IEEE Open Journal of
Industry Applications

Received 4 December 2020; revised 10 April 2021; accepted 19 June 2021. Date of publication 22 June 2021;
date of current version 8 July 2021. The review of this article was arranged by Editor Emmanuel B. Agamloh.

Digital Object Identifier 10.1109/OJIA.2021.3091557

Space-Time-Parallel 3-D Finite Element Transformer Model With Adaptive TLM and Parareal Techniques for Electromagnetic Transient Analysis

JIACONG LI (Student Member, IEEE), PENG LIU (Student Member, IEEE), AND VENKATA DINAVAHI (Fellow, IEEE)

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

CORRESPONDING AUTHOR: PENG LIU (e-mail: pliu3@ualberta.ca)

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

ABSTRACT In this paper, a novel massively space-time parallel finite element (FE) transformer model is proposed for power system electromagnetic transient simulations. The method utilizes the reduced magnetic vector potential (RMVP) formulation to solve the electromagnetic field in the transformer, and the 3-D tetrahedral edge element is applied to discretize the RMVP formulation. The discretized nonlinear system is solved by adaptive transmission line modeling (TLM) and matrix-free conjugate gradient iterator, which allows parallelism at the elemental-level and eliminates the need for Newton-Raphson (NR) iterations over the large-scale global matrix. The paper also integrates the parallel-in-time method with the finite element solver to further parallelize the model at space-time level, and the GPU implementation of the model realized a speedup of over 55 and high accuracy, compared with a commercial FE software.

INDEX TERMS Domain decomposition, eddy current field, field-circuit coupling, finite element method, graphics processors, nonlinear behavior, parallel processing, parareal, reduced magnetic vector potential, 3-D edge element.

I. INTRODUCTION

Transformers are one of the most utilized components in energy and power systems, due to their ability to change voltages and isolate currents between different coils. The electromagnetic field inside a transformer establishes the connections between these voltages, currents, and the geometry of the transformer materials. Such relations are subject to Maxwell's equations. How to accurately and efficiently model the relation is an important topic in electromagnetic transient (EMT) simulations of power systems.

Traditionally, EMT simulation simplifies the field coupling between each coil into lumped network representation. These lumped network models can be classified into the admittance matrix model [1] and the magnetic equivalent circuit model [2], [3], which have already been widely applied in EMT simulations for power system transformers with multiple coils [4], [5]. However, since some information is lost during the field-to-lumped-network simplification process, these models have limited application for high-accuracy wide-band EMT simulation.

Finite element (FE) models [6], [7], in contrast, are known for their superior precisions and ability to handle the EM field for complex geometry. However, due to material nonlinearity, traditional FE solvers require repetitive assembly and factorization of the large-scale global Jacobian matrix during each NR iteration step, which can dramatically slow down the simulation. On the other hand, during the recent decade, parallel computation has witnessed a dramatic increase of heavy-weight processors.

Considering the high accuracy of EM field models, researchers have been developing massively-parallelized algorithms to boost finite element computation with the help of parallel compute devices. Recently, a highly-efficient 2-D finite element transformer model was introduced [8]. The

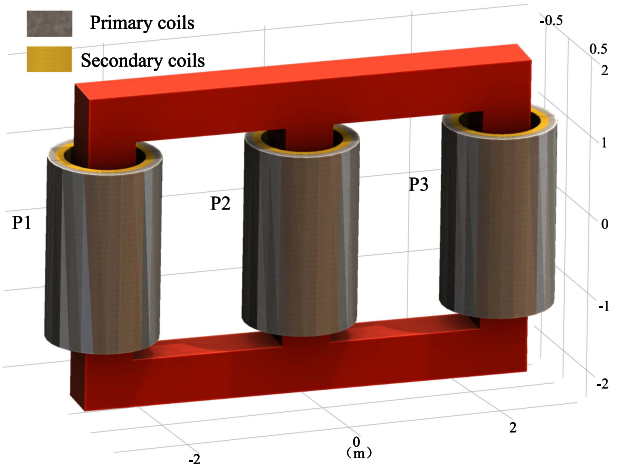

FIGURE 1. Geometry of the three-phase transformer in meters.

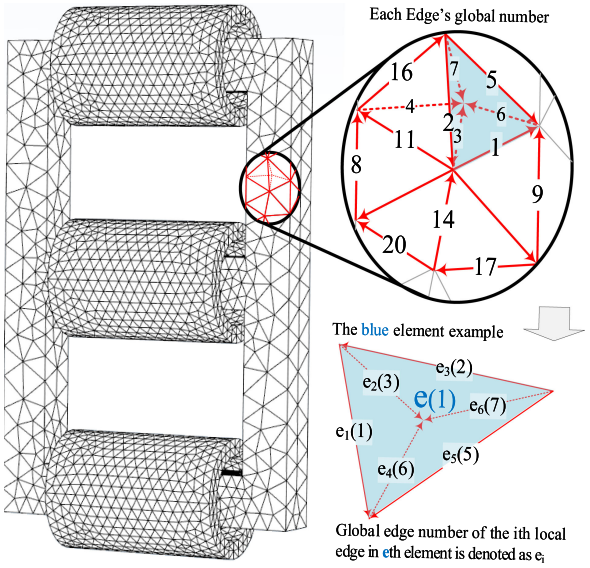

FIGURE 2. Solution domain divided into many small tetrahedral elements and the local and global number notations.

model utilized the transmission line modeling (TLM) technique to isolate local nonlinearity of the transformer iron core from the global matrix. Such isolation lead to massive parallelism and avoided global NR iterations; thus, the transformer model was able to run in real-time on the field programmable gate array (FPGA). A later work introduced a matrix-free solver to reduce the number of TLM iterations needed [9].

Despite the excellent parallelism, the above transformer model is only based on 2-D nodal finite elements. However, in high precision simulations, the geometry of a power transformer is always modeled in 3-D, and nodal elements, compared with edge elements, cause a major decrease of precision near sharp corners of the geometry [15]. Also, the degrees of freedom (DOFs) in a 2-D FE transformer model are usually lower than in the 3-D counterpart, and in some cases, a 2-D FE model cannot even fully utilize the hardware resources on a single GPU [9]. Therefore, based on the above facts, it has become necessary to explore accurate algorithms to integrate the adaptive TLM finite element method with a 3-D edge FE transformer model and to further expand the scope of parallelism.

In this paper, improvements were made on our former efforts on the matrix-free adaptive TLM decoupling technique so that the method can be applied in both static and transient scenarios, and the iterations required are further reduced. The improved adaptive TLM method is extended to the 3-D FE transformer model (Fig. 2) with space-level parallelism. In addition, the Parareal algorithm [10] is, for the first time, integrated with 3-D edge finite elements to further parallelize the model in the time dimension. The space-time parallelized adaptive TLM transformer model is then implemented and verified on the Tesla V100 GPU, and a comparison with commercial FE software *Comsol*^{TM} is conducted. Note that [11]–[13] describe a general semi-explicit nonlinear eddy-current solver that could be applied to the suggested transformer problem. Furthermore [14] describes an alternate way to avoid solving large 3-D FEM problems by using a detailed electric circuit similar to the proposed TLM solver.

