FFT-PEEC: A Fast Tool From CAD to Power Electronics Simulations

Riccardo Torchio, Francesco Lucchini, J-L. Schanen, Olivier Chadebec,
Gerard Meunier

HAL Id: hal-03273227
https://hal.archives-ouvertes.fr/hal-03273227

Submitted on 29 Jun 2021

FFT-PEEC: A Fast Tool From CAD to Power Electronics Simulations

R. Torchio, F. Lucchini, J.-L. Schanen, O. Chadebec, G. Meunier

Abstract—A fast and general Partial Element Equivalent Circuit (PEEC) method based on the Fast–Fourier–Transform (FFT) is proposed for the first time. The numerical tool only requires common CAD data input files (e.g. *.stl* format), then the discretization process is performed automatically by means of a fast voxelization technique based on ray intersection, thus drastically reducing the human effort required to setup the model. The method allows for considering at the same time inductive and capacitive effects, and is focused on power electronics applications where propagation effects can be neglected, whereas all the other electromagnetic phenomena are considered. Specifically, the proposed method is particularly suited for problems where both electric and magnetic fields are equally important and therefore quasistatic approximations do not apply. An ad–hoc preconditioner which significantly speeds-up the solver is also proposed and, thanks to the FFT, both memory and computation time are significantly reduced, without the need of applying data compression. Both linear and non–linear materials are considered by the proposed FFT–PEEC method. Sample implementation of the method is made publicly available.

Index Terms—PEEC, FFT, Non–Linear Materials, Capacitive Effects.

I. INTRODUCTION

The need of reducing the weight, size and cost of power electronics converters has been a constant trend over the years. For this purpose, the switching frequency has always been increasing, in order to reduce the passive components' dimensions and therefore the weight, size and cost of the overall system. For instance, in automotive applications, the introduction of new generation Silicon Carbide MOSFETs has allowed for significantly increasing the switching frequency of dc–dc resonant converters [1]–[3]. Devices such as wide band-gap semiconductors have reasonably low commutation losses, however when they reach frequencies of several MHz they still requires soft switching in order to further reduce switching losses. For this purpose, resonant converters are used, exciting L-C circuits. Moreover, to improve the power-density of converter components, these passive devices are often integrated into a single sophisticated electromagnetic (EM) component, mixing capacitive and magnetic effects to reduce the overall size [4], [5]. However, designing such integrated passive devices necessitates having very precise modeling capabilities, which allow for evaluating magnetizing

R. Torchio was with the Department of Industrial Engineering, Università degli Studi di Padova, 35131 Padova, Italy, e-mail: riccardo.torchio@unipd.it.

F. Lucchini was with Consorzio RFX (CNR, ENEA, INFN, Università degli Studi di Padova, Acciaierie Venete SpA), Corso Stati Uniti 4, 35127 Padova, Italy.

J.-L. Schanen, O. Chadebec, and G. Meunier were with Univ. Grenoble Alpes, CNRS, Grenoble INP, G2Elab, 28000 Grenoble, France.

and leakage inductances, as well as capacitive effects, to analyse and optimise the proposed geometry and material characteristics. The Finite Elements Method (FEM) is sometimes used [6], but the particular geometry of such kind of devices, which includes very thin conductors, makes the meshing a complicated task. Furthermore, the high memory requirements and the long computation times do not facilitate optimisation. For these reasons, engineers often use analytical formula (see, e.g., [4], [7], [8]) for the design and optimisation of such components, since they allow short computation times and low memory requirements. However, solving Maxwell's equations with analytical approaches necessitates approximations, which are not always valid. For instance, in [7] and [8], magnetic and dielectric problems are solved separately, thus significantly simplifying the study but, at the same time, approximating the actual behaviour of the real working component. For many cases this approach offers a valid and useful approximation which, however, is not valid in general [9].

Commercially available numerical tools for electromagnetic simulations are nowadays either focused on low frequency (electric motors, cables, power transformers, etc.) or high frequency applications (e.g. scattering, SAR quantification, etc.). However, as previously discussed, the increase in operating frequencies necessitates electromagnetic field problems solution in the model range where resistive, capacitive and inductive effects need to be considered simultaneously, but radiation effects can still be neglected. The modelling and simulation of applications that fit into this regime is now a challenging topic of ongoing research efforts in the field of computational electromagnetics. Since these applications are actual industrial cases subject to optimisation, complex geometries must be addressed and computation times reduced. Moreover, besides the purely computational cost, the numerical tool should also reduce the human effort required to setup the computational model, which is often the most demanding part of the process since it requires man-hours instead of CPU-time.

For these reasons, in recent years, the interest and the research in Integral Equation (IE) methods has significantly increased. Contrary to FEM, IE methods avoid the air discretization by exploiting the Green's function of the background. Among all IE methods, the Partial Element Equivalent Circuit (PEEC) method, first introduced by A. Ruehli in 1972, has captured the interest of the scientific community for its versatility and its useful and natural circuit interpretation of the electromagnetic problem [9]–[14]. However, as all classical IE approaches, the PEEC method generates dense matrices that may easily lead to prohibitive storage and computation time. For this reason, many authors have proposed several numerical techniques to reduce the computational cost and

Fig. 1. Representation of Ω_e, Ω_m and Ω_m. In this example, the electric domain Ω_e is the electric trace of the L–C–T component [4] and Ω_m is the magnetic core. The red box represents the detail reported in Fig. 3

Fig. 2. Left: Conceptual representation of the L-C-T component. Right: Corresponding CAD model scaled along the vertical direction to improve the clarity. The device is fed by connecting the voltage source to the P1 and P2′ terminals.

improve the range of applicability of the PEEC methods. For instance, PEEC has been coupled with Hierarchical Matrices and with cross approximation in [15] and [16], with the Fast Multiple Method in [17] and [12], with Voronoi decomposition in [18], and other methods in, e.g., [19]. All these approaches provide a general computational cost reduction. However, they are somehow intrusive and mostly based on matrix compression (therefore introducing unavoidable, hopefully small, approximations), which usually requires some expertise from the user which *a priori* has to choose suitable values for the compression tolerances. Moreover, all these methods only focus on computational cost reduction, and they do not reduce the effort required by the user to setup the computational model.

In this paper, a fast and general PEEC method based on the Fast–Fourier–Transform (FFT) is proposed for the first time, although it is fair to say that the possibility of combining PEEC with FFT has been preliminarily investigated in [20]. Specifically, the underlying formulation of the proposed FFT–PEEC is the one of [21], where a standard Volume Integral Equation (VIE) method is re-interpreted in the PEEC framework. Then, the FFT approach proposed in [22] for very high frequency problems (EM scattering by biological tissues) and in [23] and [24] for eddy current problems (i.e. neglecting the capacitive effects) is applied to the PEEC formulation, drastically reducing the memory requirements and the computation times, without introducing any data compression (and therefore without loss of accuracy). The proposed numerical tool only requires common CAD data input files (e.g. .*stl* format), then the discretization process is performed automatically by means of a fast voxelization technique based on ray intersection [25]. This feature allows for considerably reducing the pre–processing time, which is often underestimated but, instead, may require several man–hours to convert the CAD data information to a model which is suitable for electromagnetic simulations and then its discretization. The FFT–PEEC allows for considering resisitive, inductive and capacitives effects, and is mostly focused on low to mid–frequency applications, i.e. for electrically short devices where inductive and capacitive effects must be considered in a coupled
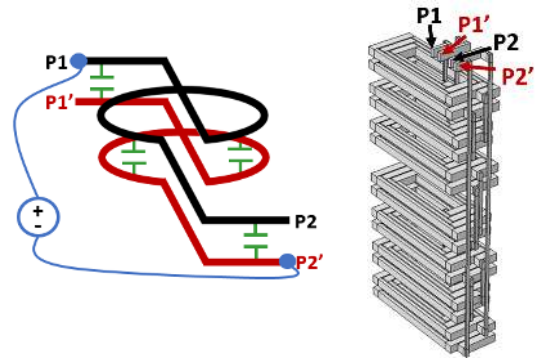
way, but propagation effects can be neglected. Considering the typical size of PE devices the FFT–PEEC method is mostly focused on a frequency range from 0 Hz to 300 MHz (or higher for devices smaller than 5 cm). In more detail, the proposed tool is mainly focused on the study of devices where electric and magnetic fields are equally important, and therefore quasistatic approximations cannot be exploited. For instance, this is the case of devices for which closed inductive loops cannot be identified or when the target frequency and/or the device structure necessitate a model based on, e.g., distributed (self and mutual) inductances and capacitances obtained from a full-Maxwell's study. The solution of the EM problem is sped-up by means of an ad–hoc preconditioner, which is fundamental to solve the EM problem for the target frequency range. Non–linear materials, which have been rarely treated in the literature [26], are also considered by the proposed FFT–PEEC.

II. FFT–PEEC FORMULATION AND DISCRETIZATION

In this section, the theoretical background of the FFT–PEEC method is thoroughly described. The background PEEC formulation adopted in this work is the general one proposed in [21], where a standard VIE formulation is re–interpreted resulting in an equivalent circuit interpretation, equal to the one of the traditional PEEC scheme. In the following, for the sake of completeness, such formulation is shortly reported. Instead, the discretization procedure is modified according to [22], [23] in order to exploit the translational invariant property of the integral kernel and speed–up the PEEC simulation.

A. Target Problem Description

Since conductive, dielectric, and magnetic media are all considered, it is convenient to introduce the following definitions:

- Ω_c is the conductive domain, i.e. all the regions where the electric conductivity σ_c is greater than zero,
- Ω_d is the dielectric domain, i.e. all the regions where the relative permittivity ε_r is different from one,
- Ω_m is the magnetic domain, i.e. all the regions where the relative permeability μ_r is different from one,
- Ω_e is the electric domain, i.e. the union of Ω_c and Ω_d.

The intersection between Ω_c, Ω_d, and Ω_m can be non-empty.

To illustrate the application of the method for power electronics components, the example illustrated in Fig. 1 is considered, i.e. an integrated LCT device [4] used in resonant power electronics converters. This device can be interpreted as a high frequency transformer with a series capacitance and a series inductance, originating from the leakage of the transformer. This device consists of a gapped magnetic core (Ω_m), and wound conductors (Ω_c) forming simultaneously the integrated capacitance and the transformer. Dielectric materials (Ω_d) are used to increase the capacitance. Fig. 2 shows a conceptual representation of the L-C-T primary winding with the corresponding CAD model. Distributed capacitances (which are fundamental for the working principle of the device) are also represented in Fig. 2 and they account for the effects due to the time derivative of the electric field. As can be seen, a closed inductive path cannot be identified, therefore the magnetoquasistatic approximation cannot be exploited, thus making the LCT component a perfect candidate for using the FFT-PEEC method. The aim is to obtain from this complex 3D geometry both inductive and capacitive behaviour of the component, thus it is not possible to separate the study into magnetoquasistatic and electrostatic analyses. Indeed, as shown by the conceptual representation of Fig. 2, it is worth noting that primary winding of the LCT component is actually divided into two parts and the currents can *jump* from one part to the other by means of capacitives effects, therefore closing the primary current path. Thus, for this particular geometry, capacitive effects related to the time derivative of the electric field *cannot* be neglected and it is not even possible to split the electromagnetic simulation into separate magnetoquasistatic and electrostatic analyses. Therefore, to correctly model the behaviour of the device, a full-Maxwell study is necessary.

