

Matrix-Free Edge-Domain Decomposition Method for Massively Parallel 3-D Finite Element Simulation With Field-Circuit Coupling

Jiacong Li¹, Peng Liu¹, and Venkata Dinavahi¹, *Fellow, IEEE*

Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada

In this article, a novel edge-domain decomposition (EDD) method is proposed to solve 3-D nonlinear finite element (FE) problems of electromagnetic devices and transient field circuit co-simulation. The method applies reduced magnetic vector potential formulation to discretize the physical problem based on 3-D edge elements, and the solution region is divided into many sub-domains that only contain one edge unknown. The solution of lightweight nonlinear sub-domain systems can be massively parallelized, and the neighbor-to-neighbor communication scheme eliminates the need to assemble the global FE matrix. This article also introduces an indirect coupling scheme to handle large eddy currents to interface the EDD FE system with external circuits. The abovementioned algorithms are then implemented on a many-core GPU for transient field circuit co-simulation. The result shows an auto-gauging property, and the comparison with a commercial FE software indicates a speedup of over 43 times with relative error less than 2%.

Index Terms—3-D edge element, domain decomposition, Eddy current field, field-circuit coupling, finite element (FE) method, graphics processors, nonlinear, parallel processing, reduced magnetic vector potential.

I. INTRODUCTION

THE finite element method (FEM) has become one of the most commonly utilized tools in the modeling and design of electro-magnetic apparatus, such as transformers and power inductors, due to its superior precision and ability to handle complex geometries. Such good accuracy gives raise to finite element (FE) and circuit co-simulations [1], where the computation for nonlinear device behaviors is always encountered.

Traditionally, the physical problem of the device is converted to a large global FE matrix system. The Newton–Raphson (NR) method is then implemented to deal with material nonlinearity in the global matrix. However, the NR scheme requires repetitive assembling and solution of the global Jacobian matrix at each iteration step. Moreover, if the external circuit equations are integrated into the global Jacobian matrix, the solvability of the matrix system can be easily damaged [2]. These properties lead to a severe computational burden.

On the other hand, the development of computation hardware has reached a bottleneck on the increment of single-core clock speed during the past decade, and the prevalent trend is to increase the number of processing cores to achieve better performance. Multi-core CPUs and many-core GPUs are widely applied for high-performance computing. For example, the recently released NVIDIA Tesla V100 GPU is equipped with 5120 Cuda cores and 16 GB HBM2 memory [3], which encourages researchers to develop massively parallelized algorithms to exploit such computational power [4].

Under such a trend, many different methods were proposed to adapt the time-consuming nonlinear FE algorithm for parallel architectures. For example, the domain decomposition methods, including the overlapping Schwartz scheme [5] and the non-overlapping FETI [6] scheme, divide the global system into several smaller sub-domain problems so that the lighter sub-domain matrices can be solved in parallel. To accelerate the solution of those matrices, the super LU/paralleled conjugate gradient method was introduced. To avoid frequent updating for nonlinearity in those matrices, the transmission line modeling method [7] was implemented.

However, the abovementioned methods still need large matrices, and massive parallelism is hardly achieved since piecewise information, generated from each domain element, is integrated into one big system and is solved simultaneously in a single step. This is a centralized way of thinking. A decentralized thinking pattern, in contrast, could fully unlock the computational power of massively parallelized architectures. For example, the GPU-based algorithm achieved element-wise parallelism to accelerate matrix-vector multiplication [8], and a decent speedup was observed.

In addition, a novel decentralized scheme called nodal-domain decomposition relaxation (NDDR) was introduced recently for 2-D nodal triangular FEM [9]. The method has the following key features. All sub-domains are shrunk into a minimum size at a single-node level, and each sub-domain has only 1 degree of freedom (DOF) that can be independently solved. Also, each sub-domain only communicates with its neighboring domains in a distributed manner. Benefitting from the abovementioned properties, the method allows the handling of material nonlinearity during iterations, as well as node-level parallelism and matrix-free computation.

Despite the excellent modularity for massively parallelized architectures, the abovementioned nonlinear NDDR method was only developed for 2-D triangular elements. While,

Manuscript received May 21, 2020; revised July 8, 2020; accepted July 18, 2020. Date of publication July 30, 2020; date of current version September 18, 2020. Corresponding author: P. Liu (e-mail: pliu3@ualberta.ca).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMAG.2020.3013143

in field-circuit co-simulations, the geometry of a nonlinear EM device is always modeled in 3-D for better precision, a 3-D time-domain nonlinear FEM with similar technique is rarely seen. Furthermore, the abovementioned 2-D NDDR FEM was designed based on nodal elements in a scalar form. However, nodal scalar elements may reduce accuracy around sharp corners of the geometry [10]. If the material involves large permeability differences, the numerical error becomes substantially large, regardless of the selection of the penalty factor. The finite edge elements with vector interpolation function, in contrast, do not have the above problems [11]. Based on the aforementioned facts, it has become critically important to explore new algorithms to integrate the abovementioned decentralized-domain decomposition scheme with 3-D edges elements, especially for nonlinear transient field-circuit co-simulation where a huge amount of computational resource is needed.

In this article, we extend our former effort of the 2-D NDDR scheme [9] further to the nonlinear 3-D edge elements. It shares the similar decentralized idea of applying domain-decomposition to each node and element. However, rather than solving a simple 2-D problem with scalar nodal unknowns, we achieved edge-level parallelism on the reduced magnetic vector potential (RMVP) for 3-D vector elements with much larger DOFs. Different convergence and gauging behaviors are also explored. Furthermore, we purpose a field-circuit coupling scheme that allows large eddy current and separate solutions of circuit and FE systems with high accuracy. Inductor and transformer cases are studied to verify the accuracy and efficiency of the proposed edge-domain decomposition (EDD) and field-circuit coupling scheme, and comparison with Comsol indicates a good accuracy and decent speedup of over 43.

This article is arranged as follows. Section II introduces the eddy current RMVP formulation discretized by edge element interpolation functions. In Section III, the decentralized idea is extracted from traditional schemes and is extended as the new EDD method. Section IV proposes the field-circuit coupling technique, which is then integrated with the EDD method in the case studies shown in Section V. Finally, Sections VI and VII give the discussion and the conclusion, respectively, of the work.

