution collection process is terminated.

Let the mode extracted from the aforementioned procedure be $\overline{\tilde{\mathbf{V}}}_h$. It may contain a $\overline{\mathbf{V}}_0$ component due to numerical error, i.e., it is not purely a high order mode we look for. Writing it as

$$\overline{\tilde{\mathbf{V}}}_{h,i} = \overline{\mathbf{U}}_0 \mathbf{u}_{0,i} + \overline{\mathbf{D}}^{-1} \overline{\mathbf{V}}_{h,i} \mathbf{u}_{h,i}, \tag{5.55}$$

the second component is the one we want to find. Here, the $\overline{\mathbf{U}}_0$ is comprised of only two vectors,

$$\overline{\mathbf{U}}_0 = [\mathbf{U}_{0,C} \quad \mathbf{U}_{0,R}] \tag{5.56}$$

where $\mathbf{U}_{0,C}$ is the C-component of the layout solution

$$\mathbf{U}_{0,C} = \overline{\mathbf{V}}_{0d} \overline{\mathbf{D}}_{\epsilon,0}^{-1} \overline{\mathbf{V}}_{0da}^{T} / (-j\omega) \mathbf{J}, \tag{5.57}$$

and $\mathbf{U}_{0,R}$ is the R-component of the layout solution

$$\begin{aligned} \mathbf{U}_{0,R} = {} & \overline{\mathbf{V}}_{0d} \overline{\mathbf{D}}_{\epsilon,0}^{-1} \overline{\mathbf{V}}_{0da}^{T} diag\{\epsilon\} \overline{\mathbf{V}}_{0c} \overline{\mathbf{D}}_{\sigma,0}^{-1} \overline{\mathbf{V}}_{0ca}^{T} \times \\ & (\bar{\mathbf{I}} - diag\{\epsilon\} \overline{\mathbf{V}}_{0d} \overline{\mathbf{D}}_{\epsilon,0}^{-1} \overline{\mathbf{V}}_{0da}^{T}) \mathbf{J} \\ & + \overline{\mathbf{V}}_{0c} \overline{\mathbf{D}}_{\sigma,0}^{-1} \overline{\mathbf{V}}_{0ca}^{T} (diag\{\epsilon\} \overline{\mathbf{V}}_{0d} \overline{\mathbf{D}}_{\epsilon,0}^{-1} \overline{\mathbf{V}}_{0da}^{T} - \bar{\mathbf{I}}) \mathbf{J}, \end{aligned} \tag{5.58}$$

both of which have been found when computing the $\overline{\mathbf{V}}_0$-component of the field solution.

Multiplying (5.55) by the left nullspace $\overline{\mathbf{U}}_{0a}^{T}$, which is similar to $\overline{\mathbf{U}}_{0}^{T}$ except that it is in the $\overline{\mathbf{V}}_{0a}$ space, and thus

$$\mathbf{U}_{0a,C} = \overline{\mathbf{V}}_{0da} \overline{\mathbf{D}}_{\epsilon,0}^{-1} \overline{\mathbf{V}}_{0da}^{T} / (-j\omega) \mathbf{J}, \tag{5.59}$$

and

$$\begin{aligned}\mathbf{U}_{0a,R} = {}& \overline{\mathbf{V}}_{0da}\overline{\mathbf{D}}^{-1}_{\epsilon,0}\overline{\mathbf{V}}^{T}_{0da} diag\{\epsilon\}\overline{\mathbf{V}}_{0c}\overline{\mathbf{D}}^{-1}_{\sigma,0}\overline{\mathbf{V}}^{T}_{0ca}\times \\ & (\bar{\mathbf{I}} - diag\{\epsilon\}\overline{\mathbf{V}}_{0d}\overline{\mathbf{D}}^{-1}_{\epsilon,0}\overline{\mathbf{V}}^{T}_{0da})\mathbf{J} \\ & + \overline{\mathbf{V}}_{0ca}\overline{\mathbf{D}}^{-1}_{\sigma,0}\overline{\mathbf{V}}^{T}_{0ca}(diag\{\epsilon\}\overline{\mathbf{V}}_{0d}\overline{\mathbf{D}}^{-1}_{\epsilon,0}\overline{\mathbf{V}}^{T}_{0da} - \bar{\mathbf{I}})\mathbf{J}.\end{aligned} \tag{5.60}$$

The multiplication results in

$$\overline{\mathbf{U}}^{T}_{0a}\overline{\mathbf{D}}\overline{\tilde{\mathbf{V}}}_{h,i} = \overline{\mathbf{U}}^{T}_{0a}\overline{\mathbf{D}}\overline{\mathbf{U}}_{0}\mathbf{u}_{0,i}. \tag{5.61}$$

Hence,

$$\mathbf{u}_{0,i} = \left(\overline{\mathbf{U}}^{T}_{0a}\overline{\mathbf{D}}\overline{\mathbf{U}}_{0}\right)^{-1}\overline{\mathbf{U}}^{T}_{0a}\overline{\mathbf{D}}\overline{\tilde{\mathbf{V}}}_{h,i}. \tag{5.62}$$

As a result, we can obtain a pure $\overline{\mathbf{V}}_h$-part from the contaminated $\overline{\tilde{\mathbf{V}}}_h$ as the following

$$\overline{\tilde{\mathbf{V}}}_h = \overline{\tilde{\mathbf{V}}}_h - \overline{\mathbf{U}}_0(\overline{\mathbf{U}}^{T}_{0a}\overline{\mathbf{D}}\mathbf{u}_0)^{-1}(\overline{\mathbf{U}}^{T}_{0a}\overline{\mathbf{D}}\overline{\tilde{\mathbf{V}}}_h), \tag{5.63}$$

which satisfies $\overline{\mathbf{V}}^{T}_{0a}\overline{\mathbf{D}}\overline{\tilde{\mathbf{V}}}_h = 0$, and thereby $\overline{\mathbf{V}}^{T}_{0a}\overline{\mathbf{V}}_h = 0$. Then we can apply this $\overline{\tilde{\mathbf{V}}}_h$ in (5.28) to obtain the $\overline{\mathbf{V}}_h$ part of the solution. Since $\overline{\mathbf{V}}^{T}_{0}\overline{\mathbf{V}}_h = 0$ is not satisfied but $\overline{\mathbf{V}}^{T}_{0a}\overline{\mathbf{V}}_h = 0$ is satisfied, if we use $\overline{\tilde{\mathbf{V}}}_h$ as the testing column space in (5.28), there is a term $\overline{\mathbf{V}}^{T}_{h}\overline{\mathbf{V}}_0\mathbf{y}_0$ left which cannot be vanished. But this term is known, which can be moved to the right hand side of (5.28). Hence, $\mathbf{y}_h$ can still be readily solved. After we solve $\mathbf{y}_h$, the final solution is combined as

$$\mathbf{e} = \overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0c}\mathbf{y}_{0c} + \overline{\tilde{\mathbf{V}}}_h\mathbf{y}_h, \tag{5.64}$$

which contains the complete R-, C-, L-, and full-wave components. The number of $\overline{\mathbf{V}}_h$ modes is usually small for IC layouts, thus (5.28) has a very small dimension, whose solution can be readily computed.

5.5 Layout Modeling and Simulation Results

In this section, we simulate a variety of IC layouts to examine the performance of the proposed work.

Table 5.1.: The capacitance computed at the near (port 1) and far end (port 2).

Capacitance (F)	This Method	Reference
C_{11}	1.0356e-15	1.0356e-15
C_{12}	1.0356e-15	1.0356e-15

5.5.1 Bus Wire

A 3-D on-chip interconnect example is simulated, which is shown in Fig. 5.2. The sizes along x-, y-, and z-directions are 31 μm, 10 μm, and 3 μm, respectively. The yellow regions are conductors. Their conductivity is 5.7×10^7 S/m. The material and geometrical data are specified in Fig. 5.2. The current source is imposed across the red line. In Table 5.1, we list the capacitance obtained from the proposed inverse model in comparison with the reference result obtained from a brute-force finite-difference solution in frequency domain. Excellent agreement is observed.

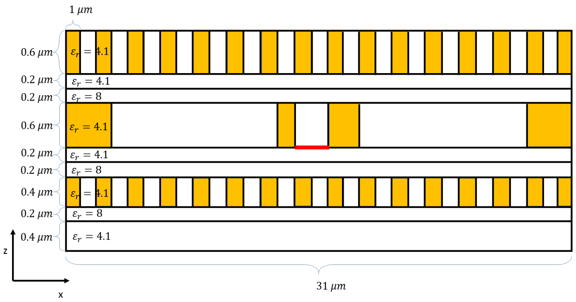


Fig. 5.2.: Illustration of a 3-D on-chip interconnect layout.

5.5.2 Test-chip Interconnect

A test-chip interconnect is simulated, whose structure is shown in Fig. 6.1. The yellow regions are conductors, the conductivity of which is $5.7 \times 10^7 S/m$. The dimension along the x-, y- and z-direction is 300, 100, and 3.912 μm respectively. The current source is launched from the bottom ground plane to the conductor in the metal-3 layer. We compare the S-parameters extracted from our method from 45 MHz to 10 GHz with the measured data in Fig. 5.4. As can be seen, they agree very well with each other. This verifies the accuracy of the proposed method.

The S-parameters shown in Fig. 5.4 are generated from the $\overline{\mathbf{V}}_0$-, i.e., RC-component of the layout solution. When we increase the frequency up to 100 GHz, we find that the RC-component is not sufficient any more in producing good accuracy. As can be seen from Fig. 5.5, without considering $\overline{\mathbf{V}}_h$, the entire solution error becomes worse and worse when frequency is increased, and it becomes 16.9% at 50 GHz, and even exceeds 40% at 100 GHz. Here, the entire solution error is measured by

$$\text{Entire Solution Error} = \frac{\|\mathbf{e} - \mathbf{e}_{ref}\|}{\|\mathbf{e}_{ref}\|}, \tag{5.65}$$

where $\mathbf{e}$ is from the proposed solution which contains all electric field unknowns in the layout, whereas $\mathbf{e}_{ref}$ is a brute-force solution obtained by solving (5.8) as it is.

We hence employ the algorithm described in Section 5.4 to extract $\overline{\mathbf{V}}_h$ modes and add the $\overline{\mathbf{V}}_h$-component into the solution of e. In the time marching procedure, we use a Gaussian derivative source with $\tau = 10^{-11}$ s. The dt is chosen to be $10^{-15}s$ for time-domain stability. Other simulation parameters are chosen as $\epsilon_1 = 10^{-5}$, $\epsilon_2 = 10^{-2}$, and $SG = 100$. From this procedure, we identify 30 $\overline{\mathbf{V}}_h$ modes. After adding the $\overline{\mathbf{V}}_h$ part of the solution, we obtain the entire solution error shown by the red line in Fig. 5.5. The error is significantly reduced from 16.9% to 1.06% at 50 GHz, and from 40% to 2% at 100 GHz. Only 32 steps of sampling are performed in the time marching procedure, and hence the time window simulated is short, and thereby the overall simulation is efficient. The proposed algorithm also allows one to achieve even higher accuracy by performing the time marching in a longer time window, and hence extracting more $\overline{\mathbf{V}}_h$ modes. For example, using 400 sampling, we find 231 $\overline{\mathbf{V}}_h$ modes, using which and the accuracy of the field solution is further reduced from 1.06% to 0.04% at 50 GHz.

In Table 5.2, we list the first 20 eigenvalues, sorted based on the imaginary part's magnitude, found from the proposed fast solution in comparison with those computed from a brute-force eigenvalue solution of (5.43). Excellent agreement can be observed, which validates the proposed fast algorithm for finding physically important eigenvectors.

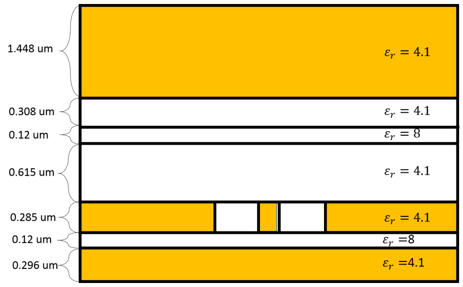
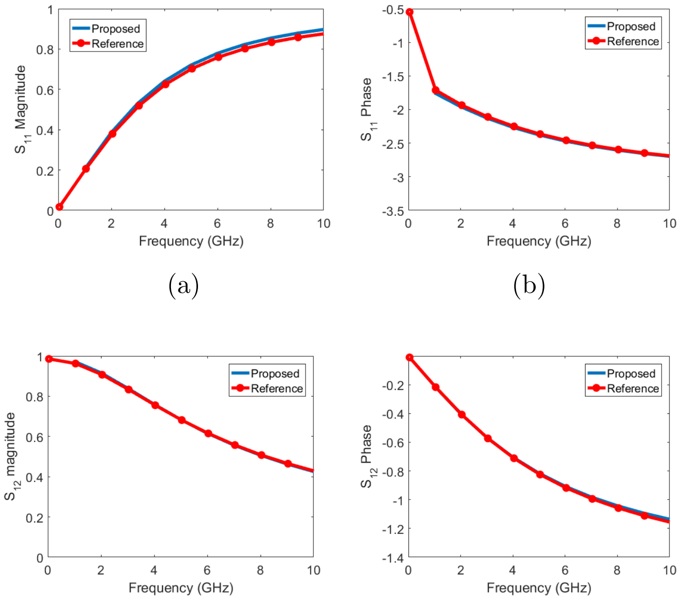


Fig. 5.3.: Structure of a test-chip interconnect.



(a)

(b)

(c)

(d)

Fig. 5.4.: (a) $|S_{11}|$; (b) S_{11} phase; (c) $|S_{12}|$; (d) S_{12} phase.

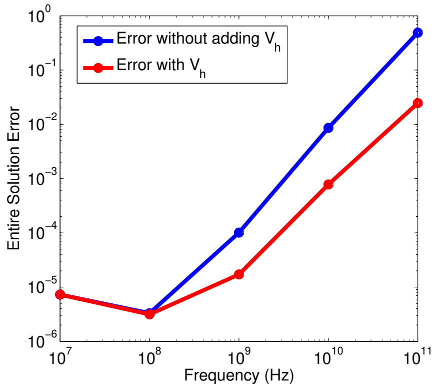


Fig. 5.5.: Entire solution error before and after adding the $\overline{\mathbf{V}}_h$ part of the solution as a function of frequency for the test-chip interconnect example.

5.5.3 On-Chip Power Grid

In this example, we simulate an on-chip power grid shown in Fig. 6.6. The power wire is colored in yellow, while the ground wire is in blue. The regions other than conductors are dielectrics. The current source is a Gaussian derivative with $\tau = 10^{-12}$ s, and it is injected from the ground wire to the power wire, which is shown by the red line in Fig. 6.6. In this example, the $\overline{\mathbf{V}}_h$ part is also needed to obtain good accuracy at high frequencies. To find $\tilde{\overline{\mathbf{V}}}_h$, the parameters used are $\epsilon_1 = 10^{-5}$, $\epsilon_2 = 10^{-2}$, and $SG = 100$. The time step used in the explicit time marching is $dt = 10^{-15}s$. There are 4 $\overline{\mathbf{V}}_h$ modes found with 5 steps of sampling. Their eigenvalues are $-5.3506e10 \pm 2.0334e13i$ and $-1.2362e11 \pm 5.2297e + 13i$. Without adding the 4 modes, the error of the entire solution is 34.48% at 1000 GHz; adding them, the error is greatly reduced to 4.4%. The accuracy before and after adding the 4 $\overline{\mathbf{V}}_h$ modes is shown in Fig. 5.7. Again, when we increase the time window for time marching, we find more $\overline{\mathbf{V}}_h$ modes, and also they are more accurate. In Fig. 5.8a and Fig. 5.8b, we plot the error of the complex eigenvalues of the $\overline{\mathbf{V}}_h$ modes extracted using 5 steps of sampling with $SG = 100$, and 300 steps of sampling with $SG = 5$, respectively. As can be seen, although the accuracy of both is good, the latter case

Table 5.2.: The first 20 eigenvalues from the proposed fast eigenvalue solution compared to those from the original eigenvalue solution for the interconnect example.

Eigenvalue from original problem	Eigenvalue from fast solution
-5.2774e+11+1.3077e+12j	-5.2812e+11+1.3075e+12j
-5.2774e+11-1.3077e+12j	-5.2774e+11-1.3075e+12j
-4.7621e+11+1.3428e+12j	-4.7623e+11-1.3428e+12j
-4.7621e+11-1.3428e+12j	-4.7623e+11-1.3428e+12j
-5.4773e+11+2.8958e+12j	-5.2391e+11+2.9010e+12j
-5.4773e+11-2.8958e+12j	-5.2391e+11-2.9010e+12j
-5.1701e+11+2.9430e+12j	-5.1427e+11+2.9423e+12j
-5.1701e+11-2.9430e+12j	-5.1427e+11-2.9423e+12j
-5.5160e+11+4.3458e+12j	-5.3063e+11+4.3972e+12j
-5.5160e+11-4.3458e+12j	-5.3063e+11-4.3972e+12j
-5.2938e+11+4.3995e+12j	-5.3063e+11+4.3972e+12j
-5.2938e+11-4.3995e+12j	-5.3063e+11-4.3972e+12j
-6.3684e+10+5.3430e+12j	-6.3678e+10+5.3431e+12j
-6.3684e+10-5.3430e+12j	-6.3678e+10-5.3431e+12j
-5.3620e+11+5.7451e+12j	-5.3561e+11+5.7433e+12j
-5.3620e+11-5.7451e+12j	-5.3561e+11-5.7433e+12j
-5.4049e+11+6.9653e+12j	-5.4235e+11+6.9603e+12j
-5.4049e+11-6.9653e+12j	-5.4235e+11-6.9603e+12j
-5.4333e+11+8.0397e+12j	-5.4189e+11+8.0346e+12j
-5.4333e+11-8.0397e+12j	-5.4189e+11-8.0346e+12j

is more accurate, and having more $\overline{\mathbf{V}}_h$ modes identified. For the latter case, 299 $\overline{\mathbf{V}}_h$ modes are identified, using which the entire solution error is reduced to 1.7131×10^{-6} at 1000 GHz.

In Table 5.3, we list the first 20 eigenvalues found from the proposed fast algorithm in comparison with those of the original eigenvalue solution. As can be seen, they match each other very well, which validates the proposed method for finding high order modes.

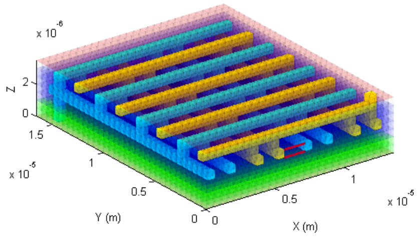


Fig. 5.6.: Structure of the power grid.

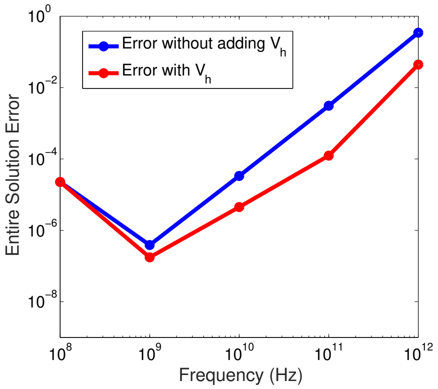
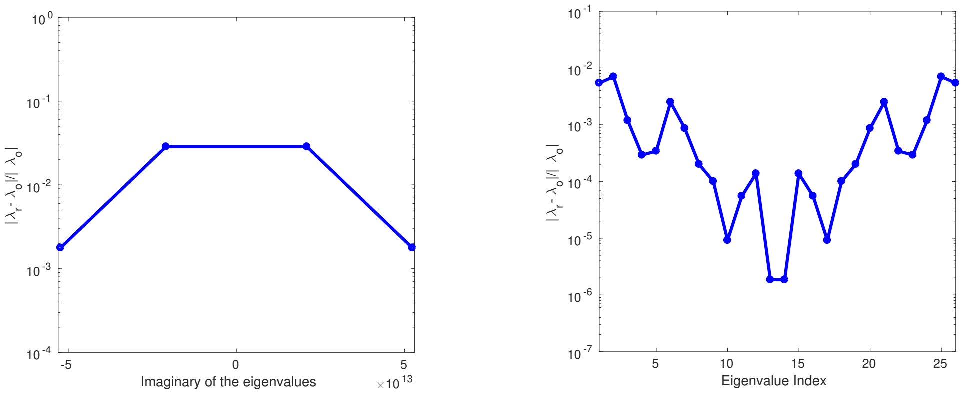


Fig. 5.7.: Entire solution error before and after adding the $\overline{\mathbf{V}}_h$ part of the solution for different frequencies for the power grid structure.