The paper is arranged as follows. In Section II, the reduced magnetic vector potential formulation is introduced to calculate the transformer's electromagnetic field. The formulation is then discretized by edge elements to form a solvable algebraic equation group. Section III proposes the matrix-free adaptive TLM method to solve the equation group based on the conjugate gradient method, which is integrated with an indirect coupler for field-circuit co-simulation in Section IV. Section V combines the co-simulator with Parareal technique to achieve time-space parallelism. The transformer-circuit co-simulation model is then verified in the case studies of Section VI. Finally, Section VII gives the conclusion of the work.

II. FEM FORMULATION FOR EDDY CURRENT ANALYSIS

A. REDUCED MAGNETIC POTENTIAL FORMULATION

The electromagnetic field inside a 3-D transformer is excited by the current density inside the transformer coils, as is described by the Maxwell's equations. To simplify the equations for power system application, one usually omits dielectric (high frequency) related terms and introduce potential functions to reduce jump of solution at different material interfaces [16]. This leads to the reduced magnetic vector potential (RMVP or $\vec{A}$) formulation [16]:

$$\nabla \times (\upsilon \nabla \times \vec{A}) + \sigma \frac{\partial \vec{A}}{\partial t} - \vec{J_e} = 0, \tag{1}$$

where υ is the field-dependent reluctivity, σ is the electrical conductivity, and $\vec{J_e}$ is the impressed external current density.

B. FINITE ELEMENTS AND DISCRETIZED FORMULATION

The transformer electromagnetic field can be restored from the vector potential $\vec{A}$ after the solution of (1). However, (1) is in a continuous form; in order to obtain the numerical solution, (1) still needs to be discretized.

To achieve this, the finite element approach splits the entire solution domain into many smaller tetrahedral elements and edges between the tetrahedrons (Fig. 2). Each edge has its own direction and one DOF (A_i related to edge number i) that is equal to the projection of $\vec{A}$ onto its direction. And within each element, $\vec{A}$ field is represented through a linear combination of known interpolation functions (or pattern fields). The interpolation functions ($\vec{N_i}$) are fully determined by the position of the 4 known elemental vertexes, and $\vec{N_i}$'s projection is 1 on edge i's direction while 0 on all other edges. This means that the 6 involved linear combination terms $(A_{e1}, A_{e2}, A_{e3}] \ldots)$ of each edge is exactly the DOF on the respective edge. To find the value of these DOFs (namely, the solution of FEM problem), one can construct local restrictions between the 6 involved edge DOFs for every single element, and add these local equations to a global matrix system. The matrix system can then be solved for all edge DOFs.

The Galerkin weighted residual method is applied to obtain elemental restrictions. The residual can be obtained from the left-hand-side of (1). The 6-by-6 matrix elemental discretized restriction is then obtained by integrating the product of the residual with 6 weighting functions (chosen to be $\vec{N_i}$) and forcing the integrals to zero:

$$\begin{aligned}\left[K_{e_ie_j}\right]_{6\times6}\cdot\left[A_{e_j}\right]_{6\times1}+\left[D_{e_ie_j}\right]_{6\times6}\cdot\frac{\partial}{\partial t}\left[A_{e_j}\right]_{6\times1}\\=\left[f_{e_ik}\right]_{6\times6}\cdot\left[I_k\right]_{6\times1}+\left[\tau_{e_i}\right]_{6\times1},\end{aligned}\tag{2}$$

where

$$K_{e_ie_j}=\int_{Ve}\nu_e\left(\nabla\times\overrightarrow{N_i}^{(e)}\right)\cdot\left(\nabla\times\overrightarrow{N_j}^{(e)}\right)\mathrm{dV},\, i,\mathrm{j}\in[1,6]\,,\tag{3}$$

$$D_{e_ie_j}=\int_{Ve}\sigma_e\overrightarrow{N_i}^{(e)}\cdot\overrightarrow{N_j}^{(e)}\mathrm{dV},\,\mathrm{i},\mathrm{j}\in[1,6]\,,\tag{4}$$

$$f_{e_ik}=\int_{Ve}\overrightarrow{N_i}^{(e)}\cdot\overrightarrow{J_{kunit}}dV,\,\mathrm{i},\mathrm{k}\in[1,6]\,,\tag{5}$$

$$\tau_{e_i}=\int_{Se}\left(\nu\nabla\times\vec{A}\right)\times\overrightarrow{ds}\cdot\overrightarrow{N_i}^{(e)},\,\mathrm{i}\in[1,6]\,,\tag{6}$$

and V_e, S_e are the volume and the surface boundary of eth element, $\overrightarrow{J_{kunit}}$ is the current density distribution at 1 A current in kth coil of the transformer, and I_k is known current in 6 coils that change with time. Through the Backward Euler numerical integration, the time derivative in (2) can also be discretized:

$$\left[M_{e_ie_j}\right]_{6\times6}\cdot\left[{A_{e_j}}^t\right]_{6\times1}=\left[b_{e_i}\right]_{6\times1}+\left[\tau_{e_i}\right]_{6\times1},\tag{7}$$

where

$$\left[b_{e_i}\right]_{6\times1}=\left[f_{e_i}\right]_{6\times1}\cdot\left[I_k\right]_{6\times1}+\frac{\left[D_{e_ie_j}\right]_{6\times6}}{\Delta t}\cdot\left[{A_{e_j}}^{t-1}\right]_{6\times1},\tag{8}$$

$$\left[M_{e_ie_j}\right]_{6\times6}=\left[K_{e_ie_j}\right]_{6\times6}+\left[D_{e_ie_j}\right]_{6\times6}/\Delta t,\tag{9}$$

and ${A_{e_j}}^t$ is the unknowns to solve at current time step, and ${A_{e_j}}^{t-1}$ is the knownA_{e_j} at the previous time step. I_k is the changing coil current that is given by problem definition, and $M_{e_ie_j}$, $K_{e_ie_j}$, $D_{e_ie_j}$, f_{e_i} are noted as M_e, K_e, D_e, f_e for simplicity.

To obtain a consistent solution over the entire domain for all elements, the local restrictions of 6 edge DOFs in (7) related to each element need to be assembled into a global system. During this process, all elements interact with each other at surfaces between different elements with respect to Maxwell's equations [17]:

1) Continuity of tangential $\vec{A}$ is automatically ensured by the edge-based DOF discretization format. As a result, $\vec{B}$ is continuous between elements.

2) On a triangular surface between 2 elements, the 2 τ terms in the 2 elements are opposite numbers for the same edge, which leads to the continuity of tangential $\vec{H}$. For ith edge, this relation can be expressed as:

$$\sum_e\tau_{e_k}=0|e_k=i,\;\;k\in[1,6]\,,\tag{10}$$

where e represents all elements sharing edge i.

During assembly, local restrictions (7) for all elements are directly added together to make the global matrix equation (11) for all global edge DOFs, and the surface integration term is eliminated in the global equation. Thus (7) and (10) are simultaneously satisfied:

$$M\cdot A^t=\left(K+\frac{D}{dt}\right)\cdot A^t=b,\tag{11}$$

where $K=\sum_{e=1}^{E}K_e$, $D=\sum_{e=1}^{E}D_e$, $b=f\cdot{I_k}^t+\frac{D}{dt}\cdot A^{t-1}$, $f=\sum_{e=1}^{E}f_e$, and E is the total number of elements.