II. FEM FORMULATION FOR EDDY CURRENT ANALYSIS

A. Reduced Magnetic Potential Formulation

Quasi-static Maxwell's equations describe the physical restriction of magnetic flux density (\vec{B}) in the eddy current analysis. However, directly solving the Maxwell's equations leads to inconvenience in numerical computation since the parallel component of \vec{B} jumps at material interfaces [12]. To avoid such discontinuity, potential functions (\vec{A} - ϕ formulation) were applied to achieve smooth solution between different materials [12]. The \vec{A} - ϕ formulation can be further simplified to the reduced \vec{A} formulation when edge element is used, which allows perpendicular jump of \vec{A} between different conductivities

$$\nabla \times (v \nabla \times \vec{A}) + \sigma \frac{\partial \vec{A}}{\partial t} - \vec{J}_e = 0 \quad (1)$$

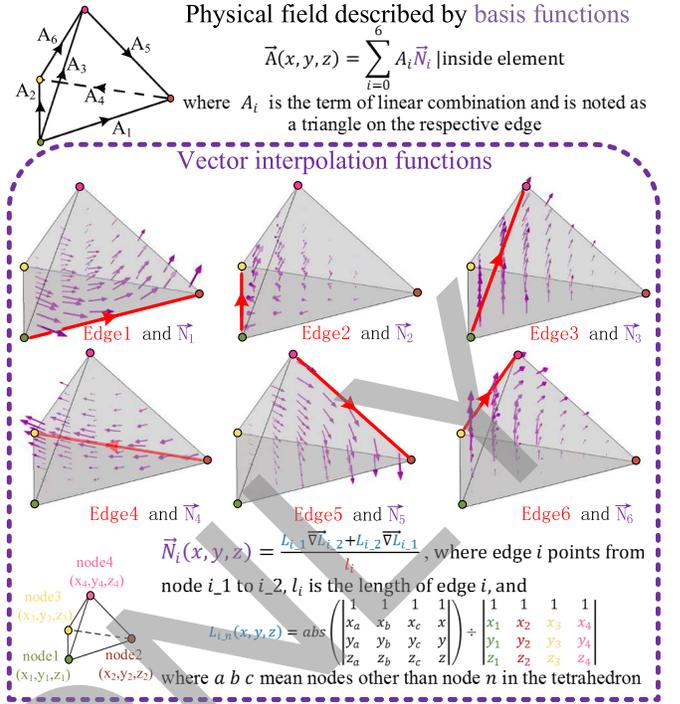


Fig. 1. Vector edge interpolation functions.

where v is the field-dependent reluctivity, σ is the electrical conductivity, and \vec{J}_e is the impressed current density.

B. Finite Elements and Discretized Formulation

When the FEM is applied, the abovementioned equation can be simplified into limited DOFs. To achieve this, the method divides the whole solution domain into many smaller elements and represents the unknown field with a linear combination of known pattern fields (interpolation functions) inside each element. As shown in Fig. 1, the interpolation functions (\vec{N}_i) are fully determined by the coordinates of the four elemental nodes (known), and \vec{N}_i 's projection is 1 on edge i while 0 on all other edges. Due to such property, the linear combination terms (A_i) are also equal to the projection of \vec{A} on the respective edges. Once these terms (DOFs) are settled properly, the unknown field is successfully solved in the element.

The Galerkin weighted residual method can be used to solve those terms. The residual is defined by discretizing the left-hand side of (1) through the upper equation in Fig. 1. If one multiplies the residual with six weighing functions (chosen to be \vec{N}_i) separately and force the integration of the resultant product to be zero, the 6×6 matrix-form elemental discretized formulation is obtained

$$[K_{ij}]_{6 \times 6} [A_j]_{6 \times 1} + [D_{ij}]_{6 \times 6} \frac{\partial}{\partial t} [A_j]_{6 \times 1} = I[f_i]_{6 \times 1} + [\tau_i]_{6 \times 1} \quad (2)$$

where

$$K_{ij} = \int_{V_e} v^e (\nabla \times \vec{N}_i) \cdot (\nabla \times \vec{N}_j) dV, \quad i, j \in [1, 6] \quad (3)$$

$$D_{ij} = \int_{V_e} \sigma^e \vec{N}_i \cdot \vec{N}_j dV, \quad i, j \in [1, 6] \quad (4)$$

$$f_i = \int_{V_e} \vec{N}_i \cdot \vec{J}_{\text{unit}} dV, \quad i \in [1, 6] \quad (5)$$

$$\tau_i = \int_{S_e} (\nu \nabla \times \vec{A}) \times \vec{ds} \cdot \vec{N}_i, \quad i \in [1, 6] \quad (6)$$

where V_e is the volume, S_e is the surface boundary of the element, \vec{J}_{unit} is the coil current density at 1-A coil current, and I is the known current in the coil that changes with time.

The time derivative in (2) can also be discretized using the backward Euler method

$$[M_{ij}]_{6 \times 6} \cdot [A_j^t]_{6 \times 1} = [b_i]_{6 \times 1} + [\tau_i]_{6 \times 1} \quad (7)$$

where

$$[b_i]_{6 \times 1} = I[f_i]_{6 \times 1} + [D_{ij}]_{6 \times 6} / \Delta t \cdot [A_j^{t-1}]_{6 \times 1} \quad (8)$$

$$[M_{ij}]_{6 \times 6} = [K_{ij}]_{6 \times 6} + [D_{ij}]_{6 \times 6} / \Delta t \quad (9)$$

where Δt is the time-step length, A_j^t is the unknowns to solve for at current time-step, and A_j^{t-1} is A_j at the previous time-step.

When the element is inside a ferromagnetic material region, ν_e become dependent on magnetic flux density \vec{B} , and \vec{B} is a function of A_i . This means that (7) becomes a nonlinear equation of A_i . The common NR iteration can be used to solve the nonlinear system. According to the method, (7) becomes a form of the Jacobian matrix and residual. The following equations are necessary to form the Jacobian matrix:

$$\vec{B} = \nabla \times \vec{A} = \sum_{i=1}^6 A_i \cdot \nabla \times \vec{N}_i = \sum_{i=1}^6 A_i \cdot \frac{2\nabla L_{i1} \times \nabla L_{i2}}{l_i} \quad (10)$$

$$\frac{\partial \nu_e}{\partial A_i} = \frac{\partial \nu_e}{\partial B^2} \cdot \frac{\partial B^2}{\partial A_i} \quad (11)$$

where $(\partial \nu_e / \partial B^2)$ is determined by B - H curve of the material.

However, the abovementioned discussion only gives the restriction of six local A_i 's inside a single element. To achieve solutions over the entire domain, all elements must communicate with each other. The communication is subject to Maxwell's equations.

