Anisotropic and Optimized FFT-Based Iterative Electromagnetic Solver for the PEEC Method

Daniele Romano, Ivana Kovacevic-Badstuebner, *Senior Member, IEEE*, Giulio Antonini, *Senior Member, IEEE*, and Ulrike Grossner, *Member, IEEE*

***Abstract*— Fast Fourier transform (FFT)-accelerated integral-equation-based electromagnetic (EM) simulators have gained attraction for their capability to compute parasitics of arbitrarily shaped and large-scale voxelized structures on a desktop computer. Yet, FFT-based solvers have limitations due to the necessity of using voxels of the same size in all three Cartesian dimensions and suffer in the case of geometries with far-apart objects that require meshing also the air between them, resulting in a huge number of voxels and, thus, of unknowns. This work aims to overcome both these limitations by developing a systematic anisotropic strategy to compute matrix–vector products using the FFT-based approach and to remove the air existing between objects without sacrificing the desirable features of the FFT-based approach. The proposed approach is presented in the framework of the partial element equivalent circuit (PEEC) method, but it is well suited to be used also with other integral equation-based methods. The accuracy, efficiency, and applicability of the proposed anisotropic and optimized FFT (aoFFT)-based PEEC solver are demonstrated in the example of two structures requiring to use voxels of different sizes along the three Cartesian dimensions and with a large portion of air between the objects.**

***Index Terms*— Computational electromagnetics (EMs), fast Fourier transform (FFT), parasitic extraction, partial element equivalent circuit (PEEC) method, voxelization.**

I. Introduction

With the ever-increasing operating frequencies and shrinking electronic systems and device sizes, tools that allow for accurate model parasitic effects have become indispensable for design explorations.

Over the last decades, many numerical algorithms and computational methods have been developed for handling more complex electromagnetic (EM) problems. Among them, we list the finite-difference time-domain (FDTD) method [1], the finite-element method (FEM) [2], the method of moments (MoM) [3], and the finite integration technique (FIT) [4].

The partial element equivalent circuit (PEEC) method can also be listed among the methods well suited to solving Maxwell's equations for complex EM problems. It has been introduced by Ruehli [5] combining the concept of partial inductances [6] and partial capacitances [7]. Since then, it has been developed through a series of remarkable developments over the years, which have made it more robust and reliable [8]. Indeed, it has been used to model EM problems in various fields including power plane-pair [9], [10], decoupling capacitors [11], [12], PCB prelayout power integrity analysis [13], power modules [14], [15], on-chip interconnections [16], shielding [17], and antennas [18], [19].

Similar to the MoM [3], the PEEC method requires meshing only the regions made of conductors, dielectrics, or magnetic materials which is clearly an advantage over differential methods such as the FDTD technique [1]. Furthermore, assuming that currents and charges radiate in the free space, according to the volume equivalence principle, the free space Green's function can be used, analytically enforcing zero-field boundary conditions at infinity.

The enforcement of Kirchhoff voltage and current laws (KVL and KCL) to the equivalent circuit leads to a system of delayed differential equations of the neutral type assuming also propagation delays [8]. For geometrically complex and/or electrically large problems, such a system of equations can be easily very large, and iterative solvers are needed. Although the computational complexity of iterative solvers is $O(n^2)$ and is significantly smaller than that of direct methods $O(n^3)$, it can still be too high for problems of industrial interest. Thus, many techniques have been developed over the years to accelerate the matrix–vector product which is the bottleneck of iterative solvers. Among the others, the most popular methods are the fast multipole method [20], [21], [22], the QR-based approach [23], and the adaptive cross-approximation (ACA) method [24], [25], [26], which is based on representing the matrices as a product of two matrices of a lower rank such that the matrix–vector product can be performed much more efficiently. Furthermore, the hierarchical $\mathcal{H}$ class of methods [27] has been employed successfully as well.

In the specific context of the PEEC method, algebraic methods such as the ACA [28], [29], multiscale decomposition techniques [30], and hierarchical matrices [31] have been used in practice. However, algebraic iterative methods need good preconditioning to keep the number of iterations reasonable.

The matrix–vector products associated with the iterative solvers can also be accelerated by fast Fourier transform (FFT), as shown in [32] for a volume integral equation formulation. Indeed, such a method exploits the translational

Manuscript received 19 July 2023; revised 18 September 2023; accepted 6 October 2023. Date of publication 31 October 2023; date of current version 7 May 2024. This work was supported by the Swiss National Science Foundation under Grant 209501. *(Corresponding author: Daniele Romano.)*

Daniele Romano and Giulio Antonini are with the UAq EMC Laboratory, Department of Industrial and Information Engineering and Economics, University of L'Aquila, 67100 L'Aquila, Italy (e-mail: daniele.romano@univaq.it; giulio.antonini@univaq.it).

Ivana Kovacevic-Badstuebner and Ulrike Grossner are with the Advanced Power Semiconductor Laboratory, ETH Zürich, 8092 Zürich, Switzerland (e-mail: kovacevic@aps.ee.ethz.ch; ulrike.grossner@ethz.ch).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMTT.2023.3326280.

Digital Object Identifier 10.1109/TMTT.2023.3326280

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

invariance of elementary magnetic and electric field interactions providing significant memory saving and an excellent improvement of the computation performances. The FFT-based acceleration also implies a representation of the 3-D modeling geometries with voxels. A parasitic inductance extractor for voxelized structures was presented in [33]. Furthermore, a Tucker-enhanced and the FFT-accelerated version of the method was proposed for capacitance and inductance extraction [34] and [35], respectively. An FFT-PEEC solver accounting for both inductances and capacitances was presented in [36]. Furthermore, an efficient iterative solver with the matrix–vector products involving the PEEC dense matrices performed via FFT was presented in [37], demonstrating that it was possible to solve large problems with millions of unknowns achieving a fast convergence (low number of iterations of the iterative solver) based on an efficient sparse preconditioner.

A bottleneck of the FFT acceleration technique is that all the voxels have the same size in each Cartesian direction (rectangular prisms) and also that the free space between 3-D geometries with multiple nonconnected objects must be meshed and properly handled through a zero padding strategy [32]. This implies that the method is less suited to handle problems involving far-apart objects because the number of voxels quickly becomes huge.

The aim of this work is to present a new version of the FFT acceleration method that allows overcoming both limitations, permitting the use of voxels of different sizes along each Cartesian direction and making it possible to handle far-apart objects efficiently.

The work is organized as follows. Section II briefly summarizes the PEEC method describing the formulation for which the FFT technique is used. The new approach that allows us to overcome the limitations of the standard FFT-based technique in performing the matrix–vector products in the framework of an iterative solver is presented in Section III. In Section IV, the proposed technique is applied to two test cases requiring both a nonuniform voxelization; the second example also considers well-separated objects that could not be handled by the standard FFT approach. The conclusions are drawn in Section V.

II. Basic PEEC Formulation

The PEEC method [8] is based on the volume equivalence principle of Maxwell's equations. It uses the electric field integral equation (EFIE) and the continuity equation (CE) [38], providing a circuit interpretation of these equations in terms of partial elements, namely resistances, partial inductances, and coefficients of potential. For this reason, the possibility of studying the equivalent circuit in both the time and frequency domains into a SPICE-like circuit solver [39] distinguishes the PEEC method from other integral equation-based techniques such as the MoM [40].

In the PEEC method [8], volumes and surfaces are discretized into hexahedra and patches, respectively, where the current and charge densities are expanded into a series of basis functions. Usually, rectangular basis functions are chosen, meaning that the current and charge densities are assumed to be constant over the elementary volume and surface cells. Then, by applying the standard Galerkin's testing procedure, branches and nodes are generated, and electrical lumped elements are identified modeling both the magnetic and electric field coupling. Dielectrics are modeled by their excess capacitance taking the dielectric polarization into account [41] while conductors can be modeled by their ohmic resistance or through the introduction of a surface impedance [42]. Magnetic and electric field couplings are modeled by partial inductances and coefficients of potential, respectively.

Enforcing KVL and KCL to the PEEC circuit yields

$$\begin{bmatrix} \mathbf{Z}_s + j\omega\mathbf{L}_p & \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \mathbf{P} & -\mathbf{M}^T \\ -\mathbf{A}^T & j\omega\mathbf{M} & \mathbf{Y}_{\ell e} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} \\ \mathbf{Q} \\ \phi \end{bmatrix} = \begin{bmatrix} \mathbf{V}_s \\ \mathbf{0} \\ \mathbf{I}_s \end{bmatrix} \quad (1)$$

where

1) $\mathbf{A}$ is the incidence matrix;
2) $\mathbf{M}$ is a selection matrix introduced in [28];
3) $\mathbf{Y}_{\ell e}$ is the admittance matrix containing lumped elements;
4) $\mathbf{P}$ is the coefficients of the potential matrix;
5) $\mathbf{L}_p$ is the partial inductance matrix;
6) $\mathbf{Z}_s$ is a diagonal matrix with the self-impedances of elementary volumes computed as described in [37] to properly model the skin effect;
7) $\mathbf{V}_s$ is the distributed voltage sources vector due to external fields;
8) $\mathbf{I}_s$ is the lumped current sources vector;
9) $\mathbf{I}$, $\mathbf{Q}$, and ϕ are the unknown vectors representing the edge currents, the surface charges, and the potentials to infinity, respectively.