It is worth noting that several well–know and established tools are used for PE devices. For instance, the well-known FastHenry [27] and FastCap [28] tools exploit the magneto-quasistatic and electrostatic approximations, respectively, and therefore they are well suited for extracting equivalent inductance and capacitance of devices which, in the frequency range of interest, exhibit a predominant inductive (e.g. a closed printed trace) or capacitive (e.g. two parallel traces) behavior. However, such tools cannot be used for general cases (as the LCT component described above) and they only consider conductive structures, completely or partially neglecting the presence of dielectric and magnetic media. Similar considerations also hold for the Flux PEEC (Inca3D) software [29], which also exploits the magnetoquasistatic assumption. Another well–known tool typically used for printed circuit boards with surface mounted conductive traces is ANSYS HFSS 3D Layout, which exploits the dominant 2D current density distribution in such structures, thus reducing the computational cost but at the same time reducing its generality. ANSYS Q3D is another extensively used tool, which provides equivalent circuits in the form of R-L-C-G lumped elements, where R-L and C-G are calculated by decoupling the electric and magnetic fields in the Maxwell's equations [9], thus allowing for reducing the problem complexity but again leading to approximations which are not always valid. A thorough discussion concerning the above mentioned tools for

PE applications can be found in [9], where pros and cons are discussed both from a theoretical and a numerical point of view. Differently from the above mentioned tools, the proposed FFT-PEEC allows for considering the full set of Maxwell's equations without making approximations as done by other FEM or IE based software (e.g. COMSOL and FEKO, respectively). However, with respect to the existing full-Maxwell's equations software, thanks to the use of FFT and automatic voxelization techniques, the proposed method is fast and efficient during both the pre-processing and computation phases.

B. PEEC as a Reinterpretation of Volume Integral Equation

As typically done in IE methods, the full–Maxwell's equations are recast into the following well–known equations [21]:

$$\mathbf{E}(\mathbf{r}) = -i\omega\mathbf{A}_e(\mathbf{r}) - \nabla\varphi_e(\mathbf{r}) - \frac{1}{\varepsilon_0}\nabla\times\mathbf{A}_m(\mathbf{r}) + \mathbf{E}_{ext}(\mathbf{r}), \quad (1)$$

$$\mathbf{H}(\mathbf{r}) = -i\omega\mathbf{A}_m(\mathbf{r}) - \nabla\varphi_m(\mathbf{r}) + \frac{1}{\mu_0}\nabla\times\mathbf{A}_e(\mathbf{r}) + \mathbf{H}_{ext}(\mathbf{r}), \quad (2)$$

where $\mathbf{E}$ and $\mathbf{H}$ are the electric and the magnetic fields, respectively, $\mathbf{A}_e$ and $\mathbf{A}_m$ are the electric and magnetic vector potentials, respectively, and φ_e and φ_m are the scalar electric and magnetic potentials. $\mathbf{E}_{ext}$ and $\mathbf{H}_{ext}$ are incident external fields, $\mathbf{r}$ is the field point, i is the imaginary unit, ω is the angular frequency and ε_0 and μ_0 are the permittivty and permeability of vacuum. Vector and scalar potentials in (1) and (2) are given by their integral expression in terms of $\mathbf{J}_e$, and $\mathbf{J}_m$, which are the electric and magnetic current densities, respectively, due to the conduction and polarization currents and the magnetization. These expressions can be found, e.g., in [21] and they exploit the Green's function of vacuum. In this paper, propagation effects are neglected, the *static* Green's function is used, however, theoretically, the *dynamic* Green's function can also be used, thus considering also propagation effects. However, in the case of very high frequency problems, which are not so relevant for power electronics applications, more suited formulations exist [22].

Then, as in [21], constitutive equations which locally link $\mathbf{E}$ with $\mathbf{J}_e$ and $\mathbf{H}$ with $\mathbf{J}_m$ by means of equivalent electric and magnetic resistivities are introduced.

Finally, combing such constitutive relations with (1) and (2), together with the continuity equations of $\mathbf{J}_e$ and $\mathbf{J}_m$, one finally obtains the four continuum equations where $\mathbf{J}_e$, $\mathbf{J}_m$, φ_e, and φ_m are the problem unknowns, as in the traditional PEEC scheme [21].

C. Discretization: Voxelization

In this work, the *dual* PEEC discretization [21], [30], [31] is adopted. However, the mesh creation procedure is fundamentally modified and the main steps are described in the following and reported in Algorithm 1. Indeed, in traditional unstructured PEEC methods like the ones in the above mentioned papers, the mesh generation is performed in a standard way, such as in FEM approaches (with the only exception that the vacuum/air domains are not discretized). Thus, the CAD input files must first be handled in order

---

Algorithm 1 FFT–PEEC Discretization

Input: .*stl* files of the devices to be simulated
1) Create Ω_{box} identified by the corners C1 and C2
2) Define the discretization steps Δ_x, Δ_y, and Δ_z and generate the 3–D grid
3) Identify index sets $\mathbb{S}_{IDe}$ and $\mathbb{S}_{IDm}$ of voxels discretizing Ω_e and Ω_m, respectively
Output: Mesh for FFT–PEEC

---

to make them suitable for EM simulations (e.g. deleting electromagnetically insignificant details and adjusting critical parts), then a mesh is generated. When complex devices such as the LCT component of Fig. 1 are considered, these two steps may require several iterations and a considerable human time and effort, that can even dominate the computational cost required by the EM solver. The proposed FFT-PEEC aims precisely at reducing the time and effort required by these two steps, making full Maxwell's EM simulations of complex devices possible also for novice users.

First, a box domain Ω_{box} which contains Ω_e and Ω_m is introduced as exemplified in Fig. 1. The box domain is identified by defining the corners C1 and C2 of Fig. 1 (step 1 of Algorithm 1).

According to step 2 of Algorithm 1, once Ω_{box} is identified, a uniform grid in the 3–D coordinate system is introduced, with Δ_x, Δ_y, and Δ_z being the three discretization steps along the x, y, and z axes, respectively. The x, y, and z dimensions of Ω_{box} are discretized by N_x, N_y, and N_z voxels, respectively. Each ijkth voxel of the grid (with $i = 1, \cdots, Nx$, $j = 1, \cdots, Ny$, and $k = 1, \cdots, Nz$) has $\Delta_x \times \Delta_y \times \Delta_z$ dimension and is identified by the triplet of indices i, j, k, or, equivalently, by the global index t. The triplet i, j, k can be mapped into the global index t as

$$t = i + (j - 1)N_x + (k - 1)N_xN_y. \tag{3}$$

Such map will be hereafter indicated as $t = map\_ijk(i, j, k)$ and the inverse map is $[i, j, k] = map\_t(t)$.

Successively, in step 3 of Algorithm 1, the index sets of voxels belonging to Ω_e and Ω_m are identified. These two index sets are defined as $\mathbb{S}_{IDe}$ and $\mathbb{S}_{IDm}$, respectively, and their dimension is N_{ve} and N_{vm}, respectively. N_{ve} and N_{vm} are, indeed, the number of voxels discretizing Ω_e and Ω_m, respectively.

The identification of $\mathbb{S}_{IDe}$ and $\mathbb{S}_{IDm}$ can be performed by using several methods, such as ray intersection [25] or conservative voxelization [32] methods. These methods are generally very fast and only require common CAD data input files (e.g. .*stl* format) where Ω_e and Ω_m are described by means of polygon-based 3D models. In this paper, with the aim of developing a pure MATLAB® tool, the MATLAB® code implemented in [33] (with only some non-substantial modifications) has been used. In Appendix A, the ray intersection approach is described. However, other more sophisticated and robust methods based on GPUs can also be used [34].

Obviously, when a voxelization method is adopted, the price to pay is that the original geometry is not perfectly represented by the voxelized structure. However, it is worth noting that in any standard VIE method the discretization process always
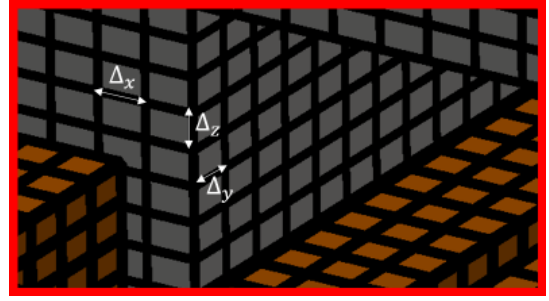

Fig. 3. Voxelization of Ω_e and Ω_m. The picture represents the detail in the red box of Fig. 1.

introduces some (hopefully small) distortion of the original geometry. Moreover, the FFT approach described in Section II-D makes it possible to consider meshes with a very large amount of elements, therefore leading to a very good definition of the voxelized structure, and, as shown by the numerical results, significantly alleviating the errors introduced by the imperfect representation of the original geometry due to the voxelization. Another important remark is that the (brute-force) voxelization approach allows for using standard CAD input files which are not directly suited for EM simulations, e.g. with many tiny and electromagnetically insignificant details that would have to manually be removed if standard simulation tools were used. On the contrary, such details are automatically treated by the ray intersection voxelization.

One of the main goals of the proposed FFT-PEEC tool is to reduce the *human* effort required for the model preparation and meshing of complex devices, such as the LCT component of Fig. 1, and, at the same time, allow for solving the full Maxwell's equations without introducing approximations. The proposed voxelization techniques allow for reducing practically to zero the user time and effort required to create a model which is suited for EM simulations having standard CAD models as input files. Clearly, one should accept to loose the total control in the model construction and to introduce some small distortion with respect to the original geometry.

Once $\mathbb{S}_{IDe}$ and $\mathbb{S}_{IDm}$ are identified, the meshes of Ω_e and Ω_m have been actually generated, as shown in Fig. 3. Now, following [21], $\mathbf{J}_e$ and $\mathbf{J}_m$ are expanded as

$$\mathbf{J}_e(\mathbf{r}) = \sum_{h=1}^{N_{fe}} \mathbf{w}_h(\mathbf{r})j_{e,h}, \tag{4}$$

$$\mathbf{J}_m(\mathbf{r}) = \sum_{h=1}^{N_{fm}} \mathbf{w}_h(\mathbf{r})j_{m,h}, \tag{5}$$

where $\mathbf{w}_h$ is the vector face shape function related to the hth face of the mesh and $j_{e,h}$ and $j_{m,h}$ are the Degrees of Freedom (DoFs) related to the hth face corresponding to the fluxes of $\mathbf{J}_e$ and $\mathbf{J}_m$ through that face, respectively. N_{fe} and N_{fm} are the number of internal faces of the mesh of Ω_e and Ω_m, respectively.

In the PEEC framework, scalar potentials are also considered as problem unknowns. Therefore, the DoFs related to

scalar electric and magnetic potentials are associated to the barycenters of the voxels belonging to $\mathbb{S}_{IDe}$ and $\mathbb{S}_{IDm}$, i.e.

$$\varphi_e(\mathbf{r}) = \sum_{h=1}^{N_{ve}} p_k(\mathbf{r})\phi_{e,h}, \qquad (6)$$

$$\varphi_m(\mathbf{r}) = \sum_{h=1}^{N_{vm}} p_k(\mathbf{r})\phi_{m,h}, \qquad (7)$$

where p_h is the hth pulse function equal to $\frac{1}{\Delta_x \Delta_y \Delta_z}$ when $\mathbf{r}$ belongs to the hth voxel in $\mathbb{S}_{IDe}$ (or $\mathbb{S}_{IDm}$) and zero elsewhere. $\phi_{e,h}$ and $\phi_{m,h}$ are the degrees of freedom corresponding to the average electric and magnetic scalar potentials in the hth voxel of $\mathbb{S}_{IDe}$ and $\mathbb{S}_{IDm}$, respectively.