- 1) Every edge has only one global direction and value A_k . When one edge is shared by multiple elements, all local A_i 's of such an edge in different elements are equal to A_k . All local \vec{N}_i 's adjust signs so that their components on the edge all point in the global direction. This ensures the continuity of tangential \vec{A} ; thus, normal \vec{B} is continuous between elements [11].
- 2) On a triangular surface shared by two elements, τ_i 's in the two elements are opposite to each other. As a result, tangential $\nu \nabla \times \vec{A}$ or \vec{H} is continuous between elements.

How to effectively handle the abovementioned elemental nonlinear equation and inter-element communication consistency presents a challenge to different methods. This will be discussed in Section III.

III. EDGE-DOMAIN DECOMPOSITION TO SOLVE THE GLOBAL NONLINEAR SYSTEM

As shown in Fig. 2(a), the traditional method only has one domain for the whole solution area. The method explicitly enforces elemental restrictions and global consistency

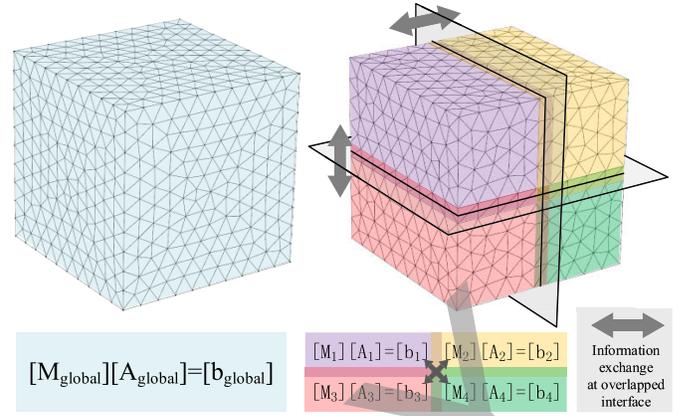


Fig. 2. Left: traditional single-domain FEM. Right: overlapping-domain decomposition method with four sub-domains marked in different colors.

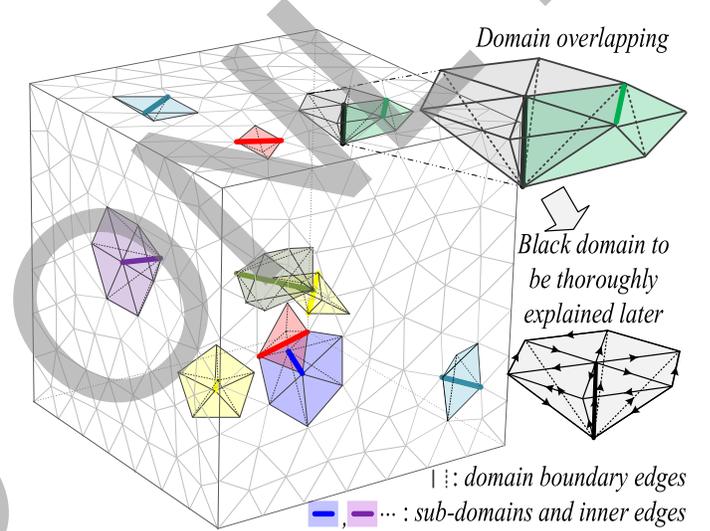


Fig. 3. EDD scheme with sub-domains containing only one unknown edge element.

through the assembly process. During assembly, one global matrix system is formed, where each edge only has one global unknown (the global counterpart of multiple elemental unknowns sharing the edge). Elemental matrices and vector are added directly to the global matrices/vector at the position of the global counterparts, and the surface integration of (6) does not appear in the system since they are canceled out. Thus, all local and inter-element restrictions are respected. After the application of a proper global-domain boundary condition, the matrix system can be solved. However, the sparse global matrix may have several millions of unknowns (total number of edges in the domain). In addition, the matrix system must be globally assembled and solved at every NR iteration, which could be computationally expensive even for efficient matrix solvers.

Such costs can be reduced using a traditional-domain decomposition method. As shown in Fig. 2(b), the whole solution region is divided into several sub-domains. A similar assembly happens in each sub-domain to generate smaller matrix systems. These sub-domain systems can be solved in parallel, and different schemes [5], [6] may be applied to exchange information and reach consistency (inter-element consistency) on sub-domain interfaces.


```

typedef struct
{
    int Type; //boundary Edge?
    int TotalCountNeiborElems;
    int NeiborElemsIDNum[20];
    int LocalNumInNeiborElem[20];
    //1,2,3,4,5,6
    float f; float b;
    //see Eq.(3),(8)
    float Ed; //eddy current term
    float A0; float A1;
    //edge values for iteration
} Edge[TotalCount_Edges];

typedef struct
{
    int EdgeID[6];
    //6 edges' global numbers
    int Type;
    //air or iron core?
    float K[6][6]; //Eq. (3)
    float D[6][6]; //D/dt Eq.(3)
    float DdotA[6];
    //eddy current buffer
    float A_2_Bxyz_term[3][6];
    //6 edges' Ai -> Bx By Bz
} Elem[TotalCount_Elements];

```

Fig. 5. Data arrangement of the EDD scheme in C language.

Figs. 5 and 6. Each element and edge consumes a fixed memory space. As a result, the total memory needed is linear to the problem size.

It is also worth mentioning that the proposed 3-D-EDD scheme is equivalent to Jacobi iteration under linear cases, which is similar to the N-scheme in [13]. However, the 3-D-EDD can easily integrate nonlinearity and matrix solution process without significantly affecting the convergence rate. Also, since the EDD method is originated from domain decomposition, replacement or modification of the sub-domain boundary condition may dramatically increase the convergence speed.

IV. COUPLING SCHEME FOR FIELD-CIRCUIT CO-SIMULATION

The abovementioned EDD scheme can provide electromagnetic field distribution based on the input coil current amplitude. However, coil currents in a power device (such as transformers) are always from a power system circuit, and a proper field-circuit coupling scheme is still necessary to interface with the external drive circuit.

There are two types of coupling schemes [2]: direct and indirect methods. The direct method simultaneously solves the entire FE and circuit system in one global matrix, which is intuitive and precise, and a symmetric matrix may be generated [14]. However, the method requires the FE matrix, and it may destroy the iteration convergence of the global matrix, which is not suitable for the matrix-free iterative EDD scheme. On the other hand, indirect methods allow separate solutions to the circuit and FE systems. However, some of the previous work cannot precisely handle strong eddy currents [1].