5.5.4 Scan D Flip-Flop Layout

Next, to examine the capability of the proposed work, we take a GDSII file from a 45 nm Scan D flip-flop design (`https://www.cs.upc.edu/~jpetit/CellRouting/`



(a)

(b)

Fig. 5.8.: Comparison of the eigenvalues from the proposed fast solution and the original eigenvalue solution with (a) 5 steps of sampling with $SG = 100$, and (b) 300 steps of sampling with $SG = 5$ for the power grid example.

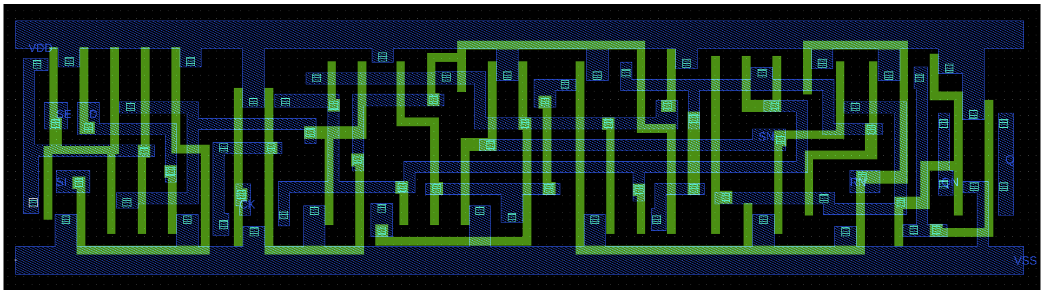


Fig. 5.9.: Top view of the Scan D flip-flop layout in layer 9, 10, and 11.

nangate/Front_End/Doc/Databook/Cells/SDFFRS_X2_NangateOpenCellLibrary_typical_typical.html) and analyze its layout performance. The top view of the structure in layer 9, 10, and 11 is shown in Fig. 5.9, with each layer plotted in different colors. The blue and green regions are occupied by conductors, whose conductivity is 5.7×10^7 S/m. The current is injected from the ground plane at the bottom to a power port, the waveform of which is a Gaussian derivative with $\tau = 10^{-10}$ s. In this example, there are around 3,616,773 unknowns. To simulate this example, the time step of a traditional FDTD must be less than 10^{-16} s to ensure stability. The proposed method

Table 5.3.: The first 20 eigenvalues from the proposed fast eigenvalue solution compared to those from the original eigenvalue solution for the power gird example.

Eigenvalue from original problem	Eigenvalue from fast solution
-2.0932e+10+1.4133e+13j	-2.0915e+11+1.4133e+13j
-2.0932e+10-1.4133e+13j	-2.0915e+11-1.4133e+13j
-2.5515e+10+2.0933e+13j	-2.2350e+10+2.0932e+13j
-2.5515e+10-2.0933e+13j	-2.2350e+10-2.0932e+13j
-2.2928e+10+2.3214e+13j	-2.3428e+10+2.3210e+13j
-2.2928e+10-2.3214e+13j	-2.3428e+10-2.3210e+13j
-2.2120e+10+2.6723e+13j	-2.1924e+10+2.6723e+13j
-2.2120e+10-2.6723e+13j	-2.1924e+10-2.6723e+13j
-2.1983e+10+2.9874e+13j	-2.7626e+10+2.9874e+13j
-2.1983e+10-2.9874e+13j	-2.7626e+10-2.9874e+13j
-2.2730e+10+3.5079e+13j	-2.3174e+10+3.5086e+13j
-2.2730e+10-3.5079e+13j	-2.3174e+10-3.5086e+13j
-2.5862e+10+3.9890e+13j	-6.5502e+10+3.9903e+13j
-2.5862e+10-3.9890e+13j	-6.5502e+10-3.9903e+13j
-2.4912e+10+4.0165e+13j	-1.0058e+11+4.0227e+13j
-2.4912e+10-4.0165e+13j	-1.0058e+11-4.0227e+13j
-2.3851e+10+4.4811e+13j	-3.5384e+10+4.4819e+13j
-2.3851e+10-4.4811e+13j	-3.5384e+10-4.4819e+13j
-2.3552e+10+4.7854e+13j	-1.8570e+10+4.7841e+13j
-2.3552e+10-4.7854e+13j	-1.8570e+10-4.7841e+13j

is able to use an arbitrarily large time step since its time-dependence is analytically derived in the inverse model. Here, we use a time step of 10^{-11} s solely determined by accuracy. The proposed method only takes around 50 seconds, whereas the FDTD requires 5.017×10^6 s to finish the simulation of the whole structure in the same time window.

5.5.5 Intel 4004

The last example is an Intel 4004 processor (a 4-bit central processing unit), the layout of which is shown in Fig. 5.10. It has 7 layers and over 86,220 objects, the discretization of which results in 115,455,658 unknowns, which is over 115 million. The GDSII file of the processor is directly loaded into the software developed based on the proposed algorithm, and the layout analysis is fully automated. It only takes the proposed method 729 s to finish the extraction and analysis of the entire layout for one circuit stimulus, which demonstrates the efficiency and high capacity of this work.

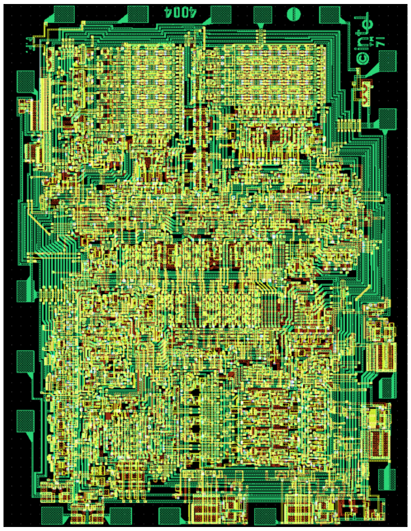


Fig. 5.10.: Layout of Intel 4004 processor.

5.6 Conclusion

In this work, a closed-form model of the inverse of full-wave Maxwell's system of equations is found for an arbitrary physical layout in both frequency and time domain. The advantages of the proposed inverse model of IC layouts are multifaceted. First, it is accurate from zero to high frequencies; Second, the layout response is explicitly decomposed into R-, C-, L-, and full-wave components, without computation or approximation, each of which can be obtained independently, and then superposed to obtain the final layout response. This not only is much more efficient than a brute-force simulation of (6.1), but also provides circuit designers with key insights for layout automation. In addition, neither time marching nor point-by-point frequency sweep needs to be performed for the RC-component as its time and frequency dependence are analytically known from (5.29). Moreover, the full-wave component is also efficiently represented by $\overline{\mathbf{V}}_h$ modes whose number is small. Hence, both its time- and frequency-domain representations can be readily obtained. The proposed work has been applied to large scale layout modeling and simulation. Superior performance in efficiency, accuracy, and capacity has been demonstrated. In addition to ICs, this work also provides package and board designers with a rapid, accurate, and circuit-intuitive tool for layout automation. Its broadband inverse model is also applicable to the electromagnetic analysis of other physical problems.

6. FAST TIME-DOMAIN METHOD FOR COMPUTING THE FULL-WAVE SOLUTION OF INTEGRATED CIRCUIT LAYOUTS BY CHANGING THE CURL-CURL OPERATOR TO LAPLACIAN

6.1 Introduction

Fast solution of full-wave Maxwell's equations in integrated circuits (ICs) is challenging because of multiple reasons. First, the problem is large and deeply multiscaled. Second, the materials are inhomogeneous, and the conductors are lossy. Using perfect conductors or impedance boundary conditions may yield significant errors in predicting the circuit performance. Third, a broad band of frequencies from DC to high frequencies is encountered. In this band, both static and full-wave components co-exist in the field solution. The two also couple with each other. We cannot solve static equations alone. Meanwhile, the frequency is not that high; we cannot ignore the static component (the gradient field) either in the electric field solution.

There have been a large number of efforts addressing the aforementioned challenges [9, 15, 22, 23, 47], including both frequency- and time-domain methods. In frequency domain, the system matrix resulting from the discretization of ICs has a large condition number, and it is indefinite, which renders an iterative solution difficult to converge. In time domain, the system matrix can be made positive definite. However, the time step is restricted by the smallest space step in a conventional explicit time marching. Comparing to the time step determined by the operating frequency of the ICs, the time step restricted by the stability criterion is orders of magnitude smaller. This makes a traditional explicit simulation not feasible for analyzing large-scale ICs. In an implicit unconditionally stable time domain method, the time step is enlarged, however, one has to solve a system matrix. In addition, this matrix becomes ill-conditioned when a large time step is used.

In this work, we analyze the property of the system matrix resulting from an implicit FDTD-based full-wave analysis. We find that the system matrix representing the curl-curl operator is the root cause of its slow convergence. This matrix's smallest eigenvalue is zero, whereas its largest one is inversely proportional to the square of the smallest feature size in the problem being studied. In an IC problem, this translates to a huge condition number. To overcome this problem, we find a way to decompose the discretized curl-curl operator into a gradient divergence operator and a Laplacian. We also achieve such a decomposition without any numerical computation, by constructing both operators via an analytical means. The gradient operator vanishes when acting on the high-frequency (full-wave) component having a nonzero curl. Hence, we can replace the curl-curl operator by the Laplacian when it operates on the full-wave component of the field solution. Since the Laplacian is full-rank and well-conditioned, the updated system matrix can be converged in a very small number of iterations, whose number does not grow with the matrix size either. Meanwhile, no theoretical approximation is made, and the accuracy is retained. We show how to use the proposed idea to solve IC problems where static and full-wave field solutions co-exist, and conductors are highly lossy. Numerical experiments have validated the accuracy and efficiency of the proposed work. The rest of the Chapter is organized as follows. In Section 6.2, we elaborate the proposed method. In Section 6.3, we present extensive numerical results to demonstrate the accuracy and efficiency of the proposed method in on-chip modeling and simulation. We summarize this work in Section 6.4.

6.2 Proposed Method

To solve full-wave Maxwell's equations, we discretize the entire physical layout of an IC into a grid to model lossy conductors and inhomogeneous materials. Let e be a vector of all electric field unknowns in the grid, the length of which is N_e. Using the patch-based single-grid formulation of the FDTD [48], we can find the following linear system of equations of e:

$$\overline{\mathbf{D}}_\epsilon \frac{d^2\mathbf{e}}{dt^2} + \overline{\mathbf{D}}_\sigma \frac{d\mathbf{e}}{dt} + \overline{\mathbf{S}}\mathbf{e} = -\frac{d\mathbf{J}}{dt}, \tag{6.1}$$

where $\mathbf{J}$ is a vector of current density, $\overline{\mathbf{D}}_\epsilon$ and $\overline{\mathbf{D}}_\sigma$ are diagonal matrices of permittivity, and conductivity respectively, whose i-th entry represents the permittivity and conductivity at the i-th edge.

The $\overline{\mathbf{S}}$ in (6.1) is a sparse matrix representing a discretized $\nabla \times \mu^{-1} \nabla \times$ operation. To generate $\overline{\mathbf{S}}$, based on [48], for each patch in the grid, we generate one row vector $\overline{\mathbf{S}}_e(i,:)$ shown in the following

$$\overline{\mathbf{S}}_e(i,:) = \left[-\frac{1}{L_i} \ \frac{1}{L_i} \ \frac{1}{W_i} \ \ -\frac{1}{W_i} \right] \oplus zeros(1, N_e), \tag{6.2}$$

where i denotes the patch index, L_i and W_i are, respectively, the two side lengths of patch i, and $\oplus$ denotes an operation of adding the four nonzero entries to a global vector of length N_e. Where to add the nonzero entries is based on the global indexes of the four local $\mathbf{E}$ unknowns of patch i. We also generate one column vector $\overline{\mathbf{S}}_h(:,i)$, which is the transpose of $\overline{\mathbf{S}}_e(i,:)$ in a uniform grid. If a non-uniform grid is used, we replace the length (width) in $\overline{\mathbf{S}}_h(:,i)$ by an average length (width) across the two patches sharing the $\mathbf{E}$ edge. This results in better accuracy in the FDTD, since $\mathbf{H}$ is not centered by $\mathbf{E}$ in a non-uniform grid. As a result, we obtain

$$\overline{\mathbf{S}}_h(:,i)^T = \left[-\frac{1}{L_{i,1a}} \ \frac{1}{L_{i,2a}} \ \frac{1}{W_{i,3a}} \ \ -\frac{1}{W_{i,4a}} \right] \oplus zeros(1, N_e), \tag{6.3}$$

where $L_{i,1a}$ ($L_{i,2a}$) represents the length averaged between the two patches sharing the first (second) local $\mathbf{E}$ edge on patch i, while $W_{i,3a}$ ($W_{i,4a}$) denote the same but for the third (fourth) $\mathbf{E}$ unknown. The two patches used for each $\mathbf{E}$ unknown are on the same plane as patch i. Using $\overline{\mathbf{S}}_h$ and $\overline{\mathbf{S}}_e$, $\overline{\mathbf{S}}$ can be written as the following

$$\overline{\mathbf{S}} = \overline{\mathbf{S}}_h \overline{\mathbf{D}}_\mu \overline{\mathbf{S}}_e, \tag{6.4}$$

where $\overline{\mathbf{S}}_h$ has N_h columns, the i-th of which is shown in (6.3), $\overline{\mathbf{S}}_e$ has N_h rows whose i-th row is shown in (6.2), and N_h is the total number of patches, which is also the number of magnetic field unknowns. The $\overline{\mathbf{D}}_\mu$ is nothing but a diagonal matrix of permeability. From (6.4), it can be seen that $\overline{\mathbf{S}}$ is rank deficient, whose smallest eigenvalue is zero; and largest one is inversely proportional to the square of the smallest feature size. This renders the condition number of $\overline{\mathbf{S}}$ extremely large when modeling micro- and nano-meter scale structures.

Inspired by the vector identity $\nabla \times \nabla \times \mathbf{A} = \nabla(\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A}$, we find that matrix $\overline{\mathbf{S}}$ can be decomposed into the following form

$$\overline{\mathbf{S}} = -\overline{\mathbf{V}}_0 \overline{\mathbf{V}}_{0a}^T / \mu + \overline{\mathbf{L}}, \tag{6.5}$$

where $\overline{\mathbf{V}}_0$ denotes the right nullspace of $\overline{\mathbf{S}}$, thus

$$\overline{\mathbf{S}}\overline{\mathbf{V}}_0 = 0; \tag{6.6}$$

and $\overline{\mathbf{V}}_{0a}^T$ denotes the left nullspace of $\overline{\mathbf{S}}$, satisfying

$$\overline{\mathbf{V}}_{0a}^T \overline{\mathbf{S}} = 0; \tag{6.7}$$

and $\overline{\mathbf{L}}$ denotes the discretized Laplacian $-\nabla^2$ divided by μ in the grid. If a uniform grid is used, then $\overline{\mathbf{V}}_0 = \overline{\mathbf{V}}_{0a}$ since $\overline{\mathbf{S}}$ is symmetric. The $\overline{\mathbf{V}}_0$'s column number is equal to the number of nodes minus 1 in the grid. The i-th column of $\overline{\mathbf{V}}_0$ corresponds to the i-th node. The number of nonzero entries in this vector, $\overline{\mathbf{V}}_0(:, i)$, is equal to the number of edges connected to node i. They can be generated as the following without any need for numerical computation [49]:

- For the edge whose electric field reference direction enters the node, the nonzero entry is $\frac{1}{l_i}$, at the row corresponding to the global index of the edge (and thereby its $\mathbf{E}$ unknown);

- For the edge whose electric field reference direction leaves the node, the nonzero entry is $-\frac{1}{l_i}$;

where l_i is the length of the i-th edge. Using (6.2), it can be seen that $\overline{\mathbf{S}}_e\overline{\mathbf{V}}_0(:,i) = 0$, and hence (6.6) is satisfied. The left nullspace $\overline{\mathbf{V}}_{0a}$ can also be generated without any need for computation. It is the same as $\overline{\mathbf{V}}_0$ but using an average length. Specifically, on an edge that is connected to the node, its corresponding nonzero entry in $\overline{\mathbf{V}}_{0a}$ vector is $\frac{1}{l_{i,ave}}$, where $l_{i,ave}$ is the length averaged from the edge and the other edge connected to the node along the same direction. Such a nullspace vector satisfies $\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{S}} = 0$ as can be seen from (6.4) and (6.3).

Let $\overline{\mathbf{V}}_h$ be $\overline{\mathbf{S}}$'s eigenvector whose eigenvalue λ is nonzero, then $\overline{\mathbf{S}}\overline{\mathbf{V}}_h = \lambda\overline{\mathbf{V}}_h$ holds true, and hence

$$\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{V}}_h = \overline{\mathbf{V}}_{0a}^T\overline{\mathbf{S}}\overline{\mathbf{V}}_h/\lambda = 0 \tag{6.8}$$

using (6.7). As a result, we find the following important property:

$$\overline{\mathbf{S}}\overline{\mathbf{V}}_h = -\overline{\mathbf{V}}_0\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{V}}_h/\mu + \overline{\mathbf{L}}\overline{\mathbf{V}}_h = \overline{\mathbf{L}}\overline{\mathbf{V}}_h. \tag{6.9}$$

This means when operating on the high-frequency modes whose curl is not zero, we can replace $\overline{\mathbf{S}}$ by $\overline{\mathbf{L}}$! Compared to $\overline{\mathbf{S}}$, $\overline{\mathbf{L}}$ is of full rank and well-conditioned with a low condition number, as it is a Laplace operator. Many iterative solvers can solve a Laplace operator fast, and even converge it in a constant number of operations independent of matrix size [50].

If the solution of (6.1) is dominated by the full-wave component, then we can directly replace $\overline{\mathbf{S}}$ therein by $\overline{\mathbf{L}}$, and solve it efficiently. But for problems involving both static and full-wave solutions, more steps need to be taken to take advantage of (6.9). Next, we show for such a situation how to use the property of (6.9) to develop a fast solution in time domain.

We discretize (6.1) in time domain as the following

$$\begin{aligned}&\overline{\mathbf{D}}_\epsilon(\mathbf{e}^{n+1} + \mathbf{e}^{n-1} - 2\mathbf{e}^n) + \Delta t\overline{\mathbf{D}}_\sigma(\mathbf{e}^{n+1} - \mathbf{e}^n) + \Delta t^2\overline{\mathbf{S}}\mathbf{e}^{n+1}\\ &= -\Delta t^2\frac{\partial \mathbf{J}^{n+1}}{\partial t},\end{aligned} \tag{6.10}$$

where $\mathbf{e}^{n+1}$, $\mathbf{e}^{n}$ and $\mathbf{e}^{n-1}$ denotes the electrical field solutions at the $(n+1)$-th, n-th and $(n-1)$-th time step, respectively, and Δt is the time step. Moving all the terms associated with previous time steps to the right hand side, we obtain

$$\begin{aligned}&(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma + \Delta t^2\overline{\mathbf{S}})\mathbf{e}^{n+1}\\ &= -\Delta t^2\frac{\partial \mathbf{J}^{n+1}}{\partial t} - \overline{\mathbf{D}}_\epsilon(\mathbf{e}^{n-1} - 2\mathbf{e}^{n}) - \Delta t\overline{\mathbf{D}}_\sigma(-\mathbf{e}^{n}).\end{aligned} \tag{6.11}$$

The above is so-called a backward difference scheme [13], which is unconditionally stable, and hence allowing for the use of an arbitrarily large time step irrespective of space step. However, as can be seen, one has to solve the following system matrix

$$(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma + \Delta t^2\overline{\mathbf{S}}). \tag{6.12}$$

This matrix is ill-conditioned especially when Δt is large. This is because the ratio of $\Delta t^2\overline{\mathbf{S}}$ to $\overline{\mathbf{D}}_\epsilon$ is the ratio of $\Delta t^2/\Delta t_a^2$, where Δt_a is the time step restricted by the space step as that determined by the CFL condition. Clearly, the larger the Δt, the more important the term of $\Delta t^2\overline{\mathbf{S}}$ as compared to $\overline{\mathbf{D}}_\epsilon$ term. For a time step 10 times larger than the CFL condition, the $\overline{\mathbf{D}}_\epsilon$ term already becomes negligible, as it is 100 times smaller than $\Delta t^2\overline{\mathbf{S}}$. Since $\overline{\mathbf{S}}$ is singular, numerically, its solution is difficult to converge. When there is conductor loss, $\overline{\mathbf{D}}_\sigma$ term exists and also it is orders of magnitude different from $\overline{\mathbf{D}}_\epsilon$, which makes an iterative solution of (6.12) even more difficult to converge.