Then, substituting (4)–(7) in the continuum equations obtained in Section II-B and projecting the resulting equations with a Galerkin scheme, the final, traditional, PEEC system of equations is obtained

$$\begin{bmatrix} \mathbf{R}_e + i\omega\mathbf{L}_e & \mathbf{K} & \mathbf{A}_e^T & \mathbf{0} \\ -\mathbf{K}^T & \mathbf{R}_m + i\omega\mathbf{L}_m & \mathbf{0} & \mathbf{A}_m^T \\ \mathbf{P}_e\mathbf{A}_e & \mathbf{0} & -i\omega\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_m\mathbf{A}_m & \mathbf{0} & -i\omega\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{j}_e \\ \mathbf{j}_m \\ \boldsymbol{\phi}_e \\ \boldsymbol{\phi}_m \end{bmatrix} = \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{h}_0 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$
$$(8)$$

where $\mathbf{j}_e$, $\mathbf{j}_m$, $\boldsymbol{\phi}_e$, and $\boldsymbol{\phi}_m$ are arrays storing the DoFs $j_{e,h}$, $j_{m,h}$, $\phi_{e,h}$, and $\phi_{m,k}$, respectively, whereas $\mathbf{e}_0$ and $\mathbf{h}_0$ are arrays corresponding to $\mathbf{E}_{ext}$ and $\mathbf{H}_{ext}$. The PEEC matrices $\mathbf{R}$, $\mathbf{L}$, and $\mathbf{P}$ are the standard resistance, inductance, and potential PEEC matrices, respectively, (where the subscript e indicates electric matrices and m indicates magnetic matrices). Their expressions can be found in [21]. Instead, the coefficients of $\mathbf{K}$ can be obtained following [4] (taking care that in [4] different, i.e scaled, current unknowns are used). It is important to note that the magnetic and dielectric properties of materials (i.e. the conductivity σ_c, the relative permittivity ε_r and the relative permeability μ_r) only appear in the sparse $\mathbf{R}_e$ and $\mathbf{R}_m$. Matrices $\mathbf{A}_e$ and $\mathbf{A}_m$ are the volumes–faces incidence matrices related to Ω_e and Ω_m, respectively. Their dimensions are $N_{ve} \times N_{fe}$ and $N_{vm} \times N_{fm}$, respectively.

The solution of (8) provides an (approximate) solution of the electromagnetic problem. Often, in literature, (8) coupled with lumped component network equations is solved by Spice–like solvers based on Modified Nodal Analysis (MNA) procedures. However, it has been clearly shown in [35] and [36] that dedicated sparse circuit solvers are extremely inefficient for the solution of PEEC systems when objects with a relatively fine discretization are considered. Indeed, $\mathbf{L}$, $\mathbf{P}$, and $\mathbf{K}$ are dense matrices whose memory storage grows quadratically with the number of the unknowns. Moreover, as noticed in [35], when even a few hundred unknowns are considered Spice–like solvers would require a considerable time just to read the input file. Furthermore, the *net list* itself would require a considerable amount of disk space. For these reasons, many authors have developed different solution strategies based, for instance, on LU factorisation [37] (when the problem dimension is reasonable), iterative solvers [36], or more sophisticated methods such as Hierarchical Matrices coupled with cross approximation [15], [16], Fast Multiple [17], Voronoi decomposition [18], and others [19], [38], [39].

In this work, thanks to the voxelization of the geometry, the coefficients of $\mathbf{L}$, $\mathbf{P}$, and $\mathbf{K}$ posses the property of being translationally invariant. Therefore, it is possible to exploit this property in order to significantly reduce the memory requirement for the storage of the matrices and speed–up the solution.

In Section II-D, it is shown how to exploit the translational invariance property in order to efficiently store $\mathbf{L}$, $\mathbf{P}$, and $\mathbf{K}$ matrices and speed up matrix–vector products involving such matrices. Then, in Section II-E, a general preconditioner is proposed to solve (8) by means of iterative solvers. Subsequently, in Section II-G, the case of connections with external lumped networks is discussed and in Section II-H the case of non–linear materials is also considered, with emphasis on non–linear magnetic media.

D. Acceleration of PEEC via FFT

Thanks to translational invariance property of the Green's function, the coefficients of $\mathbf{L}$, $\mathbf{P}$, and $\mathbf{K}$ related to two pairs of unknowns having the same mutual position are equal. Namely, whenever the mesh is obtained through a voxelization process, considering four unknowns with support barycenters equal to $\mathbf{r}_i$, $\mathbf{r}_j$, $\mathbf{r}_k$, and $\mathbf{r}_h$, when $\mathbf{r}_i - \mathbf{r}_j = \mathbf{r}_h - \mathbf{r}_k$ the related mutual inductance coefficients are equal, i.e. $L_{ij} = L_{hk}$ (and the same holds for $\mathbf{P}$ and $\mathbf{K}$). As shown in [22], [40], and [23], this property can be exploited in order to represent $\mathbf{L}$, $\mathbf{P}$, and $\mathbf{K}$ in an efficient format and speed-up matrix vector products. The main steps for doing so are described in the following for the $\mathbf{L}$, $\mathbf{P}$, and $\mathbf{K}$ matrices. Mathematical considerations are neglected in the following discussion and they can be found in other works [22], [40]. Instead, for the sake of completeness, all the steps required for the implementation of the algorithm are described and the specific operations to apply such technique to the PEEC formulation are described.

For the sake of conciseness, all the steps are given considering the electric matrices only. However, the steps are exactly the same for the magnetic matrices.

1) Matrix $\mathbf{P}_e$: Matrix $\mathbf{P}_e$, of dimension $N_{ve} \times N_{ve}$, is the easiest to handled. We now introduce the auxiliary *total* electric potential matrix $\mathbf{P}_{e,t}$ of dimension $N_{v,box} \times N_{v,box}$, where $N_{v,box} = N_x N_y N_z$ is the total number of voxels in Ω_{box}. Matrix $\mathbf{P}_e$ can be seen as a row–column selection of $\mathbf{P}_{e,t}$, i.e., by using the MATLAB® notation hereafter,

$$\mathbf{P}_e = \mathbf{P}_{e,t}(\mathbb{S}_{IDe}, \mathbb{S}_{IDe}). \qquad (9)$$

Thanks to the translational invariance property, $\mathbf{P}_{e,t}$ is a Block Toeplitz matrix with Toeplitz blocks (BTTB) and all the coefficients of $\mathbf{P}_{e,t}$ appear in its first row (or first column). The idea is to exploit such property in order to efficiently represent $\mathbf{P}_{e,t}$ and perform matrix–vector products. The computational and mathematical literature concerning Toeplitz and Block Toeplitz matrices is vast and many algorithms and numerical methods have been proposed [41], [42]. In this work, we follow the procedure described in [22] where the Block Toeplitz matrix is first represented as a $N_x \times N_y \times N_z$ tensor and then embedded into a circulant tensor. Matrix vector products are then efficiently performed by means of the FFT [43], [44].

First, as in [22], the $N_x \times N_y \times N_z$ tensor $\mathcal{P}_{e,t}$ is constructed:

$$\mathcal{P}_{e,t}(map\_t(t)) = \mathbf{P}_{e,t}(1,t), \text{ for } t = 1, N_{v,box}. \quad (10)$$

Of course, the entry $\mathbf{P}_{e,t}(1,t)$ in (10) can be generated *on the fly*, therefore the whole $\mathbf{P}_{e,t}$ is never actually generated.

A Toeplitz matrix can be always embedded in a circulant matrix. Analogously, $\mathcal{P}_{e,t}$ can be embedded in the circulant tensor $\mathcal{CP}_{e,t}$ of dimension $2N_x \times 2N_y \times 2N_z$:

$$\begin{aligned}
\mathcal{CP}_{e,t}(1:N_x, 1:N_y, 1:N_z) = \\
\mathcal{P}_{e,t}(1:N_x, 1:N_y, 1:N_z), \\
\mathcal{CP}_{e,t}(N_x+2:2N_x, 1:N_y, 1:N_z) = \\
\mathcal{P}_{e,t}(N_x:-1:2, 1:N_y, 1:N_z), \\
\mathcal{CP}_{e,t}(1:N_x, N_y+2:2N_y, 1:N_z) = \\
\mathcal{P}_{e,t}(1:N_x, N_y:-1:2, 1:N_z), \\
\mathcal{CP}_{e,t}(1:N_x, 1:N_y, N_z+2:2N_z) = \\
\mathcal{P}_{e,t}(1:N_x, 1:N_y, N_z:-1:2), \quad (11) \\
\mathcal{CP}_{e,t}(N_x+2:2N_x, N_y+2:2N_y, 1:N_z) = \\
\mathcal{P}_{e,t}(N_x:-1:2, N_y:-1:2, 1:N_z), \\
\mathcal{CP}_{e,t}(N_x+2:2N_x, 1:N_y, N_z+2:2N_z) = \\
\mathcal{P}_{e,t}(N_x:-1:2, 1:N_y, N_z:-1:2), \\
\mathcal{CP}_{e,t}(1:N_x, N_y+2:2N_y, N_z+2:2N_z) = \\
\mathcal{P}_{e,t}(1:N_x, N_y:-1:2, N_z:-1:2), \\
\mathcal{CP}_{e,t}(N_x+2:2N_x, N_y+2:2N_y, N_z+2:2N_z) = \\
\mathcal{P}_{e,t}(N_x:-1:2, N_y:-1:2, N_z:-1:2).
\end{aligned}$$

Once the circulant tensor $\mathcal{CP}_{e,t}$ is generated, the matrix–vector product between $\mathbf{P}_e$ and a general vector $\mathbf{v}$ can be efficiently obtained by using $\mathcal{CP}_{e,t}$ and the FFT. Indeed, the vector resulting from the matrix–vector product $\mathbf{x} = \mathbf{P}_e\mathbf{v}$ can be obtained from the following steps:

- embed $\mathbf{v}$ into the $N_x \times N_y \times N_x$ tensor $\mathcal{V}$, i.e.

$$\mathcal{V}(map\_t(t)) = \mathbf{v}(t), \text{ for any } t \in \mathbb{S}_{IDe}, \quad (12)$$

- using the 3–dimensional discrete FFT and Inverse-FFT to efficiently perform tensor products by means of convolution,

$$\begin{aligned}
\mathcal{X} = \text{IFFTN}(\text{FFTN}(\mathcal{CP}_{e,t}).* \\
\text{FFTN}(\mathcal{V}, [2N_x, 2N_y, 2N_z])), \quad (13)
\end{aligned}$$

where $\mathcal{X}$ is a $2N_x \times 2N_y \times 2N_x$ tensor, FFTN and IFFTN are the 3–dimensional discrete MATLAB® FFT and its inverse, $.*$ indicates the MATLAB® entry–by–entry multiplication,

- extract the solution $\mathbf{x}$ from $\mathcal{X}$:

$$\begin{aligned}
\mathcal{Y} = \mathcal{X}(1:N_x, 1:N_y, 1:N_z), \\
\mathbf{x}(t) = \mathcal{Y}(map\_t(t)), \text{ for any } t \in \mathbb{S}_{IDe}. \quad (14)
\end{aligned}$$

The standard matrix–vector product requires $\mathcal{O}(N_{v,e}^2)$ multiplications and additions and demands for the generation of all the $N_{v,e}^2$ entries of $\mathbf{P}_e$, therefore leading to a prohibitive memory requirement and computation time if several matrix–products must be performed. The steps reported in (12)–(14), instead, require a reduced memory requirement and
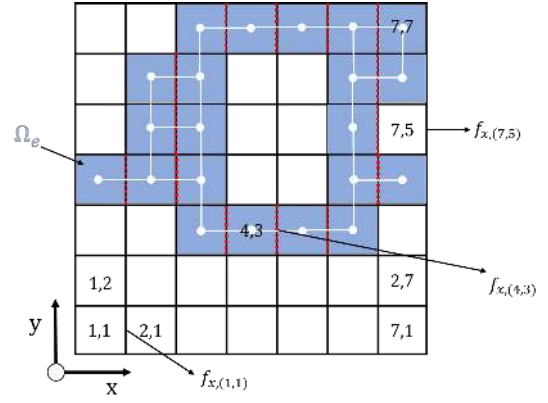

Fig. 4. 2–D exemplification of voxel and face IDs. Blue voxels discretize Ω_e whereas withe voxel discretize $\Omega_{box} \setminus \Omega_e$. Red dashed faces constitute $\mathbb{S}_{IDe,fx}$ and the white dots and lines represents the graph of the equivalent circuit.

computation time. Specifically, in (13), $8N_{v,box}$ multiplication are performed and FFTN($\mathcal{CP}_{e,t}$) can be generated once at the beginning of the algorithm and then re–used to perform several matrix–vector multiplications. FFT and IFFT have a $\mathcal{O}(Nlog(N))$ complexity, that is generally much lower than $\mathcal{O}(N_{v,e}^2)$ required by the standard matrix-vector product in IE methods.