III. SPACE PARALLELISM: THE REFINED NONLINEAR FEM SOLVER

Since the transformer iron core material has a nonlinear B-H curve, the ν_e in (2) is highly dependent on the 6 edge DOFs, which result in nonlinearity in the global system (11). Traditionally, the global NR iteration is utilized to deal with the nonlinearity. This requires the frequent updating of the local Jacobian matrix and assembling of the global sparse matrix (11) at every NR iteration, which could be both time and memory consuming [9].

In contrast, the proposed solver can avoid assembly with proper handling of the nonlinearity. The solver finds A^t with a given b in (11), and the solution is achieved by the transmission line technique (subsection B and C) which iterates towards a nonlinear solution by solving many linear subproblems using the technique introduced in subsection A.

A. LINEAR MATRIX-FREE PRECONDITIONED CONJUGATE GRADIENT SOLVER WITH PARALLELISM AT ELEMENT LEVEL

Through observation of the assembly process, one can find that the global matrix is purely the summation of the elemental matrix. Thus, multiplying a vector with the global matrix is equivalent to the following element-by-element (EbE) process.

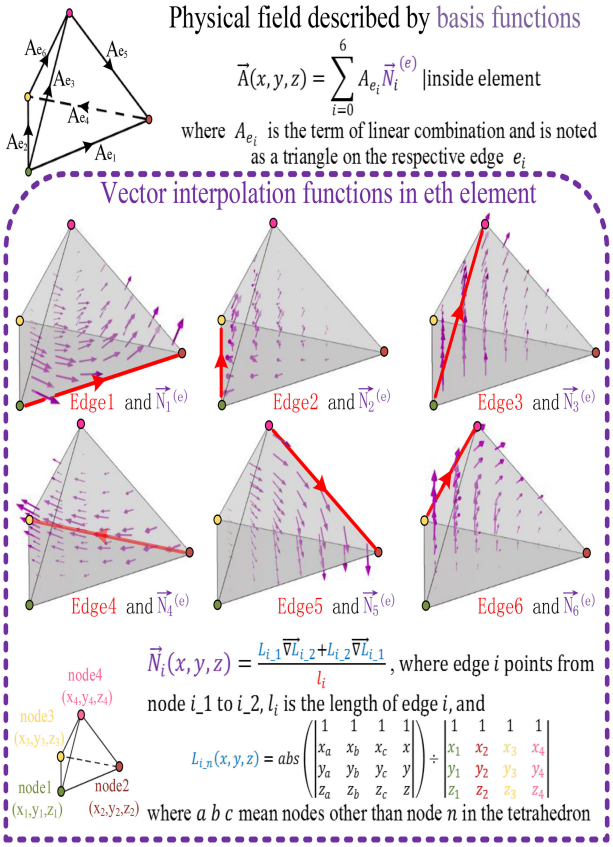

FIGURE 3. Vector edge interpolation functions. Note: upper/lower index e represents variables associated with e^{th} element.

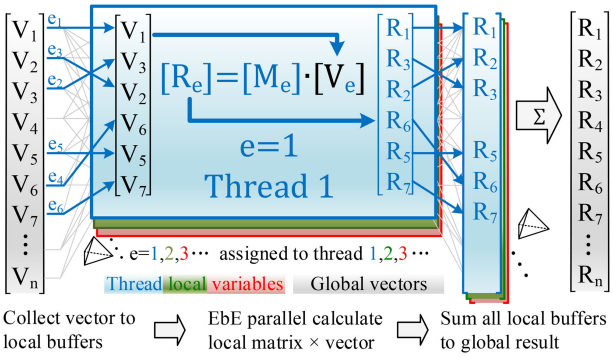

FIGURE 4. Element-by-element (EbE) technique to calculate $R = M \cdot V$ (M is in (11)).

As is shown in Fig. 4, each element is assigned a compute thread and each elemental thread collects the values associated with its 6 edges from the global vector (position $e_1 \sim e_6$) into its local buffer, and then generates its local result by multiplying the local buffer with its elemental matrix. Finally, all elements add their local result to the respective position of their edges to form the global result vector. Thus, the multiplication is finished without assembling a global matrix.

The EbE process seems redundant at the first look. However, due to light-weight elemental computation and suitability for massively paralleled architectures, this algorithm

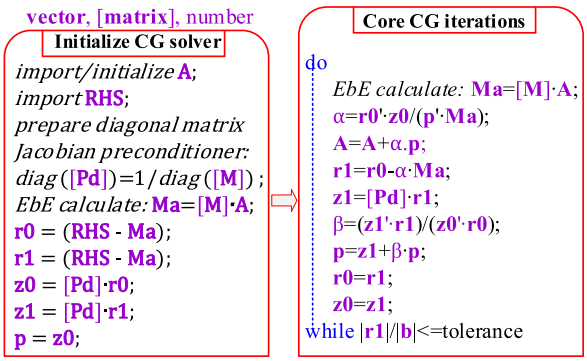

FIGURE 5. Preconditioned conjugate gradient flowchart to solve matrix system: $\mathbf{A} = \mathbf{M}^{-1} \cdot \mathbf{RHS}$ (M explained in (11)) at a linear operating point.

is actually much faster (over 50 speed-up) than the traditional centralized global matrix-vector multiplication [18], especially for nonlinear cases since the multiplication speed is not affected by the change of elemental ν_e and matrix.

With a decent speed-up, the EbE technique could boost many matrix-vector-multiplication requiring FEM iterators. For simplicity, the Jacobian preconditioned conjugate gradient method is applied in this work, and the flow chart is shown in Fig. 5. As can be seen the sub-procedures only involve EbE multiplication and vector reduction and dot product. Thus the algorithm can be massively parallelized on GPU architectures. Note that parallelization of conjugate gradient type solvers on GPUs has been widespread in the literature [19]–[22].

B. ISOLATING THE NONLINEARITY WITH THE ADAPTIVE TRANSMISSION LINE DECOUPLING TECHNIQUE

The transmission line (TL) decoupling method [23] can be explained by the following phenomenon shown in Fig. 6(a). When one places a transmission line between a current source (with internal resistance) and a diode, the voltage/current at the diode or source terminal will fluctuate and stabilize at the same value as if the source and diode are directly connected. During this process, the diode or the source can only see the TL characteristic impedance and traveling EM waves at the TL terminals. Therefore, the diode and source cannot directly 'see' each other (decoupled) but they can reach the same 'directly connected' state after EM waves reflect/superimpose to reach a steady-state.

To illustrate the how the equilibrium is achieved, the following observation is needed with reference of the transmission line impedance Z_c: once the incident wave and reflection wave are known, the current and voltage (or the V-I operating point) on the TL can be determined, and vice versa, according to the control equation:

$$\mathrm{I}_{line} = \frac{V_{inc} - V_{ref}}{Z_c}, V_{line} = V_{inc} + V_{ref}, \qquad (12)$$

where I_{line}, V_{line} are the current and voltage on the TL, and V_{inc}, V_{ref} are the incident and reflection waves on the TL.