As it may happen that different surface patches share the same node, for example, as it happens at the corner of a conductor, it becomes necessary to enforce their equipotentiality and sum up their charges. Matrix $\mathbf{M}$ allows mapping surface patches, on which the charges are assumed to be located, to the circuit nodes, as described in detail in [28]. The use of the matrix $\mathbf{M}$ allows for avoiding the inversion of the matrix $\mathbf{P}$, which is, in general, time-consuming or even impossible for large problems and a potential cause of numerical inaccuracies. Furthermore, this formulation has been found very robust and reliable in terms of convergence properties when iterative solvers are used [28]. In (1), the dependency from $j\omega$ of $\mathbf{Z}_s, \mathbf{L}_p, \mathbf{P}$ matrices and of the right-hand sides is omitted for simplicity.

As previously mentioned, the major limitation of the FFT acceleration technique is that all the voxels must have the same size in each Cartesian direction (rectangular prisms), and, in addition, the air must be meshed and properly handled through a zero padding strategy when performing the matrix–vector products (the voxels made by air do not contribute to the number of unknowns but increase only the size of the circulant tensors used to perform the matrix–vector products). In Section III, a novel anisotropic and optimized FFT (aoFFT) technique is presented with the goal of lowering the number of empty voxels and decreasing the number of unknowns by using voxels having different sizes. When an iterative solver is used to solve the PEEC system of

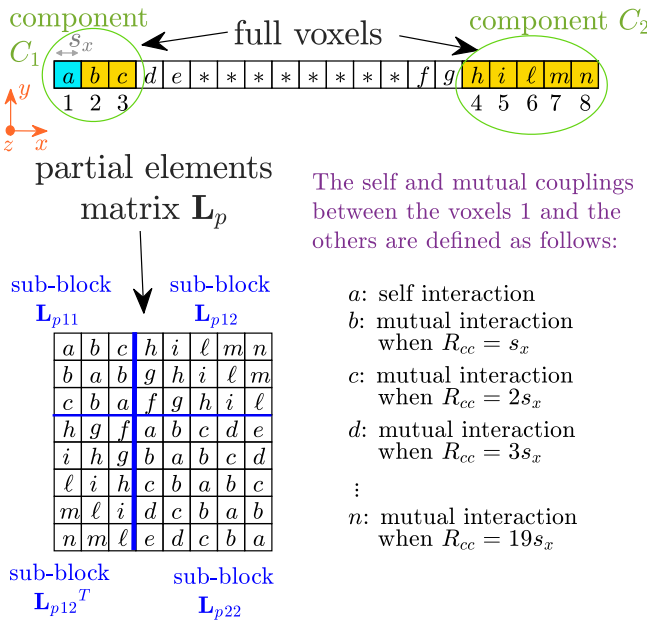


Fig. 1. Matrix $\mathbf{L}_p$ for a simple voxelized structure example.

equation (1), the convergence is reached through the following block partitioned preconditioner introduced in [37]:

$$\begin{bmatrix} \mathbf{D}_Z^{-1} - \mathbf{D}_Z^{-1}\mathbf{A}\mathbf{K}_1 & j\omega\mathbf{D}_Z^{-1}\mathbf{A}\mathbf{K}_2 & -\mathbf{D}_Z^{-1}\mathbf{A}\mathbf{K}_3^{-1} \\ \mathbf{D}_P^{-1}\mathbf{M}^T\mathbf{K}_1 & \mathbf{D}_P^{-1} - j\omega\mathbf{D}_P^{-1}\mathbf{M}^T\mathbf{K}_2 & \mathbf{D}_P^{-1}\mathbf{M}^T\mathbf{K}_3^{-1} \\ \mathbf{K}_1 & -j\omega\mathbf{K}_2 & \mathbf{K}_3^{-1} \end{bmatrix} \tag{2}$$

where

$$\mathbf{K}_1 = \mathbf{K}_3^{-1}\mathbf{A}^T\mathbf{D}_Z^{-1} \tag{3a}$$

$$\mathbf{K}_2 = \mathbf{K}_3^{-1}\mathbf{M}\mathbf{D}_P^{-1} \tag{3b}$$

$$\mathbf{K}_3 = \mathbf{Y}_{\ell e} + \mathbf{A}^T\mathbf{D}_Z^{-1}\mathbf{A} + j\omega\mathbf{M}\mathbf{D}_P^{-1}\mathbf{M}^T \tag{3c}$$

and where $\mathbf{D}_{L_p}$ and $\mathbf{D}_P$ are diagonal matrices having on the diagonals the diagonal elements of $\mathbf{L}_p$ and $\mathbf{P}$, respectively. Finally, since matrix $\mathbf{K}_3$ is sparse, its inversion can be performed by resorting to the sparse multifrontal LU factorization [37], [43] but, for large problems, this inversion requires large computational cost and time. In the numerical examples, that will be presented in Section IV, we will see how the proposed technique allows for reducing the CPU time and the memory required for the inversion of the matrix $\mathbf{K}_3$.

III. Matrix–Vector Product Through an aoFFT

The technique will be presented for the case of the partial matrix $\mathbf{L}_p$. Let us consider a simple structure voxelized into 20 voxels, for simplicity only along the x-direction, with a step size s_x as shown in Fig. 1, where eight voxels are defined by a material different than air and 12 voxels are empty (i.e., by air). Namely, the modeling domain consists of two components C_1 and C_2, where C_1 is defined by three voxels and C_2 by five voxels.

In Fig. 1, the matrix $\mathbf{L}_p$ is filled by exploiting the fact that a mutual coupling coefficient is the same for all the pairs of voxels having the same relative position (in this example, the knowledge of only 12 coefficients is required while the useless coefficients are marked as "*"). When an iterative solver is used to solve a linear system (1), the computational cost is dominated by the product between the dense matrices $\mathbf{L}_p$ and $\mathbf{P}$ with the vectors $\mathbf{I}^{\text{est}}$ and $\mathbf{Q}^{\text{est}}$, respectively, where $\mathbf{I}^{\text{est}}$ and $\mathbf{Q}^{\text{est}}$ are the estimated solution vectors for the currents and charges, respectively, at a generic iterative solver step. By referring to the example shown in Fig. 1, let us consider the following matrix–vector product:

$$\mathbf{X} = \mathbf{L}_p\mathbf{I}^{\text{est}} \tag{4}$$

where $\mathbf{X}$ simply denotes the vector result of the matrix–vector product and $\mathbf{I}^{\text{est}}$ can be written as

$$\mathbf{I}^{\text{est}} = \begin{bmatrix} i_1 & i_2 & i_3 & i_4 & i_5 & i_6 & i_7 & i_8 \end{bmatrix}^T. \tag{5}$$

When the matrix–vector product $\mathbf{L}_p\mathbf{I}^{\text{est}}$ is accelerated through the FFT-based approach [32], [36], first, two 3-D matrices (circulant tensors) of size $2N_x \times 2N_y \times 2N_z$ are stored for both $\mathbf{L}_p$ and $\mathbf{I}^{\text{est}}$ through their multidimensional FFT where N_x, N_y, and N_z denote the number of voxels along the x, y, and z Cartesian axis, respectively. Then, a 3-D matrix $\mathbf{X}_{\text{3-D}}$ is computed as the result of the member-to-member product between these two circulant tensors. Finally, the vector $\mathbf{X}$ is restored from $\mathbf{X}_{\text{3-D}}$ through its inverse FFT (IFFT). In this process, it is sufficient to compute only one row of matrix $\mathbf{L}_p$ including also empty voxels (voxels made by air). More precisely, the underlying behavior of the FFT-acceleration approach can be given by using a 2-D representation for the matrices by skipping the multidimensional FFT and IFFT steps. By referring to the example shown in Fig. 1, first, the vector $\mathbf{I}^{\text{est}}$ defined in (5) is expanded in the vector $\mathbf{I}^{\text{est}}_{\text{exp}}$, of size 20×1, as follows:

$$\mathbf{I}^{\text{est}}_{\text{exp}} = \begin{bmatrix} i_1 & i_2 & i_3 & 0 & \cdots & 0 & i_4 & i_5 & i_6 & i_7 & i_8 \end{bmatrix}^T \tag{6}$$

to include also the empty voxels. Such expansion can be simply achieved by introducing a sparse expansion matrix $\mathbf{E}$ of size 20×8 (total number of voxels $\times$ number of voxel defined by a material different than air) such that

$$\mathbf{I}^{\text{est}}_{\text{exp}} = \mathbf{E}\mathbf{I}^{\text{est}}. \tag{7}$$

Then, the row of $\mathbf{L}_p$ is filled with the self-interaction of voxel 1 and with all the mutual couplings between the voxel 1 with all the other voxels. At this point, the 2-D matrix $\mathbf{L}_p^{\text{FFT}}$, which describes in a comprehensive way the underline behavior of the FFT matrix-products acceleration, is filled as shown in Fig. 2.