2) Matrix $\mathbf{L}_e$: Due to the orthogonality of the face functions and the regularity of the mesh, matrix $\mathbf{L}_e$, of dimension $N_{f,e} \times N_{f,e}$ has the following structure:

$$\mathbf{L}_e = \begin{bmatrix} \mathbf{L}_{e,xx} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{e,yy} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L}_{e,zz} \end{bmatrix} \quad (15)$$

where $\mathbf{L}_{e,xx}$, $\mathbf{L}_{e,yy}$, and $\mathbf{L}_{e,zz}$ have dimensions $N_{f,e,x} \times N_{f,e,x}$, $N_{f,e,y} \times N_{f,e,y}$, and $N_{f,e,z} \times N_{f,e,z}$, respectively. $N_{f,e,x}$, $N_{f,e,y}$, and $N_{f,e,z}$ are the number of faces shared by voxels in $\mathbb{S}_{IDe}$ pointing in the x, y, and z directions, respectively. As for $\mathbf{P}_e$, also $\mathbf{L}_{e,xx}$, $\mathbf{L}_{e,yy}$, and $\mathbf{L}_{e,zz}$ are seen as row–column selections of BTTB matrices. For instance,

$$\mathbf{L}_{e,xx} = \mathbf{L}_{e,xx,t}(\mathbb{S}_{IDe,fx}, \mathbb{S}_{IDe,fx}) \quad (16)$$

where $\mathbf{L}_{e,xx,t}$ is the total BTTB inductance matrix and $\mathbb{S}_{IDe,fx}$ is the index set of faces (of dimension $N_{f,e,x}$) shared by voxels in $\mathbb{S}_{ID,e}$ pointing in the x direction. Index sets $\mathbb{S}_{IDe,fy}$ and $\mathbb{S}_{IDe,fz}$, with obvious meaning, are also introduced. $\mathbf{L}_{e,xx,t}$ has dimension $N_{v,box} \times N_{v,box}$. Indeed, as exemplified in Fig. 4 for a 2–D case, each face in Ω_{box} pointing in the x direction is identified by the ijk-id of the the voxel at its left. Therefore, all the faces pointing in the x direction at the far left of Ω_{box} are not considered (since there are no voxels at their left). Equivalent considerations hold also for $\mathbf{L}_{e,yy,t}$ and $\mathbf{L}_{e,zz,t}$.

Then, all the steps described above for matrix $\mathbf{P}_e$ can be also applied to efficiently perform matrix–vector products involving $\mathbf{L}_e$, significantly reducing computation time and memory.

3) Matrix $\mathbf{K}$: Due to the orthogonality of the face functions and the regularity of the mesh, matrix $\mathbf{K}$, of dimension $N_{f,e} \times$

$N_{f,m}$, has the following structure:

$$\mathbf{K} = \begin{bmatrix} \mathbf{0} & \mathbf{K}_{xy} & \mathbf{K}_{xz} \\ \mathbf{K}_{xy}^T & \mathbf{0} & \mathbf{K}_{yz} \\ \mathbf{K}_{xz}^T & \mathbf{K}_{yz}^T & \mathbf{0} \end{bmatrix} \qquad (17)$$

where $\mathbf{K}_{xy}$, $\mathbf{K}_{xz}$, and $\mathbf{K}_{yz}$ have dimensions $N_{f,e,x} \times N_{f,m,y}$, $N_{f,e,x} \times N_{f,m,z}$, and $N_{f,e,y} \times N_{f,m,z}$, respectively. Again, $\mathbf{K}_{xy}$, $\mathbf{K}_{xz}$, and $\mathbf{K}_{yz}$ are seen as row–column selections of BTTB matrices, e.g.

$$\mathbf{K}_{xy} = \mathbf{K}_{xy,t}(\mathbb{S}_{IDe,fx}, \mathbb{S}_{IDm,fy}), \qquad (18)$$

where $\mathbf{K}_{xy,t}$ has dimension $N_{v,box} \times N_{v,box}$. While $\mathbf{P}_{e,t}$, $\mathbf{L}_{e,xx,t}$, $\mathbf{L}_{e,yy,t}$, and $\mathbf{L}_{e,zz,t}$ are symmetric matrices, $\mathbf{K}_{xy,t}$ is antisymmetric (and the same holds for $\mathbf{K}_{xz,t}$ and $\mathbf{K}_{yz,t}$). However, $\mathbf{K}_{xy,t}$ is still a BTTB matrix and matrix–vector products involving $\mathbf{K}$ can be handled similarly to $\mathbf{P}_e$ and $\mathbf{L}_e$. The only difference is in the construction of the circulant tensor.

First, the three $N_x \times N_y \times N_z$ tensors $\mathcal{K}_{xy,t}$, $\mathcal{K}_{xz,t}$, and $\mathcal{K}_{yz,t}$ are constructed, i.e.

$$\mathcal{K}_{xy,t}(map\_t(t)) = \mathbf{K}_{xy,t}(1,t), \text{ for } t = 1, N_{v,box}, \qquad (19)$$

and similarly for $\mathcal{K}_{xz,t}$ and $\mathcal{K}_{yz,t}$. Then, $\mathcal{K}_{xy,t}$, $\mathcal{K}_{xz,t}$ and $\mathcal{K}_{yz,t}$ are embedded in the circulant tensors $\mathcal{CK}_{xy,t}$, $\mathcal{CK}_{xz,t}$, $\mathcal{CK}_{yz,t}$ of dimension $2N_x \times 2N_y \times 2N_z$:

$$\begin{aligned} &\mathcal{CK}_{\alpha,t}(1:N_x, 1:N_y, 1:N_z) = \\ &\quad \mathcal{K}_{\alpha,t}(1:N_x, 1:N_y, 1:N_z), \\ &\mathcal{CK}_{\alpha,t}(N_x+2:2N_x, 1:N_y, 1:N_z) = \\ &\quad s_1(\alpha)\mathcal{K}_{\alpha,t}(N_x:-1:2, 1:N_y, 1:N_z), \\ &\mathcal{CK}_{\alpha,t}(1:N_x, N_y+2:2N_y, 1:N_z) = \\ &\quad s_2(\alpha)\mathcal{K}_{\alpha,t}(1:N_x, N_y:-1:2, 1:N_z), \\ &\mathcal{CK}_{\alpha,t}(1:N_x, 1:N_y, N_z+2:2N_z) = \\ &\quad s_3(\alpha)\mathcal{K}_{\alpha,t}(1:N_x, 1:N_y, N_z:-1:2), \qquad (20) \\ &\mathcal{CK}_{\alpha,t}(N_x+2:2N_x, N_y+2:2N_y, 1:N_z) = \\ &\quad s_4(\alpha)\mathcal{K}_{\alpha,t}(N_x:-1:2, N_y:-1:2, 1:N_z), \\ &\mathcal{CK}_{\alpha,t}(N_x+2:2N_x, 1:N_y, N_z+2:2N_z) = \\ &\quad s_5(\alpha)\mathcal{K}_{\alpha,t}(N_x:-1:2, 1:N_y, N_z:-1:2), \\ &\mathcal{CK}_{\alpha,t}(1:N_x, N_y+2:2N_y, N_z+2:2N_z) = \\ &\quad s_6(\alpha)\mathcal{K}_{\alpha,t}(1:N_x, N_y:-1:2, N_z:-1:2), \\ &\mathcal{CK}_{\alpha,t}(N_x+2:2N_x, N_y+2:2N_y, N_z+2:2N_z) = \\ &\quad -\mathcal{K}_{\alpha,t}(N_x:-1:2, N_y:-1:2, N_z:-1:2). \end{aligned}$$

where $\alpha = xy, xz, yz$ and $s_k(\alpha)$, with $k = 1, \cdots, 6$ is given in Tab. I.

Once the circulant tensors are constructed, matrix–vector products involving $\mathbf{K}$ can be performed following the steps given for $\mathbf{P}_e$. It is worth noting that, thanks to the antisymmetric property, matrix–vector products involving $\mathbf{K}_{xy}^T$, $\mathbf{K}_{xz}^T$, and $\mathbf{K}_{yz}^T$ are easily obtained from $\mathbf{K}_{xy}$, $\mathbf{K}_{xz}$, and $\mathbf{K}_{yz}$ by changing the signs.

Finally, following the above instructions to efficiently perform matrix–vector products involving $\mathbf{P}$, $\mathbf{L}$, and $\mathbf{K}$, one can efficiently perform matrix–vector products involving the whole system (8). Since they are very sparse, multiplications with

the resistance and the incidence matrices are performed in a classical way.

E. Preconditioning the PEEC System

In the previous Section, a method based on FFT and convolution to efficiently perform matrix–vector products involving the system of equations in (8) has been described. This procedure can actually be used in an iterative solver to speed–up matrix vector products and significantly reduce the memory required to solve (8). However, it is well–known that, for integral equation problems, the convergence of iterative solvers is very slow if no preconditioning is used. The preconditioning of IE methods is still an open issue and some consolidated approaches have been found only for very high frequency problems, e.g. [45]. Moreover, especially for the kind of EM problems which are the target of this paper the literature is scarce.

In this section, a preconditioner based on [23] is constructed. However, with respect to [23], the preconditioner is extended to include dielectric and magnetic media and, more importantly for the goal of the paper, capacitive effects. Indeed, the formulation in [23] only considers eddy current problems with possible lumped voltage excitation, but completely neglects capacitive effects.

System (8) can be represented in a more compact form, i.e.

$$\begin{bmatrix} \mathbf{Z}_{em} & \mathbf{A}_{em}^T \\ \mathbf{P}_{em}\mathbf{A}_{em} & -i\omega\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{j}_{em} \\ \boldsymbol{\phi}_{em} \end{bmatrix} = \begin{bmatrix} \mathbf{eh}_0 \\ \mathbf{0} \end{bmatrix}, \qquad (21)$$

where matrices and vectors have been grouped:

$$\mathbf{Z}_{em} = \begin{bmatrix} \mathbf{R}_e + i\omega\mathbf{L}_e & \mathbf{K} \\ -\mathbf{K}^T & \mathbf{R}_m + i\omega\mathbf{L}_m \end{bmatrix}, \qquad (22)$$

$$\mathbf{j}_{em} = \begin{bmatrix} \mathbf{j}_e \\ \mathbf{j}_m \end{bmatrix}, \ \boldsymbol{\phi}_{em} = \begin{bmatrix} \boldsymbol{\phi}_e \\ \boldsymbol{\phi}_m \end{bmatrix}, \ \mathbf{eh}_0 = \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{h}_0 \end{bmatrix} \qquad (23)$$

and equivalently for the other quantities in (21).

The preconditioner proposed in this paper is obtained starting from the following system

$$\boldsymbol{\Lambda} = \begin{bmatrix} diag(\mathbf{Z}_{em}) & \mathbf{A}_{em}^T \\ diag(\mathbf{P}_{em})\mathbf{A}_{em} & -i\omega\mathbf{1} \end{bmatrix}, \qquad (24)$$

where $diag()$ extracts the diagonal of the matrices. System (24) somehow approximates the original system (21) since, by extracting the diagonals of $\mathbf{Z}_{em}$ and $\mathbf{P}_{em}$, all the mutual electromagnetic interactions are neglected and only the self interactions are considered.

Then, as in [23], the Schur complement is applied to efficiently perform $\mathbf{y} = \boldsymbol{\Lambda}^{-1}\mathbf{x}$, i.e.

$$\begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} diag(\mathbf{Z}_{em}) & \mathbf{A}_{em}^T \\ diag(\mathbf{P}_{em})\mathbf{A}_{em} & -i\omega\mathbf{1} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \qquad (25)$$

TABLE I
VALUE OF s_k, WITH $k = 1, \cdots, 6$ FOR THE CONSTRUCTION OF THE
CIRCULANT TENSORS $\mathcal{CK}_{xy,t}$, $\mathcal{CK}_{xz,t}$, AND $\mathcal{CK}_{yz,t}$.