During the scattering phase, the V_{inc} on the TL is considered fixed, and the V_{ref} needs to be found at the diode end. Due

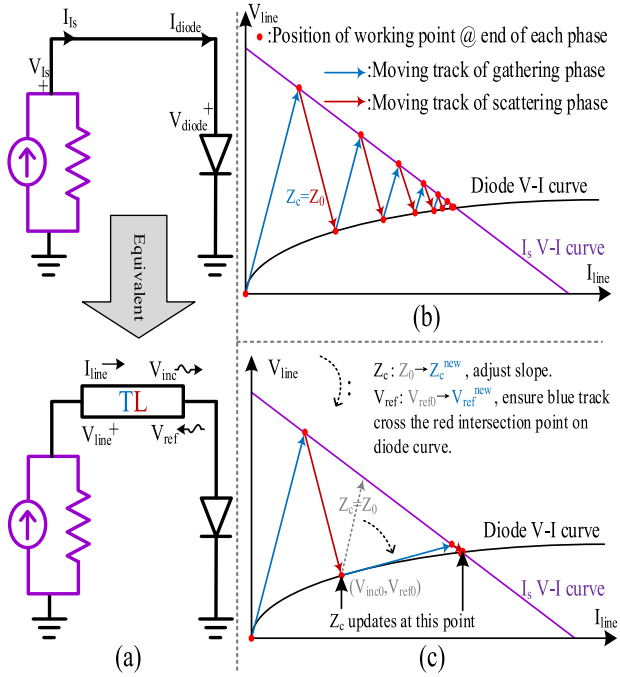

FIGURE 6. (a) Inserting a transmission line does not affect a circuit's static-state V and I. (b) Traditional transmission line iterations in V-I graph. (c) Adaptive transmission line iterations in $V-I$ graph.

to the restriction (12), the V-I operating point has to move from the previous position at a slope of $-Z_c$. Also, since the wave is reflected by the diode, the operating point must move forward to intersect with the diode's V-I curve, which requires the solution of nonlinear equations.

At the gathering phase, the V_{ref} on the TL is considered unchanged while the V_{inc} needs to be solved at the source end, and similarly, the operating point has to move from the previous position towards the voltage source's IV curve at a slope of Z_c. This process only requires solution of a linear equation, since the operating point's moving track is equivalent to a Norton source. Therefore, the source end only sees the Norton source, and nonlinearity is abstracted away:

$$G_{norton} = 1/Z_c \tag{13}$$

$$I_{norton} = 2V_{ref}/Z_c \tag{14}$$

With injection and reflection waves respectively restoring information from the last operating point, the scattering and gathering alternately repeat to move the operating point and converge to the final solution.

However, for the above conventional method, the convergence can be very slow depending on the selection of Z_c. In contrast, the adaptive TL decoupling method can converge much faster by adaptively changing the Z_c according to the diode curve, as is shown in Fig 6. At the end of each scattering phase, the moving track rotates around the red intersection point so that it becomes tangent with the diode curve (the gray track rotates into the blue track). To achieve this, Z_c and V_{ref}

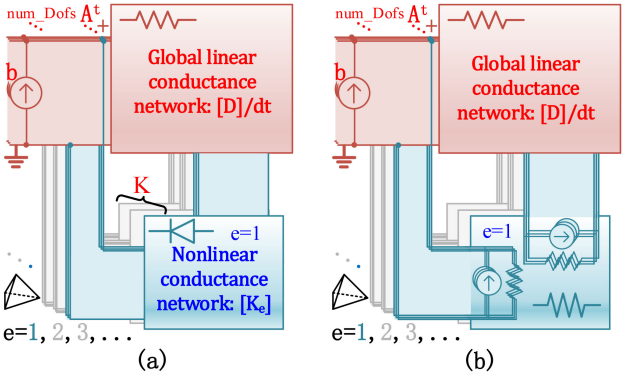

FIGURE 7. (a) A^t on each edge is mapped to node voltage on the global conductance network that represents K, D, and b in (11). (b) TL convert nonlinearity of K_e in (9) into Norton sources to form a global linear system with D.

need the following updates:

$$Z_c = \partial I_{diode}/\partial V_{diode}, \tag{15}$$

$$V_{ref} = Z_c/Z_{c0}\left(V_{ref0} - V_{inc0}\right) + V_{ref0} + V_{inc0}, \tag{16}$$

where the Z_{c0} is the characteristic impedance before rotation. V_{inc0} and V_{ref0} are the coordinate of the intersection point with reference to the previous TL impedance Z_{c0}. In this way, the TL mismatch is reduced by dynamically adjusting the TL characteristic impedance. Thus the adaptive technique needs fewer iterations.

C. APPLYING THE ADAPTIVE TRANSMISSION LINE DECOUPLING TECHNIQUE TO THE FINITE ELEMENT EQUATIONS

Through basic circuit analysis principles [24], the FEM equation (11) can be converted into an equivalent circuit network (Fig. 7(a)), where the nonlinear admittances network (K or K_e) connects in parallel with the linear current sourced network (D/dt and b). Thus, the above circuit decoupling method between the diode and resistive current source can also be extended between the nonlinear and linear FEM networks. As is shown in Fig. 7(b), due to the TL decoupling of the global conductance (D) and elemental conductance (K_e), each element's local nonlinearity is isolated and converted into linear Norton equivalent sources connected in parallel with the global conductance to form a global linear network, which can be solved by the linear solver proposed in Section II(A). The detailed elemental nonlinear Norton source model and FEM TL iteration process are discussed below.

Fig. 8(a) shows the structure of the elemental equivalent network of the local K_e matrix in (9) and Fig. 7(a). The voltage on the circuit red nodes are equivalent to the value of DOFs associated with the 6 edge DOFs in the element, and the blue and yellow lines represent respectively the self-ground conductance and mutual conductance on the 6 circuit nodes, which are defined by the following equations:

$$G_{ij} = -K_{e_i e_j}, \tag{17}$$

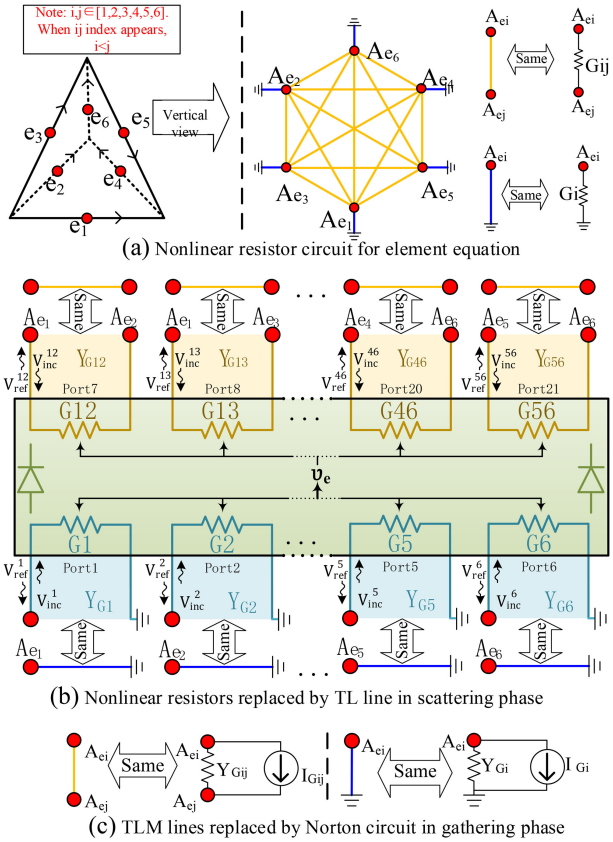

FIGURE 8. (a) Equivalent circuit of the elemental nonlinear network K_e in Fig. 7(a). (b) Nonlinear resistors replaced by coupled nonlinear elemental model (green box) and transmission lines. (c) TL line model becomes linear Norton source at gathering phase.

$$G_i = \sum_{j=1}^{6} -K_{e_i e_j}. \tag{18}$$

where G_i is the self-ground of edge i, and G_{ij} is the mutual conductance between edge i and j.