It is important to remark that in the FFT acceleration, a 3-D circulant tensor is built for $\mathbf{L}_p$ by using only one row of $\mathbf{L}_p^{\text{FFT}}$, while matrix $\mathbf{L}_p^{\text{FFT}}$ is never explicitly computed and stored (here it is introduced only to give a comprehensive explanation). Finally, the result of the product in (4) is achieved as

$$\mathbf{X} = \mathbf{E}^T\left(\mathbf{L}_p^{\text{FFT}}\mathbf{I}^{\text{est}}_{\text{exp}}\right). \tag{8}$$

In Fig. 2, the subblocks of $\mathbf{L}_p$, shown in Fig. 1, are highlighted to show the correlation between $\mathbf{L}_p$ and $\mathbf{L}_p^{\text{FFT}}$. From Fig. 2, it is evident that if we have a large number of empty voxels

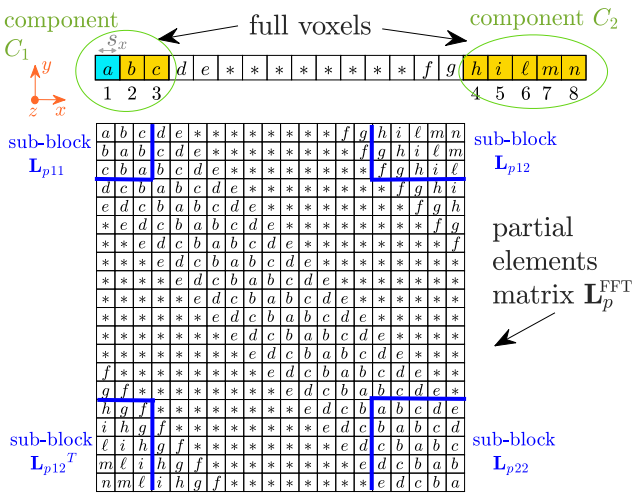


Fig. 2. Matrix $\mathbf{L}_p^{\text{FFT}}$ for a simple voxelized structure example.

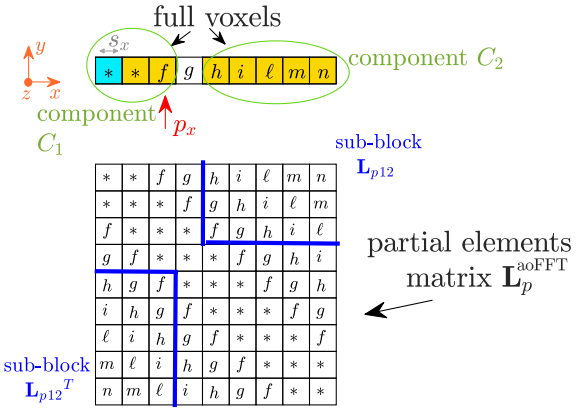


Fig. 3. Reduced voxelization and reduced partial elements matrix $\mathbf{L}_p^{\text{aoFFT}}$ for a simple structure example.

their inclusion can be avoided to save memory and accelerate the matrix–vector products. To this aim, the idea is to split the matrix–vector product (4) as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{11} + \mathbf{X}_{12} \\ \mathbf{X}_{21} + \mathbf{X}_{22} \end{bmatrix} \tag{9a}$$

$$\mathbf{X}_{11} = \mathbf{L}_{p_{11}} \mathbf{I}_1^{\text{est}} \tag{9b}$$

$$\mathbf{X}_{12} = \mathbf{L}_{p_{12}} \mathbf{I}_2^{\text{est}} \tag{9c}$$

$$\mathbf{X}_{21} = \mathbf{L}_{p_{12}}^T \mathbf{I}_1^{\text{est}} \tag{9d}$$

$$\mathbf{X}_{22} = \mathbf{L}_{p_{22}} \mathbf{I}_2^{\text{est}} \tag{9e}$$

$$\mathbf{I}_1^{\text{est}} = \begin{bmatrix} i_1 & i_2 & i_3 \end{bmatrix}^T \tag{9f}$$

$$\mathbf{I}_2^{\text{est}} = \begin{bmatrix} i_4 & i_5 & i_6 & i_7 & i_8 \end{bmatrix}^T. \tag{9g}$$

From (9), it follows that the computation of $\mathbf{X}_{11}$ and $\mathbf{X}_{22}$ can be trivially performed by applying the standard FFT acceleration twice: one time considering only the component 1 and one time considering only the component 2. The efficient computation of $\mathbf{X}_{12}$ and $\mathbf{X}_{21}$ is the main idea of this work and it will be presented in the rest of this section.

A. Mutual Coupling Products Between Different Components—1-D Case

By considering two components C_1 and C_2, the number of voxels in the x-direction for the first and the second component is N_{x_1} and N_{x_2}, respectively. The minimal number of voxels N_{T_x} required to build the circulant tensor, associated with matrix $\mathbf{L}_p$, that must be used to compute $\mathbf{X}_{12}$ or $\mathbf{X}_{21}$ in (9) is given by

$$N_{T_x} = 2\max(N_{x_1}, N_{x_2}) - 1. \tag{10}$$

By applying (10) to the example of Figs. 1 and 2, it follows that $N_{T_x} = 9$. Then, the starting position in which the mutual couplings for the subblock $\mathbf{L}_{p_{12}}$ must be placed is given by

$$p_x = N_{T_x} - N_{x_2} - N_{x_1} + 2. \tag{11}$$

For the example of Figs. 1 and 2, it follows that $p_x = 3$. At this point, the reduced voxelization is built as shown in Fig. 3 where in this case also the 2-D matrix $\mathbf{L}_p^{\text{aoFFT}}$ is introduced only to explain the technique in a comprehensive way because in practice a 3-D circulant tensor is built through the multidimensional FFT of one row of $\mathbf{L}_p^{\text{aoFFT}}$ [36].

It is important to underline that only $N_{T_x} = 9$ voxels are required for any distance between the components C_1 and C_2. For the fill-in of the row vector associated with the reduced voxelization of Fig. 3, the computation of the mutual coupling coefficients must be done preserving the original distance between the voxels. In particular, for the example of Figs. 1 and 2, the center-to-center distance between the first voxels of C_1 and C_2 is $R_{cc} = 15s_x$. Let us denote this original distance with $O_{R_{cc}}$. A mutual coefficient h is associated at the distance of $R_{cc} = 15s_x$ that must be preserved also when computing the coefficient in the reduced voxelization as shown in Fig. 3 where the first voxel of the component C_2 is located now in position $N_{T_x} - N_{x_2} + 1$.

Hence, it is straightforward to observe that a mutual coefficient between the first voxel of C_1 and another voxel in position $N_{T_x} - N_{x_2} + 1 + \Delta$, where Δ is an integer number (also negative), must be computed assuming a distance equal to $O_{R_{cc}} + s_x\Delta$.

Finally, $\mathbf{X}_{12}$ and $\mathbf{X}_{21}$, of (9), can be computed with the introduction of the matrix $\mathbf{L}_p^{\text{aoFFT}}$, instead of $\mathbf{L}_p^{\text{FFT}}$, in the following way:

$$\begin{bmatrix} \mathbf{X}_{12} \\ * \end{bmatrix} = \mathbf{L}_p^{\text{aoFFT}} \begin{bmatrix} \mathbf{0}_{4\times 1} \\ \mathbf{I}_2^{\text{est}} \end{bmatrix} \tag{12a}$$

$$\begin{bmatrix} * \\ \mathbf{X}_{21} \end{bmatrix} = \mathbf{L}_p^{\text{aoFFT}} \begin{bmatrix} \mathbf{I}_1^{\text{est}} \\ \mathbf{0}_{6\times 1} \end{bmatrix} \tag{12b}$$

where the not required subblocks are marked as "∗." In conclusion, from this simple 1-D example, it can be observed that $N_{T_x} = 9$ is not the minimum number of voxels since for this example the minimum number of voxels is 7. This little overhead is paid to have generic formulas (10) and (11) that can be used for any combination of values for N_{x_1} and N_{x_2}. Indeed, if we consider the case in which $N_{x_1} = N_{x_2} = N_x$, from (10) to (11), it follows that $N_{T_x} = 2N_x - 1$ and $p_x = 1$ that represents the minimum size of the reduced

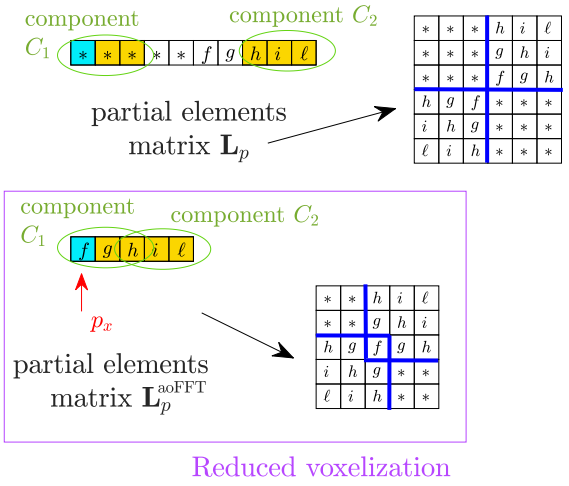


Fig. 4. Reduced voxelization for a 1-D case with $N_{x_2} = N_{x_1}$.