α \ s_k	s_1	s_2	s_3	s_4	s_5	s_6
xy	+1	+1	−1	+1	−1	−1
xz	+1	−1	+1	−1	+1	−1
yz	−1	+1	+1	−1	−1	+1

leading to

$$\mathbf{Sd} = \mathbf{b} - diag(\mathbf{P}_{em})\mathbf{A}diag(\mathbf{Z}_{em})^{-1}, \tag{26}$$

$$\mathbf{c} = diag(\mathbf{Z}_{em})^{-1}(\mathbf{a} - \mathbf{A}^T\mathbf{d}), \tag{27}$$

where $\mathbf{S}$ is the Schur complement

$$\mathbf{S} = -i\omega\mathbf{1} - diag(\mathbf{P}_{em})(\mathbf{A}diag(\mathbf{Z}_{em})^{-1}\mathbf{A}^T). \tag{28}$$

Since $\mathbf{S}$ is very sparse, $\mathbf{d}$ can be cheaply obtained by using an efficient sparse LU factorisation available in MATLAB®, then $\mathbf{c}$ is obtained according to (27).

F. Dealing with Heterogeneous Resistivity

When regions with very different resistivity values are involved, (8) is badly scaled and this may reduce the numerical performances of the iterative solver. It is worth noting that this issue naturally occurs when highly conductive media and dielectric media are considered since, generally, $\sigma_c \gg i\omega\varepsilon_0(\varepsilon_r - 1)$.

As shown by numerical results (Section III), a possible solution to speed-up the solver is the use of a standard Gauss–Seidel iteration scheme [46], i.e. by separately solving the electromagnetic problem (i.e. (8)) for each domain and updating the right–hand–side according to the mutual interactions between the domains. A similar procedure has been proposed in [47] to reduce the overall computational cost of VIE methods, however only focusing on multiple disconnected conductive regions with equal resistivity.

G. Connections with Lumped Circuit Components and Parameters Extraction

Thanks to the circuit interpretation provided by the PEEC formulation, it is very easy to couple external circuits consisting of lumped components with (21). This leads to

$$\begin{bmatrix} \mathbf{Z}_{em} & \mathbf{0} & \mathbf{A}_{em}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_l & \mathbf{A}_{l-em}^T & \mathbf{A}_l^T \\ \mathbf{P}_{em}\mathbf{A}_{em} & \mathbf{P}_{em}\mathbf{A}_{l-em} & -i\omega\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_l & \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{j}_{em} \\ \mathbf{j}_l \\ \phi_{em} \\ \phi_l \end{bmatrix} = \begin{bmatrix} \mathbf{eh}_0 \\ \mathbf{e}_{0l} \\ \mathbf{P}_{em}\mathbf{j}_0 \\ \mathbf{j}_{0l} \end{bmatrix} \tag{29}$$

where $\mathbf{Z}_l$ is the impedance matrix of the lumped components, $\mathbf{A}_l$ is the incidence matrix of the external lumped circuit and $\mathbf{A}_{l-em}$ is the incidence matrix describing connections between the equivalent circuit representing the discretized devices and the external lumped circuit. $\mathbf{j}_l$ and ϕ_l are the vectors of currents and potentials of the external lumped circuit and $\mathbf{j}_0$, $\mathbf{j}_{0l}$, and $\mathbf{e}_{0l}$ are vectors storing possible currents and voltage excitation. System (29) can be solved by means of an iterative solver using the FFT method described in Section II-D. Moreover, a preconditioner like the one proposed in Section II-E can still be constructed by grouping together the matrix blocks of (29).

It is worth noting that the proposed FFT-PEEC method can be also used to extract equivalent circuit parameters such as impedances, and self and mutual inductances and capacitances. Indeed, once the solution of the EM problem is found, equivalent circuit parameters can be extracted from both circuit inspections (involving currents and potentials) or derived from the electric and magnetic energies which can be easily obtained from the solution in terms of DoFs by using

the dense matrix operators. For instance, when only conductive media are present, the magnetic energy, W_m, and the electric energy, W_e, can be obtained as

$$W_m = \frac{1}{4}\mathbf{j}_e^*\mathbf{L}_e\mathbf{j}_e, \tag{30}$$

$$W_e = \frac{1}{4}\mathbf{q}_e^*\phi_e, \tag{31}$$

where $*$ denotes the conjugate transpose and $\mathbf{q}_e = \frac{1}{i\omega}\mathbf{A}_e\mathbf{j}_e$ is the charge density array. Obviously, (30) can be computed efficiently by using the related circulant tensor operator. From (30) and (31) one can easily extract *equivalent* inductance and capacitance, respectively.

H. Non–Linear Materials

Magnetic materials used in Power Electronics are designed to reduce high frequency losses. However, saturation occurs at quite low flux densities, especially for ferrites. Therefore, this phenomenon must be taken into account. This section shows how to extend the FFT-PEEC method to the case of non-linear media. Even if we treat only the case of non–linear magnetic materials, equivalent considerations also hold for non-linear dielectric media.

When non–linear magnetic media are taken into account, the relative permeability is considered as a function of the magnetic field $\mathbf{H}$. Therefore, the equivalent magnetic resistivity ρ_m is a function of $\mathbf{H}$ too (see [21] for the definition of ρ_m).

Usually, data-sheets of materials provide information concerning the B–H curve. Therefore, it is convenient to translate the B–H curve into a function of the type $\rho_m(\mathbf{J}_m)$.

When non–linear materials are involved, only static or time–domain solutions can be solved (non–linear frequency domain problems can be solved only by imposing the first harmonic approximation and using the *effective* B–H curve). Here, for the sake of clarity, only static problems are considered but the extension to the other cases is straightforward.

The (static) non–linear PEEC problem is given by

$$\begin{bmatrix} \mathbf{R}_e & \mathbf{0} & \mathbf{A}_e^T & \mathbf{0} \\ -\mathbf{K}^T & \mathbf{R}_m(\rho_m(\mu_0\mathbf{M})) & \mathbf{0} & \mathbf{A}_m^T \\ \mathbf{A}_e & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_m\mathbf{A}_m & \mathbf{0} & -\mathbf{1} \end{bmatrix}\begin{bmatrix} \mathbf{j}_e \\ \mathbf{m} \\ \phi_e \\ \phi_m \end{bmatrix} = \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{h}_0 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{32}$$

where, with respect to (8), $\mu_0\mathbf{M}$ has been considered as unknown instead of $\mathbf{J}_m$. $\mu_0\mathbf{M}$ is expanded as in (5) and $\mathbf{m}$ stores the corresponding DoFs. Also note that the expression of ρ_m for a static problem is

$$\rho_m(\mathbf{r}) = \frac{1}{\mu_0(\mu_r(\mu_0\mathbf{M}, \mathbf{r}) - 1)}. \tag{33}$$

The non–linear problem (32) can be re–written as: find $\mathbf{x} = [\mathbf{j}_e, \mathbf{m}, \phi_e, \phi_m]^T$ such that

$$\boldsymbol{\mathcal{S}}(\mathbf{x})\mathbf{x} - \mathbf{b} = \mathbf{0} \tag{34}$$

where $\boldsymbol{\mathcal{S}}$ is the matrix in (32), and $\mathbf{b} = [\mathbf{e}_0, \mathbf{h}_0, \mathbf{0}, \mathbf{0}]^T$.

The non–linear problem (34) can be solved by using the Newton–Raphson method which requires the solution of a system of equations involving the Jacobian of (34) several

times. The generic ijth entry of the Jacobian, $\mathcal{J}$, of (34) is given by

$$\mathcal{J}_{ij}(\mathbf{x}) = \mathcal{S}_{ij}(\mathbf{x}) + \sum_{a=1}^{N} \frac{\partial \mathcal{S}_{ia}(\mathbf{x})}{\partial x_j} x_a - \frac{\partial b_i}{\partial x_j}, \quad (35)$$

where x_j is the jth entry of $\mathbf{x}$ and b_i is the ith entry of $\mathbf{b}$.

In the right–hand–side of (35), the most difficult term to evaluate is the second one. However, since only matrix $\mathbf{R}_m$ depends on $\mu_0 \mathbf{M}$ (and therefore from $\mathbf{x}$), its computation reduces to the computation of

$$\mathcal{H}_{ij} = \sum_{a}^{N_{v,m}} \frac{\partial R_{ia}(\mathbf{m})}{\partial m_j} m_a. \quad (36)$$

In the above expression, it is required to evaluate $\frac{\partial \rho_m}{\partial m_j}$. It is convenient to do that as follows

$$\frac{\partial \rho_m}{\partial m_j} = \frac{\partial \rho_m}{\partial \|\mu_0 \mathbf{M}\|} \frac{\partial \|\mu_0 \mathbf{M}\|}{\partial m_j}, \quad (37)$$

where

$$\frac{\partial \|\mu_0 \mathbf{M}\|}{\partial m_j} = 2 \sum_{b=1}^{N_{v,box}} \mathbf{w}_j \cdot \mathbf{w}_b x_b. \quad (38)$$

Then, $\mathcal{H}_{ij}$ is given by

$$\mathcal{H}_{ij} = \sum_{a=1}^{N_{v,box}} \sum_{b=1}^{N_{v,m}} \int_{\Omega_m} \mathbf{w}_i \cdot \mathbf{w}_a \mathbf{w}_j \cdot \mathbf{w}_b 2 x_b x_a \frac{\partial \rho_m}{\|\mu_0 \mathbf{M}\|} d\Omega. \quad (39)$$

It is worth noting that, in general, due to the local support of the shape functions, $\mathcal{H}_{ij}$ can be computed according to (39) with only one *for–loop* on the voxels belonging to $\mathbb{S}_{IDm}$ and adding the local contributions. Thus, the double sum in (39) is converted into the more efficient

$$\sum_{a=1}^{N_{v,box}} \sum_{b=1}^{N_{v,m}} (\cdots) = \sum_{k=1}^{N_{v,box}} \sum_{a=v_f(k,1)}^{v_f(k,6)} \sum_{b=v_f(k,1)}^{v_f(k,6)} (\cdots), \quad (40)$$

where $v_f(k,h)$ provides face ID of the hth face of the kth voxel, with $h = 1, \cdots, 6$.

Finally, the non–linear problem can still be solved by using an iterative solver. At each Newton–Rhapson iteration, a system of equations involving $\mathcal{J}$ must be solved. Actually, the structure of $\mathcal{J}$ is very similar to the original system of equations (32) (matrix $\mathcal{H}$ is added to $\mathbf{R}_m$). Therefore, at each Newton–Rhapson iteration, the same procedure described in Section II-D and Section II-E can be used to efficiently solve

$$\mathcal{J}\Delta\mathbf{x} = \mathbf{f}, \quad (41)$$

by means of an iterative solver and without actually assembling and storing $\mathcal{J}$. The preconditioner is constructed following the same steps of Section II-E and thus selecting the diagonal of $\mathcal{H} + \mathbf{R}_m$.
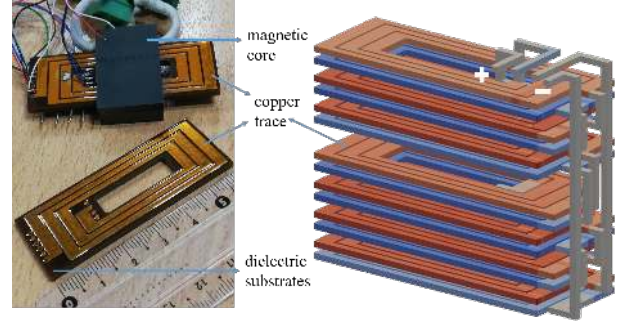

Fig. 5. L–C–T prototype (left) and L–C–T copper trace scaled along the z direction of a factor of 20 (right). Symbols $+$ and $-$ indicate the terminals where the voltage source is connected.