In view of (6.9), we decompose the field solution into two parts:

$$\mathbf{e} = \overline{\mathbf{V}}_0\mathbf{y}_0 + \mathbf{e}_h, \tag{6.13}$$

where $\overline{\mathbf{V}}_0\mathbf{y}_0$ is its gradient component, and $\mathbf{e}_h$ is its $\overline{\mathbf{V}}_h$ component, so that (6.9) can be utilized to speed up the solution of high-frequency components.

Using (6.13) in (6.11), and multiplying the resultant by $[\overline{\mathbf{V}}_{0a}\ \overline{\mathbf{I}}]^T$, we obtain

$$\begin{aligned}&\begin{bmatrix}\overline{\mathbf{V}}_{0a}^T(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_0 & \overline{\mathbf{V}}_{0a}^T(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma)\\ (\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_0 & \overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma + \Delta t^2\overline{\mathbf{S}}\end{bmatrix}\begin{bmatrix}\mathbf{y}_0^{n+1}\\ \mathbf{e}_h^{n+1}\end{bmatrix}\\ &= \begin{bmatrix}\overline{\mathbf{V}}_{0a}^T\mathbf{b}\\ \mathbf{b}\end{bmatrix},\end{aligned} \tag{6.14}$$

where $\mathbf{b} = -\Delta t^2 \frac{\partial \mathbf{J}^{n+1}}{\partial t} - \overline{\mathbf{D}}_\epsilon(\mathbf{e}^{n-1} - 2\mathbf{e}^n) - \Delta t \overline{\mathbf{D}}_\sigma(-\mathbf{e}^n)$, and (6.6) and (6.7) are utilized to vanish $\overline{\mathbf{S}}$-related terms when $\overline{\mathbf{S}}$ is multiplied with nullspace vectors.

The only $\overline{\mathbf{S}}$ left in (6.14) operates on $\mathbf{e}_h^{n+1}$, which is in the space of $\overline{\mathbf{V}}_h$. Using (6.9), we can replace $\overline{\mathbf{S}}$ by $\overline{\mathbf{L}}$, obtaining

$$\begin{bmatrix} \overline{\mathbf{V}}_{0a}^T(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_0 & \overline{\mathbf{V}}_{0a}^T(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma) \\ (\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_0 & \overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma + \Delta t^2\overline{\mathbf{L}} \end{bmatrix} \begin{bmatrix} \mathbf{y}_0^{n+1} \\ \mathbf{e}_h^{n+1} \end{bmatrix} \tag{6.15}$$

As a result, the lower right block of the system matrix becomes a well-conditioned matrix to solve. Notice that $\overline{\mathbf{L}}$ is positive definite and of full rank.

If we solve (6.15) as it is, although the entire system matrix is composed of well-conditioned submatrices, the convergence of its iterative solution can be slow because the submatrices are very different in magnitude, and hence unbalanced. We therefore propose to use the following $\overline{\mathbf{P}}$ as a preconditioner to solve (6.15), thus

$$\overline{\mathbf{P}} = \begin{bmatrix} \overline{\mathbf{V}}_{0a}^T(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_0 & \overline{\mathbf{V}}_{0a}^T(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma) \\ & \overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma + \Delta t^2\overline{\mathbf{L}} \end{bmatrix} \tag{6.16}$$

Using the above, the solution of (6.15) is found to converge in a very small number of iterations. This is because the off-diagonal block is much smaller than the diagonal one in the second block row of (6.15) in an integrated circuit layout.

In (6.16), in order to solve $\overline{\mathbf{V}}_{0a}^T(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_0$ fast, we transform it to the solution of two Laplacians as follows. We expand the static component $\overline{\mathbf{V}}_0\mathbf{y}_0$ into

$$\overline{\mathbf{V}}_0\mathbf{y}_0 = \overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0c}\mathbf{y}_{0c}, \tag{6.17}$$

where $\overline{\mathbf{V}}_{0d}$ are the $\overline{\mathbf{V}}_0$ columns associated with the dielectric nodes, and $\overline{\mathbf{V}}_{0c}$ are those associated with conductor nodes. Then the $\overline{\mathbf{V}}_{0a}^T(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_0$ subsystem of equations can be rewritten as

$$\begin{aligned} &\begin{bmatrix} \overline{\mathbf{V}}_{0da}^T(\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0d} & \overline{\mathbf{V}}_{0da}^T(\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0c} \\ \overline{\mathbf{V}}_{0ca}^T(\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0d} & \overline{\mathbf{V}}_{0ca}^T(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_{0c} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{0d} \\ \mathbf{y}_{0c} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{b}_{0d} \\ \mathbf{b}_{0c} \end{bmatrix}. \end{aligned} \tag{6.18}$$

where $\overline{\mathbf{V}}_{0d}$, $\overline{\mathbf{V}}_{0da}$, $\overline{\mathbf{V}}_{0c}$, $\overline{\mathbf{V}}_{0ca}$ are all normalized, and $\overline{\mathbf{V}}_{0da}$ and $\overline{\mathbf{V}}_{0ca}$ are left nullspace vectors. The above can be written in short as

$$\begin{bmatrix} \overline{\mathbf{M}}_{dd} & \overline{\mathbf{M}}_{dc} \\ \overline{\mathbf{M}}_{cd} & \overline{\mathbf{M}}_{cc} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{0d} \\ \mathbf{y}_{0c} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{0d} \\ \mathbf{b}_{0c} \end{bmatrix}, \tag{6.19}$$

where $\overline{\mathbf{M}}_{dd} = \overline{\mathbf{V}}^T_{0da}(\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0d}$, $\overline{\mathbf{M}}_{dc} = \overline{\mathbf{V}}^T_{0da}(\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0c}$, $\overline{\mathbf{M}}_{cd} = \overline{\mathbf{V}}^T_{0ca}(\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0d}$ and $\overline{\mathbf{M}}_{cc} = \overline{\mathbf{V}}^T_{0ca}(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_{0c}$. Notice that $\overline{\mathbf{M}}_{cc}$ is inside conductors, and hence the displacement current is much smaller than the conduction current. Then the term $(\overline{\mathbf{D}}_\epsilon)$ can be ignored. Therefore, we have $\overline{\mathbf{M}}_{cc} \approx \overline{\mathbf{V}}^T_{0ca}(\Delta t\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_{0c}$.

To solve (6.19), first, we solve the Schur complement of the $\overline{\mathbf{M}}_{dd}$, which is

$$(\overline{\mathbf{M}}_{dd} - \overline{\mathbf{M}}_{dc}\overline{\mathbf{M}}_{cc}^{-1}\overline{\mathbf{M}}_{cd})\mathbf{y}_{0d} = \mathbf{b}_{0d} - \overline{\mathbf{M}}_{dc}\overline{\mathbf{M}}_{cc}^{-1}\mathbf{b}_{0c}. \tag{6.20}$$

Because in this matrix, $\mathbf{D}_\sigma$ dominates in magnitude and the other matrix blocks' magnitude is much smaller, we have $\|\overline{\mathbf{M}}_{dd}\| >> \|\overline{\mathbf{M}}_{dc}\overline{\mathbf{M}}_{cc}^{-1}\overline{\mathbf{M}}_{cd}\|$. Therefore, (6.20) can be rewritten accurately as

$$\overline{\mathbf{M}}_{dd}\mathbf{y}_{0d} = \mathbf{b}_{0d} - \overline{\mathbf{M}}_{dc}\overline{\mathbf{M}}_{cc}^{-1}\mathbf{b}_{0c}. \tag{6.21}$$

After $\mathbf{y}_{0d}$ is obtained, substitute it into the first equation, and $\mathbf{y}_{0c}$ can be computed as

$$\overline{\mathbf{M}}_{cc}\mathbf{y}_{0c} = \mathbf{b}_{0c} - \overline{\mathbf{M}}_{cd}\mathbf{y}_{0d}. \tag{6.22}$$

Notice that the two diagonal blocks, denoted by $\overline{\mathbf{M}}_{dd}$ and $\overline{\mathbf{M}}_{cc}$, are nothing but discretized $\nabla \cdot \epsilon\nabla$ and $\nabla \cdot \sigma\nabla$ in the grid. Hence, (6.21) and (6.22) can be rapidly solved by fast Laplacian solvers such as a multigrid iterative method, which can converge in a constant number of steps for Laplace operators.

The preconditioner shown in (6.16) can further be simplified to the following upper triangular matrix

$$\overline{\mathbf{P}} = \begin{bmatrix} \overline{\mathbf{V}}^T_{0da}(\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0d} & \overline{\mathbf{V}}^T_{0da}(\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0c} & \overline{\mathbf{V}}^T_{0da}(\overline{\mathbf{D}}_\epsilon) \\ & \overline{\mathbf{V}}^T_{0ca}(\Delta t\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_{0c} & \overline{\mathbf{V}}^T_{0ca}(\overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma) \\ & & \overline{\mathbf{D}}_\epsilon + \Delta t\overline{\mathbf{D}}_\sigma + \Delta t^2\overline{\mathbf{L}} \end{bmatrix} \tag{6.23}$$

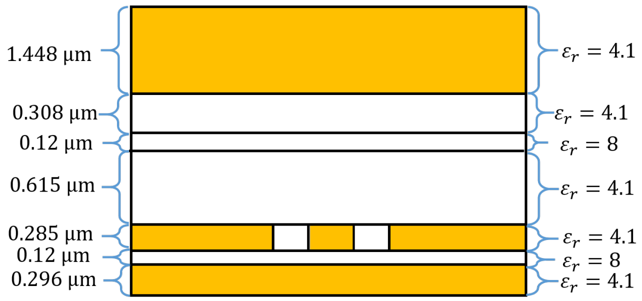


Fig. 6.1.: Structure of a test-chip interconnect.

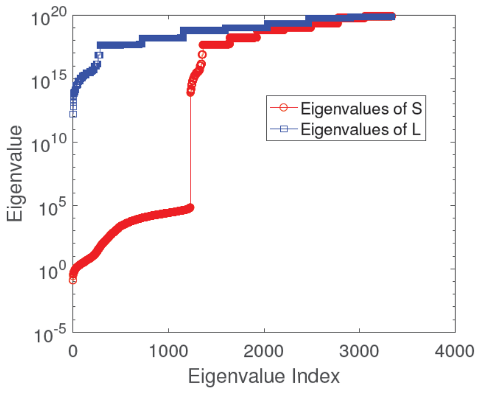


Fig. 6.2.: Eigenvalues of $\overline{\mathbf{L}}$ as compared to those of $\overline{\mathbf{S}}$.

It is effective to solve (6.15) because the blocks omitted are orders of magnitude smaller than the matrix blocks residing in the same block row of equations in an integrated circuit layout. Furthermore, the solution of the above preconditioner can be computed efficiently since via a backward substitution procedure, it only requires solving the three diagonal blocks, and each of which is a Laplacian, and hence can be solved fast.

6.3 Simulation Results

6.3.1 Test-chip Interconnect

A test-chip interconnect is simulated, whose structure is shown in Fig. 6.1. The yellow regions are conductors, the conductivity of which is 5.8×10^7 S/m. The dimensions along the x-, y- and z-directions are 300, 2000, and 3.19 μm respectively. We use a non-uniform grid to discretize the structure. The current source is launched from the bottom ground plane to the middle conductor. The value of $\frac{\|\overline{\mathbf{SV}}_h - (\overline{\mathbf{LV}}_h)\|}{\|\overline{\mathbf{SV}}_h\|}$ is evaluated and found to be 3.3239×10^{-10}. This verifies (6.9). Another check we did is to evaluate $\frac{\|\overline{\mathbf{S}} - (-\overline{\mathbf{V}}_0 \overline{\mathbf{V}}_{0a}^T/\mu + \overline{\mathbf{L}})\|}{\|\overline{\mathbf{S}}\|}$, which is found to be 3.2806×10^{-16}, and hence validating our theoretical understanding of the $\overline{\mathbf{S}}$ matrix. In Fig. 6.2, we also plot the eigenvalues of $\overline{\mathbf{S}}$, and $\overline{\mathbf{L}}$ respectively. It is obvious that $\overline{\mathbf{S}}$ is ill-conditioned while $\overline{\mathbf{L}}$ is not. All those eigenvalues whose magnitude is 10^5 or smaller are actually zero eigenvalues of $\overline{\mathbf{S}}$. They cannot be computed as exact zeros because of machine precision: their values are about 15 or 16 orders of magnitude smaller than the largest one. The condition number of $\overline{\mathbf{S}}$ is found to be 4.6737×10^{27}, whereas that of $\overline{\mathbf{L}}$ is only 2.5×10^8.

If we do not apply the preconditioner, we find to solve (6.15), GMRES takes 3000 iterations to converge at a relative residual of 0.0011, which is very inefficient. After applying the preconditioner of (6.16) and also using its diagonal blocks only, the solution converges in 14 steps with a relative residual smaller than 10^{-5}. This shows the effectiveness of the preconditioner. Meanwhile, the preconditioner itself has a fast solution because the matrix $(\overline{\mathbf{D}}_\epsilon + \Delta t \overline{\mathbf{D}}_\sigma + \Delta t^2 \overline{\mathbf{L}})$ is Laplacian based, and it can be solved within a small iteration number using the multigrid method. The same is true to the first diagonal block of (6.16). From our numerical experiments, solving the original equation $(\overline{\mathbf{D}}_\epsilon + \Delta t \overline{\mathbf{D}}_\sigma + \Delta t^2 \overline{\mathbf{S}})$ takes 400 steps, while $(\overline{\mathbf{D}}_\epsilon + \Delta t \overline{\mathbf{D}}_\sigma + \Delta t^2 \overline{\mathbf{L}})$ only takes 1 step to reach a relative residual of 10^{-15}. When solving (6.19), it only takes 3 steps to achieve a relative residual of 1e-5 to solve $\overline{\mathbf{M}}_{dd}$. And solving $\overline{\mathbf{M}}_{cc}$ takes 9 steps to achieve a relative residual of 1e-5. The comparison of the time domain

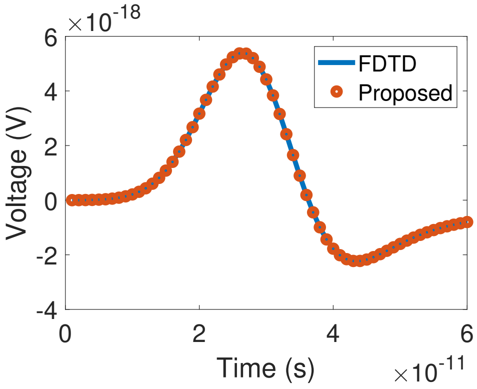


Fig. 6.3.: Comparison between the proposed method and the traditional FDTD in simulating an on-chip interconnect.

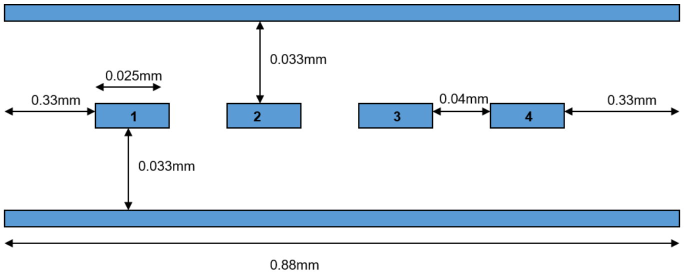


Fig. 6.4.: Structure of an IBM plasma package interconnect.

voltage simulated using the traditional FDTD and the proposed method is shown in Fig. 6.3, which reveals good agreement. We use a Gaussian derivative as the current source with $\tau = 10^{-11}s$. We choose $\Delta t = 10^{-12}$ s, which is solely determined by accuracy, while a conventional explicit marching must use a time step as small as 10^{-16} s in this example.

6.3.2 IBM Plasma Interconnect

The second example is an IBM Plasma Package interconnect, whose cross section is shown in Fig. 6.4. The blue regions are conductors, with a conductivity of 5.8×10^7 S/m. The dimension along x-, y- and z-direction is 0.88, 10 and 0.165 mm respectively.

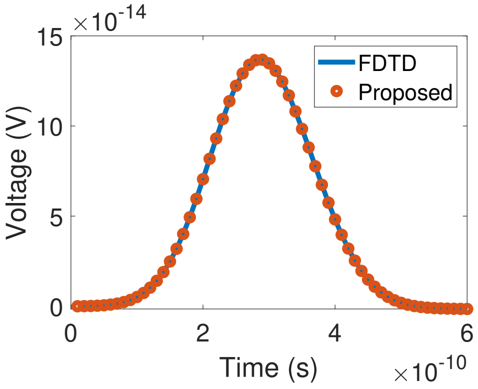
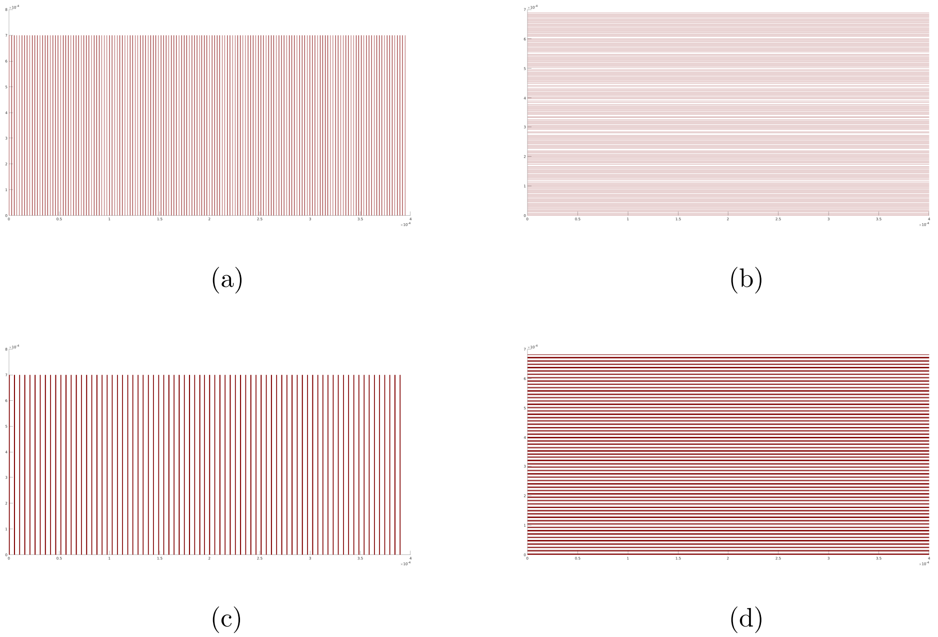


Fig. 6.5.: Comparison between the proposed method and the traditional FDTD in simulating a package interconnect.


(a)

(b)

(c)

(d)

Fig. 6.6.: Illustration of an on-chip power grid example. (a) M5 layer. (b) M6 layer. (c) M7 layer. (d) M8 layer.

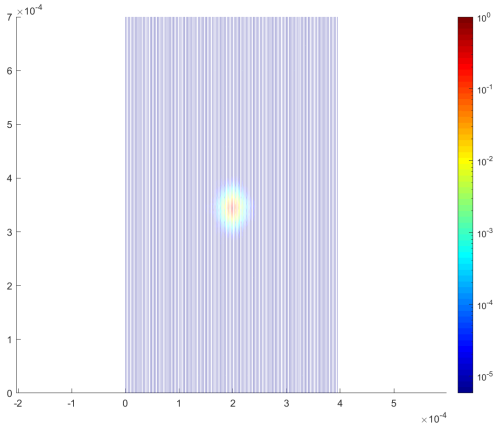


Fig. 6.7.: Simulated voltage distribution of an on-chip power grid at 10 GHz in M5 layer.

The current source is injected from the bottom plane to the conductor 1. Firstly, we calculated $\frac{\|\overline{\mathbf{S}}\overline{\mathbf{V}}_h - \overline{\mathbf{L}}\overline{\mathbf{V}}_h\|}{\|\overline{\mathbf{S}}\overline{\mathbf{V}}_h\|}$ and found it to be 1.6572×10^{-12}, which again verifies (6.9). The $\frac{\|\overline{\mathbf{S}} - (-\overline{\mathbf{V}}_0 \overline{\mathbf{V}}_{0a}^T/\mu + \overline{\mathbf{L}})\|}{\|\overline{\mathbf{S}}\|}$ is found to be 2.8933×10^{-16}.