In this work, we propose an indirect coupling scheme by observing the time-discretized restrictions on coil current and voltage through traditional FE matrices. The scheme solves field and circuit equations separately and can produce accurate results under strong eddy currents.

The voltage of a 3-D coil results from the electric field at the direction of coil wire, which is also equivalent to time derivative of coil magnetic flux φ

$$V = \frac{N_{\text{coil}}}{S_{\text{coil}}} \cdot \int_{V_{\text{coil}}} \frac{\partial \vec{A}}{\partial t} \cdot \vec{n}_{\text{coil}} dv = \frac{\partial \varphi}{\partial t} \quad (14)$$

where N_{coil} is the number of coil windings, S_{coil} is the wire intersection area, and \vec{n}_{coil} is the unit vector of predefined coil wire direction. When the \vec{A} field is discretized by known

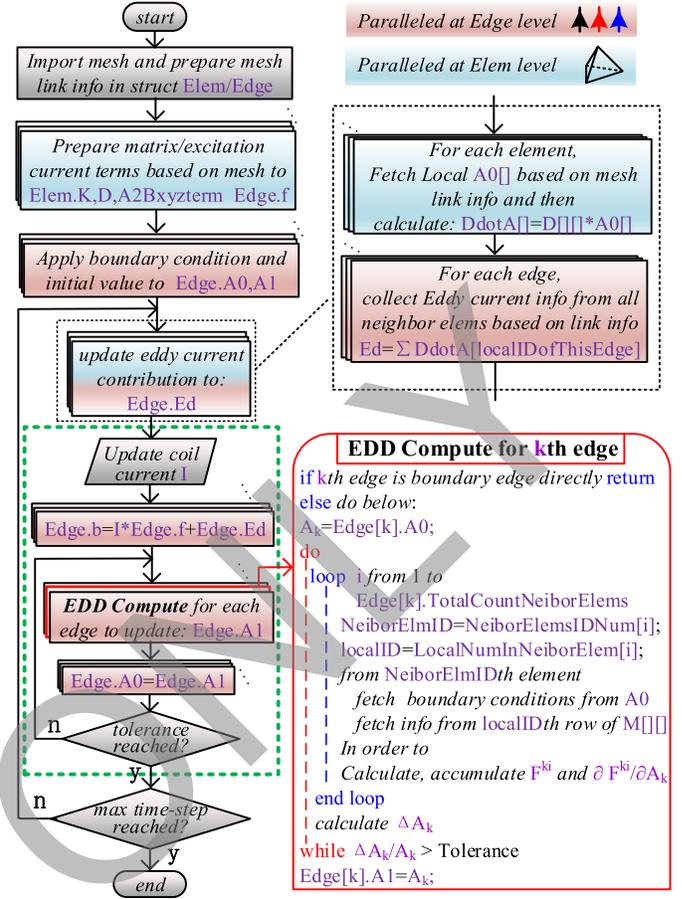


Fig. 6. Flowchart of the EDD scheme.

interpolation functions, (14) becomes

$$V = \sum_{k=1}^Q \frac{\partial A_k}{\partial t} \left(\int_{V_{\text{coil}}} \frac{N_{\text{coil}}}{S_{\text{coil}}} \cdot \vec{N}_k \cdot \vec{n}_{\text{coil}} dv \right) \quad (15)$$

where Q is the total number of edges, A_k is the field unknown on the k th global edge, and \vec{N}_k is interpolation functions associated with the k th edge. As a result, the voltage becomes a linear combination of unknowns on each edge noted as

$$V = \mathbf{A}_{-\varphi} \cdot \frac{\partial \mathbf{A}}{\partial t} \quad (16)$$

where \mathbf{A} is a column vector of all edge unknowns, and $\mathbf{A}_{-\varphi}$ is the row vector of the integration terms in (15), which can be found before FEM solution. Note that $\mathbf{A}_{-\varphi}$ also maps \mathbf{A} into coil flux φ .

Equation (16) and the FEM global matrix system establish the link between coil current and voltage, with \mathbf{A} as bridge. This relation is used to abstract the FEM model into a circuit component by several steps

$$\begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & -\mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{A} \\ \mathbf{V} \end{bmatrix} + \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{A}_{-\varphi} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \partial \mathbf{A} / \partial t \\ \mathbf{V} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} I(t) \quad (17)$$

where \mathbf{K} is the $N \times N$ global stiffness matrix, and \mathbf{f} is the $N \times 1$ global excitation column vector of the coil, assembled by elemental equations (2). When the backward Euler method

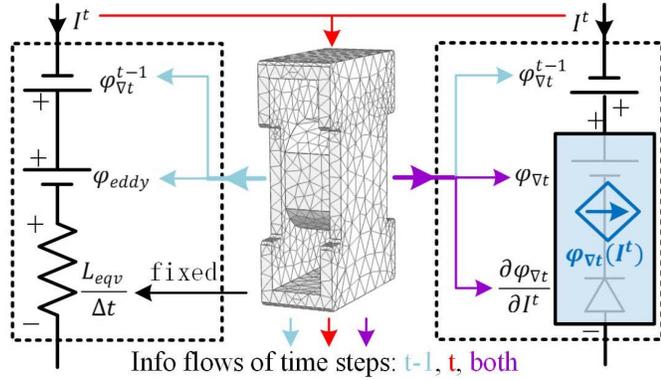


Fig. 7. FE circuit models for circuit coupling: left-hand side is for linear cases, and right-hand side is for nonlinear ones.

is used to discretize the time derivative, (17) becomes

$$\begin{bmatrix} \mathbf{K} + \frac{\mathbf{D}}{\Delta t} & 0 \\ \frac{\mathbf{A}_\varphi}{\Delta t} & -1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{A}^t \\ \mathbf{V}^t \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix} I^t + \begin{bmatrix} \frac{\mathbf{D}}{\Delta t} & 0 \\ \frac{\mathbf{A}_\varphi}{\Delta t} & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{A}^{t-1} \\ \mathbf{V}^{t-1} \end{bmatrix} \quad (18)$$

where the upper index t means the unknowns of current time-step, and $t-1$ means known from the previous time-step. After basic linear algebra operations, one can obtain

$$\mathbf{V}^t = \boldsymbol{\psi}(\mathbf{f} \times I^t + \mathbf{D}/\Delta t \cdot \mathbf{A}^{t-1}) - \varphi_{dt}^{t-1} \quad (19)$$

where $\varphi_{dt}^{t-1} = \mathbf{A}_\varphi/\Delta t \cdot \mathbf{A}^{t-1}$, and the row vector $\boldsymbol{\psi} = [\mathbf{A}_\varphi/\Delta t] \cdot [\mathbf{K} + \mathbf{D}/\Delta t]^{-1}$. Note that multiplying the row vector $\boldsymbol{\psi}$ with a column vector can be interpreted in another way: solve the FEM problem with the column vector as excitation and extract the coil flux term from the solution vector \mathbf{A}^t . This process is defined as $\boldsymbol{\psi}$ operation to a column vector. The equation explicitly describes the link between \mathbf{V}^t and I^t , which directly gives rise to the following circuit models.