III. NUMERICAL RESULTS AND EXAMPLES OF APPLICATION

The FFT–PEEC method was implemented in MATLAB® coupled with MEX–FORTRAN functions combined with OpenMP libraries. Simulations were run on a Linux machine equipped with a Xeon E5-2643 v4 processor (dual 6-core/12-thread, @3.40 GHz) and 512 GB of RAM. To run the FFT-PEEC code the only required input files are: 1) the *.stl* files of each part with a flag to indicate if they are conductive, dielectric, magnetic media or ports with assigned potential, 2) the material properties and/or the assigned potentials (if ports), 3) the voxel sizes for the vozelization, and 4) the frequencies for the simulation. Then, the mesh generation and the solution is performed automatically by the tool. Sample MATLAB® implementations of the FFT–PEEC method are available at https://github.com/UniPD-DII-ETCOMP/FFT-PEEC.

A. L–C–T Component

The L-C-T component already introduced in Fig. 1 was designed using analytical formulas in [4] and then manufactured, as illustrated in Fig. 5. At that time, due to the complexity of the geometry, the device was too complex to be easily simulated with a numerical tool. The device is a significant test–case since capacitive effects are fundamental for its operation and conductive, dielectric, and magnetic media are present.

All the geometrical and material details of the device are reported in [48]. The device (Fig. 5) consists of 16 copper layers (70 μm thickness each) that compose the first and the second part of the primary winding of the transformer. The 16 copper layers are printed on dielectric substrates composed of Kapton ($\varepsilon_r = 3.3$, 50 μm thick) or Preg ($\varepsilon_r = 4.5$, 140 μm thick). The magnetic core is composed of Ferrite 3F3, which exhibits a complex frequency dependent relative permeability.

The device is fed by a voltage source connected to only two of the four terminals of the primary winding. At low frequency, the device behaves as a capacitor. The primary winding is an open circuit and the current flows from the first part to the second part of the winding thanks to capacitive effects.

The design of this kind of devices is a complex process since they must resonate at a prescribed working frequency and both geometry and material parameters affect their self resonant frequency. The FFT–PEEC method can simplify the design
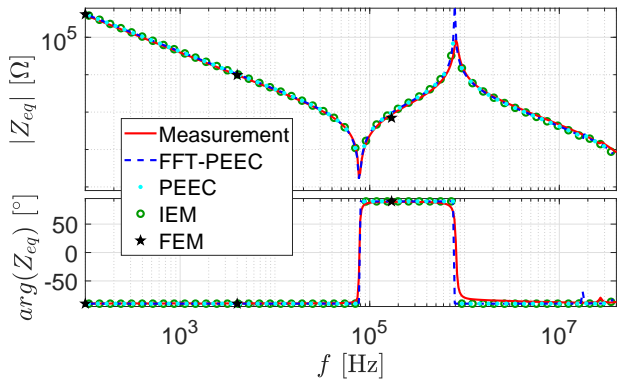
Fig. 6. Magnitude and phase of Z_{eq} of the L–C–T component.
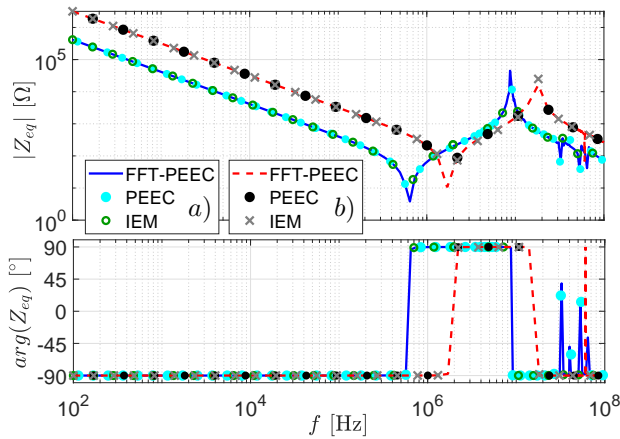

Fig. 7. Magnitude and phase of Z_{eq} of the L–C–T component without magnetic core (a) and with the copper trace only (b).

process of these devices since the electromagnetic simulation only requires the CAD file as input, therefore reducing the human effort. The FFT–PEEC model of the L–C–T component is discretized by $N_x = 67$, $N_y = 127$, and $N_z = 138$ voxels, resulting into $N_{ve} + N_{vm} = 470\,703$ electric and magnetic voxels.

Fig. 6 shows the magnitude and phase of the equivalent impedance (Z_{eq}) obtained from the FFT–PEEC method and compared with: 1) measurements, 2) the unstructured PEEC method proposed in [37], 3) a commercial software based on FEM, and 4) a commercial software based on IE methods. To highlight the importance of considering dielectric and magnetic media, Fig. 7 show the magnitude and phase of Z_{eq} for the device without the magnetic core (a) and with the copper trace only (b). In Table II, the computational details of the numerical methods are reported for a single frequency simulation.

As can be seen from Fig. 6, Fig. 7, and Table II, the results of FFT–PEEC method are in good agreement with measurements and the other numerical approaches. Moreover, the FFT–PEEC method required a smaller computational cost compared to the other methods. Furthermore, although it is very difficult to make a precise quantitative comparison, the FFT-PEEC method required a much lower user time with respect the one required by the FEM method (minutes vs several hours/working-days

TABLE II
COMPUTATIONAL DATA FOR THE L–C–T CASE.

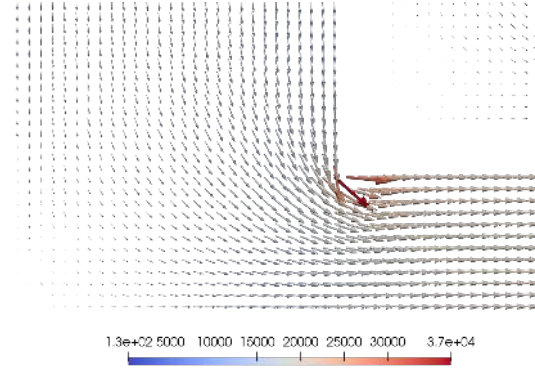|	DoFs	Time (s)	Memory (GB)
FFT–PEEC	1 849 228	647	18
PEEC	18 429	1 278	15
FEM	29 582 652	18 713	430
IEM	35 243	1 892	19

Fig. 8. Current density distribution in the corner of the external turn of the bottom layer copper trace (in A/m^2) at 100 kHz.
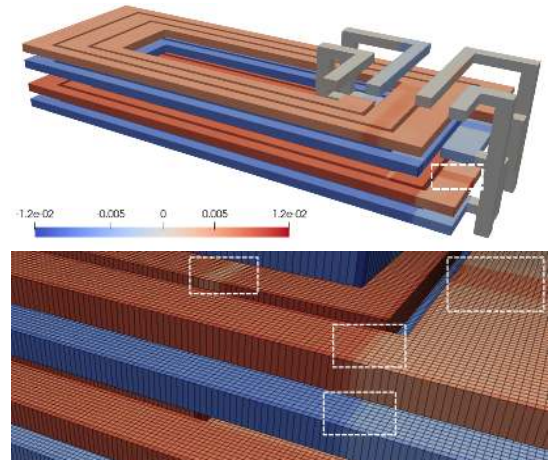

Fig. 9. Charge density distribution in the four top layers of the transformer (top). Detail of the charge density distribution with mesh (bottom). Results are in C/m^3. For the sake of clarity, geometry has been scaled in the z direction of a factor of 20. The white dashed boxes show abrupt changes in charge density due to the presence of nearby corners of adjacent layers. Results are at 100 kHz.

in the authors' experience). Indeed, as can be seen from Fig. 5, the copper trace has a very complex geometry, with very thin thickness and air gaps. The discretization of this kind of geometry, generally, requires a non–negligible time and human effort, especially for FEM models. Instead, with the FFT–PEEC method, the meshing time and the human effort for the discretization is eliminated since the method automatically creates the mesh from the CAD file.

Fig. 8 shows the current density distribution in the corner of the external turn of the bottom layer copper trace. As can be seen, thanks to the high number of mesh elements which can be considered with FFT-PEEC, hot-spots of the current density can be better quantified, which can provide useful information
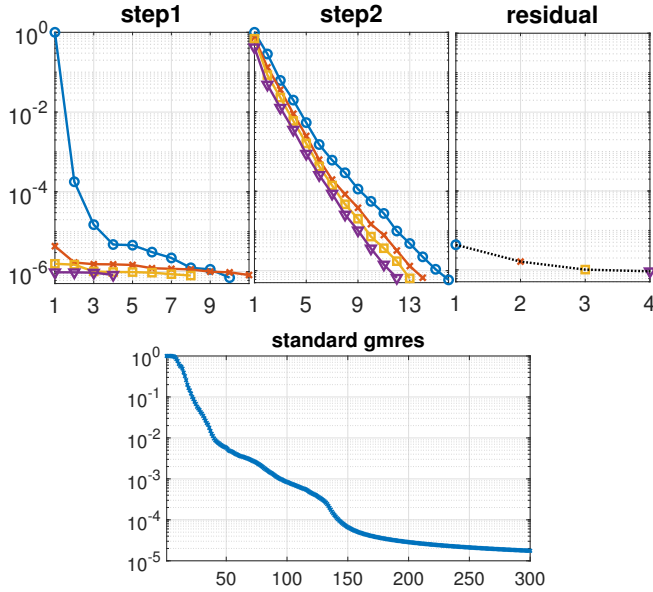
Fig. 10. Convergence of the solver based on the Gauss–Seidel iteration scheme (top) and convergence of a standard GMRES approach (bottom) applied to the L–C–T case at 50 kHz. Blue circles: first iteration, orange crosses: second iteration, yellow squares: third iteration, purple triangles: fourth iteration.
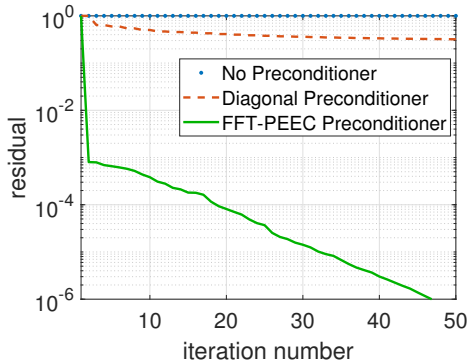

Fig. 11. Convergence of the solver for the L-C-T case at 10 MHz without the preconditioner, with a diagonal preconditioner, and with the proposed preconditioner.

during the design of such devices. Fig. 9, instead, shows the charge density distribution in the four top layers of the primary winding of the transformer. As can be seen in the bottom of Fig. 9, FFT-PEEC is able to accurately model the charge density distribution which determines the capacitive behaviour of the device. Clearly, such kind of information cannot be obtained with analytical formula or too coarse discretizations.

Finally, it is worth noting that, due to the presence of highly conductive media and dielectric media, the numerical approach proposed in Section II-F has been adopted. In particular, the solution is reached by separately solving the conductive trace plus the magnetic core in one step and the dielectric substrate in the second step. Then, the solver iterates between the two steps until the convergence is reached (here fixed to 10^{-6}). The convergence plots (for a frequency equal to 50 kHz) are reported in Fig. 10 and compared with the convergence profile obtained
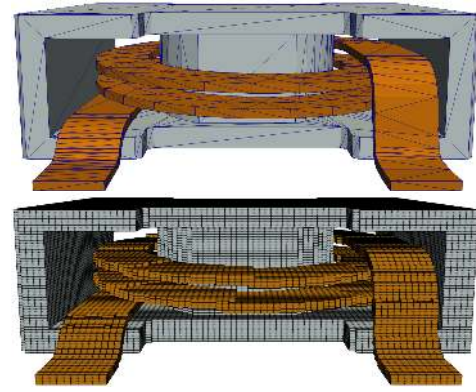

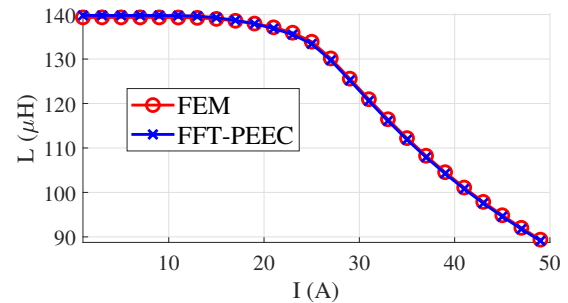Fig. 12. Non–linear Power Inductor *.stl* model (top) and its voxelization (bottom).