When solving the problem in time domain, we use GMRES with diagonal part of $\overline{\mathbf{P}}$ being the preconditioner. The solution is shown to converge in 14 steps, achieving a relative residual of 10^{-7}. In the preconditioner, we solve $(\overline{\mathbf{D}}_\epsilon + \Delta t \overline{\mathbf{D}}_\sigma + \Delta t^2 \overline{\mathbf{L}})$ instead of $(\overline{\mathbf{D}}_\epsilon + \Delta t \overline{\mathbf{D}}_\sigma + \Delta t^2 \overline{\mathbf{S}})$. The multigrid method solves $(\overline{\mathbf{D}}_\epsilon + \Delta t \overline{\mathbf{D}}_\sigma + \Delta t^2 \overline{\mathbf{L}})$ with only 1 step achieving accuracy of 10^{-11}. In contrast, if $(\overline{\mathbf{D}}_\epsilon + \Delta t \overline{\mathbf{D}}_\sigma + \Delta t^2 \overline{\mathbf{S}})$ is solved, it takes 500 steps to achieve an accuracy of 0.001. When solving (6.19), $\overline{\mathbf{M}}_{dd}$ only takes 3 steps to reach 10^{-5} accuracy and $\overline{\mathbf{M}}_{cc}$ takes no greater than 17 steps to reach 10^{-5}. The time domain voltage drops between conductor 1 and the ground plane from the proposed method and the FDTD are shown in Fig. 6.5. Excellent agreement is observed. A Gaussian derivative pulse with $\tau = 10^{-10}$ s is used here as the current source. Based on the sampling accuracy, $\Delta t = 10^{-11}$ s is chosen as the time step, which is independent of the space step.

6.3.3 On-Chip Power Grid

The third example is a large-scale on-chip power grid. It consists of 4 metal layers and 3 dielectric layers, which is illustrated in Fig. 6.6. The dimension along x-, y- and z-direction is 400 μm, 700 μm, and 2.838 μm respectively. The power and ground rails are interleaved, and vias are located at the intersection of like rails in adjacent metal layers. The discretization of the structure results in 4,397,222 unknowns. A current source is injected from a ground rail to a power one in the middle of the grid at the bottom metal layer, which is a Gaussian derivative pulse with $\tau = 10^{-9}$ s. The proposed method uses a large time step of $\Delta t = 10^{-10}$ s to perform time marching, which is solely determined from the input spectrum instead of space step. The voltage distribution at 10 GHz across the power grid on metal 6 layer is shown in Fig. 6.7. The voltage sampled at the input terminal is plotted in Fig. 6.8, and

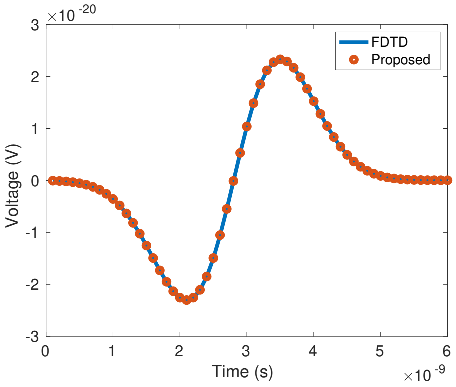


Fig. 6.8.: Comparison between the proposed method and the traditional FDTD in simulating an on-chip power grid.

compared with the FDTD result. Good agreement is observed. This problem is solved using an upper triangular preconditioner shown in (6.23). It only takes 6 steps to reach a relative residual of 0.007. In the preconditioner, solving $(\overline{\mathbf{D}_\epsilon} + \Delta t\overline{\sigma} + \Delta t^2\overline{\mathbf{L}})$ takes 3 step to achieve an accuracy of 10^{-5}. And solving $\overline{\mathbf{M}}_{dd}$ takes 1 steps to reach 10^{-5} accuracy, while for $\overline{\mathbf{M}}_{cc}$, it takes no greater than 15 steps to achieve the same accuracy. Comparing the time consumption, running the traditional implicit FDTD takes about 56820 s, while the proposed method takes 8989.2 s, which is more efficient.

6.4 Conclusion

In this work, a fast time-domain method is developed to accelerate the computation of fullwave solution of integrated circuit layouts. In this method, we expand the field solution into a gradient field and a full-wave component. The former is in the nullspace of the curl-curl operator, which is analytically found in this work. The latter's governing system matrix can be changed to a Laplace counterpart rigorously, and also without any computation. Since the Laplace operator is well-conditioned and positive definite, we are able to significantly accelerate the convergence of an implicit FDTD method for analyzing IC layouts. Numerical experiments have demonstrated its accuracy and efficiency. The proposed method is equally applicable to other time domain methods.

7. FAST METHOD FOR ACCELERATING CONVERGENCE IN ITERATIVE SOLUTION OF FREQUENCY-DOMAIN PARTIAL DIFFERENTIAL EQUATION METHODS

7.1 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 is involved, the system matrix is complex valued, which is even more difficult to be solved. The traditional way to solve this problem is to apply direct solvers, using Gaussian Elimination or LU factorization. This kind of method has complexity as $O(N^3)$, which is especially expensive when the size of the problem is large. So for large-scale electromagnetic problems this type of direct solvers is not suitable.

Considering efficient usage of time and memory, iterative methods are another appropriate way to solve to the complex matrix resulted from the partial differential equations. Existing techniques for expediting iterative solutions are mainly based on finding a good preconditioner for the full-wave system matrix. For example, in [31], a diagonal matrix is used as the preconditioner to improve the iterative performance of solving the Electric Field Integral Equation (EFIE); in [7], the symmetric successive overrelaxation (SSOR) is found to be a very effective preconditioner with Conjugate Gradient (CG) method to solve the full-wave matrix from the Finite-Element Method (FEM); [32] proposed a triangular matrix preconditioners to solve both static and harmonic problems resulted from the partial differential equations; [33] developed a preconditioner from an approximate system to solve the finite element-boundary integral system to achieve fast convergence; [34] applies preconditioned generalized minimal residual method (GMRES) to fast solve the differential equations for the planar circuits. Among the existing iterative methods, there are pros and cons. For the diagonal, block diagonal, SSOR and some other preconditioners, they are easy to construct and can have fast convergence for some problems. But, the performance is problem dependent and not always reliable. Another way is to use approximate inverse preconditioners or incomplete factorization preconditioners. This type of preconditioners is robust even for poorly conditioned system and can have faster convergence. The shortcomings are that it requires additional computational cost and sometimes even breaks down during the construction of the preconditioner. Other preconditioners suffer from similar performance and cost trade-off. Instead of applying a preconditioner, we propose to replace the original singular matrix with its Laplacian counterpart when operating on the full-wave components. This Laplacian matrix can be built analytically without computational cost. Due to its good property, the iterative solution of this modified system converges fast irrespective of the matrix size, which achieves high performance and trivial cost.

Our approach is based on the following finding. The curl-curl operator (stiffness matrix) in the frequency-domain system can be rigorously decomposed into a gradient divergence operator and a Laplacian operator, both of which can be constructed from the mesh information without any need for computation. The gradient divergence operator vanishes when operating on the full-wave (non-gradient) component of the field solution, and hence the singular curl-curl operator can be replaced by the Laplacian operator without any approximation. This Laplacian operator shares the same non-zero eigenvalues and eigenvectors of the curl-curl operator. Since the Laplacian operator is positive definite and well-conditioned, the resultant iterative solution has guaranteed convergence, and also it can converge in a small number of steps. Based on the aforementioned idea, we develop a fast iterative solution to solve large-scale integrated circuits in frequency domain. In such a problem, the field solution is dominated by both a static component, which is a gradient field, and a full-wave component whose curl is not zero. The conductor cannot be treated as perfect conductor, and the field inside conductors need to be simulated. Both difficulties are overcome in this work. We use the finite difference method (FDM) to demonstrate the essential idea of the proposed method, but the same method is applicable to other PDE methods such as the finite element method (FEM).

In this section, the background of this problem is introduced. Then the analytical and theoretical way to generate the nullspace of the system matrix is illustrated. The relationship between the system matrix and the Laplacian matrix is derived. Moreover, the algorithm to accelerate the convergence of the partial differential equation is demonstrated. At the end, numerical experiments on the full-wave analysis of large-scale layouts validate the accuracy and efficiency of the proposed method.

7.2 Background

The Maxwell's equation determining the physical phenomenon of a layout of analog and mix-signal ICs, packages, and boards is expressed as:

$$\begin{aligned} \nabla \times \mathbf{E} &= -\mu_r \mu_0 \frac{\partial \mathbf{H}}{\partial t} \\ \nabla \times \mathbf{H} &= \sigma \mathbf{E} + \epsilon \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J}, \end{aligned} \tag{7.1}$$

where μ_0 is free-space permeability, μ_r is relative permeability, ϵ is the permittivity, σ is conductivity, and $\mathbf{J}$ is the current source.

Due to the geometrical properties of the physical layout, it is natural to use the Cartisian grid as the mesh to discretize the entire area. In addition, the Cartisian grid gets rid of the computational cost by the irregular mesh. From that, a frequency-domain PDE based solution of Maxwell's equations, in general, results in the following linear system of equations:

$$-\omega^2 \overline{\mathbf{D}}_\epsilon \mathbf{e} + j\omega \overline{\mathbf{D}}_\sigma \mathbf{e} + \overline{\mathbf{S}} \mathbf{e} = -j\omega \mathbf{J}, \tag{7.2}$$

where ω is the angular frequency, $\mathbf{e}$ denotes a vector of electric field unknowns, $\overline{\mathbf{D}}_\epsilon$-term is associated with permittivity, $\overline{\mathbf{D}}_\sigma$-term is with conductivity, $\overline{\mathbf{S}}$ represents a discretized $\nabla \times \mu^{-1} \nabla \times$ operator, and $\mathbf{J}$ denotes a current source vector. The $\overline{\mathbf{S}}$'s smallest eigenvalue is zero, and its largest one is inversely proportional to the square of the smallest mesh size. This renders an iterative solution of (7.2) difficult to converge, especially when the problem is multiscaled in both geometry and frequency.

From the new single-grid patch-based formulation of the FDTD method in [48], the $\overline{\mathbf{S}}$ matrix is generated by the summation of rank-1 matrices of patches in the mesh. The formulation of $\overline{\mathbf{S}}$ is

$$\overline{\mathbf{S}} = \sum_{i=1,\ldots,N_p} \frac{1}{\mu_i} \overline{\mathbf{S}}_h^{(i)} \overline{\mathbf{S}}_e^{(i)}, \tag{7.3}$$

where N_p is the number of patches, μ_i is the permeability of patch i. For each patch, $\overline{\mathbf{S}}_e^{(i)}$ can be written as

$$\overline{\mathbf{S}}_e^{(i)} = \begin{bmatrix} -\frac{1}{L_i} & \frac{1}{L_i} & \frac{1}{W_i} & -\frac{1}{W_i} \end{bmatrix} \oplus zeros(1, N_e), \tag{7.4}$$

where L_i and W_i are the length and width of patch i, and $\oplus$ denotes the extended addition based on the global indexes of the four local $\mathbf{E}$ unknowns of patch i. For a uniform grid, the formula of $\overline{\mathbf{S}}_h$ is just the transpose of $\overline{\mathbf{S}}_e$, which is

$$\overline{\mathbf{S}}_h = (\overline{\mathbf{S}}_e)^T. \tag{7.5}$$

For a non-uniform grid, this $\overline{\mathbf{S}}_h$ should be replaced by a more accurate expression, that is all the lengths should be the averaged length across two patches sharing the same $\mathbf{E}$. The formula of $\overline{\mathbf{S}}_h^{(i)}$ is changed to

$$\overline{\mathbf{S}}_h^{(i)} = \begin{bmatrix} -\frac{1}{L_{i1}^{avg}} & \frac{1}{L_{i2}^{avg}} & \frac{1}{W_{i1}^{avg}} & -\frac{1}{W_{i2}^{avg}} \end{bmatrix}^T \oplus zeros(N_e, 1), \tag{7.6}$$

where L_{i1}^{avg}, L_{i2}^{avg}, W_{i3}^{avg} and W_{i4}^{avg} are the averaged lengths across the patches sharing patch edges. (7.3) can also be rewritten as

$$\overline{\mathbf{S}} = \overline{\mathbf{S}}_h \overline{\mathbf{D}}_{1/\mu} \overline{\mathbf{S}}_e. \tag{7.7}$$

where $\overline{\mathbf{S}}_h$ ith column is $\overline{\mathbf{S}}_h^{(i)}$, $\overline{\mathbf{S}}_e$ ith row is $\overline{\mathbf{S}}_e^{(i)}$ and $\overline{\mathbf{D}}_{1/\mu}$ is the diagonal matrix of μ^{-1}.

Let the $\overline{\mathbf{V}}_0$, $\overline{\mathbf{V}}_{0a}$ be the right and left nullspace of $\overline{\mathbf{S}}$ satisfying $\overline{\mathbf{S}}\overline{\mathbf{V}}_0 = 0$ and $\overline{\mathbf{V}}_{0a}^T \overline{\mathbf{S}} = 0$. It is a gradient field, which can be analytically generated from a mesh used to discretize the problem [49]. The method to generate the nullspace is illustrated in Section 5.3. We find that $\overline{\mathbf{S}}$ can be analytically decomposed into $\overline{\mathbf{S}} = (-\overline{\mathbf{V}}_0\overline{\mathbf{V}}_{0a}^T + \overline{\mathbf{L}})/\mu$ in a material having permeability μ, where $-\overline{\mathbf{V}}_0\overline{\mathbf{V}}_{0a}^T$ denotes the discretized form of $\nabla(\nabla\cdot\,)$ and $\overline{\mathbf{L}}$ denotes the discretized form of $-\nabla^2$. This decomposition helps to analyze the solution into two part and modify the original ill-conditioned matrix to a well-performed matrix when operating on the full-wave component, which makes it much easier to solve.

7.3 The relationship between the $\overline{\mathrm{S}}$ matrix and the Laplacian matrix

From the vector identity $\nabla \times (\nabla \times \mathbf{A}) = \nabla(\nabla \cdot \mathbf{A}) - (\nabla \cdot \nabla)\mathbf{A}$, we find that $\overline{\mathbf{S}}$ can be analytically decomposed into $\overline{\mathbf{S}} = (-\overline{\mathbf{V}}_0\overline{\mathbf{V}}_{0a}^T + \overline{\mathbf{L}})/\mu$ in a material having permeability μ, where $-\overline{\mathbf{V}}_0\overline{\mathbf{V}}_{0a}^T$ denotes the discretized form of $\nabla(\nabla\cdot\,)$ and $\overline{\mathbf{L}}$ denotes the discretized form of $-\nabla^2$. Therefore, we can have

$$\overline{\mathbf{L}} = \mu\overline{\mathbf{S}} + \overline{\mathbf{V}}_0\overline{\mathbf{V}}_{0a}^T. \tag{7.8}$$

Let $\overline{\mathbf{V}}_h$ be the eigenspace of $\overline{\mathbf{S}}$ formed by eigenvectors of nonzero eigenvalues $\overline{\Lambda}_h$ and they satisfy $\overline{\mathbf{S}}\overline{\mathbf{V}}_h = \overline{\mathbf{V}}_h\overline{\Lambda}_h$. Since $\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{S}} = 0$, we have $\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{S}}\overline{\mathbf{V}}_h = \overline{\mathbf{V}}_{0a}^T\overline{\mathbf{V}}_h\overline{\Lambda}_h = 0$. Thus, $\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{V}}_h = 0$ is satisfied. Multiply $\overline{\mathbf{V}}_h$ to both sides of (7.8), we obtain

$$\overline{\mathbf{S}}\overline{\mathbf{V}}_h = \overline{\mathbf{L}}\overline{\mathbf{V}}_h/\mu. \tag{7.9}$$

This means when operating on $\overline{\mathbf{S}}$'s eigenmodes corresponding to nonzero eigenvalues, i.e., full-wave modes, we can replace $\overline{\mathbf{S}}$ by $\overline{\mathbf{L}}$. In an FDM, the $\overline{\mathbf{L}}$ can be readily constructed. One way to generate the matrix $\overline{\mathbf{L}}$ is as the following:

- For edge i, find all the surrounding edges j.
- For each edge in j, if this edge j_m is on the same line with edge i, at the entry (i, j_m) put $-1/l_i \times 1/l_{i,j_m}^{avg}$, where l_i is the length of edge i, and l_{i,j_m}^{avg} is the averaged length of edge i and edge j_m.

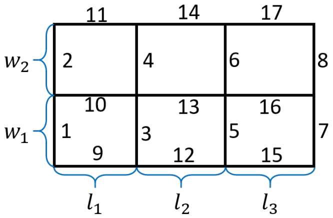


Fig. 7.1.: Illustration of the generation of the Laplacian matrix.

- For each edge in j, if this edge j_m is not on the same line with edge i, at the entry (i, j_m) put $-1/l_{i,j_m} \times 1/l_i^{avg}$, where l_{i,j_m} is the distance between the centers of edge i and edge j_m, and l_i^{avg} is the averaged length of of the two patches sharing edge i, where the patches are on the plane of edge i and j_m.

- For the entry (i,i), if this edge j_m is not on the same line with edge i, add $1/l_{i,j_m} \times 1/l_i^{avg}$ to the entry (i,i); else, add $1/l_i \times 1/l_{i,j_m}^{avg}$ to the entry (i,i) for both ends of edge i.

To illustrate this, take a simple mesh in Fig. 7.1 as an example. In the figure, the number on the edges is the global edge number. l_1, l_2, l_3 are the lengths of the cells along x axis and w_1, w_2 are the lengths of the cells along y axis. Take the edge 13 as an example, the edges around it are edge 10, edge 12, edge 14, edge 16. So on the entry $(13, 10)$ we have $-\frac{2}{l_1+l_2} \cdot \frac{1}{l_2}$; $(13, 12)$ is $-\frac{1}{w_1} \cdot \frac{2}{w_1+w_2}$; $(13, 14)$ is $-\frac{1}{w_2} \cdot \frac{2}{w_1+w_2}$; $(13, 16)$ is $-\frac{2}{l_2+l_3} \cdot \frac{1}{l_2}$; $(13, 13)$ is $\frac{2}{l_1+l_2} \cdot \frac{1}{l_2} + \frac{1}{w_1} \cdot \frac{2}{w_1+w_2} + \frac{1}{w_2} \cdot \frac{2}{w_1+w_2} + \frac{2}{l_2+l_3} \cdot \frac{1}{l_2}$. After visiting all the surrounding edges, the final $\overline{\mathbf{L}}$ is generated.

Another way to generate $\overline{\mathbf{L}}$ is to use (7.8). From (7.7), $\overline{\mathbf{S}}$ is constructed analytically. The nullspace vectors $\overline{\mathbf{V}}_0$, $\overline{\mathbf{V}}_{0a}$ are also known from Section 5.3. Therefore, the Laplacian matrix $\overline{\mathbf{L}}$ is ready to be obtained.

In an FEM, the $\overline{\mathbf{L}}$ can be obtained from $\overline{\mathbf{S}} + \overline{\mathbf{V}}_0\overline{\mathbf{V}}_0^T/\mu$. If different permeabilities are involved, $\overline{\mathbf{S}} = (-\overline{\mathbf{V}}_0\overline{\mathbf{V}}_{0a}^T + \overline{\mathbf{L}})/\mu$ can be modified to suit the scenario.