For linear cases, the operator $\boldsymbol{\psi}$ is fixed. Therefore, (19) degenerates into a pure linear restriction

$$\mathbf{V}^t = L_{eqv}/\Delta t \times I^t + \varphi_{eddy} - \varphi_{dt}^{t-1} \quad (20)$$

where $L_{eqv} = \boldsymbol{\psi} \cdot \mathbf{f}$ and $\varphi_{eddy} = \boldsymbol{\psi} \cdot (\mathbf{D}/\Delta t) \cdot \mathbf{A}_k^{t-1}$. The FEM system becomes a fixed linear resistor and voltage sources (extracted from history FEM solution) in the circuit, as shown in Fig. 7 (left). For nonlinear cases, the operator $\boldsymbol{\psi}$ changes with its input. However, since \mathbf{A}^{t-1} is known from previous time-step, $\boldsymbol{\psi}$ becomes a nonlinear function (φ_{dt}) of I^t , and (19) degenerates into

$$\mathbf{V}^t = \varphi_{dt}(I^t) - \varphi_{dt}^{t-1}. \quad (21)$$

Thus, the FE system is regarded as a nonlinear current-controlled voltage source and a fixed voltage bias in the circuit shown in Fig. 7 (right).

With the abovementioned FE circuit model, the FE and circuit systems can be solved separately. From the circuit solver's perspective, the complex FEM solution process is abstracted away as different circuit components, and the FE information comes back to the circuit as voltages on those components.

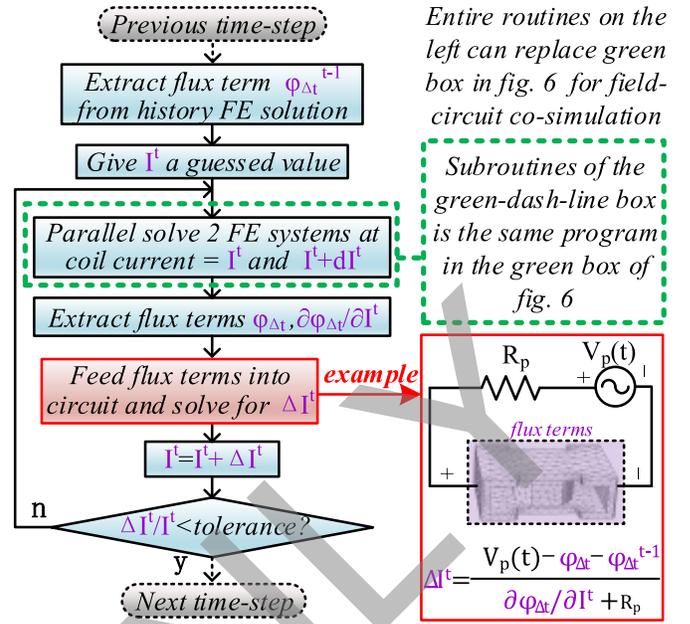


Fig. 8. Field-circuit coupling iteration flowchart at time-step t .

On the other hand, the FE solving process only sees current I^t from the circuit as input (operator $\boldsymbol{\psi}$ only sees I^t).

However, despite the convenience of isolation, the model still needs iterations (usually two to four times) to handle the nonlinearity. The purpose of the iterations is to find an I^t that leads to consistent voltages on both the FE circuit model and the circuit connected to it. If the NR scheme is applied, the circuit solver needs to know $(\partial \varphi_{dt}(I^t))/\partial I^t$ and $\varphi_{dt}(I^t)$ to calculate the increment of I^t . $\varphi_{dt}(I^t)$ can be obtained by directly solving the FEM problem at coil current I^t , and there are different methods to find $(\partial \varphi_{dt}(I^t))/\partial I^t$. For simplicity, the small probing increment method shown in (22) is used in this work

$$\frac{\partial \varphi_{dt}(I^t)}{\partial I^t} = \frac{\varphi_{dt}(I^t + dI^t) - \varphi_{dt}(I^t)}{dI^t}. \quad (22)$$

Note that the calculation of $\varphi_{dt}(I^t + dI^t)$ and $\varphi_{dt}(I^t)$ can be parallelized due to independence, and the iteration flowchart is shown in Fig. 8.

The iteration converges to the same result of the direct coupling method, regardless of the eddy current strength. The reason is that the proposed scheme simultaneously enforces the field, circuit, and the voltage/current consistency equations, and the same restrictions are explicitly assembled into the global matrix in the direct coupling methods. Also, note that although the scheme derives from matrices, it is still valid for matrix-free EDD FEM due to independent solutions of field and circuit.

V. CASE STUDIES

To demonstrate the efficiency and precision of the abovementioned methods, two case studies were carried out, and the results were compared with Comsol on the same mesh. The first one verified the nonlinear handling ability of the EDD scheme in a deep-saturated static scenario, and the second one combined field-circuit coupling and EDD in a

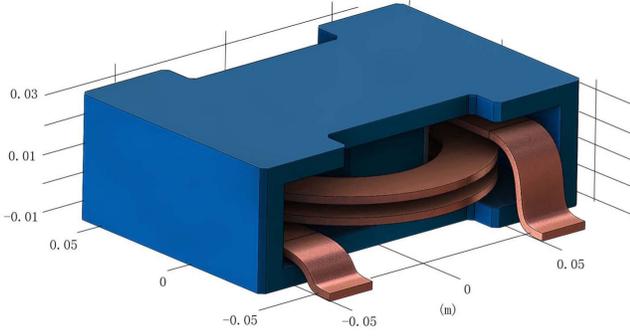
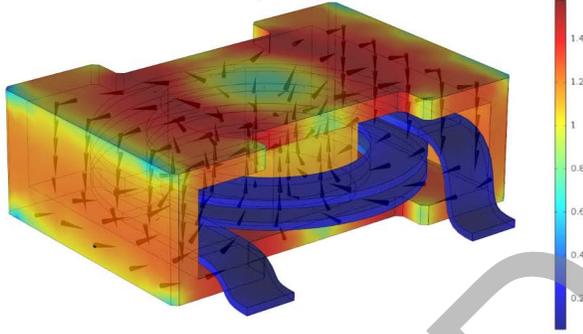


Fig. 9. Geometry of the power inductor in meters.