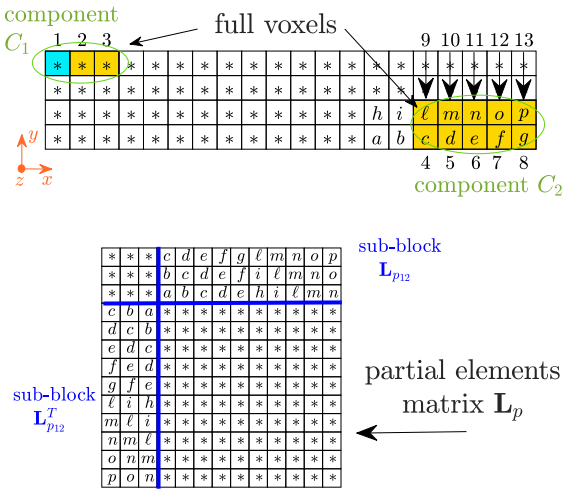


Fig. 5. Partial elements matrix $\mathbf{L}_p$ for a simple 2-D voxelized structure example.

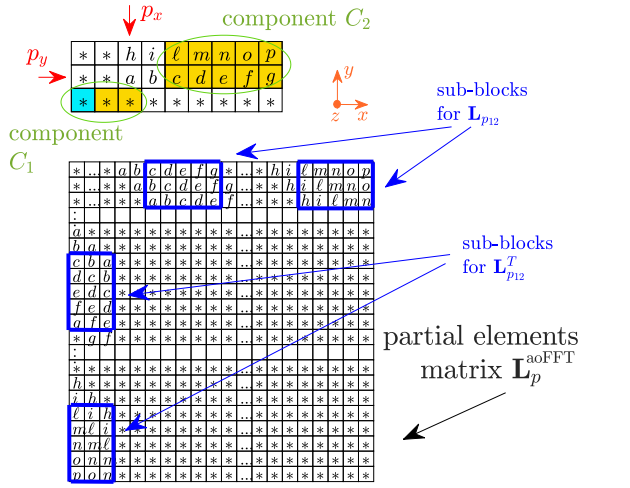


Fig. 6. Reduced voxelization and reduced partial elements matrix $\mathbf{L}_p^{\text{aoFFT}}$ for a simple 2-D voxelized structure example.

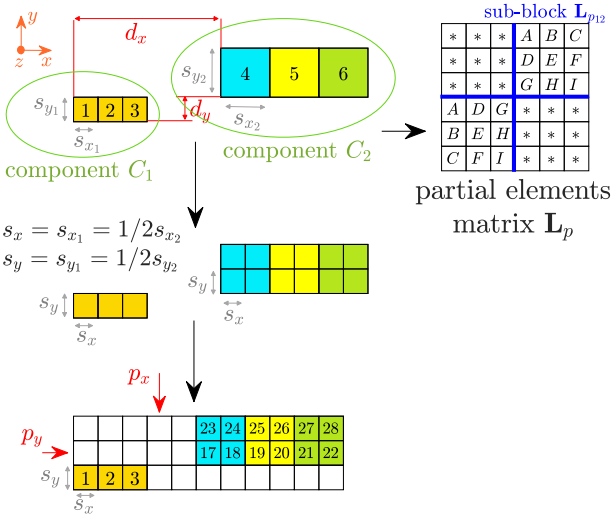


Fig. 7. Mutual coupling products between components with different sizes for a simple 2-D voxelized structure example.

voxelization as clearly can be seen from a simple example with $N_{x_1} = N_{x_2} = 3$ shown in Fig. 4.

In the next section, the 2-D case is analyzed to generalize the reduction strategy.

B. Mutual Coupling Products Between Different Components—2-D Case

Let us consider the voxelization shown in Fig. 5 and its mutual coupling matrix $\mathbf{L}_p$.

The reduction strategy introduced in Section III-A must be applied now in the x- and y-directions. Hence, applying (10) and (11), it follows that the reduced voxelization will have $N_{T_x} = 9$, $p_x = 3$, $N_{T_y} = 3$, and $p_y = 2$ as shown in Fig. 6.

It can be noted that, in this more general case, the matrix $\mathbf{L}_p^{\text{aoFFT}}$ has a partitioned subblock $\mathbf{L}_{p_{12}}$. For this reason, in the general case, the computation of $\mathbf{X}_{12}$ and $\mathbf{X}_{21}$ in (9) must be performed with a straightforward partitioning strategy also for the right-hand sides $\mathbf{I}_1^{\text{est}}$ and $\mathbf{I}_2^{\text{est}}$.

It is important to underline that if in one direction the components C_1 and C_2 have overlapped coordinates, the proposed reduction strategy can be applied but there will be no advantages in such operation (the proposed examples for the 1-D case in Section III-A are applied only in the x-direction and not in the y-direction).

In conclusion, the extension of the proposed reduction strategy, introduced in Section III-A, to all three directions, is simply achieved by applying the 1-D case in each direction with the same considerations made for the 2-D case. In addition, the reduction process is applied only for two components C_1 and C_2 but in the general case it must be applied in the same way for each pair of components.

C. Mutual Coupling Products Between Components With Different Sizes

Let us consider the case of two different 2-D components as shown in the upper part of Fig. 7 where the first component meshes with voxels having sizes s_{x_1} and s_{y_1} while the second one meshes with voxels having sizes s_{x_2} and s_{y_2}.

If the matrix–vector product is split as reported in (9), it follows that the self-products $\mathbf{L}_{p_{11}}\mathbf{I}_1^{\text{est}}$ and $\mathbf{L}_{p_{22}}\mathbf{I}_2^{\text{est}}$ can be

performed by applying the standard FFT matrix–vector product acceleration inside each component. Hence, in this case, also, the mutual coupling coefficients between different components must be properly handled. In particular, the matrix–vector products can be accelerated through the use of the FFT if, for each pair of components, the voxel sizes, in each Cartesian direction, can be expressed as multiple of a certain common size s_x, s_y, and s_z

$$k_{x_1} s_x = s_{x_1}; \quad k_{x_2} s_x = s_{x_2} \tag{13a}$$
$$k_{y_1} s_y = s_{y_1}; \quad k_{y_2} s_y = s_{y_2} \tag{13b}$$
$$k_{z_1} s_z = s_{z_1}; \quad k_{z_2} s_x = s_{z_2} \tag{13c}$$

where k_{x_1}, k_{x_2}, k_{y_1}, k_{y_2}, k_{z_1}, and k_{z_2} are integer numbers. In the example of Fig. 7, we simply assume that $k_{x_1} = k_{y_1} = 1$ and $k_{x_2} = k_{y_2} = 2$. Let us consider now the following standard evaluation of the matrix $\mathbf{L}_p$ coefficients between two inductive cells γ and δ under the quasistatic hypothesis and by assuming for simplicity a 0 thickness along the z-axis:

$$L_{p_{\gamma\delta}} = \frac{\mu_0}{4\pi \ell_\gamma \ell_\delta} \int_{S_\gamma} \int_{S_\delta} \frac{1}{R_{\gamma\delta}} dS_\gamma dS_\delta \tag{14}$$

where $R_{\gamma\delta}$ is the distance between any two points on surfaces γ and δ, while ℓ_γ and ℓ_δ denote the length of the edges that are orthogonal to the current direction. By assuming that the currents flow in the x-direction and by referring to Fig. 7, we can write the mutual coefficient A between the voxels 1 and 4 as

$$A = L_{p_{1,4}} = \frac{\mu_0}{4\pi s_{y_1} s_{y_2}} \int_0^{s_{x_1}} \int_0^{s_{y_1}} \int_{d_x}^{d_x+s_{x_2}} \int_{d_y}^{d_y+s_{y_2}} \times \frac{1}{R(x, y, x', y')} dx dy dx' dy'. \tag{15}$$