7.4 Proposed algorithm to accelerate convergence in iterative solution of the frequency-domain partial differential equation

Based on the above finding, instead of solving (7.2) directly, we separate the static and full-wave component of $\mathbf{e}$ as $\mathbf{e} = \overline{\mathbf{V}}_0\mathbf{y}_0 + \mathbf{e}_h$, and transform (7.2) to the following to solve

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

where $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_{0a}$ are normalized for each vector, and we replace the lower right block $(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma + \overline{\mathbf{S}})$ by $(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma + \overline{\mathbf{L}})$ since that block operates on $\mathbf{e}_h$. $\mathbf{e}_h$ is in $\overline{\mathbf{V}}_h$ space and then we can transfer $\overline{\mathbf{S}}\mathbf{e}_h$ to $\overline{\mathbf{L}}\mathbf{e}_h$. Choosing normalized $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_{0a}$ is to make sure that the above matrix is balanced in magnitudes for all the blocks. Denoting the above in short by

$$\begin{bmatrix} \overline{\mathbf{A}}_{00} & \overline{\mathbf{A}}_{0h} \\ \overline{\mathbf{A}}_{h0} & \overline{\mathbf{A}}_{hh} \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{e}_h \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{V}}_{0a}^T(-j\omega\mathbf{J}) \\ -j\omega\mathbf{J} \end{bmatrix}. \tag{7.10}$$

The matrix $\overline{\mathbf{A}}_{hh}$ has good properties since the original ill-conditioned matrix $\overline{\mathbf{S}}$ is changed to $\overline{\mathbf{L}}$, which is Laplacian and can be solved fast by using either fast direct solver or advanced iterative methods, such as GMRES or multigrid method. To fully take advantage of the good property of $\overline{\mathbf{A}}_{hh}$, we solve it in the following way. We first use Schur complement to solve

$$(\overline{\mathbf{A}}_{hh} - \overline{\mathbf{A}}_{h0}\overline{\mathbf{A}}_{00}^{-1}\overline{\mathbf{A}}_{0h})\mathbf{e}_h = (-j\omega\mathbf{J}) - \overline{\mathbf{A}}_{h0}\overline{\mathbf{A}}_{00}^{-1}\overline{\mathbf{V}}_{0a}^T(-j\omega\mathbf{J}), \tag{7.11}$$

which can be accurately approximated as

$$(\overline{\mathbf{A}}_{hh})\mathbf{e}_h = (-j\omega\mathbf{J}) - \overline{\mathbf{A}}_{h0}\overline{\mathbf{A}}_{00}^{-1}\overline{\mathbf{V}}_{0a}^T(-j\omega\mathbf{J}), \tag{7.12}$$

due to the physical dimension of on-chip problems. With the scale of the on-chip problem, $\overline{\mathbf{A}}_{hh}$, which contains $\overline{\mathbf{L}}$ with norm $\frac{1}{\mu l^2}$, is usually much larger than other terms, which get rid of $\overline{\mathbf{S}}$ due to $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_{0a}$. Therefore, it is accurate to do the approximation in (7.12). Since $\overline{\mathbf{A}}_{hh}$ is made of a Laplacian, iterative solvers such as GMRES or multigrid can be used to solve it fast. After obtaining $\mathbf{e}_h$, we substitute it into the first equation of (7.10), and solve $\mathbf{y}_0$ from

$$\overline{\mathbf{A}}_{00}\mathbf{y}_0 = \overline{\mathbf{V}}_{0a}^T(-j\omega\mathbf{J}) - \overline{\mathbf{A}}_{0h}\mathbf{e}_h. \tag{7.13}$$

In (7.13), $\overline{\mathbf{A}}_{00}$ can also be transformed to the solution of two Laplacians as follows. We expand the static component $\overline{\mathbf{V}}_0\mathbf{y}_0$ into $\overline{\mathbf{V}}_0\mathbf{y}_0 = \overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0c}\mathbf{y}_{0c}$, 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. The generation of $\overline{\mathbf{V}}_{0d}$, $\overline{\mathbf{V}}_{0c}$, $\overline{\mathbf{V}}_{0da}$, $\overline{\mathbf{V}}_{0ca}$ is illustrated in Section 5.3. Then $\overline{\mathbf{A}}_{00}$ can be rewritten as

$$\begin{aligned}&\begin{bmatrix} \overline{\mathbf{V}}_{0da}^T(-\omega^2\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0d} & \overline{\mathbf{V}}_{0da}^T(-\omega^2\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0c} \\ \overline{\mathbf{V}}_{0ca}^T(-\omega^2\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0d} & \overline{\mathbf{V}}_{0ca}^T(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_{0c} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{0d} \\ \mathbf{y}_{0c} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{b}_{0d} \\ \mathbf{b}_{0c} \end{bmatrix}.\end{aligned} \tag{7.14}$$

where $\overline{\mathbf{V}}_{0d}$, $\overline{\mathbf{V}}_{0da}$, $\overline{\mathbf{V}}_{0c}$, $\overline{\mathbf{V}}_{0ca}$ are all normalized. (7.14) can be written in short as

$$\begin{bmatrix} \overline{\mathbf{M}}_{dd} & \overline{\mathbf{M}}_{dc} \\ \overline{\mathbf{M}}_{cd} & \overline{\mathbf{M}}_{cc} \end{bmatrix}, \tag{7.15}$$

where $\overline{\mathbf{M}}_{dd} = \overline{\mathbf{V}}_{0da}^T(-\omega^2\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0d}$, $\overline{\mathbf{M}}_{dc} = \overline{\mathbf{V}}_{0da}^T(-\omega^2\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0c}$, $\overline{\mathbf{M}}_{cd} = \overline{\mathbf{V}}_{0ca}^T(-\omega^2\overline{\mathbf{D}}_\epsilon)\overline{\mathbf{V}}_{0d}$ and $\overline{\mathbf{M}}_{cc} = \overline{\mathbf{V}}_{0ca}^T(-\omega^2\overline{\mathbf{D}}_\epsilon + j\omega\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_{0c}$. Notice that $\overline{\mathbf{M}}_{cc}$ is inside conductors, and hence the displacement current is much smaller than the conduction current. Then the term $(-\omega^2\overline{\mathbf{D}}_\epsilon)$ can be ignored. Therefore, we have $\overline{\mathbf{M}}_{cc} \approx \overline{\mathbf{V}}_{0ca}^T(j\omega\overline{\mathbf{D}}_\sigma)\overline{\mathbf{V}}_{0c}$.

First, we solve the Schur complement of the $\overline{\mathbf{M}}_{cc}$, which is the following

$$(\overline{\mathbf{M}}_{cc} - \overline{\mathbf{M}}_{cd}\overline{\mathbf{M}}_{dd}^{-1}\overline{\mathbf{M}}_{dc})\mathbf{y}_{0c} = \mathbf{b}_{0c} - \overline{\mathbf{M}}_{cd}\overline{\mathbf{M}}_{dd}^{-1}\mathbf{b}_{0d}. \tag{7.16}$$

Because in this matrix, $\overline{\mathbf{D}}_\sigma$ dominates in magnitude and the other blocks' magnitudes are much smaller, we have $\|\overline{\mathbf{M}}_{cc}\| >> \|\overline{\mathbf{M}}_{cd}\overline{\mathbf{M}}_{dd}^{-1}\overline{\mathbf{M}}_{dc}\|$. Therefore, (7.16) can be rewritten accurately as

$$(\overline{\mathbf{M}}_{cc})\mathbf{y}_{0c} = \mathbf{b}_{0c} - \overline{\mathbf{M}}_{cd}\overline{\mathbf{M}}_{dd}^{-1}\mathbf{b}_{0d}. \tag{7.17}$$

After $\mathbf{y}_{0c}$ is obtained, substitute it into the first equation, and $\mathbf{y}_{0d}$ can be computed as

$$\overline{\mathbf{M}}_{dd}\mathbf{y}_{0d} = \mathbf{b}_{0d} - \overline{\mathbf{M}}_{dc}\mathbf{y}_{0c}. \tag{7.18}$$

Notice that the two diagonal blocks, denoted by $\overline{\mathbf{M}}_{dd}$ and $\overline{\mathbf{M}}_{cc}$, are nothing but discretized $\nabla \cdot \epsilon \nabla$ and $\nabla \cdot \sigma \nabla$ in the grid. Hence, the above (7.17) and (7.18) can be rapidly solved by solving $\overline{\mathbf{M}}_{dd}$ and $\overline{\mathbf{M}}_{cc}$, both of which are Laplacian matrices, with some advanced fast direct solvers or the iterative methods, such as multigrid iterative method.

After all these computation, the final solution is obtained by

$$\mathbf{e} = \overline{\mathbf{V}}_{0d}\mathbf{y}_{0d} + \overline{\mathbf{V}}_{0c}\mathbf{y}_{0c} + \mathbf{e}_h. \tag{7.19}$$

7.5 Further development of the proposed algorithm to solve general problems

From the previous section, if (7.11) is approximated to (7.12), the physical dimension is in the level of on-chip problems. This idea can also be used to solve other problems, such as the board, package and antenna problems, where $\overline{\mathbf{S}}$'s spectral radius is smaller due to larger size or the problem's electrical size is larger. In order to do that, we have a further development of this algorithm.

The problem we are solving is still the 2 by 2 system in (7.10). The first equation to be solved is the schur complement in (7.11). Instead of approximating this to (7.12), we solve

$$(\overline{\mathbf{D}} + \overline{\mathbf{L}} - \overline{\mathbf{DV}}_0(\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{DV}}_0)^{-1}\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}})\mathbf{u}_h = (\bar{\mathbf{I}} - \overline{\mathbf{DV}}_0(\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{DV}}_0)^{-1}\overline{\mathbf{V}}_{0a}^T)(-j\omega\mathbf{J}) \tag{7.20}$$

with GMRES or other iterative solvers because the matrix $(\overline{\mathbf{A}}_{hh} - \overline{\mathbf{A}}_{h0}\overline{\mathbf{A}}_{00}^{-1}\overline{\mathbf{A}}_{0h})$ is nothing but a numerical Laplacian matrix, with a Laplacian matrix $(\overline{\mathbf{D}} + \overline{\mathbf{L}})$ minus a term $\overline{\mathbf{DV}}_0(\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{DV}}_0)^{-1}\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}}$. By deducting the term $\overline{\mathbf{DV}}_0(\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{DV}}_0)^{-1}\overline{\mathbf{V}}_{0a}^T\overline{\mathbf{D}}$, the resultant matrix removes the solution space on $\overline{\mathbf{V}}_0$, making sure the result is in $\overline{\mathbf{V}}_h$ space. After applying GMRES or other iterative solvers to get $\mathbf{e}_h$, substitute it into (7.13) and we can get $\mathbf{y}_0$.

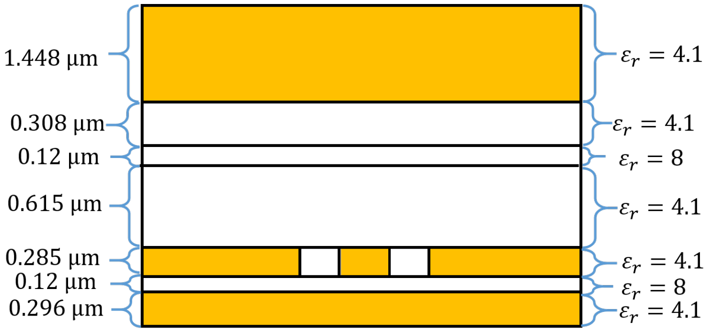


Fig. 7.2.: Structure of a test-chip interconnect.

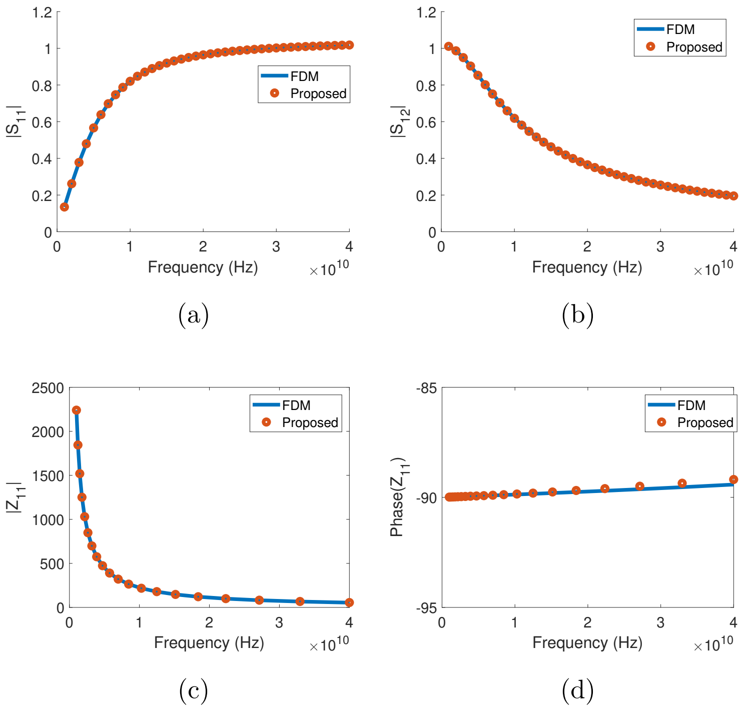
7.6 Numerical results

7.6.1 Test-chip Interconnect

The first example is a test-chip interconnect, which is shown in Fig. 7.2. The conductors have conductivity 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 $\frac{\|\overline{\mathbf{L}}\mathbf{V}_h - \overline{\mathbf{S}}\mathbf{V}_h\|}{\|\overline{\mathbf{S}}\mathbf{V}_h\|}$, 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 (7.2) in Fig. 7.3. They are shown to agree well with each other. In this example, we employ GMRES to solve (7.12) , which takes only 35 steps to reach a relative residual of 0.0077 at frequency 10 GHz. In contrast, if we solve (7.2) by GMRES directly, after 5001 steps, the relative residual is still at 0.253 and fails to converge. Fig. 7.4 shows the relative residual vs. the iteration number. As we can see, solving $\overline{\mathbf{A}}_L(=\overline{\mathbf{D}}+\overline{\mathbf{L}})$ is much more efficient than solving the original matrix $\overline{\mathbf{A}}(=\overline{\mathbf{D}}+\overline{\mathbf{S}})$. When solving $\overline{\mathbf{M}}_{dd}$ and $\overline{\mathbf{M}}_{cc}$, the multigrid solver only takes about 4 to 8 steps to reach a relative residual of 10^{-5}.

7.6.2 NAND Gate

The second example is a NAND gate, which is shown in Fig. 7.5. This structure has 17 layers. We discretize the structure with a non-uniform grid mesh yielding



(a)

(b)

(c)

(d)

Fig. 7.3.: (a) and (b): S-parameters of a test-chip interconnect; (c) and (d): Z-parameters of a NAND Gate.

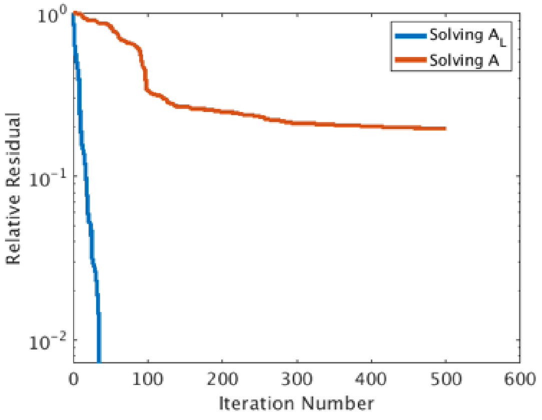


Fig. 7.4.: The relative residual vs. the iteration number when solving $\mathbf{A}$ and $\mathbf{A}_L$.

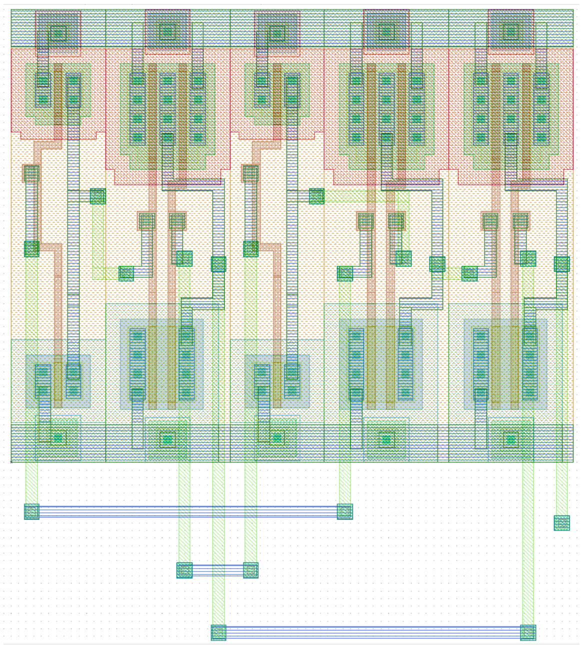


Fig. 7.5.: Layout of a NAND gate.

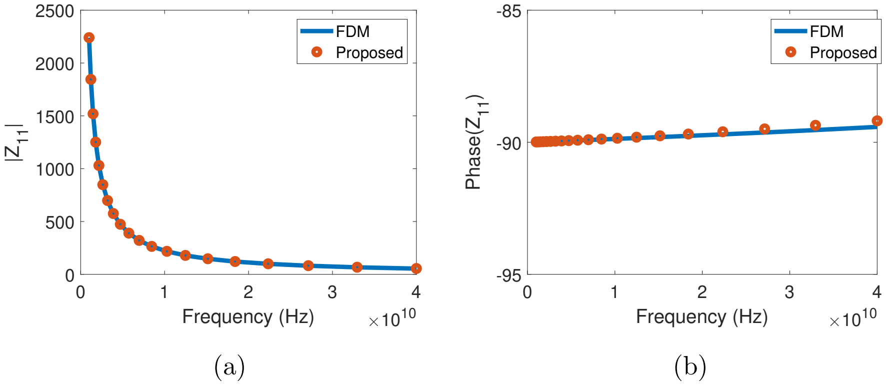


Fig. 7.6.: Z-parameters of a nand2 gate

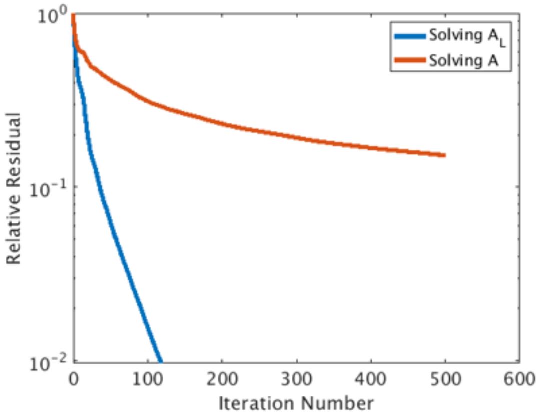


Fig. 7.7.: The relative residual vs. the iteration number when solving $\overline{\mathbf{A}}$ and $\overline{\mathbf{A}}_L$.

665,940 unknowns. The extracted Z parameters agree well with those from the brute-force solution of (7.2), as shown in Fig. 7.6. When solving $\overline{\mathbf{A}}_L$, GMRES takes 102 steps to reach a relative residual of 0.00989 at 10 GHz. If we solve the original matrix $\overline{\mathbf{A}}$ directly, it takes 5001 steps to reach a relative residual of 0.0793 without converging. The relationship between the relative residual and the iteration number for solving $\overline{\mathbf{A}}$ and $\overline{\mathbf{A}}_L$ is shown in Fig. 7.7. From this, we can see that solving $\overline{\mathbf{A}}_L$ is much more efficient. The $\overline{\mathbf{M}}_{dd}$ and $\overline{\mathbf{M}}_{cc}$ solution takes at most 43 steps to achieve a relative residual of 10^{-5}.

7.6.3 Single-ended Microstrip

The third example is a single-ended microstrip, which is shown in Fig. 7.8. It contains 3 layers. The bottom conductor is the ground plane. There are two GSG launchers on the two ends of the stripline, where the currents runs through. The length of the stripline is 14.854 mm. The S parameters are extracted from the far and near ends of the stripline. Due to the larger physical size of this problem, we apply GMRES to solve (7.20). We compare the results from the proposed method and the Finite Difference Method in Fig. 7.9. As can be seen, the accuracy of our results is guaranteed. As for the efficiency, at 10 GHz, GMRES takes 44 steps to relative

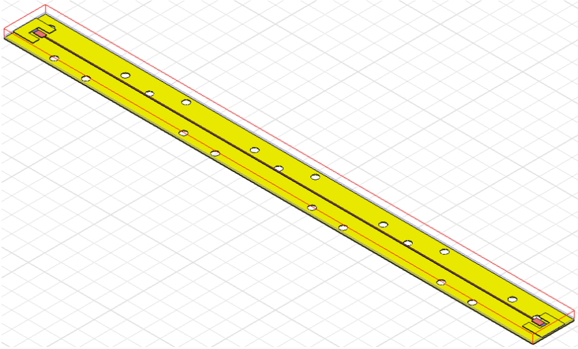
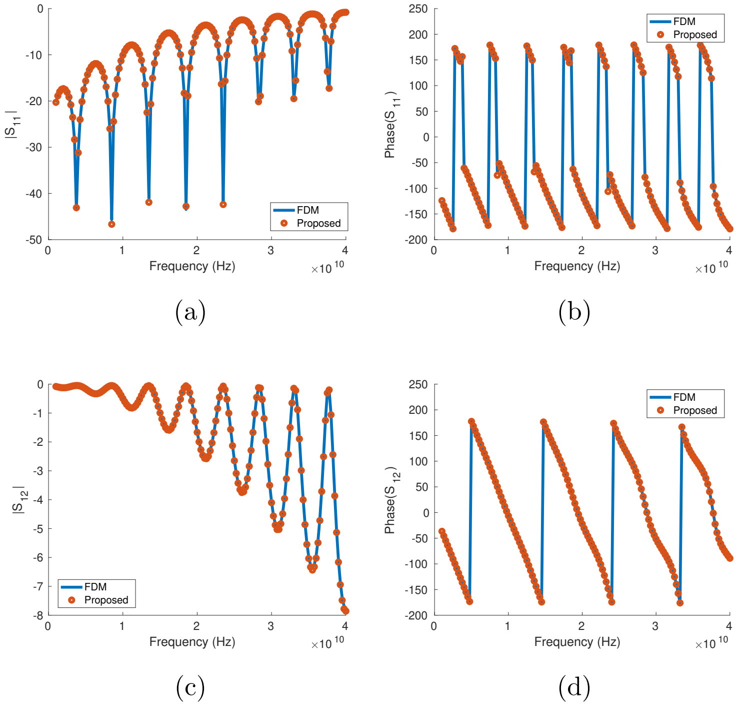


Fig. 7.8.: The structure of the single-ended microstrip example.