TABLE I
INDUCTOR PROBLEM DEFINITION

Coil/air permeability	μ_0	Coil current/winding turns	16A/380
Iron core B-H curve	$H = \begin{cases} \nu_0 B / 2e3, & B \leq 0.6 \\ \nu_0 B / 2e3 + 4e4(B - 0.6)^4 B / B, & B > 0.6 \end{cases}$		

Fig. 10. Saturated static magnetic flux density (T) of the power inductor.

transient simulation. The algorithm was developed in CUDA-C language, and the test was implemented on Intel Xeon E5-2698 v4 CPU (Comsol) and NVIDIA Tesla V100-PCIR-16-GB GPU (EDD scheme).

For the computation parameters, the termination condition is set to a global relative change of $1e-6$ and a global relative residual of $1e-5$, and all domains have the Dirichlet boundary condition for the tangential component of \vec{A}

$$\vec{A}_{\parallel} = 0. \quad (23)$$

A. Nonlinear Static EDD Simulation

Fig. 9 shows a power inductor with a blue iron core and a copper coil. Since it is studied in a static case, the D matrix in (3) was set to 0. The material and coil parameters are given in Table I.

After computation and post-processing, the field is displayed in Fig. 10, and the comparison with Comsol over different mesh size is shown in Table II. The result shows good accuracy with an average relative error of less than 2% over the space domain. Meanwhile, a significant speedup can be seen due to excellent parallelism and a fast convergence rate over DOFs. However, readers should notice that the speedup may vary for different $B-H$ curves and excitation amplitudes since the convergence rate will be affected by the system spectral radius.

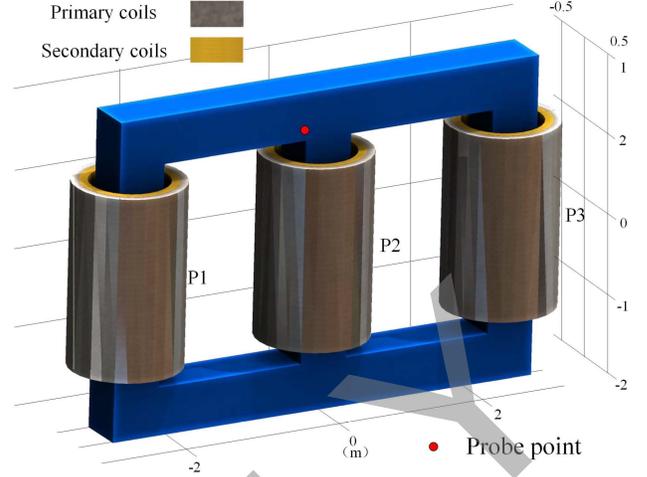


Fig. 11. Geometry of the three-phase transformer in meters.

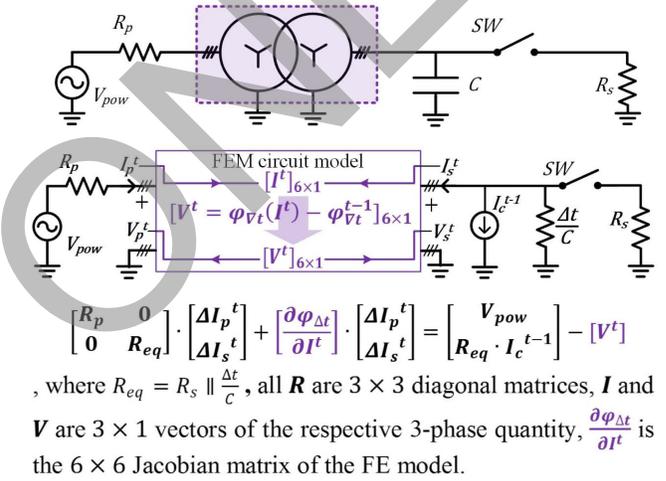
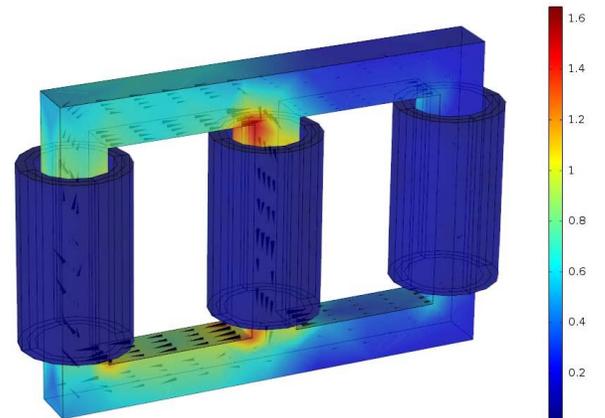


Fig. 12. Test circuit with the FE model and the equation to calculate current increment.

Fig. 13. Magnetic flux density (B) field of a three-phase transformer at 0.00733 s.

It is also worth mentioning that the static version of (2) does not have a full rank because it cannot specify the divergence of \vec{A} . However, the static case study still converges to the correct solution. This means that the iteration process can auto-gauge the problem like the conjugate gradient method [15].