In the case, in which we split the voxels of the second component (as reported in the middle of Fig. 7, it follows that (15) can be rewritten as

$$\begin{aligned} A = \frac{\mu_0}{4\pi s_{y_1} s_{y_2}} \Bigg[& \int_0^{s_x} \int_0^{s_y} \int_{d_x}^{d_x+s_x} \int_{d_y}^{d_y+s_y} \frac{1}{R(x, y, x', y')} \times dx dy dx' dy' \\ & + \int_0^{s_x} \int_0^{s_y} \int_{d_x+s_x}^{d_x+2s_x} \int_{d_y}^{d_y+s_y} \frac{1}{R(x, y, x', y')} \times dx dy dx' dy' \\ & + \int_0^{s_x} \int_0^{s_y} \int_{d_x}^{d_x+s_x} \int_{d_y+s_y}^{d_y+2s_y} \frac{1}{R(x, y, x', y')} \times dx dy dx' dy' \\ & + \int_0^{s_x} \int_0^{s_y} \int_{d_x+s_x}^{d_x+2s_x} \int_{d_y+s_y}^{d_y+2s_y} \frac{1}{R(x, y, x', y')} \times dx dy dx' dy' \Bigg]. \end{aligned} \tag{16}$$

Hence, by splitting the integral (15) as shown in (16), it follows that the decoupling strategy can be adopted as reported in Sections III-A and III-B and in the lower part of Fig. 7 with the only difference that the voxels of the starting mesh must be properly mapped into the split voxels. Practically, the standard mutual matrix–vector product

$$\mathbf{L}_{p_{12}} \mathbf{I}_2^{\text{est}} = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} \cdot \begin{bmatrix} i_4 \\ i_5 \\ i_6 \end{bmatrix} \tag{17}$$

can be performed via FFT as

$$\underset{3\times 28}{\mathbf{G}^T} \left(\underset{28\times 28}{\mathbf{L}_p^{\text{aoFFT}}} \left(\underset{28\times 3}{\mathbf{G}} \underset{3\times 1}{\mathbf{I}_2^{\text{est}}} \right) \right) \tag{18}$$

where $\mathbf{L}_p^{\text{aoFFT}}$ is built on the uniform voxelization shown in the lower part of Fig. 7 and where $\mathbf{G}$ is a sparse matrix having nonzeros elements, equal to 1, only at:

1) column 1 and rows 17, 18, 23, 24;
2) column 2 and rows 19, 20, 25, 26;
3) column 3 and rows 21, 22, 27, 28.

It is evident that having components meshed with different voxel sizes does not give any speed-up for off-diagonal products but it allows us to have a reduced number of unknowns (the number of voxels full of material) of the system (in the example of Fig. 7, we have six unknowns in the case of components meshed with different sizes while in the case of components meshed with same sizes, we have 15 unknowns). A low number of unknowns allows having a faster preconditioner setup, fewer memory requirements for its storage, and, usually, a lower number of iterative solver iterations (fewer unknowns must reach the convergence). Such advantages will be shown in the examples reported in Section V.

As a final mark, it is important to underline that it is not required to split explicitly the voxels when performing the proposed FFT matrix–vector product, but it is sufficient to apply only a proper mapping of the voxels (in the example of Fig. 7, the voxel 4 in the proposed FFT approach will be represented by voxels 17, 18, 23, and 24).

IV. Numerical Results

To validate the proposed aoFFT acceleration technique introduced in Section III and the preconditioning and scaling strategies, for iterative solvers, reported in [37], two practical applications are presented in this section. All the simulations have been carried out on a computer equipped with 1 TB of RAM and a 32-core AMD processor operating at 3 GHz.

A. PCB Loop Example

In this example, a PCB loop presented in [37] is analyzed again by resorting to the aoFFT acceleration technique introduced in Section III. The model is shown in Fig. 8. The blue line represents a lumped current port modeled as equipotential squares of 0.1 mm^2 centered around the red dots. The relative permittivity of the dielectric is 4.4.

The entire structure has been analyzed, from 100 Hz to 1 GHz, with the following methods.

1) The FW-PEEC method [37] with and without the scaling strategy proposed in [37] reported also in Table I for completeness. In this case, the mesh used for the standard FFT approach is made by a 3-D grid having $N_x = 67$, $N_y = 110$, and $N_z = 92$, where N_x, N_y, and

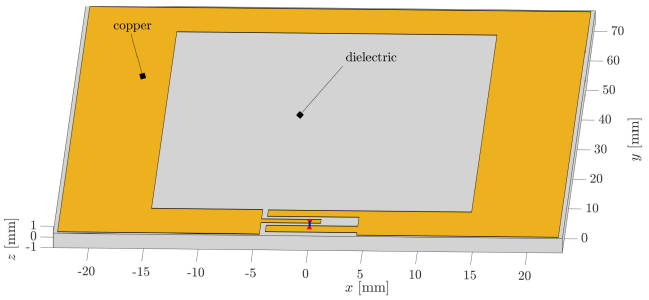


Fig. 8. Geometry of the PCB loop example. The thickness of the conductors is 0.0354 mm, the thickness of the dielectric is 1.55 mm, and the shoulders is 0.381 mm. More geometrical details can be found in [37].

TABLE I
SCALED UNITS

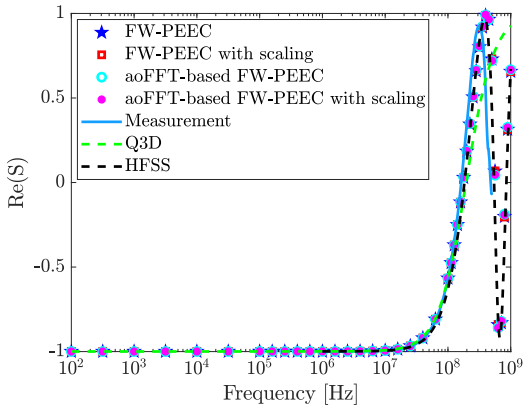
Voltage	Current	Charge	$\mathbf{P}$	$\mathbf{C}_d$	$\mathbf{R}$	$\mathbf{L}_p$	f
V	mA	pC	pF^{-1}	pF	kΩ	μH	GHz

N_z are the number of voxels along the x-, y-, and z-axis, respectively.

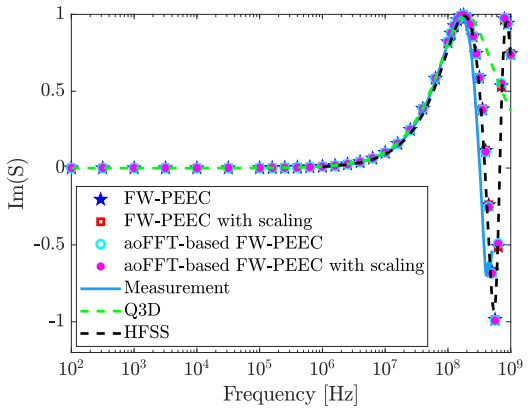
2) The aoFFT-based FW-PEEC method with and without the scaling units of Table I with a mesh made by a 3-D grid having $N_x = 67$, $N_y = 110$, and $N_z = 8$. It implements the aoFFT strategy reported in Section III. In particular, we identify three connected components: the ground (meshed with $N_z = 2$), the dielectric (meshed with $N_z = 4$), and the upper copper loop part (meshed with $N_z = 2$). To satisfy the conditions (13), the thickness of the dielectric is increased from 1.55 to 1.5576 mm (in this way, the thickness of the dielectric is exactly 44 times greater than the thickness of the conductors). It is important to underline that since the maximum frequency is 1 GHz, the thickness of 0.3894 mm for the voxels of the dielectric satisfies the $\lambda/40$ criterion condition, that is, $0.3894 \cdot 10^{-3} < (c_0/40 \cdot 10^9)$.
3) Q3D denotes the quasistatic R-L solution obtained with the commercial software ANSYS Q3D extractor that assumes that the resistance increases with frequency as $\sqrt{f}$.
4) The commercial software ANSYS HFSS, based on the FEM, computes the solution for frequencies larger than 1 MHz.

The thresholds for the GMRES convergence have been set to 10^{-4} for all the PEEC methods. The scattering parameters, evaluated using all these methods, are reported in Fig. 9 in which also the comparison with the measurements, performed using a Keysight E4991B Impedance Analyzer operating from 1 to 500 MHz, is shown.

As can be clearly seen, a very good agreement between all the methods and the measurements is achieved (the little difference in measurements is due to the excitation provided via a BNC connector, which is not included in the simulated model). It can be noted that the skin effect is well caught by using the surface impedance introduced in [37] and that the PEEC methods are accurate over the whole frequency range.



(a) Real part.



(b) Imaginary part.

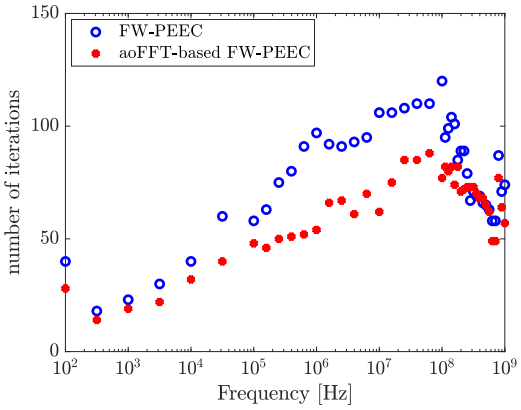
Fig. 9. Scattering parameters for the PCB loop example shown in Fig. 8. (a) Real part. (b) Imaginary part.