(a) (b)

(c) (d)

Fig. 7.9.: (a) and (b): S_{11}'s magnitude and phase; (c) and (d): S_{12}'s magnitude and phase.

residual 0.00073. The relationship between the relative residual and the iteration number is shown in Fig. 7.10 for solving $\overline{\mathbf{A}}$ and $\tilde{\overline{\mathbf{A}}}_L$, where $\tilde{\overline{\mathbf{A}}}_L$ denotes the matrix in (7.20).

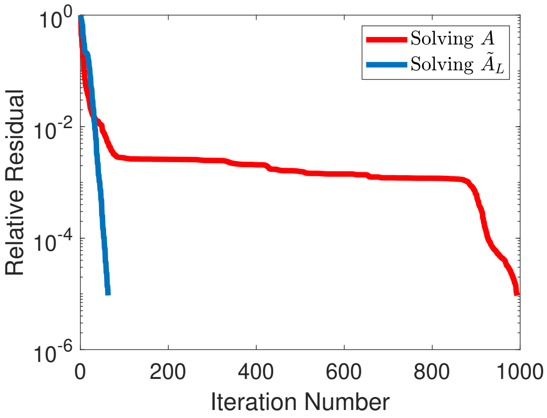


Fig. 7.10.: The relative residual vs. the iteration number for solving $\overline{\mathbf{A}}$ and $\overline{\tilde{\mathbf{A}}}_L$

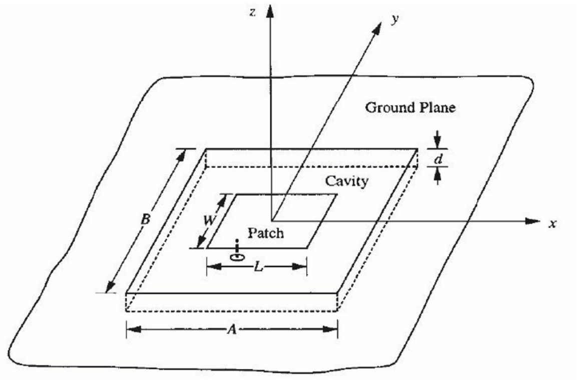


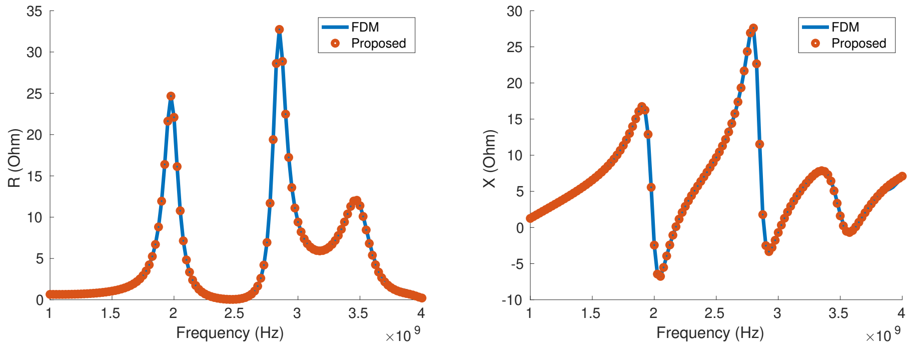
Fig. 7.11.: The structure of the antenna example.

7.6.4 Cavity-Backed Microstrip Patch Antenna

The fourth example is a path antenna example. The structure is shown in Fig. 7.11 [13]. The patch size is W = 3.4 cm, L = 5 cm and d = 0.08779 cm. There is a 50 Ω load at the position x_L=-2.2 cm and y_L=-1.5 cm. The current is injected at the position x_f=1.22 cm and y_f=0.85 cm. We extract the Z parameters and compare that with the Finite Difference Method. The comparison result is in Fig. 7.12. They can match very well. We also record the iteration number it takes to solve (7.11). At 1 GHz, it takes 53 steps to relative residual 0.00082. Fig. 7.13 shows the iteration number vs. the relative residual for solving $\overline{\mathbf{A}}$ and $\overline{\tilde{\mathbf{A}}}_L$ and we can see that solving $\overline{\tilde{\mathbf{A}}}_L$ converges much faster.

7.6.5 IBM plasma package

The fifth example is to simulate an IBM plasma package. The top view structure is shown Fig. 7.14. We use a non-uniform grid mesh to discretize the structure. The current is injected from the ground plane to the pin on the top right corner. S parameters are extracted and compared with the reference result from other reference tool, shown in Fig. 7.15, and they are very close to each other. The performance of the iterative solver is also checked. At 10 GHz, solving $\overline{\tilde{\mathbf{A}}}_L$ takes 274 steps to relative



(a) (b)

Fig. 7.12.: Z_{11}'s real and imaginary parts.

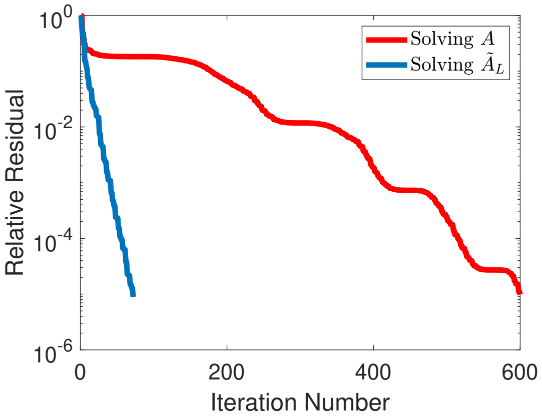


Fig. 7.13.: The relative residual vs. the iteration number for solving $\overline{\mathbf{A}}$ and $\overline{\tilde{\mathbf{A}}}_L$

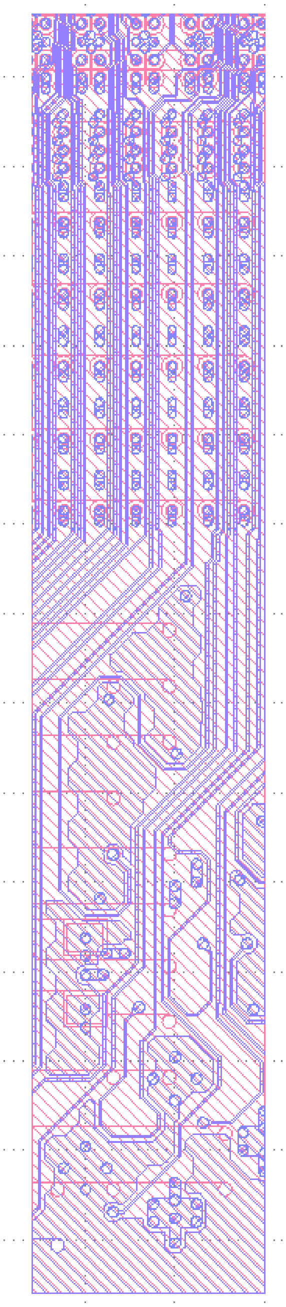


Fig. 7.14.: The structure of the plasma package.

residual 0.01. While, solving $\overline{\mathbf{A}}$ takes 1001 steps to the same relative residual 0.0644. This shows the efficiency of this method is better.

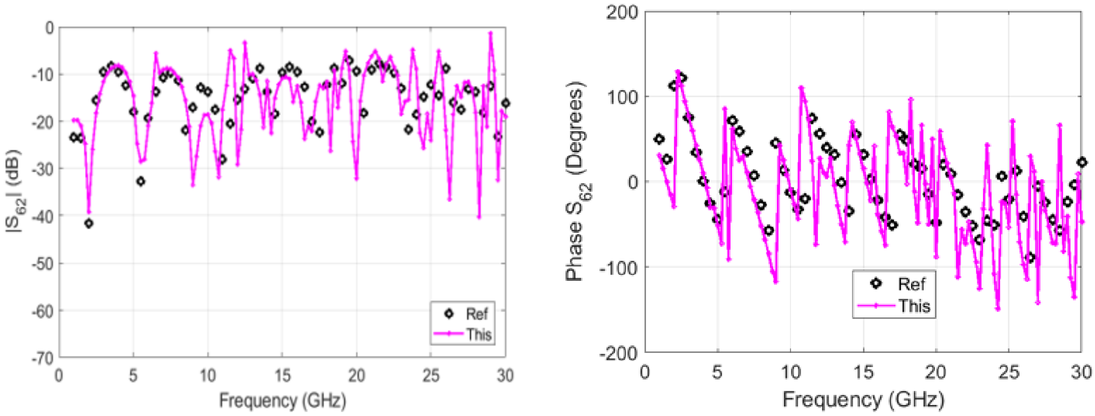


Fig. 7.15.: $\mathbf{S}_{62}$'s magnitude and phase for the plasma package.

8. FAST METHOD FOR ACCELERATING CONVERGENCE IN ITERATIVE SOLUTION OF FREQUENCY-DOMAIN PARTIAL DIFFERENTIAL EQUATION METHODS FOR PROBLEMS WITH PERFECT ELECTRICAL CONDUCTOR (PEC) OBJECTS

8.1 Introduction

In previous chapter, we propose a method to accelerate the iterative method by replacing the original matrix with a Laplacian matrix. Due to the good property of the Laplacian matrix, the iterative solution has a guaranteed convergence, and it also converges in a much smaller number of steps compared with a conventional solution of the original matrix. The method provides a very effective solution for on-chip, board, package and antenna problems. In this chapter, we show it can also be extended to the type of problems where conductors are treated as perfect conductors such as traditional high-frequency problems in which the skin depth is negligible. For this kind of problems, the challenge resides on how to find the Laplacian counterpart of the curl curl operator in an analytical way, since the original problem has holes created in a continuous domain. This chapter explains how the difficulty is overcome so that a Laplacian can be analytically constructed for problems with PEC objects.

8.2 Background

Similar to previous chapter, we solve the following the system matrix resulting from discretizing the Maxwell's Equations

$$-\omega^2\overline{\mathbf{D}}_\epsilon\mathbf{e} + j\omega\overline{\mathbf{D}}_\sigma\mathbf{e} + \overline{\mathbf{S}}\mathbf{e} = -j\omega\mathbf{J}, \tag{8.1}$$

where ω is the angular frequency, $\mathbf{e}$ denotes a vector of unknown electric field unknowns, $\overline{\mathbf{D}}_\epsilon$-term is associated with permittivity, $\overline{\mathbf{D}}_\sigma$-term is with conductivity, $\overline{\mathbf{S}}$ represents a discretized $\nabla \times \mu^{-1}\nabla\times$ operator, and $\mathbf{J}$ denotes a current source vector.

For a problem in which conductors are perfect, we can focus on its dielectric part. Therefore, (8.1) is reduced to

$$-\omega^2\overline{\mathbf{D}}_{\epsilon oo}\mathbf{e} + \overline{\mathbf{S}}_{oo}\mathbf{e} = -j\omega\mathbf{J}_o, \tag{8.2}$$

where $\overline{\mathbf{D}}_{\epsilon oo}$ is the diagonal matrix of permittivity, $\overline{\mathbf{S}}_{oo}$ is the discretized $\nabla \times \mu^{-1}\nabla\times$ operator, and $\mathbf{J}_o$ is the vector of the current source, in the dielectric region only. The $\overline{\mathbf{S}}_{oo}$ matrix is generated as the following:

$$\overline{\mathbf{S}}_{oo} = \sum_{j=1,\ldots,N_p} \frac{1}{\mu_o^j}\overline{\mathbf{S}}_{h,o}^{(j)}\overline{\mathbf{S}}_{e,o}^{(j)}, \tag{8.3}$$

where j denotes a patch index, subscript o denotes outside conductors, N_p is the number of patches outside conductors, and other parameters are similar to those in (7.3). The matrices $\overline{\mathbf{D}}_{\epsilon oo}$ and $\overline{\mathbf{S}}_{oo}$ can be generated by removing the rows and columns corresponding to the conductor edges from $\overline{\mathbf{D}}_\epsilon$ and $\overline{\mathbf{S}}$.

8.3 Analytical method for finding the nullspace of the $\overline{\mathbf{S}}_{oo}$ matrix

The nullspace of $\overline{\mathbf{S}}_{oo}$ can be found analytically. The number of vectors in the nullspace is the number of dielectric nodes plus the number of conductors minus 1. For the nullspace corresponding to the dielectric nodes, denoted by $\overline{\mathbf{V}}_{0d}$, there is one nullspace vector at each dielectric node. The rule to generate it is as follows:

- Find all the edges connected to a dielectric node;
- If the edge basis enters the node, $\frac{1}{l_j}$ appears on the row corresponding to this edge basis, where l_j is the length of the corresponding edge;
- If the edge basis leaves the node, $-\frac{1}{l_j}$ appears on the row corresponding to this edge basis, where l_j is the length of the corresponding edge.

An opposite convention can also be used in the generation process. The $\overline{\mathbf{V}}_{0d}$ is generated after going through all the dielectric nodes in the mesh. For the nullspace corresponding to conductors, denoted by $\overline{\mathbf{V}}_{0g}$, its dimension is the number of free conductors. Each column vector in $\overline{\mathbf{V}}_{0g}$ is the summation of the $\overline{\mathbf{V}}_0$ vectors at all nodes on and inside each conductor. After generating $\overline{\mathbf{V}}_{0d}$ and $\overline{\mathbf{V}}_{0g}$ following the aforementioned rules, one vector should be removed from the union because the summation of all the nullspace vectors is 0 and thus one vector is redundant. The approach described in this section is also applicable to the Finite Element Method (FEM).

For a uniform grid in the Finite Difference Method, $\overline{\mathbf{S}}_{oo}$'s left and right nullspace are the same because $\overline{\mathbf{S}}_{oo}$ is symmetric. When a non-uniform grid is used, $\overline{\mathbf{S}}_{oo}$ is asymmetric, and hence the left nullspace is different from the right nullspace. For the left nullspace, we can also generate it analytically. Considering the left nullspace $\overline{\mathbf{V}}_{0da}$ on the dielectric nodes, the rule to generate it is as the following:

- Find all the edges connected to a dielectric node;
- If the edge basis enters the node, $\frac{1}{l_j^{avg}}$ appears on the row corresponding to this edge basis, where l_i^{avg} is the averaged length of the edge j and its adjacent edge around this node along the same direction;
- If the edge basis leaves the node, $-\frac{1}{l_j^{avg}}$ appears on the row corresponding to this edge basis, where l_i^{avg} is the averaged length of this edge j and its adjacent edge around this node along the same direction.

For each conductor, the nullspace $\overline{\mathbf{V}}_{0ga}$ is the weighted summation of the vectors $\overline{\mathbf{V}}_{0a}$ at all the nodes on and inside each conductor, where the weight for each vector is $l_x^{j,avg}$, $l_y^{j,avg}$, $l_z^{j,avg}$, which is the averaged length respectively along the x-, y-, z-directions around node j. One vector should be removed also because the sum of all $\overline{\mathbf{V}}_{0da}$ and $\overline{\mathbf{V}}_{0ga}$ vectors is zero.

8.4 The generation of the Laplacian matrix for problems having PEC objects

In this work, we formulate the Laplacian matrix for problems whose conductors are treated as PEC. This Laplacian matrix denotes the discretized ∇^2 operator in the dielectric region with PEC boundaries. It can be generated as the following

$$\overline{\mathbf{L}}_{oo} = \overline{\mathbf{S}}_{oo} + \frac{\overline{\mathbf{V}}_{0d}\overline{\mathbf{V}}_{0da}^T}{\mu} + \frac{\overline{\mathbf{V}}_{0b}\overline{\mathbf{V}}_{0ba}^T}{\mu}, \tag{8.4}$$

where $\overline{\mathbf{V}}_{0d}$ and $\overline{\mathbf{V}}_{0da}$ are the right and left nullspace vectors at the dielectric nodes, $\overline{\mathbf{V}}_{0b}$ and $\overline{\mathbf{V}}_{0ba}$ are the right and left nullspace vectors at the conductor surface nodes, and μ is the material permeability. The rows of all the vectors involved only have dielectric edges. (8.4) can be viewed as the windowed version of (7.8), which focuses on the dielectric part. In FEM, this $\overline{\mathbf{L}}_{oo}$ can be obtained similarly as $\overline{\mathbf{L}}_{oo} = \overline{\mathbf{S}}_{oo} + \frac{\overline{\mathbf{V}}_{0d}\overline{\mathbf{V}}_{0d}^T}{\mu} + \frac{\overline{\mathbf{V}}_{0b}\overline{\mathbf{V}}_{0b}^T}{\mu}$, where the left and right nullspace are the same.

The generated $\overline{\mathbf{L}}_{oo}$ is exactly the discretized form of the Laplacian operator ∇^2 in the dielectric system. It is sparse, and has the good property of a Laplacian matrix. Therefore, it can be solved with fast convergence when we use iterative solvers. Next, we show how it is utilized to build a fast solver.