Each of these nonlinear conductances depends on local A_{e_i} and can affect the solution of the whole FEM system. To isolate theses local nonlinearity and reach a consistent solution on the FEM system, the transmission line is placed between the red circuit nodes and each conductance (Fig. 8(a)). The information exchanges between the 2 ends of each transmission line during the following scattering and gathering phase.

Fig. 8(b) shows the FEM transmission line scattering phase. Inside each element, all incident waves and transmission line characteristic conductance are fixed and the reflection waves need to be found at the 21 nonlinear conductance ports. Because the voltage on each of the 6 blue conductance can affect the permeability inside the whole element, the 21 conductances are coupled together (green box) and the following small scale nonlinear equation group/process is designed to solve for the reflection waves.

The blue conductance's nonlinear equation group considering local B-H relation and TL control equation (12) is solved first:

$$Gi\left(V^i_{inj0} + V^i_{ref0}\right) - Y_{Gi}\left(V^i_{inj0} - V^i_{ref0}\right) = 0, \tag{19}$$

where Y_{Gi} is the transmission line characteristic conductance fixed in this equation. Since Gi is dependent on V^i_{ref0}, the solution of (19) requires a 6-by-6 NR iteration with the following restrictions to calculate the Jacobian matrix:

$$\frac{\partial Gi}{\partial V^j_{ref0}} = \frac{\partial \mathrm{Gi}}{\partial \nu_e} \cdot \frac{\partial \nu_e}{\partial B^2} \cdot \frac{\partial B^2}{\partial V^j_{ref0}}, \tag{20}$$

$$\vec{B} = \nabla \times \vec{A} = \sum_{i=1}^{6} \left(V^i_{inc0} + V^i_{ref0}\right) \cdot \nabla \times \vec{N_i}. \tag{21}$$

Once the V^j_{ref0} associated with the blue/self-ground conductances are found, the rest yellow conductance's reflection waves can be determined [25]:

$$V^{ij}_{ref0} = V^{ij}_{inj0} \frac{Gij - Y_{Gij}}{Gij + Y_{Gij}}. \tag{22}$$

At the end of the scattering phase, all reflection waves are found and the element ν_e moves from ν_{e0} to a new value ${\nu_e}^{new}$. Similar to the previous diode case, each element updates its 21 transmission line conductance and the reflection waves in order to reduce the total iteration numbers needed.

$$V_{ref} = k\left(V_{ref0} - V_{inc0}\right) + V_{ref0} + V_{inc0}, \tag{23}$$

$$Y_{Gi} = kY_{Gi0}, Y_{Gij} = kY_{Gij0}, \tag{24}$$

where $\mathrm{k} = {\nu_e}^{new}/\nu_{e0}$, and Y_{Gi0}, Y_{Gij0} are the characteristic impedance velues beofre the adptive update.

During the gathering phase, all reflection waves are fixed and the incident waves need to be updated from the global network. The reflection waves and transmission lines becomes equivalent Norton source as is shown in Fig. 8(c). These sources carry nonlinear information from each element to form a global matrix with the D/dt network in Fig. 7(a). After the solution of the global network, the ground-voltages on all edges (A_{e_i}) are found, and thus the voltage on each Norton source can be calculated to extract the incident waves:

$$V_{inc0} = V_{Norton} - V_{ref}. \tag{25}$$

The scattering and gathering alternately update the reflection and incident waves on all transmission lines until the system converges to the final solution, with the adaptive update of transmission line impedance to boost the process. The flow chart of the adaptive TLM iteration is shown in Fig. 9. With suitable parallelism achieved at element/edge-level, the entire process does not involve any global matrix.

IV. FLUX EXTRACTION TECHNIQUE TO COUPLE FINITE ELEMENTS WITH EXTERNAL DRIVING CIRCUIT

The above discussion only involves finite element computation of the three-phase transformer electromagnetic field from 6 given coil current amplitudes. However, in reality, a transformer is always connected to a driven circuit and a coupling

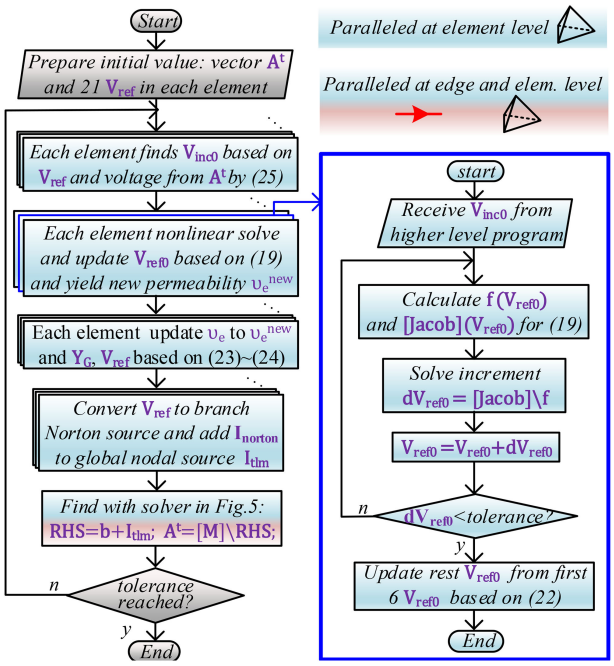

FIGURE 9. Flow chart of EbE-based adaptive-TLM finite element method to solve the nonlinear system in (11).