TABLE II
NUMBER OF UNKNOWNS AND EXPERIMENTAL RESULTS FOR THE PCB LOOP EXAMPLE

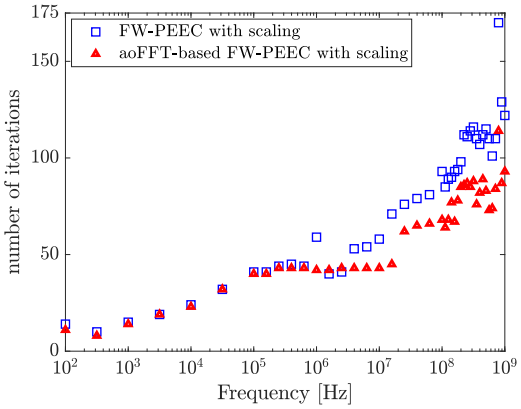
	FW-PEEC	aoFFT-based FW-PEEC
Inductive branches #	1974519	131473
Capacitive patches #	58537	28801
Nodes #	659802	39379
Setup time	9 s	32 s
$\mathbf{L}_p$ filling time	18 s	24 s
$\mathbf{P}$ filling time	3 min	4 min
$\mathbf{L}_p$ storage	450 MB	957 MB
$\mathbf{P}$ storage	1.3 GB	3 GB
$\mathbf{L}_p$-vector product time	1.2 s	2.1 s
$\mathbf{P}$-vector product time	4 s	6 s
$\mathbf{K}_3$ (3c) computation time	10 min	1 s
$\mathbf{K}_3$ (3c) computation: peak of memory reached	106 GB	400 MB

The number of unknowns and experimental results for both the FW-PEEC and aoFFT-based FW-PEEC methods are summarized in Table II, where matrix $\mathbf{K}_3$ is the sparse matrix reported into (3c) in which the inversion is performed by resorting to the sparse multifrontal LU factorization [37], [43]. In Table II, the setup time denotes the time spent to perform all the operations required to fill all the sparse matrices of the system (1) including also the time for meshing.

From these results, it follows that for geometries in which we use the aoFFT only to have different voxel sizes (without



(a) Number of iterations without the scaling units.



(b) Number of iterations with the scaling units.

Fig. 10. Number of iterations of the implemented PEEC iterative solvers for the PCB loop example shown in Fig. 8. Number of iterations (a) without the scaling units and (b) with the scaling units.

removing big regions of air), the aoFFT requires, in general, much more memory storage for the circulant tensors and CPU time for the setup and matrix–vector products than the standard FFT approach but, since it allows to have a lower number of unknowns, the inversion of $\mathbf{K}_3$ (3c), required to build the preconditioner, is accelerated both in terms of CPU time and memory requirements. In general, when FFT or aoFFT is used, the $\mathbf{P}$ fill-in is slower than $\mathbf{L}_p$ because, to model parallel and orthogonal surfaces and their combinations, nine circulant tensors are required while $\mathbf{L}_p$ requires only one circulant tensor for each Cartesian direction [34], [35], [37]).

Finally, the number of GMRES iterations for the PEEC methods are shown in Fig. 10 where it can be clearly seen that the PEEC methods adopting the aoFFT strategy require, in general, a lower number of iterations as they have a lower number of unknowns. In addition, it can be seen that the scaling strategy, reported in Table I, can be adopted until 100 MHz to improve the convergence of the iterative GMRES algorithm.

B. WPT Antennas Example

In this example, a wireless power transfer (WPT) square antenna, introduced in [44], is analyzed. The WPT system consists of two identical printed square antennas symmetric facing each other where the receiving antenna is 12 cm away

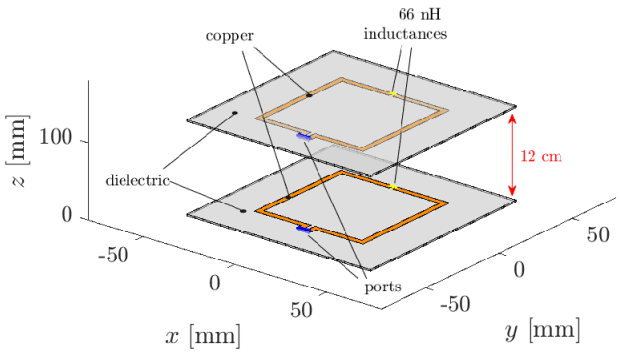


Fig. 11. Geometry of the WPT antennas example. More geometrical details can be found in [44].

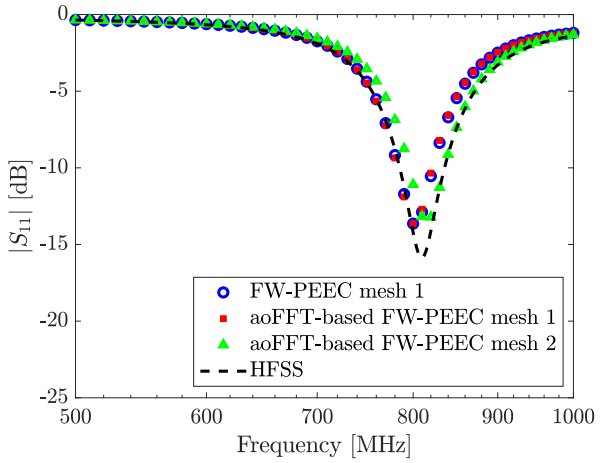
from the transmitting antenna. The model is shown in Fig. 11 where the blue lines represent the lumped current ports with 50-Ω reference impedance while the yellow lines represent two 66-nH inductors (the gap of the inductor and the lumped port is set to 0.2 cm). The thickness of the loop cooper strip is 50 μm and it is printed on a 2-mm-thick FR-4 substrate having a relative permittivity of 4.4 and loss tangent tanδ 0.2.

By considering the 66-nH inductors and the ports as edges for the structures, it follows that four connected components are identified.

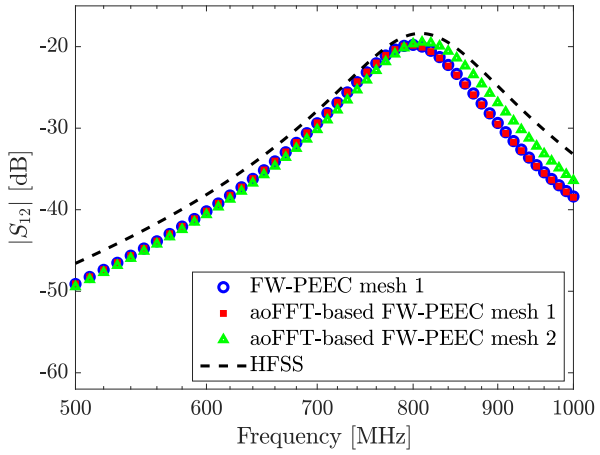
1) The dielectric substrate of the transmitter antenna.
2) The copper loop of the transmitter antenna.
3) The dielectric substrate of the receiver antenna.
4) The copper loop of the receiver antenna.

Then, the bounding boxes of the copper loops are expanded in the xy-plane to match the bounding boxes of the dielectric substrates. This is done to guarantee the matching and the continuity of the nodes between the touching connected components. The analysis was performed from 500 MHz to 1 GHz, with the following methods.

1) The FW-PEEC method [37] (without any scaling strategy since it is helpful only at low frequencies). The sizes of the voxels along the x-, y-, and z-axes are $s_x = 746$ μm, $s_y = 746$ μm, and $s_z = 25$ μm, respectively. In particular, the mesh is achieved through a rectangular 3-D grid having $N_x = 134$, $N_y = 134$, and $N_z = 4960$, and N_x, N_y, and N_z are the number of voxels along the x-, y-, and z-axes, respectively;
2) The aoFFT-based FW-PEEC method (without any scaling strategy) implements the aoFFT strategy reported in Section III. For this method, we use two meshes.
   a) The first mesh (denoted as mesh 1) where all the components are meshed with $s_x = 746$ μm and $s_y = 746$ μm ($N_x = 134$ and $N_y = 134$). Then, the conductors are meshed with $s_z = 25$ μm ($N_z = 2$), while the dielectric parts are meshed with $s_z = 125$ μm ($N_z = 16$). With these settings, the $\lambda/40$ criterion condition is satisfied. Practically, such a mesh is equivalent to that of the FW-PEEC method.
   b) The second mesh (denoted as mesh 2) is finer than mesh 1. In detail, all the components are meshed with $s_x = 500$ μm and $s_y = 500$ μm ($N_x = 200$ and $N_y = 200$). Then, the



(a) $|S_{11}|$.



(b) $|S_{12}|$.