8.5 Proposed fast algorithm to accelerate convergence in iterative solutions of problems with PEC objects

Instead of solving the original system (8.1), we decompose the full-wave system into a three by three block system by expanding $\mathbf{e}$ as $\mathbf{e} = \overline{\mathbf{V}}_0\mathbf{y}_0 + \overline{\tilde{\mathbf{V}}}_{0b}\mathbf{y}_{0b} + \mathbf{e}_h$, where $\overline{\mathbf{V}}_0 = [\overline{\mathbf{V}}_{0d}, \overline{\mathbf{V}}_{0g}]$ is the nullspace vectors of $\overline{\mathbf{S}}_{oo}$, $\overline{\tilde{\mathbf{V}}}_{0b}$ is the linearly independent nullspace vectors on the surface of the conductor, and $\mathbf{e}_h$ denotes the rest of the solution in the space complementary to $\overline{\mathbf{V}}_{0d}$ and $\overline{\mathbf{V}}_{0b}$. Therefore, (8.1) is transformed to

$$\begin{bmatrix} \overline{\mathbf{V}}_{0a}^T(-\omega^2\overline{\mathbf{D}}_{\epsilon oo})\overline{\mathbf{V}}_0 & \overline{\mathbf{V}}_{0a}^T(-\omega^2\overline{\mathbf{D}}_{\epsilon oo})\overline{\tilde{\mathbf{V}}}_{0b} & \overline{\mathbf{V}}_{0a}^T(-\omega^2\overline{\mathbf{D}}_{\epsilon oo}) \\ \overline{\tilde{\mathbf{V}}}_{0ba}^T(-\omega^2\overline{\mathbf{D}}_{\epsilon oo})\overline{\mathbf{V}}_0 & \overline{\tilde{\mathbf{V}}}_{0ba}^T(-\omega^2\overline{\mathbf{D}}_{\epsilon oo}+\overline{\mathbf{S}}_{oo})\overline{\tilde{\mathbf{V}}}_{0b} & \overline{\tilde{\mathbf{V}}}_{0ba}^T(-\omega^2\overline{\mathbf{D}}_{\epsilon oo}+\overline{\mathbf{S}}_{oo}) \\ (-\omega^2\overline{\mathbf{D}}_{\epsilon oo})\overline{\mathbf{V}}_0 & (-\omega^2\overline{\mathbf{D}}_{\epsilon oo}+\overline{\mathbf{S}}_{oo})\overline{\tilde{\mathbf{V}}}_{0b} & (-\omega^2\overline{\mathbf{D}}_{\epsilon oo}+\overline{\mathbf{L}}_{oo}) \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_{0b} \\ \mathbf{e}_h \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{V}}_{0a}^T(-j\omega\mathbf{J}) \\ \overline{\tilde{\mathbf{V}}}_{0ba}^T(-j\omega\mathbf{J}) \\ -j\omega\mathbf{J} \end{bmatrix}, \tag{8.5}$$

where $\overline{\mathbf{V}}_0 = [\overline{\mathbf{V}}_{0d}, \overline{\mathbf{V}}_{0g}]$, $\overline{\mathbf{V}}_{0a} = [\overline{\mathbf{V}}_{0da}, \overline{\mathbf{V}}_{0ga}]$, $\overline{\tilde{\mathbf{V}}}_{0b}$, $\overline{\tilde{\mathbf{V}}}_{0ba}$ are vectors of $\overline{\mathbf{V}}_{0b}$ and $\overline{\mathbf{V}}_{0ba}$ linearly independent to $\overline{\mathbf{V}}_0$, and $\overline{\mathbf{V}}_{0a}$, respectively. The space $\overline{\mathbf{V}}_0$ is just the combination of the $\overline{\mathbf{V}}_{0d}$ and $\overline{\mathbf{V}}_{0g}$ from the previous section. The same rule can be applied to the generation of $\overline{\mathbf{V}}_{0a}$. To ease the generation of linearly independent $\overline{\tilde{\mathbf{V}}}_{0b}$ easily, we set the mesh to have at least two cells in between two adjacent conductors. This is also required for a mesh to accurately capture field variations. For each conductor, construct $\overline{\tilde{\mathbf{V}}}_{0b}$ and $\overline{\tilde{\mathbf{V}}}_{0ba}$ as $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_{0a}$ on the conductor surface nodes with one surface node removed. Removing one surface node is to make sure $\overline{\tilde{\mathbf{V}}}_{0b}$ is linearly independent with $\overline{\mathbf{V}}_{0g}$. The same can be applied to $\overline{\tilde{\mathbf{V}}}_{0ba}$. The column space of $[\overline{\mathbf{V}}_0, \overline{\tilde{\mathbf{V}}}_{0b}]$ is the same space as $[\overline{\mathbf{V}}_{0d}, \overline{\mathbf{V}}_{0g}, \overline{\mathbf{V}}_{0b}]$, but with linearly independent ones removed. All the vectors in $[\overline{\mathbf{V}}_0, \overline{\tilde{\mathbf{V}}}_{0b}]$ are linearly independent with each other.

Write equation (8.5) in short as

$$\begin{bmatrix} \overline{\mathbf{A}}_{11} & \overline{\mathbf{A}}_{12} & \overline{\mathbf{A}}_{13} \\ \overline{\mathbf{A}}_{21} & \overline{\mathbf{A}}_{22} & \overline{\mathbf{A}}_{23} \\ \overline{\mathbf{A}}_{31} & \overline{\mathbf{A}}_{32} & \overline{\mathbf{A}}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_{0b} \\ \mathbf{e}_h \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}. \tag{8.6}$$

As can be seen from (8.5), $\overline{\mathbf{A}}_{11}$ and $\overline{\mathbf{A}}_{33}$ are nothing but Laplacian matrices. $\overline{\mathbf{A}}_{11}$ is the discretized $\nabla \cdot \epsilon \nabla$ operator in the dielectric region, where $\overline{\mathbf{V}}_{0a}^T$ denotes the divergence operator $\nabla\cdot$, $\overline{\mathbf{D}}_\epsilon$ represents ϵ for each edge and $\overline{\mathbf{V}}_0$ is the gradient operator ∇. $\overline{\mathbf{A}}_{33}$ can be viewed as $\overline{\mathbf{L}}_{oo}$ with some perturbation on the diagonal, hence $\overline{\mathbf{A}}_{33}$ is also a Laplacian matrix. $\overline{\mathbf{A}}_{11}$ and $\overline{\mathbf{A}}_{33}$ can be solved fast by iterative methods, such as the multigrid iterative method. For $\overline{\mathbf{A}}_{22}$, each conductor's $\overline{\mathbf{V}}_{0b}$ only interacts with itself. Therefore, $\overline{\mathbf{A}}_{22}$ is a block diagonal matrix, where the block number equals to the number of conductors. For each block, the matrix is also very sparse because each $\overline{\mathbf{V}}_{0b}$ only interacts with the surrounding nodes' $\overline{\mathbf{V}}_{0ba}$. Due to the sparse structure of $\overline{\mathbf{A}}_{22}$, its inverse is also easy to compute.

To solve (8.6), firstly, we solve the third row of equations with its schur complement as the following:

$$\left(\overline{\mathbf{A}}_{33} - [\overline{\mathbf{A}}_{31}, \overline{\mathbf{A}}_{32}] \begin{bmatrix} \overline{\mathbf{A}}_{11} & \overline{\mathbf{A}}_{12} \\ \overline{\mathbf{A}}_{21} & \overline{\mathbf{A}}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \overline{\mathbf{A}}_{13} \\ \overline{\mathbf{A}}_{23} \end{bmatrix}\right) \mathbf{e}_h = b_3 - [\overline{\mathbf{A}}_{31}, \overline{\mathbf{A}}_{32}] \begin{bmatrix} \overline{\mathbf{A}}_{11} & \overline{\mathbf{A}}_{12} \\ \overline{\mathbf{A}}_{21} & \overline{\mathbf{A}}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}. \tag{8.7}$$

This equation can be solved by GMRES with fast convergence. The left-hand matrix can be rewritten as

$$\frac{\overline{\mathbf{V}}_{0d}\overline{\mathbf{V}}_{0da}^T}{\mu} + \frac{\overline{\mathbf{V}}_{0b}\overline{\mathbf{V}}_{0ba}^T}{\mu} + \left(\bar{\mathbf{I}} - [\overline{\mathbf{A}}_{31}, \overline{\mathbf{A}}_{32}] \begin{bmatrix} \overline{\mathbf{A}}_{11} & \overline{\mathbf{A}}_{12} \\ \overline{\mathbf{A}}_{21} & \overline{\mathbf{A}}_{22} \end{bmatrix}^{-1} \begin{bmatrix} \overline{\mathbf{V}}_{0a}^T \\ \overline{\mathbf{V}}_{0ba}^T \end{bmatrix}\right) (-\omega^2 \overline{\mathbf{D}}_{oo} + \overline{\mathbf{S}}_{oo}), \tag{8.8}$$

which can be viewed as deducting the component of the original matrix in the space of $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_{0b}$, and resulting in a numerical Laplacian matrix in inhomogeneous materials. Due to this reason, GMRES or other iterative solvers can converge this matrix fast, and only the matrix-vector multiplication needs to be performed. All the matrices involved are sparse, and can be multiplied fast. As for $\begin{bmatrix} \overline{\mathbf{A}}_{11} & \overline{\mathbf{A}}_{12} \\ \overline{\mathbf{A}}_{21} & \overline{\mathbf{A}}_{22} \end{bmatrix}^{-1}$, we can solve this fast by applying the upper triangular matrix as the preconditioner. The matrix $\overline{\mathbf{A}}_{11}$ is Laplacian and $\overline{\mathbf{A}}_{22}$ is block diagonal with a very sparse structure. So these two matrices' inverse can be readily obtained. Therefore, this preconditioner can be applied efficiently.

After $\mathbf{e}_h$ is obtained, substituting it into the first two equations, we obtain

$$\begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_{0b} \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{A}}_{11} & \overline{\mathbf{A}}_{12} \\ \overline{\mathbf{A}}_{21} & \overline{\mathbf{A}}_{22} \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} - \begin{bmatrix} \overline{\mathbf{A}}_{13} \\ \overline{\mathbf{A}}_{23} \end{bmatrix} \mathbf{e}_h \right). \tag{8.9}$$

The inverse $\begin{bmatrix} \overline{\mathbf{A}}_{11} & \overline{\mathbf{A}}_{12} \\ \overline{\mathbf{A}}_{21} & \overline{\mathbf{A}}_{22} \end{bmatrix}^{-1}$ can be obtained similarly by using the upper triangular matrix as the preconditioner.

With this fast algorithm, we can also solve problems with lossy conductors by separating the conductor region from the dielectric part to solve. The entire system of equations can hence be divided as

$$\begin{bmatrix} \overline{\mathbf{A}}_{oo} & \overline{\mathbf{A}}_{oi} \\ \overline{\mathbf{A}}_{io} & \overline{\mathbf{A}}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{x}_o \\ \mathbf{x}_i \end{bmatrix} = \begin{bmatrix} \mathbf{b}_o \\ 0 \end{bmatrix}, \tag{8.10}$$

where subscript o denotes outside conductors, i is for inside conductors, $\overline{\mathbf{A}}_{oo} = -\omega^2 \overline{\mathbf{D}}_{\epsilon oo} + \overline{\mathbf{S}}_{oo}$, $\overline{\mathbf{A}}_{oi} = \overline{\mathbf{S}}_{oi}$, $\overline{\mathbf{A}}_{io} = \overline{\mathbf{S}}_{io}$ and $\overline{\mathbf{A}}_{ii} = -\omega^2 \overline{\mathbf{D}}_{\epsilon ii} + j\omega \overline{\mathbf{D}}_{\sigma ii} + \overline{\mathbf{S}}_{ii}$. First, we solve the subsystem inside conductors as

$$(\overline{\mathbf{A}}_{ii} - \overline{\mathbf{A}}_{io} \overline{\mathbf{A}}_{oo}^{-1} \overline{\mathbf{A}}_{oi}) \mathbf{x}_i = -\overline{\mathbf{A}}_{io} \overline{\mathbf{A}}_{oo}^{-1} b_o. \tag{8.11}$$

Since $\|\overline{\mathbf{A}}_{ii}\| >> \|\overline{\mathbf{A}}_{io} \overline{\mathbf{A}}_{oo}^{-1} \overline{\mathbf{A}}_{oi}\|$ and inside the conductors $\|j\omega \overline{\mathbf{D}}_{\sigma ii}\| >> \| -\omega^2 \overline{\mathbf{D}}_{\epsilon ii} + \overline{\mathbf{S}}_{ii}\|$, the schur complement part can be ignored and (8.11) becomes

$$j\omega \overline{\mathbf{D}}_{\sigma ii} \mathbf{x}_i = -\overline{\mathbf{A}}_{io} \overline{\mathbf{A}}_{oo}^{-1} \mathbf{b}_o. \tag{8.12}$$

The above can be solved easily because $\overline{\mathbf{D}}_{\sigma ii}$ is just a diagonal matrix. After $\mathbf{x}_i$ is solved, we substitute it into the second row of equation in (8.10) to solve $\mathbf{x}_o$ as

$$\overline{\mathbf{A}}_{oo} \mathbf{x}_o = \mathbf{b}_o - \overline{\mathbf{A}}_{oi} \mathbf{x}_i, \tag{8.13}$$

where $\overline{\mathbf{A}}_{oo}^{-1}$'s solution can be obtained from the fast algorithm in this Chapter.

8.6 Numerical Results

In this section, we simulate a suite of examples to validate the algorithm proposed in this chapter. Among these examples, the first example has both inside and outside conductor subsystems to validate the solution of (8.10), and the rest examples have perfect conductors to validate the solution of (8.6).

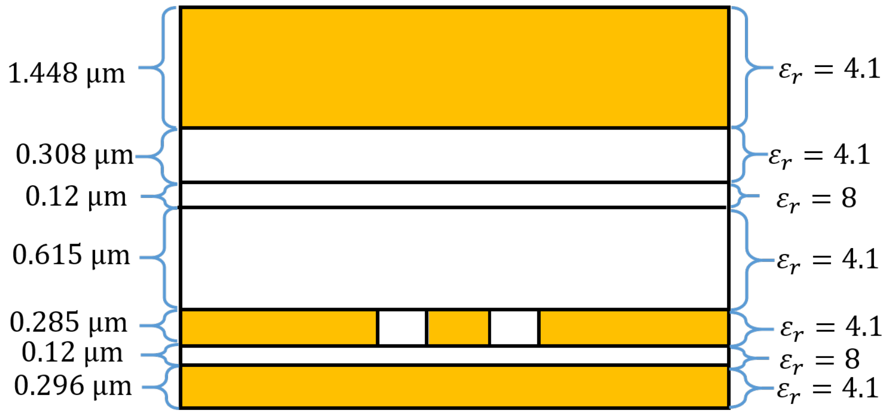


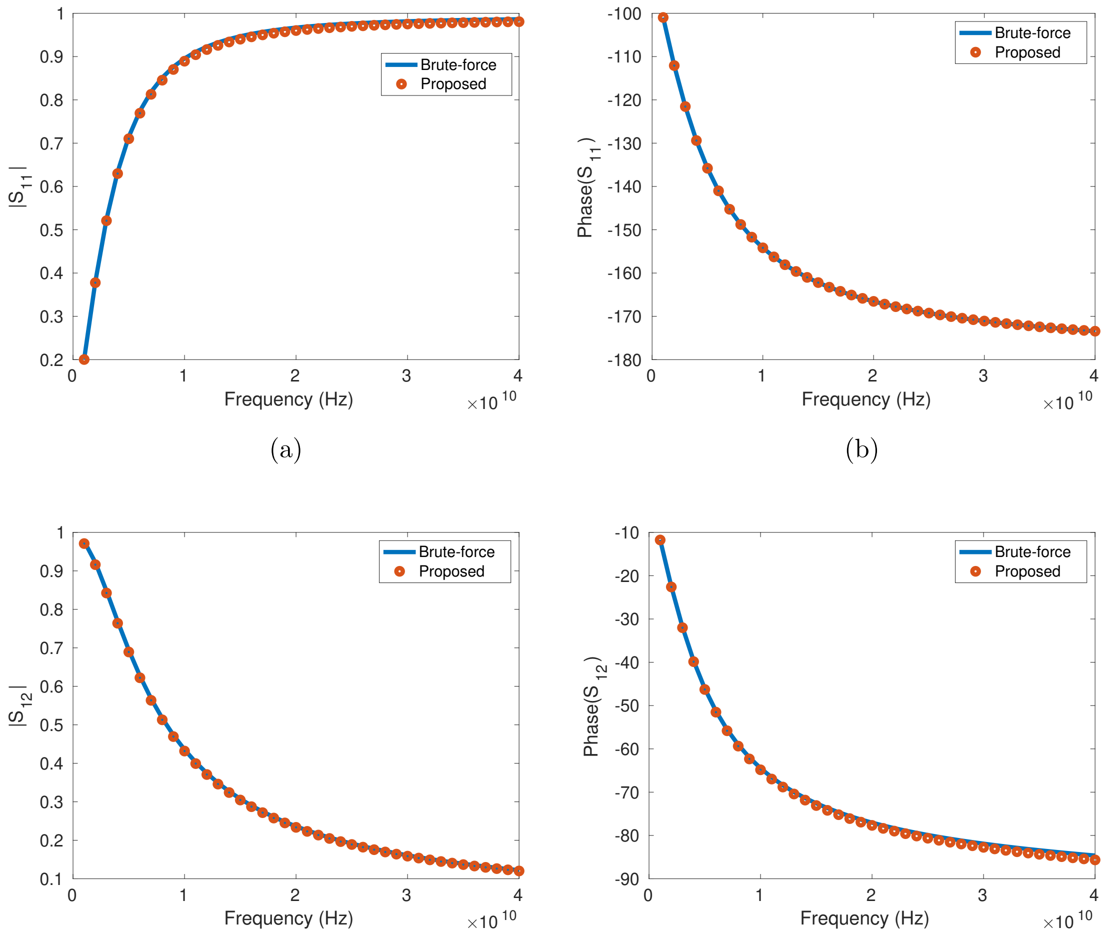
Fig. 8.1.: Structure of a test-chip interconnect.

8.6.1 Test-chip Interconnect

The first example is a test-chip interconnect, whose structure is shown in Fig. 8.1. All the configurations for this example are the same as those in previous chapter. The S-parameters are extracted at the near and far ends and compared with the reference solution, which are shown in Fig. 8.2. Note that the solution of this example is obtained from solving (8.10). The solutions from the proposed method match well with the results from the reference. As for the convergence rate, at 10 GHz, solving (8.7) takes 17 steps to relative residual 0.00491. And solving $\begin{bmatrix} \overline{\mathbf{A}}_{11} & \overline{\mathbf{A}}_{12} \\ \overline{\mathbf{A}}_{21} & \overline{\mathbf{A}}_{22} \end{bmatrix}^{-1}$ takes 2 steps to relative residual 1e-7. So the iterative solution is also very efficient. In the procedure, solving the Laplacian matrix $\overline{\mathbf{A}}_{11}$ only takes 3 steps to the relative residual of 1e-5.

8.6.2 Single-ended Microstrip

The second example is a single-ended microstrip, and the structure is shown in Fig. 8.3. The length of the stripline is 14.854 mm. On the two ends are the GSG launchers and currents are injected through the launchers. The S-parameters at the near and far ends are extracted and compared with the results from the brute-force



(a)

(b)

(c)

(d)

Fig. 8.2.: (a) and (b): $\mathbf{S}_{11}$'s magnitude and phase; (c) and (d): $\mathbf{S}_{12}$'s magnitude and phase for the test-chip interconnect.

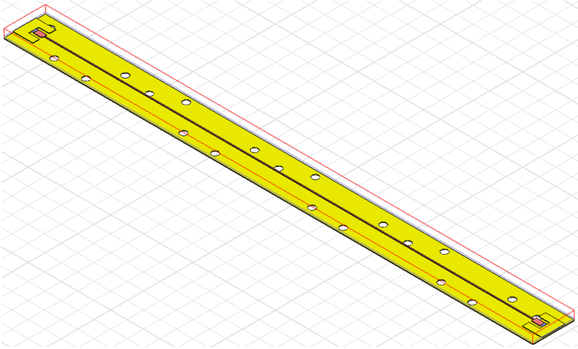
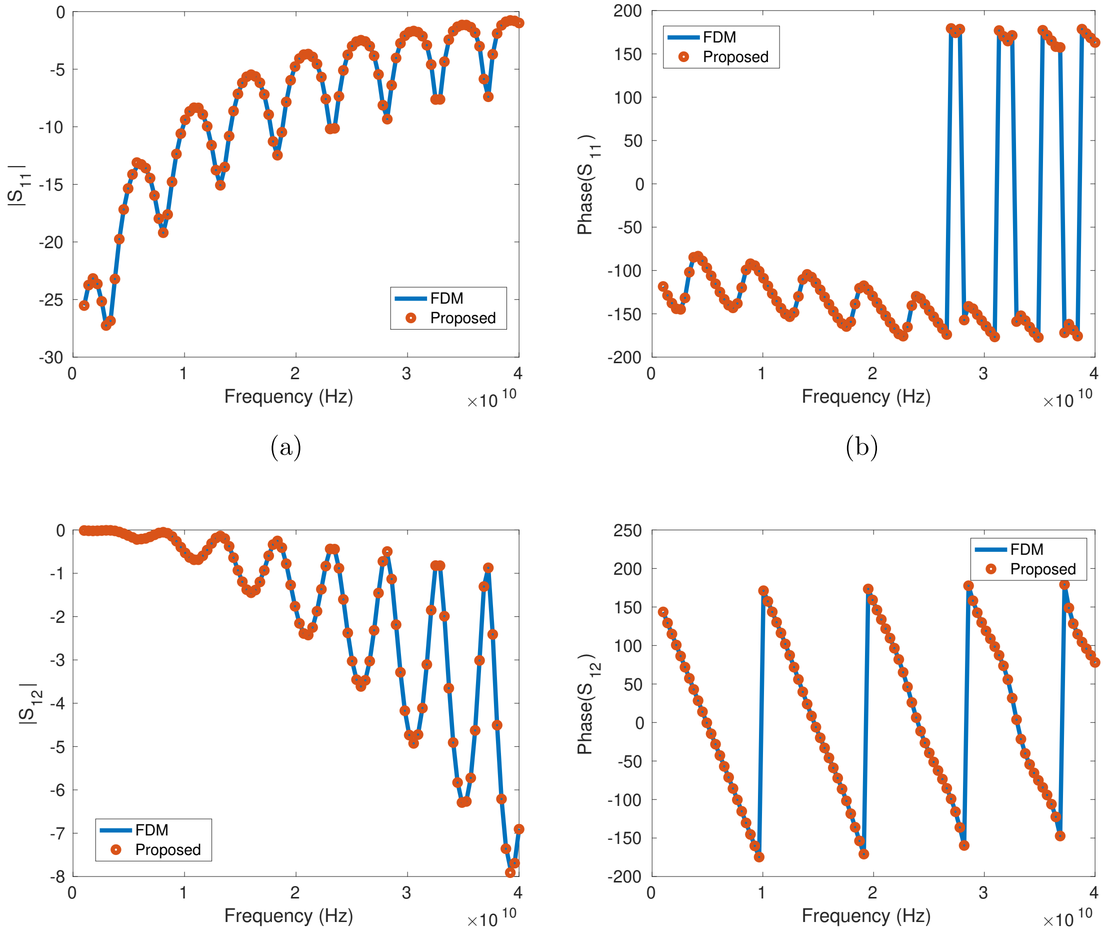


Fig. 8.3.: The structure of a single-ended microstrip.

methods, shown in Fig. 8.4. As can be seen, the results match the reference result very well. The iterative solver only takes 13 steps to solve (8.7) to relative residual 0.000287936 at 10 GHz. And solving $\begin{bmatrix} \overline{\mathbf{A}}_{11} & \overline{\mathbf{A}}_{12} \\ \overline{\mathbf{A}}_{21} & \overline{\mathbf{A}}_{22} \end{bmatrix}^{-1}$ takes 17 steps to relative residual of 6.2755e-6. And solving $\overline{\mathbf{A}}_{11}$ only takes 1 step to relative residual 1e-5.