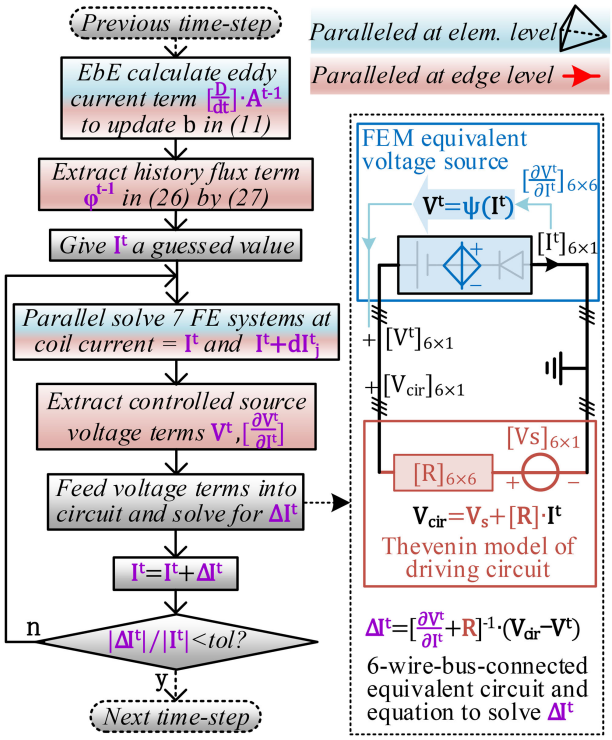

FIGURE 10. Flow chart of field-circuit co-simulation.

method is needed for the FEM transformer model to interplay with the external network. For simplicity, the indirect flux scheme [26] is applied in this work. The coupling scheme results from the below 3 observations:

1) Based on Faraday's law, the voltage of a coil is generated from the time-derivative of the flux of the 6 coils:

$$V_k = \left(\varphi_k^t - \varphi_k^{t-1}\right)/dt, \tag{26}$$

where V_k is the voltage of the kth coil, φ_k^t and φ_k^{t-1} are the coil's flux of the previous time step and current time step.

2) The coil k's flux φ_k^t can be expressed by the following equation due to Ampere's circuital law and the relation between $\vec{B}$ and $\vec{A}$.

$$\varphi^t = \frac{N_k}{S_k}\int_{V coil k} \vec{A}\cdot\vec{n_k}\,dv = \sum_{i=1}^{E} \mathrm{A_i}^t\left(\frac{N_k}{S_k}\cdot\int_{V coil k}\vec{N_i}\cdot\vec{n_k}\,dv\right), \tag{27}$$

where E is the total number of edges, $\mathrm{A_i}^t$ is the field unknown on i^{th} global edge, $\vec{N_i}$ means all interpolation functions associated with ith edge. N_k, S_k, and $\vec{n_k}$ means the number of winding turns, the wire intersection area, and the unit direction vector of the kth coil. All terms inside the blanket sign are fixed because the geometry and mesh stay unchanged. Therefore, the φ_k^t becomes purely a linear combination of A^t, and φ_k^{t-1} follows a similar discipline.

3) The FEM solution vector of the previous time step A^{t-1} is known, and the solution vector of current time step A^t is a function of the coil currents amplitude I based on (11).

Considering the above 3 descriptions, the voltages in the 6 coils are a nonlinear function of the 6 coil currents. As a result, one can treat the FEM model as a 6-port-current-controlled voltage source. With given input currents, the output of the voltage source can be calculated by solving the FEM problem and then extracting coil voltages from a fixed linear combination (26) and (27). This process is noted as:

$$V_k = \psi\left(I_j{}^t\right), \tag{28}$$

where $I_j{}^t$ is the current in coil j.

The controlled voltage source can be directly integrated with existing circuit solvers. In this work, we apply the NR scheme to find proper coil currents that produce consistent voltages between each coil and driving circuit connected to it. This process requires partial derivatives of the voltage values to form the Jacobian matrix. And the derivative can be found by small probing increment method:

$$\frac{\partial V_k\left(I_j{}^t\right)}{\partial I_j^t} = \frac{\psi_k\left(I_j{}^t + dI_j^t\right) - \psi_k\left(I_j{}^t\right)}{\Delta I}. \tag{29}$$

It is worth mentioning that the method requires solving the FEM problem 7 times. However, the 7 computations can be parallelized since they are not dependent on each other. The flow chart for the adaptive TLM field-circuit co-simulation solver is shown in Fig. 10.

V. TIME PARALLELISM: PARAREAL ALGORITHM FOR FIELD AND CIRCUIT CO-SIMULATION

The time Parareal algorithm can accelerate time-domain ordinary differential equations (ODEs) by introducing parallelism along the time span, and the method has been applied to nonlinear circuit solvers [27] and even nonlinear 2-D FEM

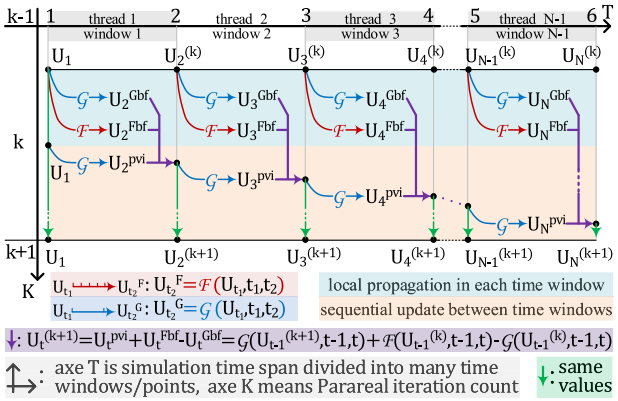

FIGURE 11. **Parallel-in-time algorithm explained in detail. Note: $U_2 \sim U_N$ are all set to 0 at the start of Parareal iteration, and U_1 is the known initial boundary value.**

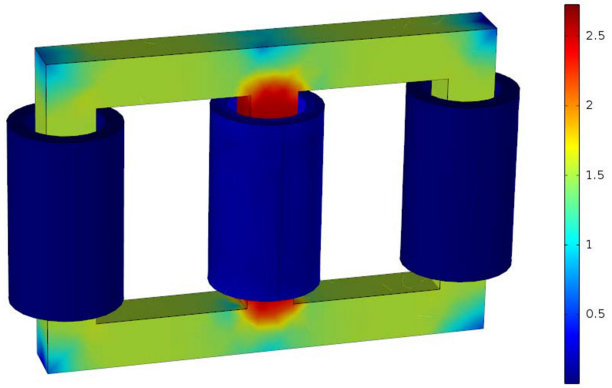

FIGURE 12. **Magnetic flux density B (T) field of the transformer under steady-state.**

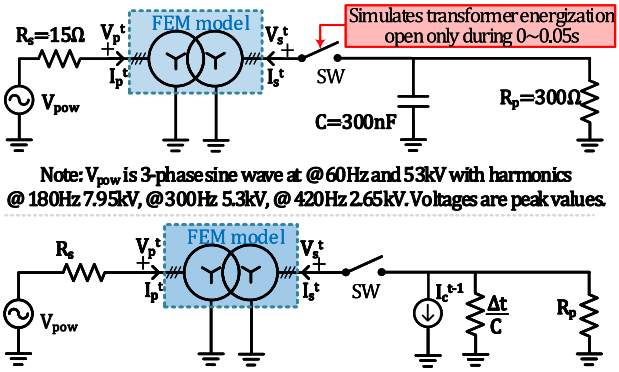

FIGURE 13. **3-phase balanced circuit and the time-discretized circuit model for the co-simulation.**

analysis [28] with decent speed-ups, since they are essentially ODE systems. On the other hand, due to the inherent ODE nature, the above adaptive TLM field-circuit solver can also be integrated with the Parareal method. As a result, the parallelism is further extended into both time and space levels so that hardware resources can be fully exploited.

To explain how the space-time parallelized field-circuit solver works, we introduce the following definitions:

1) Each edge's magnetic potential A^t, and the circuit's all nodal voltages and branch currents are noted as state variables. And these state variables are packed in one vector U_t.