Fig. 12. Scattering parameters for the WPT antennas example shown in Fig. 11. (a) $|S_11|$. (b) $|S_12|$.

conductors are meshed with $s_z = 25$ μm ($N_z = 2$), while the dielectric parts are meshed with $s_z = 25$ μm ($N_z = 16$). Again, with these settings, the $\lambda/40$ criterion condition is satisfied.

For both mesh 1 and mesh 2, the mutual products between the conductor and the dielectric components are performed with $s_z = 125$ μm, to satisfy the conditions (13), as explained in Section III-C.

3) The commercial software ANSYS HFSS.

In this case, also, the thresholds for the GMRES convergence have been set to 10^{-4} for all the PEEC methods. The scattering (S-) parameters evaluated using all these methods are reported in Fig. 12. It is evident that the proposed aoFFT-based FW-PEEC method using mesh 2 shows a better agreement with HFSS than the PEEC methods using mesh 1 but the FW-PEEC method cannot be implemented by decreasing s_x and s_y due to RAM limitations.

The number of unknowns and experimental results for both the FW-PEEC and aoFFT-based FW-PEEC methods are summarized in Table III where the same notation of Table II has been used.

From the presented results, it follows that the proposed FFT-based approach strongly outperforms the standard FFT one in computing the matrix–vector products for the modeling problems with a large portion of air between the objects and requiring voxels of different sizes.

TABLE III

NUMBER OF UNKNOWNS AND EXPERIMENTAL RESULTS FOR THE WPT ANTENNAS EXAMPLE

	FW-PEEC mesh 1	aoFFT-based FW-PEEC mesh 1	aoFFT-based FW-PEEC mesh 2
Inductive branches #	8549772	1688972	3770772
Capacitive patches #	162544	93936	194816
Nodes #	2874136	575768	1283584
Setup time	6 min	88 s	7 min
$\mathbf{L}_p$ filling time	8 min	33 s	93 s
$\mathbf{P}$ filling time	8 min	2 min	5 min
$\mathbf{L}_p$ storage	34 GB	3.5 GB	7.8 GB
$\mathbf{P}$ storage	102 GB	10.4 GB	23.4 GB
$\mathbf{L}_p$-vector product time	39 s	9 s	24 s
$\mathbf{P}$-vector product time	2 min	27 s	72 s
$\mathbf{K}_3$ (3c) computation time	73 min	94 s	6 min
$\mathbf{K}_3$ (3c) computation: peak of memory reached	760 GB	18 GB	87 GB

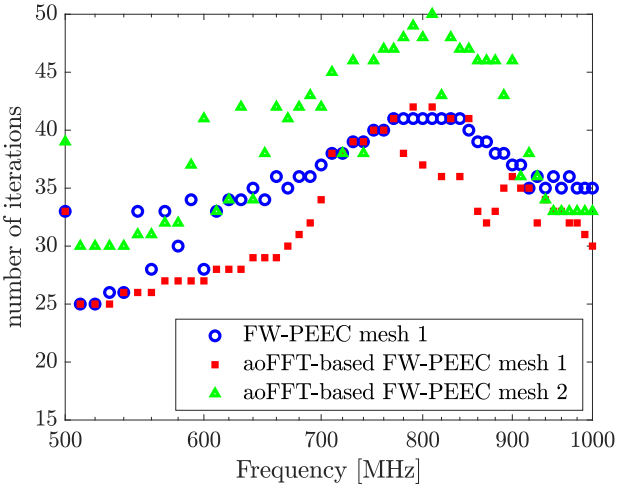


Fig. 13. Number of iterations of the implemented PEEC iterative solvers for the WPT antennas example shown in Fig. 11.

In addition, from Table III, it can be noted that the storage of the circulant tensors for matrices $\mathbf{L}_p$ and $\mathbf{P}$ requires several gigabytes. The mutual aoFFT circulant tensors, describing the mutual coupling between basis functions inside different components, have a low-rank behavior [45] and their storage can be improved by performing a Tucker decomposition combined with SVD or ACA [45], [46]. It guarantees very high compression ratios without losing accuracy (usually tensors of the order of gigabytes in compressed format occupy a few megabytes [45], [46]) but, to perform matrix–vector products via FFT, the tensors must be first decompressed. This limitation will be the subject of future work by the authors with the aim to achieve a computational speed-up in matrix–vector products via FFT by exploiting a compressed form.

It is important to underline also that the performance of the aoFFT-based FW-PEEC methods does not change for any distance between the two antennas for the reasons given in Section III.

Finally, the number of GMRES iterations for the PEEC methods is shown in Fig. 13. It can be seen that the PEEC methods adopting the aoFFT strategy require, in general, a lower number of iterations for equivalent meshes.

V. Conclusion

This work presents an aoFFT-based iterative PEEC solver introduced to speed up the matrix–vector products and overcome two major limitations of the state-of-the-art FFT-based PEEC solver, that is, a uniform mesh with voxels of the same sizes in each Cartesian direction and a requirement to mesh unnecessary portion of air between the objects. From the numerical experiments, it follows that the proposed aoFFT method allows achieving high CPU time and memory saving in the preconditioning step while a little overhead of CPU time and memory requirements is paid in performing matrix–vector products if the method is applied to problems with different mesh sizes in the three Cartesian directions. In any case, such little overhead is compensated by the fact that an iterative method requires fewer iterations to reach the convergence since the number of unknowns is smaller. Furthermore, if a significant portion of air is removed, the proposed method allows a significant speed-up in performing matrix–vector products. This means that problems that are computationally time- and memory-demanding can be analyzed faster even on computers equipped with much less-performing hardware. In conclusion, the developed aoFFT method, applied in this work for the PEEC method, is a general technique that can be applied to improve the standard FFT matrix–vector product acceleration in any simulation environment requiring iterative solvers.

References

[1] K. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propag.*, vol. AP-14, no. 3, pp. 302–307, May 1966.

[2] J. Jin, *The Finite Element Method in Electromagnetics*. Hoboken, NJ, USA: Wiley, 1993.

[3] R. F. Harrington, *Field Computation by Moment Methods*. New York, NY, USA: Macmillan, 1968.

[4] T. Weiland, "A discretization method for the solution of Maxwell's equations for six-component fields," *Electron. Commun. AEU*, vol. 31, no. 3, pp. 116–120, Mar. 1977.

[5] A. E. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," *IEEE Trans. Microw. Theory Techn.*, vol. MTT-22, no. 3, pp. 216–221, Mar. 1974.

[6] A. E. Ruehli, "Inductance calculations in a complex integrated circuit environment," *IBM J. Res. Develop.*, vol. 16, no. 5, pp. 470–481, Sep. 1972.

[7] A. E. Ruehli and P. A. Brennan, "Efficient capacitance calculations for three-dimensional multiconductor systems," *IEEE Trans. Microw. Theory Techn.*, vol. MTT-21, no. 2, pp. 76–82, Feb. 1973.

[8] A. E. Ruehli, G. Antonini, and L. Jiang, *Circuit Oriented Electromagnetic Modeling Using the PEEC Techniques*. Hoboken, NJ, USA: Wiley, 2017.

[9] J. Kim, J. Fan, A. E. Ruehli, J. Kim, and J. L. Drewniak, "Inductance calculations for plane-pair area fills with vias in a power distribution network using a cavity model and partial inductances," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 8, pp. 1909–1924, Aug. 2011.

[10] L. Wei et al., "Plane-pair PEEC model for power distribution networks with sub-meshing techniques," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 3, pp. 733–741, Mar. 2016.

[11] T. Makharashvili et al., "Circuit models for the inductance of eight-terminal decoupling capacitors," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 10, no. 1, pp. 142–150, Jan. 2020.

[12] T. Makharashvili et al., "Accurate inductance models of mounted two-terminal decoupling capacitors," *IEEE Trans. Electromagn. Compat.*, vol. 63, no. 1, pp. 237–245, Feb. 2021.

[13] Y. S. Cao et al., "Inductance extraction for PCB prelayout power integrity using PMSR method," *IEEE Trans. Electromagn. Compat.*, vol. 59, no. 4, pp. 1339–1346, Aug. 2017.

[14] C. Martin, J.-L. Schanen, J.-M. Guichon, and R. Pasterczyk, "Analysis of electromagnetic coupling and current distribution inside a power module," *IEEE Trans. Ind. Appl.*, vol. 43, no. 4, pp. 893–901, Jul. 2007.

[15] L. Horowitz, N. Pallo, S. Coday, and R. C. N. Pilawa-Podgurski, "A method of partial inductances to evaluate and optimize switching cells," in *Proc. IEEE Appl. Power Electron. Conf. Exposit. (APEC)*, Jun. 2021, pp. 1549–1554.

[16] M. W. Beattie and L. T. Pileggi, "On-chip induction modeling: Basics and advanced methods," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 6, pp. 712–729, Dec. 2002.