8.6.3 NAND Gate

The next example is a NAND gate example, which is shown in Fig. 8.5. There are 17 layers and each layer is shown with a different color. This structure is discretized by a non-uniform mesh yielding about 665,940 unknowns. Here, we plot the S-parameters from the proposed method and the brute-force method in Fig. 8.6, and they can match very well. As for the convergence rate, solving (8.7) takes 145 steps to relative residual 0.00954 at 10 GHz. Solving $\begin{bmatrix} \overline{\mathbf{A}}_{11} & \overline{\mathbf{A}}_{12} \\ \overline{\mathbf{A}}_{21} & \overline{\mathbf{A}}_{22} \end{bmatrix}^{-1}$ only takes 1 or 2 steps to a relative residual of 1e-7. Solving $\overline{\mathbf{A}}_{11}$ only takes at most 6 steps to relative residual 1e-5. which is also efficient.



(a) (b)

(c) (d)

Fig. 8.4.: (a) and (b) are S_{11}'s magnitude and phase; (c) and (d) are S_{12}'s magnitude and phase for the single-ended microstrip example.

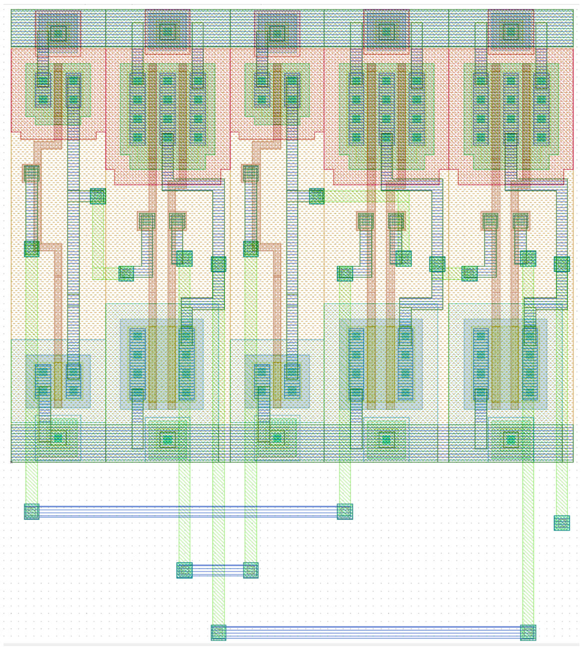
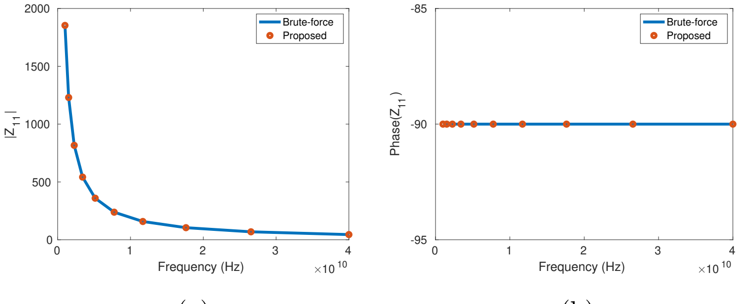


Fig. 8.5.: Structure of the NAND gate.



(a) (b)

Fig. 8.6.: Z_{11}'s magnitude and phase for the NAND gate example.

8.6.4 Plasma Package Structure

The last example is a plasma package structure from IBM. The top view of this structure is shown in Fig. 7.14. Non-uniform mesh is used to discretize the entire chip, yielding 1,762,840 unknowns. The performance of the iterative solver to solve (8.7) is 55 steps to relative residual 0.00974521 at 10 GHz. And solving $\begin{bmatrix} \overline{\mathbf{A}}_{11} & \overline{\mathbf{A}}_{12} \\ \overline{\mathbf{A}}_{21} & \overline{\mathbf{A}}_{22} \end{bmatrix}^{-1}$ only takes 64 steps to a relative residual of 1.30201e-6. Solving the Laplacian matrix $\overline{\mathbf{A}}_{11}$ by HYPRE only takes at most 3 steps to relative residual of 1e-5.

9. CONCLUSIONS AND FUTURE WORK

9.1 Conclusions

In this work, we develop rapid modeling and simulation methods for large-scale and circuit-intuitive electromagnetic analysis of integrated circuits and systems as follows:

- First, we develop an analytical method for finding the nullspace of the curl-curl operator in an arbitrary mesh for an arbitrary order of curl conforming vector basis function. It can be applied to the partial differential equation methods such as a finite difference method or a finite element method. Rather than solving a numerical eigenvalue problem or using some approximations, we find an analytical way to obtain the nullspace of the curl-curl operator regardless of whether the basis function is a zeroth-order edge element or a higher-order basis for any unstructured mesh generated from arbitrarily shaped elements. When considering the zeroth-order basis, the rule to generate the nullspace is to first find all edges connected to a node and if the edge basis enters the node, $\frac{1}{l_i}$ appears on the row corresponding to this edge basis, where l_i is the length of the edge; otherwise, we have $-\frac{1}{l_i}$. For higher-order bases, we first find the nullspace based on the homogeneous bases. The generation rule is similar to that from the zeroth-order basis, that is to generate the nullspace vectors by finding all the bases connected to one node from the distribution of the homogeneous basis functions. Then the relationship between the interpolatory basis coefficients and the homogeneous basis coefficients can be used to find out the nullspace of interpolatory bases.

- Second, we develop an analytical method for finding the complementary space of the nullspace, $\mathbf{V}_h$. By finding the independent cycle bases in the mesh, we can generate the vectors of $\mathbf{V}_h$. Therefore, l_i appears on the row corresponding to the global index of the edges around this cycle. By using the nullspace and its complementary spaces, we develop a rigorous method without using any low-frequency approximation to solve the breakdown problem involving lossy conductors embedded in inhomogeneous dielectrics. We use the analytical spaces instead of the eigensolutions, which significantly speeds up the computation. In this method, a full-wave finite-element solution is first decomposed into two components: nullspace component and its complementary one, and each space is generated analytically for an arbitrary mesh. Each component is then found without breakdown from high frequency down to any low frequency. In addition, for general inhomogeneous lossy problems, the frequency dependence of a full-wave finite-element solution is revealed at low frequencies.

- Third, we develop a fast FDTD solver to perform the layout extraction of integrated circuits. We propose a way to compute the RC part of the solution efficiently. The time step is not restricted by the small space step restricted by stability in the IC layout. Instead, the time step can be arbitrarily large and make the simulation from very low frequencies to high frequencies feasible. The computational cost is also minimized because all the column spaces we use to decompose the field solution are identified analytically. Numerical results also validate the accuracy and efficiency of this method.

- Fourth, we derive a closed-form model of the inverse of full-wave Maxwell's system of equations for an arbitrary physical layout in both frequency and time domain. The model is decomposed into R-, C-, L- and full-wave components with first-principal accuracy from zero to high frequencies. We decompose our system into a zero mode system and a high order mode system. The zero mode system is nothing but a Laplacian system. The solution from the zero mode system has an analytical frequency and time dependence and denotes the RC component. The field space used in the zero mode system is from our analytical method to generate the nullspace of the curl-curl operator. The high order mode system is a reduced small system, which can also be solved fast. The solution from the high order mode system denotes the L and full-wave component. We also develop a fast method to find those physically important high order modes to represent the field space, whose number is small. After these two solutions are obtained, we can combine them to get the final full solution. Each of the R-, C-, L- and full-wave components is found explicitly and efficiently. The efficiency, accuracy, and capacity of this work are demonstrated by real-world large-scale layout extraction.

- At the end, when solving the full-wave components, another efficient way is developed. Instead of solving the original equation we solve its Laplacian counterpart, which is positive definite and well-conditioned. We find that the curl-curl operator can be rigorously decomposed into a gradient divergence operator and a Laplacian operator, both of which can be constructed from the mesh information without any need for computation. When operating on the full-wave component of the field solution, the gradient divergence term vanishes and only the Laplacian operator is left. Therefore, we are able to replace the curl-curl matrix with the Laplacian matrix when operating on the full-wave component without any approximation. Its iterative solution has guaranteed convergence and it can converge in a small number of steps. After the static component and the full-wave component are found, they can be combined to obtain the total solution. The conductor's setting can be arbitrary. Our method can handle both perfect conductors and lossy ones. For perfect conductors, we only focus on the dielectric edges and reduce the original system to be a smaller one. This idea can be applied to FEM, FDM and other PDE methods.

9.2 Future work

The future work of this research includes but is not limited to:

- *Application of this algorithm to solve general problems.* From the previous work, we have already applied and verified the inverse model in various realistic large-scale engineering problems. For example, we successfully simulated the plasma package example and obtained an accurate result. The next step is to implement it as a general tool to handle more layout structures and solve more real-life problems.

- *Parallelization of the algorithms will be developed in this work.* From the inverse model, we can clearly see that the solution is decomposed into R-, C-, L- and full-wave components. And each component can be obtained independently. Moreover, with different ports excitation, the solutions can also be computed independently. With parallelization, the proposed method can be further accelerated in CPU run time.

- *Application to other PDE solvers.* From all the analysis, we can see that the proposed algorithm can be applied to not only Finite Difference Method but also other PDE solvers like Finite Element Method. The nullspace of the curl-curl matrix and the Laplacian matrix are constructed similarly as that from the Finite Difference Method. The system matrix is composed of similar components. Therefore, the proposed algorithms can be readily implemented in the Finite Element Method. The same is true to other PDE solvers.

REFERENCES

REFERENCES

[1] A. Ruehli and P. Brennan, "Efficient capacitance calculation for three dimensional multiconductor systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 21, no. 2, 1973.

[2] K. Nabors and J. White, "Fastcap: A multipole accelerated 3-D capacitance extraction program," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, no. 11, pp. 1447–1459, 1991.

[3] T. Chou and Z. Cendes, "Capacitance calculation of IC packages using the finite element method and planes of symmetry," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 9, pp. 1159–1166, 1994.

[4] M. Kamon, M. Tsuk, and J. White, "FASTHENRY: A multipole-accelerated 3-D inductance extraction program," *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, no. 9, pp. 1750–1758, 1994.

[5] W. Shi, J. Liu, N. Kakani, and T. Yu, "A fast hierarchical algorithm for 3-D capacitance extraction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 3, pp. 330–336, 2002.

[6] W. Chai and D. Jiao, "Dense matrix inversion of linear complexity for integral-equation based large-scale 3-D capacitance extraction," *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, no. 10, pp. 2404–2421, 2011.

[7] G. Chen, H. Zhu, T. Cui, C. Z, X. Zeng, and W. Cai, "ParAFEMCap: A parallel adaptive finite-element method for 3-D VLSI interconnect capacitance extraction," *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 2, pp. 218–231, 2012.

[8] W. Sun, W. Hong, and W. Dai, "Resistance extraction using superconvergence accelerated boundary element method," *Proceedings of Asia-Pacific Microwave Conference*, vol. 3, Nov. 2002.

[9] A. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 22, no. 3, 1974.

[10] T. Smedes, N. V. D. Meijs, and A. V. Genderen, "Extraction of circuit models for substrate cross-talk," *Proceedings of IEEE International Conference Computer-Aided Design*, Nov. 2002.

[11] M. Piket-May, A. Taflove, and J. Baron, "FD-TD modeling of digital signal propagation in 3-D circtuis with passive and active loads," *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, no. 8, pp. 1514–1523, 1994.

[12] A. Taflove and S. Hagness, *Computational electrodynamics.* Artech House, Norwood, 2000.

[13] J. Jin, *The finite element method in electromagnetics.* John Wiley and Sons, 2014, vol. 8.

[14] S. Chang, R. Coccioli, Y. Qian, and T. Itoh, "A global finite-element time domain analysis of active nonlinear microwave," *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, no. 12, pp. 2401–2416, 1999.

[15] R. Wang and J. Jin, "A symmetric electromagnetic-circuit simulator based on the extended time-domain finite element method," *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, no. 12, pp. 2875–2884, 2008.

[16] D. Jiao, "A novel technique for full-wave modeling of large-scale three-dimensional high-speed on/off-chip interconnect structures," *Proceedings of International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, pp. 39–42, 2003.

[17] D. Jiao, S. Chakravarty, and C. Dai, "A layered finite-element method for electromagnetic analysis of large-scale high-frequency integrated circuits," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 2, pp. 422–432, 2007.

[18] Q. He, H. Gan, and D. Jiao, "From layout directly to simulation: A first-principle-guided circuit simulator of linear complexity and its efficient parallelization," *IEEE Transactions on Componenets, Packaging and Manufacturing Technology*, vol. 2, no. 4, pp. 687–699, 2012.

[19] S. Kapur and D. Long, "IES3: A fast integral equation solver for efficient 3-dimensional extraction," *Proceedings of IEEE International Conference on Computer-Aided Design (ICCAD)*, pp. 448–455, 1997.

[20] A. Rong, A. Cangellaris, and L. Dong, "Comprehensive broad-band electromagnetic modeling of on-chip interconnects with a surface discretization-based generalized PEEC model," *Proceedings of IEEE 12th Topical Meeting on Electrical Performance of Electrical Packaging (EPEP)*, pp. 367–370, Oct. 2003.

[21] Z. Qian, "Crosstalk analysis by fast computational algorithm," *Proceedings of IEEE 14th Topical Meeting on Electrical Performance of Electrical Packaging (EPEP)*, pp. 367–370, 2006.

[22] S. Kapur and D. Long, "Large-scale full-wave simulation," *Proceedings of the 41st annual Design Automation Conference (DAC)*, pp. 806–809, 2004.

[23] Z. Zhu, B. Song, and J. White, "Algorithms in FastImp: A fast and wide-band impedance extraction program for complicated 3-D geometries," *Proceedings of 40th ACM/EDAC/IEEE Design Automation Conference*, pp. 712–717, 2003.

[24] W. Chai and D. Jiao, "An LU decomposition based direct integral equation solver of linear complexity and higher-order accuracy for large-scale interconnect extraction," *IEEE Transactions on Advanced Packaging*, vol. 33, no. 4, pp. 794–803, 2010.

[25] ——, "Direct matrix solution of linear complexity for surface integral-equation-based impedance extraction of complicated 3-D structures," *Proceedings of IEEE*, vol. 101, no. 2, pp. 372–388, 2013.

[26] J. Zhao and W. Chew, "Integral equation solution of Maxwell's equations from zero frequency to microwave frequencies," *IEEE Transactions on Antennas and Propagation*, vol. 48, no. 10, pp. 1635–1645, 2000.

[27] S. Lee and J. Jin, "Application of the tree-cotree splitting for improving matrix conditioning in the full-wave finite-element analysis of high-speed circuits," *Microwave and Optical Technology Letters*, vol. 50, no. 6, pp. 1476–1481, 2008.

[28] J. Zhu and D. Jiao, "A theoretically rigorous full-wave finite-element-based solution of Maxwell's equations from DC to high frequencies," *IEEE Transactions on Advanced Packaging*, vol. 33, no. 4, pp. 1043–1050, 2010.

[29] ——, "A rigorous solution to the low-frequency breakdown in full-wave finite-element-based analysis of general problems involving inhomogeneous lossless/lossy dielectrics and nonideal conductors," *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, no. 12, pp. 3294–3306, 2011.

[30] G. Vecchi, "Loop-star decomposition of basis functions in the discretization of the EFIE," *IEEE Transactions on Antennas and Propagation*, vol. 47, no. 2, pp. 339–346, 1999.

[31] F. Canning and J. Scholl, "Diagonal preconditioners for the EFIE using a wavelet basis," *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 9, pp. 1239–1246, 1996.

[32] Y. Bo, W. Hong, and S. Song, "The triangular matrix preconditioners for solutions of partial differential equations in electromagnetics," *IEEE Antennas and Propagation Society International Symposium*, pp. 222–225, 205.

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

[34] Y. Yang, Z. Fan, D. Ding, and S. Liu, "Application of the preconditioned GMRES to the Crank-Nicolson finite-difference time-domain algorithm for 3D full-wave analysis of planar circuits," *Microwave and Optical Technology Letters*, vol. 50, no. 6, pp. 1458–1463, 2008.

[35] D. Sun, J. Manges, X. Yuan, and Z. Cendes, "Spurious modes in finite-element methods," *IEEE Antennas and Propagation Magazine*, vol. 37, no. 5, pp. 12–24, 1995.

[36] L. Xue and D. Jiao, "Broadband and sparse finite-element formulation free of low-frequency breakdown," *IEEE International Symposium on Antennas and Propagation*, 2018.

[37] ——, "Rapid inverse modeling of integrated circuit layout in both frequency and time domain from zero to full-wave frequencies," *IEEE International Microwave Symposium (IMS)*, 2019.

[38] N. V. Venkatarayalu and J. Lee, "Removal of spurious DC modes in edge element solutions for modeling three-dimensional resonators," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 7, pp. 3019–3025, 2006.

[39] S. Yan, J. Jin, and Z. Nie, “Accurate and stable matrix-free time-domain method in 3-D unstructured meshes for general electromagnetic analysis,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 12, pp. 4201–4214, 2015.

[40] R. D. Graglia, D. R. Wilton, and A. F. Peterson, “Higher order interpolatory vector bases for computational electromagnetics,” *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 329–342, 1997.

[41] L. Xue and D. Jiao, “Analytical method for finding the nullspace of stiffness matrix in the finite element method,” *IEEE International Symposium on Antennas and Propagation*, 2017.

[42] A. Taflove and S. Hagness, *Computational electromagnetics: the finite-difference time-domain method.* Artech House, Norwood, 2005.

[43] R. Falgout and U. Yang, “Hypre: A library of high performance preconditioners,” *International Conference on Computational Science*, 2002. [Online]. Available: https://github.com/LLNL/hypre

[44] Q. He, H. Gan, and D. Jiao, “Explicit time-domain finite-element method stabilized for an arbitrarily large time step,” *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 11, pp. 5240–5250, 2012.

[45] J. Yan and D. Jiao, “Fast algorithm for generating a minimal-order model of a general lossy electromagnetic problem,” *IEEE Antennas and Propagation Society International Symposium (APSURSI)*, 2014.

[46] J. Lee, D. Chen, V. Balakrishnan, C. Koh, and D. Jiao, “A quadratic eigenvalue solver of linear complexity for 3-D electromagnetics-based analysis of large-scale integrated circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 3, pp. 380–390, 2012.

[47] 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 Transactions on Microwave Theory and Techniques*, vol. 68, no. 4, pp. 1270–1283, 2020.

[48] J. Yan and D. Jiao, “Fast explicit and unconditionally stable FDTD method for electromagnetic analysis,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 8, pp. 2698–2710, 2017.

[49] 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 Transactions on Microwave Theory and Techniques*, vol. 68, no. 2, pp. 456–468, 2020.

[50] A. Brandt, S. McCormick, and J. Ruge, *Algebraic multigrid (AMG) for sparse matrix equations, in Sparsity and its applications.* Cambridge University Press, 1983.