2) By repeating the algorithm in Fig. 10, the state variables can be propagated from the initial time point t_1 to a given future time point t_2. When the system time step dt is small enough, the program yields precise state variable value at t_2, and the computation process is defined as the precise propagator $\mathcal{F}$:

$$U_{t_2}{}^{F} = \mathcal{F}\left(U_{t_1}, t_1, t_2\right). \tag{30}$$

If the time step dt is much bigger compared to $\mathcal{F}$, the program in Fig. 10 can only generate a coarse estimation of state variables at t_2 with a lower computation amount. This process is noted as the coarse propagator $\mathcal{G}$:

$$U_{t_2}{}^{G} = \mathcal{G}\left(U_{t_1}, t_1, t_2\right). \tag{31}$$

Due to the time-step difference, $\mathcal{G}$ requires light computation while $\mathcal{F}$ becomes computationally expensive.

As is shown in Fig. 11, the Parareal algorithm divides the whole time span into N-1 smaller time windows. Between these time windows are the time points where the state variables need to be found. Traditionally, to achieve precise results at these time points, operator $\mathcal{F}$ needs to sequentially propagate state values between each time window, which could be time-consuming considering the dependency of later time points on previous ones. In contrast, the Parareal algorithm executes the propagators in parallel inside each time window and utilizes iterations to make the values at each time point converge to the precise result.

During each iteration, every time window is assigned to a working thread. Each time window propagates in parallel the state values from their local start time point to the local endpoint, and the result is saved in their local buffers (U_i^{Fbf} or U_i^{Gbf}). Note that these local propagations involve both $\mathcal{F}$ and $\mathcal{G}$, and the computational expensive $\mathcal{F}$ is parallel-executed in each different time windows with different initial values. After the local propagations, state values at each time point are sequentially updated by the local buffers and the information from the previous time point as follows:

$$U_t{}^{(k+1)} = \mathcal{G}\left(U_{t-1}{}^{(k+1)}, \mathrm{t}-1, \mathrm{t}\right) + \mathcal{F}\left(U_{t-1}{}^{(k)}, \mathrm{t}-1, \mathrm{t}\right) - \mathcal{G}\left(U_{t-1}{}^{(k)}, \mathrm{t}-1, \mathrm{t}\right), \tag{32}$$

where k is the Parareal iteration count, and t is the time point to be updated.

The Parareal iteration repeats until state variables at the last time point converge into a stable result. If the Parareal iterations required are less than the number of time windows, the parallel-in-time algorithm becomes faster than the traditional time-serial method, and the Parareal-adaptive-TLM transformer model is boosted with both space and time parallelism.

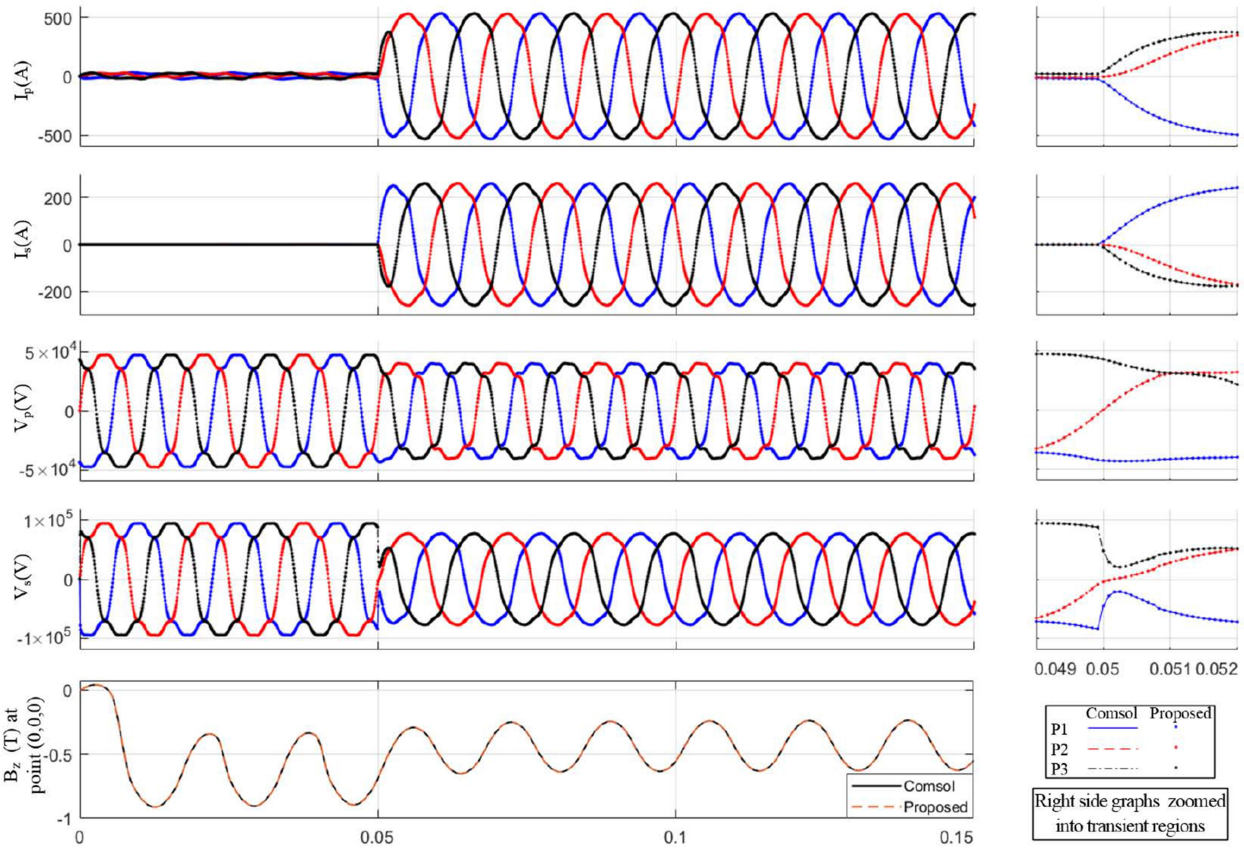

FIGURE 14. Result comparison of the space-time-parallel scheme and Comsol^{TM} results over time (s) for the 3-phase transformer model.

TABLE I Transformer Parameters

Transformer parameters					
Coil/Air permeability	μ_0	Winding turns primary/secondary	750/1500	Core σ	2000S
Iron core B-H curve	H=	$\begin{cases} 0.001\upsilon_0 B, & \lvert B\rvert \le 1.3 \\ 0.001\upsilon_0 B + 1e5(\lvert B\rvert - 1.3)^3 \times \lvert B\rvert / B, & \lvert B\rvert > 1.3 \end{cases}$			

TABLE II Comparison of *COMSOL*^{TM} and the TLM Solver in Static Case

Mesh size (DoF)		6465	15k	26.8k	85.8k	229k
Time use(s)	*Comsol*^{TM}	6.9	15.1	26.4	85.7	343.2
	Adaptive TLM	0.12	0.28	0.78	3.73	17.3
TLM iteration		3536	3054	4133	5531	7756
Speed-up		55.8	56.4	33.8	22.9	19.8

VI. CASE STUDIES

To verify the accuracy and efficiency of the proposed method, 2 case studies were carried out on the transformer model with the parameters shown in Table I. The algorithms were implemented on a workstation with Intel Xeon E5-2698 v4 CPU and NVIDIA Tesla V100-PCIR-16 GB GPU in CUDA C language. The convergence tolerance the implementation is set to a relative change of 1e-6, and the results are compared with *Comsol*^{TM}.