[17] N. Xia and Y. Du, "Reduction of PEEC unknowns for 3D metallic plates in magnetic shielding," *IEEE Trans. Magn.*, vol. 49, no. 5, pp. 2001–2004, May 2013.

[18] Y. S. Cao, L. J. Jiang, and A. E. Ruehli, "Distributive radiation and transfer characterization based on the PEEC method," *IEEE Trans. Electromagn. Compat.*, vol. 57, no. 4, pp. 734–742, Aug. 2015.

[19] Y. Dou and K.-L. Wu, "Nature of antenna radiation revealed by physical circuit model," *IEEE Trans. Antennas Propag.*, vol. 69, no. 1, pp. 84–96, Jan. 2021.

[20] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *J. Comput. Phys.*, vol. 73, no. 2, pp. 325–348, Dec. 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0021999187901409

[21] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propag. Mag.*, vol. 35, no. 3, pp. 7–12, Jun. 1993.

[22] J. M. Song, C. C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.

[23] S. Kapur and D. E. Long, "IES^3: A fast integral equation solver for efficient 3-dimensional extraction," in *Proc. Int. Conf. Comput. Aided Design*, San Jose, CA, USA, Nov. 1997, p. 448–.455.

[24] M. Bebendorf, "Approximation of boundary element matrices," *Numerische Math.*, vol. 86, no. 4, pp. 565–589, Oct. 2000.

[25] M. Bebendorf and S. Rjasanow, "Adaptive low-rank approximation of collocation matrices," *Computing*, vol. 70, no. 1, pp. 1–24, Feb. 2003.

[26] K. Zhao, M. N. Vouvakis, and J.-F. Lee, "The adaptive cross approximation algorithm for accelerated method of moments computations of EMC problems," *IEEE Trans. Electromagn. Compat.*, vol. 47, no. 4, pp. 763–773, Nov. 2005.

[27] W. Hackbusch and B. Khoromskij, "A sparse matrix arithmetic based on H-matrices. Part I: Introduction of H-matrices," *Computing*, vol. 62, pp. 89–108, Apr. 1999.

[28] G. Antonini and D. Romano, "Adaptive-cross-approximation-based acceleration of transient analysis of quasi-static partial element equivalent circuits," *IET Microw., Antennas Propag.*, vol. 9, no. 7, pp. 700–709, May 2015.

[29] D. Voltolina, R. Torchio, P. Bettini, R. Cavazzana, and M. Moresco, "PEEC modeling of planar spiral resonators," *IEEE Trans. Magn.*, vol. 56, no. 1, pp. 1–4, Jan. 2020.

[30] G. Antonini and D. Romano, "Efficient frequency-domain analysis of PEEC circuits through multiscale compressed decomposition," *IEEE Trans. Electromagn. Compat.*, vol. 56, no. 2, pp. 454–465, Apr. 2014.

[31] D. Voltolina, P. Bettini, P. Alotto, F. Moro, and R. Torchio, "High-performance PEEC analysis of electromagnetic scatterers," *IEEE Trans. Magn.*, vol. 55, no. 6, pp. 1–4, Jun. 2019.

[32] A. G. Polimeridis, J. F. Villena, L. Daniel, and J. K. White, "Stable FFT-JVIE solvers for fast analysis of highly inhomogeneous dielectric objects," *J. Comput. Phys.*, vol. 269, pp. 280–296, Jul. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999114002071

[33] A. C. Yucel, I. P. Georgakis, A. G. Polimeridis, H. Bagci, and J. K. White, "VoxHenry: FFT-accelerated inductance extraction for voxelized geometries," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 4, pp. 1723–1735, Apr. 2018.

[34] M. Wang, C. Qian, J. K. White, and A. C. Yucel, "VoxCap: FFT-accelerated and tucker-enhanced capacitance extraction simulator for voxelized structures," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 12, pp. 5154–5168, Dec. 2020.

[35] M. Wang, C. Qian, E. Di Lorenzo, L. J. Gomez, V. Okhmatovski, and A. C. Yucel, "SuperVoxHenry: Tucker-enhanced and FFT-accelerated inductance extraction for voxelized superconducting structures," *IEEE Trans. Appl. Supercond.*, vol. 31, no. 7, pp. 1–11, Oct. 2021.

[36] R. Torchio, F. Lucchini, J.-L. Schanen, O. Chadebec, and G. Meunier, "FFT-PEEC: A fast tool from CAD to power electronics simulations," *IEEE Trans. Power Electron.*, vol. 37, no. 1, pp. 700–713, Jan. 2022.

[37] D. Romano, I. Kovacevic-Badstuebner, G. Antonini, and U. Grossner, "Efficient PEEC iterative solver for power electronic applications," *IEEE Trans. Electromagn. Compat.*, vol. 65, no. 2, pp. 546–554, Apr. 2023.

[38] C. A. Balanis, *Advanced Engineering Electromagnetics*. New York, NY, USA: Wiley, 1989.

[39] L. W. Nagel, "SPICE: A computer program to simulate semiconductor circuits," Electr. Res. Lab., Univ. California, Berkeley, CA, USA, Tech. Rep. ERL M520, May 1975.

[40] R. F. Harrington, *Field Computation by Moment Methods*. Malabar: Krieger, 1982.

[41] A. E. Ruehli and H. Heeb, "Circuit models for three-dimensional geometries including dielectrics," *IEEE Trans. Microw. Theory Techn.*, vol. 40, no. 7, pp. 1507–1516, Jul. 1992.

[42] M. Bandinelli et al., "A surface PEEC formulation for high-fidelity analysis of the current return networks in composite aircrafts," *IEEE Trans. Electromagn. Compat.*, vol. 57, no. 5, pp. 1027–1036, Oct. 2015.

[43] T. A. Davis and I. S. Duff, "An unsymmetric-pattern multifrontal method for sparse LU factorization," *SIAM J. Matrix Anal. Appl.*, vol. 18, no. 1, pp. 140–158, Jan. 1997.

[44] W. Xiong, M. Jiang, G. Huang, and H. Chen, "Analysis on transfer efficiency of five different antenna configurations in short-distance wireless power transfer," in *Proc. IEEE Int. Conf. Electron Devices Solid State Circuits (EDSSC)*, Jun. 2018, pp. 1–2.

[45] I. I. Giannakopoulos et al., "Compression of volume-surface integral equation matrices via tucker decomposition for magnetic resonance applications," *IEEE Trans. Antennas Propag.*, vol. 70, no. 1, pp. 459–471, Jan. 2022.

[46] M. Wang, C. Qian, and A. C. Yucel, "Tensor decompositions applied to electromagnetics: A review," in *Proc. 3rd URSI Atlantic Asia Pacific Radio Sci. Meeting (AT-AP-RASC)*, May 2022, pp. 1–4.

Daniele Romano was born in Campobasso, Italy, in 1984. He received the Laurea degree in computer science and automation engineering and the Ph.D. degree in industrial engineering from the University of L'Aquila, L'Aquila, Italy, in 2012 and 2018, respectively.

Since 2012, he has been with the UAq EMC Laboratory, University of L'Aquila, where he is currently a Researcher, focusing on EMC modeling and analysis, algorithm engineering, and speed-up techniques applied to EMC problems.

Ivana Kovacevic-Badstuebner (Senior Member, IEEE) received the Ph.D. degree from ETH Zürich, Zürich, Switzerland, in 2012.

From 2008 to 2015, she was with the Power Electronics Systems (PES) Laboratory, ETH Zürich, focusing on the prediction of electromagnetic behavior of power electronics systems based on the developed numerical techniques and the lifetime modeling of power semiconductor modules. In March 2016, she joined the Advanced Power Semiconductor (APS) Laboratory, ETH Zürich. Her research interests include novel packaging technologies for SiC power devices, the optimization of power module layout with respect to electromagnetic interference, and multidomain modeling of power semiconductor devices and their modules.

Giulio Antonini (Senior Member, IEEE) received the Laurea degree (cum laude) in electrical engineering from the University of L'Aquila, L'Aquila, Italy, in 1994, and the Ph.D. degree in electrical engineering from the University of Rome "La Sapienza," Rome, Italy, in 1998.

Since 1998, he has been with the UAq EMC Laboratory, University of L'Aquila, where he is currently a Professor. He has authored more than 300 papers published in international journals and the proceedings of international conferences. He has coauthored the book "*Circuit Oriented Electromagnetic Modeling Using the PEEC Techniques*" (Wiley–IEEE Press, 2017). His scientific interests are in the field of computational electromagnetics.

Ulrike Grossner (Member, IEEE) received the Dipl.-Phys. and Dr.rer.nat. degrees from Friedrich-Schiller-University Jena, Jena, Germany, in 1997 and 2000, respectively.

In 2014, she was appointed as a Full Professor with ETH Zürich, Zürich, Switzerland, where she established the Advanced Power Semiconductor Laboratory (APS), working on devices and packaging for advanced power semiconductors.

Open Access funding provided by 'Università degli Studi dell'Aquila' within the CRUI CARE Agreement