Fig. 13. Equivalent inductance of the power inductor for different current values. FFT–PEEC vs FEM.

from a standard GMRES solver without using the approach described in Section II-F. Fig. 11 shows the performance of the proposed preconditioner with respect to both an un-preconditioned GMRES and a standard diagonal preconditioner. Due to the presence of capacitive effects, the FFT-PEEC system is badly scaled and/or possibly ill-conditioned, therefore the adoption of a good preconditioner is fundamental.

B. Power Inductor

In this section, for completeness, the case of a power inductor is considered, taking into account the saturation effect of the magnetic material. For the sake of reproducibility, the *Power Inductor* model provided by COMSOL® and available online has been chosen. The device consists of a current driven coil and a non–linear magnetic core (its **B**–**H** curve is the one provided by COMSOL® for lossless soft iron). The CAD model and its voxelization are shown in Fig. 12. The volume of the power inductor is $0.15 \times 0.1 \times 0.0475$ m^3, which are also the dimensions of Ω_{box}.

In this test case, the inductance of the device has been evaluated by varying the current value, and therefore the saturation of the magnetic core. The results, compared with the one obtained from FEM, are reported in Fig. 13.

The computational performances of the two methods are reported in Table III (for a current of 31 A). As can be seen from Fig. 13 and Table III, although a fully standard Newton–Rhapson method has been used for the FFT–PEEC, the results

TABLE III
COMPUTATIONAL DATA FOR THE NON−LINEAR POWER INDUCTOR CASE.

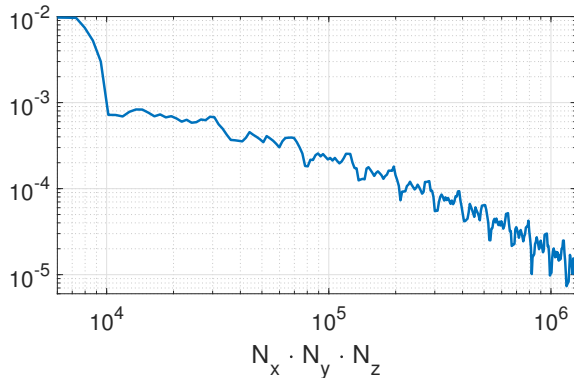|	DoFs	Time (s)	Memory (GB)
FFT–PEEC	1 823 609	1 456	6.30
FEM	2 278 001	2 583	15.63

Fig. 14. Convergence of the error on the inductance of the Power Inductor with respect to the FEM solution versus the number of voxels.

are in good agreement and the computational cost and memory of FFT–PEEC are abut half of those of FEM.

Finally, Fig. 14 shows the convergence of the error on the inductance of the device with respect to the FEM solution as a function of the number of voxels used for the discretization. As can be seen, Fig. 14 shows a rapid decrease (which is typical of IE). Fluctuations are due to the discrete representation of the device, which introduces deviations from the original geometry. The effects of such imperfect representation also decrease with the number of voxels, i.e. by reducing the sizes of the voxels.

C. L-Shaped Trace Over a Ground Plane

Finally, in this section, to further test the accuracy of the FFT-PEEC tool, the simple case of an L-shaped conductive trace over a conductive ground plane is considered. Contrary to the LCT primary winding, this component has a predominantly resistive-inductive behaviour and the equivalent impedance of the device obtained from the FFT-PEEC can be compared against the one obtained from the well-known FastHenry tool [27], which exploits the magnetoquasistatic approximation therefore neglecting capacitive effects. Moreover, results are also compared with the already mentioned full-Maxwell IE method. Fig. 15 shows the model of the component. The ground plate has a 0.01 mm thickness whereas the L-shaped trace has a 0.01 mm thickness and a 4 mm width. Both the ground plate and the L-trace have a 10^7 S/m conductivity. The distance between the ground plate and the trace is 0.5 mm. A voltage source connects the trace with the ground plate in (A), whereas a short circuit is placed between the trace and the plate in (B) to close the path.

The magnitude of the equivalent impedance seen from the voltage source computed with the three methods is reported in Fig. 16. As expected, the three methods are in good accordance for a wide frequency range (from 0 Hz to ∼145 MHz) until capacitives effects become significant. Indeed, since FastHenry
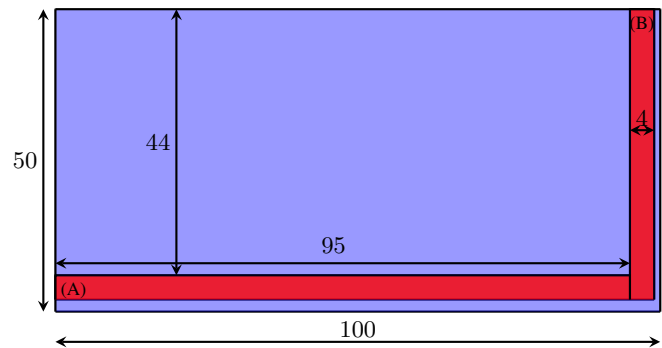

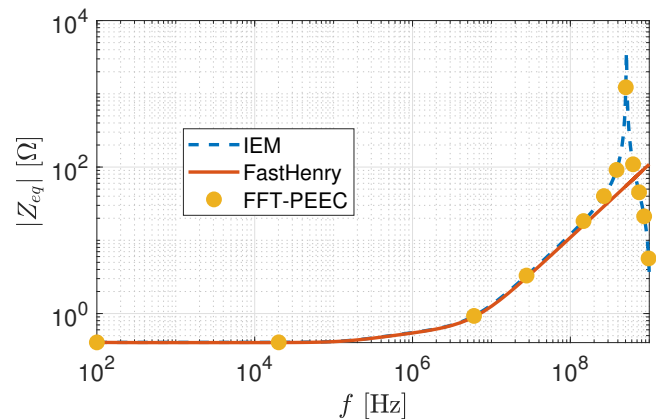Fig. 15. Top view of the L-shaped trace (red) over the ground plane blue). Dimensions are in mm.

Fig. 16. Magnitude of the equivalent impedance of the L-shape trace over the ground plate vs frequency.

is based on the magnetoquasitatic approximation, it cannot follow the reference curve (i.e. the full-Maxwell IE one) for the highest frequencies. FFT-PEEC and the IE method, instead, agree very well for the whole frequency range since they both solve the full-Maxwell equations.

For this simple case, all the three methods required less than 10 seconds for each frequency solution and less than 100 MB.

IV. CONCLUSION

A fast Partial Element Equivalent Circuit (PEEC) method based on the Fast–Fourier–Transform (FFT) has been presented. In the proposed method the discretization is performed automatically by means of a voxelization procedure based on ray intersection. Therefore, the FFT–PEEC method only requires common CAD input files, thus significantly reducing the human effort and, in general, the overall time for the pre–processing. Thanks to the FFT approach, both memory and computation time are significantly reduced, without the need of applying matrix compression (i.e. approximations). The proposed method has been developed starting from the results of [23], and then extended for considering both inductive and capacitive effects, with focus of mid–frequency applications among which, for instance, printed circuit boards and electronic components for the automotive industry. Conductive, dielectric, and magnetic media can be considered by the proposed FFT–PEEC method and both linear and non–linear media can
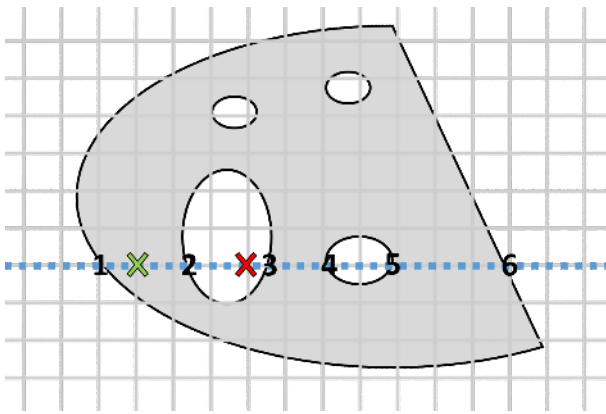
Fig. 17. 2D exemplification of ray tracing voxelization technique. The gray domain represents the device. Nodes of the superimposed gray grid are the barycenter of voxels. Numbers represent the intersection counter of the blue dashed line of the grid with the object, from left to right order. The green cross indicates a voxel which belongs to the object since it has, at its left, a odd number of intersections. The red cross indicates a voxel which does not belong to the object since it is has, at its left, a even number of intersections.

be considered. As proposed in [49], further computational improvements can be obtained with the adoption of a Graphical Processing Unit (GPU) that can speed-up the voxelization process and, mostly, the matrix-vector products via FFT. Moreover, when the considered devices have parts with very different dimensions (e.g. bulky components and bonding wires) one can adopt FFT-PEEC for the bulky components and couple it with a standard Integral Equation method for the smaller part like bounding wires. Furthermore, the FFT-PEEC can be extended to unstructured meshes (therefore eliminating the problem of using a global voxel size) by exploiting the advanced precorrected-FFT technique [50], [51]. Finally, further improvements can be also achieved by combining FFT-PEEC with Model Order Reduction (MOR) techniques, which allow for generating computationally inexpensive equivalent models of the considered devices. Starting from the proposed FFT-PEEC tool, future works will exploit such MOR techniques in order to provide fast and accurate models capable of considering all the EM effects of interest in Power Electronics applications, such as electromagnetic interference modeling and signal integrity analysis.

APPENDIX A
RAY INTERSECTION VOXELIZATION

In this work, the voxelization of the devices is obtained by using a ray intersection technique. The input of such approach (i.e. the information which the user must provide) are only the CAD file (e.g. in STL format) and the desired resolution of the voxelization. Then, the ray intersection voxelization algorithm automatically creates the regular structured mesh in an efficient way, i.e. it finds the index sets $\mathbb{S}_{IDe}$ and $\mathbb{S}_{IDm}$.

At the beginning, the algorithm generates a structured 3D grid of the desired resolution which contains the devices. Nodes of this grid correspond to the barycenters of the voxels. Then, the algorithm iterates along each xy-plane, generating rays along axes, and finding the intersection points of the rays with the boundary facets of the devices. Indeed, CAD files (like STL

format) represent the devices by means of triangular facets. Since the number of intersections of each ray with the facets of the device is always even, it is possible to define a criterion to determine if a voxel belongs or not to the device. Thus, if a voxel has at its left an odd number of intersections it actually belongs to the device. Instead, if the voxel has at its left an even number of intersections it does not belong to the device. Fig. 17 shows an exemplification of such ray intersection approach. This technique is embarrassingly parallel and is very efficient, i.e. mesh with millions of voxels are generated in only some seconds (about 4 seconds for the LCT case).

Therefore, the ray intersection technique allows for determining the index sets $\mathbb{S}_{IDe}$ and $\mathbb{S}_{IDm}$, i.e. the voxels IDs belonging to the devices. It is worth noting that a standard mesh also requires other information such as the connections of voxels with each others. However, since the mesh is structured and regular, such information can be evaluated during the construction of the matrices with a truly negligible computational effort.