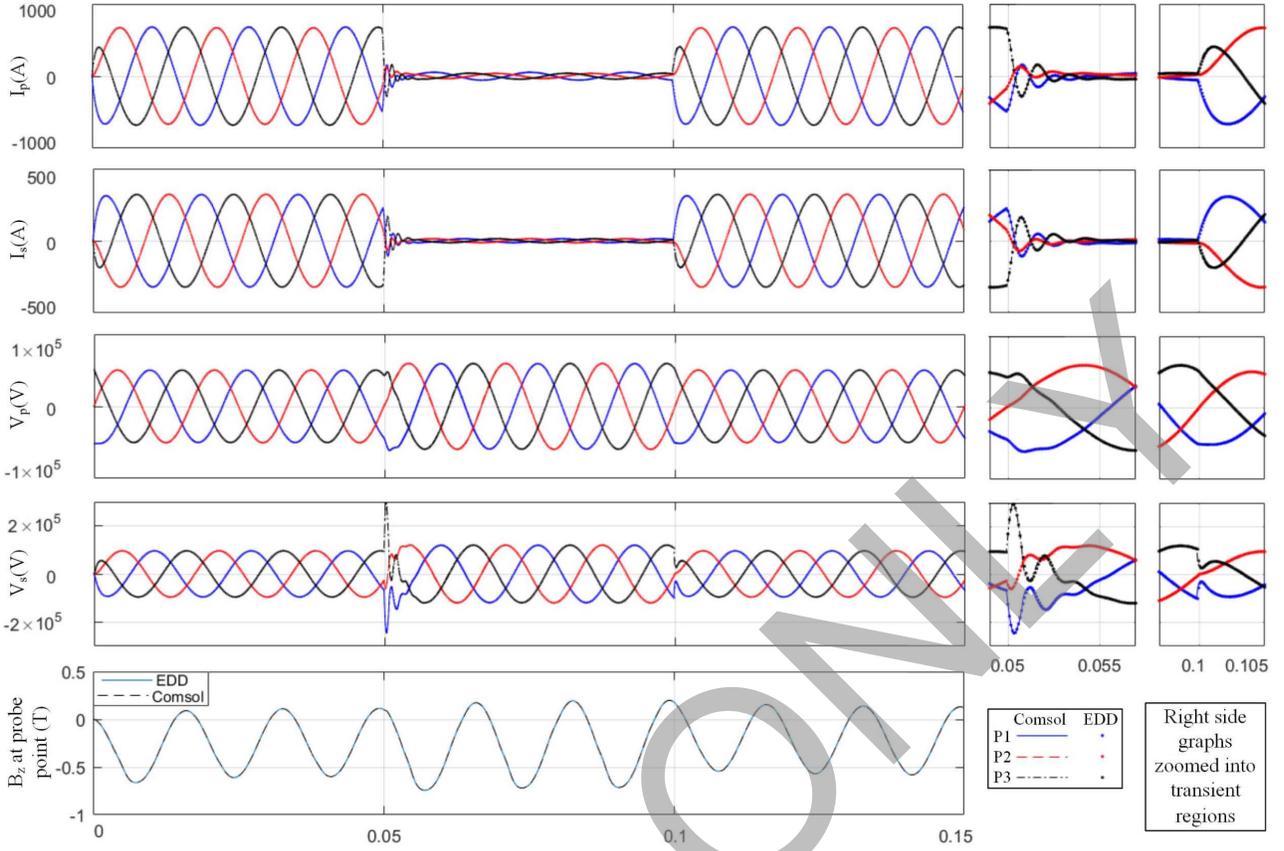


Fig. 14. Comparison of EDD scheme and Comsol results over time (s) for the three-phase transformer.

TABLE II
COMPARISON WITH COMSOL FOR THE STATIC CASE

Mesh size (DoF)	7870	24.3k	47.5k	79.8k	172k	523k
Time	<i>ComsolTM</i>					
use(s)	14.9	47.2	107.5	198.1	579.3	2274
	EDD					
	0.34	1.26	3.82	11.9	38.7	206.9
EDD iteration	5158	5529	7019	9039	13.8k	24.7k
Speed-up	43.7	37.5	28.2	16.6	14.9	10.99

B. Nonlinear Dynamic Field-Circuit Co-Simulation

As shown in Fig. 11, the studied transformer has three-phase metal-colored coils and an E-shape blue iron core. The transformer has six coils, while the abovementioned coupling scheme only describes the voltage–current relation of a single coil, which means that further extension should be made. Six coils, rather than a single one, all contribute to the total external excitation current in the FE domain. The influence of six-coil currents is packed into a total column FE excitation vector

$$\mathbf{f}_{\text{all}} = \sum_{i=1}^6 I_i^t \times \mathbf{f}_i \quad (24)$$

where I_i^t is the current inside the i th coil, and \mathbf{f}_i is the FE global excitation vector of the i th coil. \mathbf{f}_i is then fed into the FE solver to obtain the field distribution (\mathbf{A}), which generates voltages on all six coils

$$\mathbf{V}_{\text{coil}} = \mathbf{A}_{\text{coil}} \cdot \mathbf{A}. \quad (25)$$

The abovementioned process establishes a 6×6 current-to-voltage mapping between the coils. Thus, the transformer

TABLE III
TRANSFORMER PROBLEM DEFINITION

<i>Transformer parameters</i>					
Coil/air permeability	μ_0	Winding turns primary/secondary	750/1500	Core σ	2000S
Iron core B-H curve	$\mathbf{H} = \begin{cases} v_0 B / 1000, & B \leq 1.3 \\ v_0 B / 1000 + 1e5(B - 1.3)^3 \times B / B, & B > 1.3 \end{cases}$				
<i>Circuit parameters</i>					
$R_s = 15\Omega$	$R_p = 300\Omega$	$C = 300nF$			
V_{pow} are sinusoidal waves with peak amplitude 60kV at 60Hz					
<i>Simulation time set-up</i>					
Time-step length: 1/60/200s			Total time span: 0 to 0.15s		

FE system becomes a six-port nonlinear current (\mathbf{I}^t)-controlled voltage source (\mathbf{V}^t), and the Jacobian matrix of the nonlinear source is obtained by similar small probing increment separately superimposed on six coils

$$\frac{\partial \varphi_{\Delta t}}{\partial I^t_{ij}} = \frac{\varphi_{\Delta t}(\mathbf{I}^t)_i - \varphi_{\Delta t}(\mathbf{I}^t \cup dI^t_j)_i}{dI^t_j}, \quad i, j \in [1, 6] \quad (26)$$

where dI^t_j means small probing increment of current on coil j , and $\varphi_{\Delta t}(\mathbf{I}^t \cup dI^t_j)_i$ means time-discretized flux on coil i , generated from reference current \mathbf{I}^t superimposed with dI^t_j .

The extended FEM model is connected to the balanced three-phase circuit shown in Fig. 12. The circuit switch simulates an open-circuit fault at the secondary loads R_s . The fault starts at 0.05 s with a duration of 0.05 s.

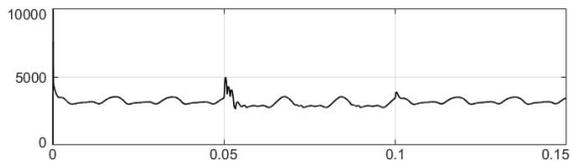


Fig. 15. Total EDD iterations versus time (s) with field-circuit interfacing. Note that the circuit-field iteration count is fixed at 5 for all time-steps. The figure gives the sum of all coil-parallel EDD iterations involved in five coil-field iterations.