REFERENCES

[1] C. Nan, R. Ayyanar, and Y. Xi, "A 2.2-MHz Active-Clamp Buck Converter for Automotive Applications," *IEEE Transactions on Power Electronics*, vol. 33, no. 1, pp. 460–472, 2018.
[2] A. Stippich, C. H. Van Der Broeck, A. Sewergin, A. H. Wienhausen, M. Neubert, P. Schülting, S. Taraborrelli, H. van Hoek, and R. W. De Doncker, "Key components of modular propulsion systems for next generation electric vehicles," *CPSS Transactions on Power Electronics and Applications*, vol. 2, no. 4, pp. 249–258, 2017.
[3] L. Han, L. Liang, Y. Kang, and Y. Qiu, "A Review of SiC IGBT," *IEEE Transactions on Power Electronics*, pp. 1–1, 2020.
[4] Y. Lembeye, P. Goubier, and J.-P. Ferrieux, "Integrated planar L-C-T component: Design,Characterization and Experimental Efficiency analysis," *IEEE Transactions on Power Electronics*, vol. 20, no. 3, pp. 593–599, 2005.
[5] E. C. W. de Jong, J. A. Ferreira, and P. Bauer, "Lct integration optimization on a printed circuit board technology, platform," in *Conference Record of the 2006 IEEE Industry Applications Conference Forty-First IAS Annual Meeting*, vol. 4, 2006, pp. 1849–1856.
[6] S. Mao and Y. Zhang, "Design and characterization of planar integrated passive component for power converters," in *Proceedings of the 2011 14th European Conference on Power Electronics and Applications*, 2011, pp. 1–6.
[7] W. Tan, X. Margueron, L. Taylor, and N. Idir, "Leakage inductance analytical calculation for planar components with leakage layers," *IEEE Transactions on Power Electronics*, vol. 31, no. 6, pp. 4462–4473, 2016.
[8] W. Tan, X. Margueron, and N. Idir, "Analytical modeling of parasitic capacitances for a planar common mode inductor in emi filters," in *2012 15th International Power Electronics and Motion Control Conference (EPE/PEMC)*, 2012, pp. DS3f.2–1–DS3f.2–6.
[9] I. Kovacevic-Badstuebner, D. Romano, G. Antonini, J. Ekman, and U. Grossner, "Broadband circuit-oriented electromagnetic modeling for power electronics: 3-d peec solver vs. rlcg-solver," *Energies*, vol. 14, no. 10, 2021.
[10] A. E. Ruehli, G. Antonini, and L. Jiang, *Circuit Oriented Electromagnetic Modeling Using the PEEC Techniques*. John Wiley & Sons, Inc., 2017.
[11] A. E. Ruehli, G. Antonini, J. Esch, J. Ekman, A. Mayo, and A. Orlandi, "Nonorthogonal PEEC formulation for time- and frequency-domain EM and circuit modeling," *IEEE Transactions on Electromagnetic Compatibility*, vol. 45, no. 2, pp. 167–176, 2003.

[12] V. Ardon, J. Aime, O. Chadebec, E. Clavel, J. Guichon, and E. Vialardi, "EMC Modeling of an Industrial Variable Speed Drive With an Adapted PEEC Method," *IEEE Transactions on Magnetics*, vol. 46, no. 8, pp. 2892–2898, 2010.

[13] T. Bauernfeind, P. Baumgartner, O. Bíró, A. Hackl, C. Magele, W. Renhart, and R. Torchio, "Multi-objective synthesis of nfc-transponder systems based on peec method," *IEEE Transactions on Magnetics*, vol. 54, no. 3, pp. 1–4, 2018.

[14] V. Cirimele, R. Torchio, A. Virgillito, F. Freschi, and P. Alotto, "Challenges in the electromagnetic modeling of road embedded wireless power transfer," *Energies*, vol. 12, no. 14, 2019. [Online]. Available: https://www.mdpi.com/1996-1073/12/14/2677

[15] D. Voltolina, P. Bettini, P. Alotto, F. Moro, and R. Torchio, "High-Performance PEEC Analysis of Electromagnetic Scatterers," *IEEE Transactions on Magnetics*, vol. 55, no. 6, pp. 1–4, 2019.

[16] D. Voltolina, R. Torchio, P. Bettini, R. Cavazzana, and M. Moresco, "PEEC Modeling of Planar Spiral Resonators," *IEEE Transactions on Magnetics*, vol. 56, no. 1, pp. 1–4, 2020.

[17] G. Antonini, "Fast multipole method for time domain PEEC analysis," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 275–287, 2003.

[18] G. Gabriadze, G. Chiqovani, A. Gheonjian, I. Oganezova, A. Demurov, Z. Kutchadze, X. Bunlon, F. Ajebbar, and R. Jobava, "Enhanced PEEC Model Based on Automatic Voronoi Decomposition of Triangular Meshes," *IEEE Transactions on Electromagnetic Compatibility*, pp. 1–13, 2020.

[19] J. He, S. Tao, and H. Wu, "A PEEC-Based Concise Broadband Physical Circuit Modeling Method With Parameter Extraction for PCB Inductive Components," *IEEE Transactions on Power Electronics*, vol. 35, no. 10, pp. 10 852–10 862, 2020.

[20] T. Schellenberg and M. Schneider, "GMRES-FFT accelerated magneto-quasistatic calculations using PEEC," in *2013 International Symposium on Electromagnetic Compatibility*, 2013, pp. 653–658.

[21] R. Torchio, "A Volume PEEC Formulation Based on the Cell Method for Electromagnetic Problems From Low to High Frequency," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 12, pp. 7452–7465, 2019.

[22] A. Polimeridis, J. Villena, L. Daniel, and J. White, "Stable FFT-JVIE solvers for fast analysis of highly inhomogeneous dielectric objects," *Journal of Computational Physics*, vol. 269, pp. 280 – 296, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999114002071

[23] A. C. Yucel, I. P. Georgakis, A. G. Polimeridis, H. Bağcı, and J. K. White, "VoxHenry: FFT-Accelerated Inductance Extraction for Voxelized Geometries," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 4, pp. 1723–1735, 2018.

[24] M. Wang, C. Qian, E. D. Lorenzo, L. J. Gomez, V. Okhmatovski, and A. C. Yucel, "Supervoxhenry tucker-enhanced and fft-accelerated inductance extraction for voxelized superconducting structures," *CoRR*, vol. abs/2105.08627, 2021. [Online]. Available: https://arxiv.org/abs/2105.08627

[25] S. Patil and B. Ravi, "Voxel-based representation, display and thickness analysis of intricate shapes," in *Ninth International Conference on Computer Aided Design and Computer Graphics (CAD-CG'05)*, 2005, pp. 6 pp.–.

[26] D. Romano, G. Antonini, and A. E. Ruehli, "Time-Domain Partial Element Equivalent Circuit Solver Including Non-Linear Magnetic Materials," *IEEE Transactions on Magnetics*, vol. 52, no. 9, pp. 1–11, 2016.

[27] M. Kamon, M. J. Tsuk, and J. K. 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.

[28] 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.

[29] E. Vialardi, E. Clavel, O. Chadebec, J.-M. Guichon, and M. Lionet, "Electromagnetic simulation of power modules via adapted modelling tools," in *Proceedings of 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010*, 2010, pp. T2–78–T2–83.

[30] F. Freschi and M. Repetto, "A General Framework for Mixed Structured/Unstructured PEEC Modelling," *Applied Computational Electromagnetics Society Journal*, vol. 23, pp. 200–206, 09 2008.

[31] P. Bettini and F. Trevisan, "Electrostatic analysis for plane problems with finite formulation," *IEEE Transactions on Magnetics*, vol. 39, no. 3 I, pp. 1127–1130, 2003.

[32] M. Schwarz and H.-P. Seidel, "Fast Parallel Surface and Solid Voxelization on GPUs," *ACM Transactions on Graphics*, vol. 29, no. 6 (Proceedings of SIGGRAPH Asia 2010), pp. 179:1–179:9, Dec. 2010.

[33] A. H. Aitkenhead, "Mesh voxelisation," https://www.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation, 2013.

[34] J. Baert, "Cuda Voxelizer: A GPU-accelerated Mesh Voxelizer," https://github.com/Forceflow/cuda_voxelizer, 2017.

[35] S. Safavi and J. Ekman, "Feasibility analysis of specialized PEEC solvers in comparison to SPICE-like solvers," *Journal of Computational Electronics*, vol. 11, no. 4, pp. 440–452, Dec. 2012.

[36] F. Freschi, "Fast Block-Solution of PEEC Equations," *IEEE Transactions on Magnetics*, vol. 49, no. 5, pp. 1753–1756, 2013.

[37] R. Torchio, F. Moro, G. Meunier, J.-M. Guichon, and O. Chadebec, "An Extension of Unstructured-PEEC Method to Magnetic Media," *IEEE Transactions on Magnetics*, vol. 55, no. 6, pp. 1–4, 2019.

[38] S. Ahyoune, J. Sieiro, T. Carrasco, N. Vidal, J. M. López-Villegas, E. Roca, and F. V. Fernández, "Quasi-static PEEC planar solver using a weighted combination of 2D and 3D analytical Green's functions and a predictive meshing generator," *Integration*, vol. 63, pp. 332 – 341, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167926017306235

[39] Y. Dou and K. Wu, "A Passive PEEC-Based Micromodeling Circuit for High-Speed Interconnection Problems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 3, pp. 1201–1214, 2018.

[40] H. Gan and W. Chew, "A discrete BCG-FFT algorithm for solving 3D inhomogeneous scatterer problems," *Journal of Electromagnetic Waves and Applications*, vol. 9, no. 10, pp. 1339–1357, 1995. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1163/156939395X00082

[41] D. Lee, "Fast multiplication of a recursive block Toeplitz matrix by a vector and its application," *Journal of Complexity*, vol. 2, no. 4, pp. 295 – 305, 1986. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0885064X86900075

[42] A. W. Bojanczyk, "Some complexity results in parallel matrix-based signal processing," *Journal of Complexity*, vol. 6, no. 3, pp. 324 – 336, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0885064X9090026A

[43] P. Zwamborn and P. M. van den Berg, "The three dimensional weak form of the conjugate gradient FFT method for solving scattering problems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 40, no. 9, pp. 1757–1766, 1992.

[44] C. Y. Shen, K. J. Glover, M. I. Sancer, and A. D. Varvatsis, "The discrete Fourier transform method of solving differential-integral equations in scattering theory," *IEEE Transactions on Antennas and Propagation*, vol. 37, no. 8, pp. 1032–1041, 1989.

[45] N. V. Budko and A. B. Samokhin, "Spectrum of the volume integral operator of electromagnetic scattering," *SIAM Journal on Scientific Computing*, vol. 28, no. 2, pp. 682–700, 2006. [Online]. Available: https://doi.org/10.1137/050630660

[46] A. Ghai, C. Lu, and X. Jiao, "A comparison of preconditioned Krylov subspace methods for large-scale nonsymmetric linear systems," *Numerical Linear Algebra with Applications*, vol. 26, no. 1, p. e2215, 2019, e2215 nla.2215. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nla.2215

[47] M. Passarotto, R. Specogna, and C. Geuzaine, "Fast Iterative Schemes for the Solution of Eddy-Current Problems Featuring Multiple Conductors by Integral Formulations," *IEEE Transactions on Magnetics*, vol. 56, no. 3, pp. 1–4, 2020.

[48] P. Goubier, "Etude et Conception de composants passifs L–C–T intégrés," *Ph.D. dissertation, Univ. Grenoble, Grenoble, France*, 2003.

[49] I. I. Giannakopoulos, M. S. Litsarev, and A. G. Polimeridis, "Memory Footprint Reduction for the FFT-Based Volume Integral Equation Method via Tensor Decompositions," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 12, pp. 7476–7486, 2019.

[50] J. R. Phillips and J. K. White, "A precorrected-fft method for electrostatic analysis of complicated 3-d structures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 10, pp. 1059–1072, 1997.

[51] V. Okhmatovski, M. Yuan, I. Jeffrey, and R. Phelps, "A three-dimensional precorrected fft algorithm for fast method of moments solutions of the mixed-potential integral equation in layered media," *IEEE Transactions on Microwave Theory and Techniques*, vol. 57, no. 12, pp. 3505–3517, 2009.