Based on the model in Fig. 12 and the parameters in Table III, the simulation was carried out at a fixed mesh size of 15006 DoFs. The computation time consumed is 501s vs. 1198s on ComsolTM. The final result is shown in Figs. 13 and 14, and a comparison with ComsolTM shows a relative error of less than 2% over time and space domain. Also, the iterations needed for each time-step are plotted in Fig. 15.

VI. DISCUSSION

As shown in Table II, the speedup decreases with the mesh DOFs, which leads to the following question: will the EDD scheme becomes slower than Comsol for millions of DOFs? The current performance is hampered by total GPU cores of 5120. This means that many sub-domain solvers are still sequentially executed. However, if all sub-domains are solved in parallel, the time needed will be proportional to the number of EDD iterations, which is slower than the growth of mesh DOFs, and this results in an increasing speedup versus DOFs compared with Comsol (note that, in reality, there might be some implementation limit, such as communication delay between a large number of cores).

It is also seen that Comsol is much faster (such as 50 times) in the time domain compared with the static scenario. This is possibly because it utilizes a pre-factorize-and-back-substitution method to save time at the linear time-steps. The system matrix may be factorized only once, and only the light back-substitutions are carried out at the time-steps of the linear material $B-H$ region. Thus, the time consumption is not comparable with the EDD-circuit scheme since the method does both factorization and back-substitution at each time-step.

Future research will focus on expending the EDD scheme on clusters with multiple GPUs to exceed the 5000-core limit. Also, we will integrate the EDD scheme with the pre-factorize-and-back-substitution method to gain better speedup for the time-domain cases.

VII. CONCLUSION

In this article, a novel EDD method was proposed to calculate the 3-D field for nonlinear electromagnetic devices. The algorithm achieved massive parallelism and was implemented on GPU architectures.

For the first time, the idea of minimum sub-domain division is implemented with the time-domain 3-D nonlinear edge elements based on RMVP formulation. Benefitting from the extreme sub-domain size, the method simultaneously solves material nonlinearity and global FE system without having to assemble the global matrix. Also, the lightweight sub-domain

task and huge sub-domain number result in excellent modularity and massive parallelism, which made the EDD scheme a perfect choice for many-core GPU implementation.

In addition, for field-circuit co-simulation, a coupling scheme was developed to interface the EDD scheme with an external circuit. The scheme can abstract away complex FE models in controlled sources while maintaining a high precision at the same time, especially under high eddy current conditions. The efficiency and accuracy of the EDD-circuit method were discussed and verified through the GPU implementation. The comparison with Comsol indicates a significant speedup of 43.7 with an error of less than 2%.

ACKNOWLEDGMENT

This work was supported by the Natural Science and Engineering Research Council of Canada (NSERC).

REFERENCES

- [1] P. Liu and V. Dinavahi, "Real-time finite-element simulation of electromagnetic transients of transformer on FPGA," *IEEE Trans. Power Del.*, vol. 33, no. 4, pp. 1991–2001, Aug. 2018.
- [2] B. Asghari, V. Dinavahi, M. Rioual, J. A. Martinez, and R. Iravani, "Interfacing techniques for electromagnetic field and circuit simulation programs IEEE task force on interfacing techniques for simulation tools," *IEEE Trans. Power Del.*, vol. 24, no. 2, pp. 939–950, Apr. 2009.
- [3] Accessed: Jun. 2018. [Online]. Available: <https://images.nvidia.com/content/technologies/volta/pdf/tesla-volta-v100-datasheet-letter-fnl-web.pdf>
- [4] P. Liu, J. Li, and V. Dinavahi, "Matrix-free nonlinear finite-element solver using transmission-line modeling on GPU," *IEEE Trans. Magn.*, vol. 55, no. 7, pp. 1–5, Jul. 2019.
- [5] O. Widlund and M. Dryja, "Towards a unified theory domain decomposition algorithms for elliptic problems," Dept., Comput. Sci., Courant Inst., New York, NY, USA, Ultracomputer Note 167, Tech. Rep. 486, Dec. 1989.
- [6] C. T. Wolfe, U. Navsariwala, and S. D. Gedney, "A parallel finite-element tearing and interconnecting algorithm for solution of the vector wave equation with PML absorbing medium," *IEEE Trans. Antennas Propag.*, vol. 48, no. 2, pp. 278–284, Feb. 2000.
- [7] J. Li, P. Liu, and V. Dinavahi, "Massively parallel computation for 3-D nonlinear finite edge element problem with transmission line decoupling technique," *IEEE Trans. Magn.*, vol. 55, no. 10, pp. 1–8, Oct. 2019.
- [8] I. Kiss, S. Gyimothy, Z. Badics, and J. Pavo, "Parallel realization of the element-by-element FEM technique by CUDA," *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 507–510, Feb. 2012.
- [9] P. Liu and V. Dinavahi, "Matrix-free nodal domain decomposition with relaxation for massively parallel finite-element computation of EM apparatus," *IEEE Trans. Magn.*, vol. 54, no. 9, pp. 1–7, Sep. 2018.
- [10] J.-M. Jin, *The Finite Element Method in Electromagnetics*. Hoboken, NJ, USA: Wiley, 2015.
- [11] M. L. Barton and Z. J. Cendes, "New vector finite elements for three-dimensional magnetic field computation," *J. Appl. Phys.*, vol. 61, no. 8, pp. 3919–3921, Apr. 1987.
- [12] R. Albanese and G. Rubinacci, "Finite element methods for the solution of 3D eddy current problems," in *Advances in Imaging and Electron Physics*, vol. 102. Amsterdam, The Netherlands: Elsevier, 1997, pp. 1–86.
- [13] J. P. A. Bastos and N. Sadowski, "A new method to solve 3-D magnetodynamic problems without assembling an $Ax = b$ system," *IEEE Trans. Magn.*, vol. 46, no. 8, pp. 3365–3368, Aug. 2010.
- [14] R. Escarela-Perez, E. Melgoza, and J. Alvarez-Ramirez, "Coupling circuit systems and finite element models: A 2-D time-harmonic modified nodal analysis framework," *IEEE Trans. Magn.*, vol. 45, no. 2, pp. 707–715, Feb. 2009.
- [15] N. T. K. Fujiwara and T. Nakata, "Advanced computational and design techniques in applied electromagnetic systems," Dept. Comput. Sci., Courant Inst., New York, NY, USA, Ultracomputer Note 167, Tech. Rep. 486, Jun. 1995.