A. STATIC CASE

This case study verifies the convergence speed of the adaptive-TLM-EbE nonlinear FEM solver (Fig. 9) in a static scenario. A static current of 1000 A is fed into the middle primary coil to drive the transformer iron core into a saturated state. The time-step was set to infinite big to eliminate the σ terms for the simulation.

After computation, the magnetic flux distribution is shown in Fig. 12, and the comparison with Comsol based on different mesh sizes is displayed in Table II. Benefitting from the massive parallelism and boosted convergence rate, the method achieved a maximum speed-up of 55.8 with a relative error of less than 2%.

B. SPACE-TIME-PARALLEL FIELD-CIRCUIT CO-SIMULATION

In this case study, the dynamic field-circuit co-simulation in Fig. 10 is implemented to achieve space or elemental level parallelism. The case study also deploys the Parareal method of Section V to further parallelize the simulation at the time level. Parallelism is expanded as much as possible to realize the complex space-time field-circuit co-simulation.

Fig. 13 displays the circuit configuration used for the co-simulation. The total time-span of the co-simulation is 0.15 s, and the Parareal algorithm divides the entire time-span into 15 smaller time windows for the implementation. Within each time window, the propagator $\mathcal{G}$ is assigned a big time-step of 0.01 s for coarse prediction, and the propagator $\mathcal{F}$ is given a much smaller time-step (1/12000 s) to precisely calculate the state variables at the time points between the time windows.

Based on the above circuit and Parareal parameters, the computation was carried out with a fixed transformer mesh size of 6465. The Parareal algorithm takes 5 iterations to converge, and the 3-phase current and voltages are shown in Fig. 14 with a relative error of 2% vs. Comsol. The execution time of the proposed space-time-parallel model is 293.7 s in comparison with $Comsol^{TM}$'s 1288 s.

The speed-up is of this case study was not as much as its counterpart in the first case-study. However, it is worth mentioning that different time windows of the Parareal algorithm are sequentially executed due to the hardware limitation of the workstation. Given enough hardware resources, the computation time will be shrunk by 15 times, yielding a theoretical speed-up of over 65.

VII. CONCLUSION

In this paper, a modified adaptive transmission line method is presented so that it can be utilized for both static and transient simulations of electromagnetic apparatus. Compared with the traditional fixed-impedance transmission line technique, the new method changes the characteristic impedance based on the operating point, which dramatically improves the convergence speed of TLM iterations. The adaptive TLM method was then integrated with 3-D edge finite elements to achieve massive parallelism at space-level for transformer simulations. In addition, the Parareal method was, for the first time, applied to the 3-D finite edge element simulations. This further increased the parallelism at the space-time level for the FEM transformer model. The space-time-parallel transformer model was implemented on the GPU and compared with Comsol^{TM} in 2 case studies, and the result indicates a significant speed-up of over 55 with an error of less than 2%.

REFERENCES

[1] V. Brandwajn, H. W. Donnel, and I. I. Dommel, "Matrix representation of three-phase n-winding transformers for steady-state and transient studies," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 6, pp. 1369–1378, Jun. 1982.

[2] A. Narang and R. H. Brierley, "Topology based magnetic model for steady-state and transient studies for three-phase core type transformers," *IEEE Trans. Power Syst.*, vol. 9, no. 3, pp. 1337–1349, Aug. 1994.

[3] J. Liu and V. Dinavahi, "A real-time nonlinear hysteretic power transformer transient model on FPGA," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3587–3597, Jul. 2014.

[4] W. H. Kersting and W. H. Phillips, "Modeling and analysis of unsymmetrical transformer banks serving unbalanced loads," *IEEE Trans. Ind. Appl.*, vol. 32, no. 3, pp. 720–725, Jun. 1996.

[5] J. Peppanen, C. Rocha, J. A. Taylor, and R. C. Dugan, "Secondary low-voltage circuit models: "how good is good enough?," *IEEE Trans. Ind. Appl.*, vol. 54, no. 1, pp. 150–159, Jan./Feb. 2018.

[6] W. S. Fonseca, D. S. Lima, A. K. F. Lima, M. V. A. Nunes, U. H. Bezerra, and N. S. Soeiro, "Analysis of structural behavior of transformer's winding under inrush current conditions," *IEEE Trans. Ind. Appl.*, vol. 54, no. 3, pp. 2285–2294, May/Jun. 2018.

[7] H. Ahn, J. Lee, J. Kim, Y. Oh, S. Jung, and S. Hahn, "Finite-element analysis of short-circuit electromagnetic force in power transformer," *IEEE Trans. Ind. Appl.*, vol. 47, no. 3, pp. 1267–1272, May/Jun. 2011.

[8] 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.

[9] 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.

[10] J-L, Lions, Y. Maday, and G. Turinci, "A 'parareal' in time discretization of PDE's," *Comptes Rendus de l'Académie des Sci.*, Série 1, vol. 332, no. 7, pp. 661–668, 2015.

[11] J. Smajic, M. Bucher, R. Christen, and Z. Tanasic, "DG-FEM for time domain h- Φ eddy current analysis," *IEEE Trans. Magn.*, vol. 53, no. 6, pp. 1–4, Jun. 2017.

[12] J. Smajic, M. Bucher, R. Christen, and Z. Tanasic, "Stability analysis of time domain discontinuous galerkin H-Φ method for eddy current simulations," *IEEE Trans. Magn.*, vol. 54, no. 3, pp. 1–4, Mar. 2018.

[13] J. Smajic, M. K. Bucher, C. Jager, and R. Christen, "Treatment of multiply connected domains in time-domain discontinuous galerkin *H* - Φ eddy current analysis," *IEEE Trans. Magn.*, vol. 55, no. 6, pp. 1–4, Jun. 2019.

[14] J. Smajic *et al.*, "Simulation and measurement of lightning-impulse voltage distributions over transformer windings," *IEEE Trans. Magn.*, vol. 50, no. 2, pp. 553–556, Feb. 2014.

[15] J.-M. Jin, *The Finite Element Method in Electromagnetics*. Hoboken, NJ, USA: Wiley, 2015.

[16] R. Albanese and G. Rubinacci, "Finite element methods for the solution of 3 D eddy current problems," in *Proc. Adv. Imag. Electron Phys.*. Elsevier., 1997, pp. 1–86.

[17] M. Barton and Z. Cendes, "New vector finite elements for three-dimensional magnetic field computation," *J. Appl. Phys.*, vol. 61, no. 8, pp. 3919–3921, 1987.

[18] 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.

[19] A. Dziekonski, A. Lamecki, and M. Mrozowski, "GPU acceleration of multilevel solvers for analysis of microwave components with finite element method," *IEEE Microw. Wireless Compon. Lett.*, vol. 21, no. 1, pp. 1–3, Jan. 2011.

[20] M. M. Dehnavi, D. M. Fernandez, and D. Giannacopoulos, "Enhancing the performance of conjugate gradient solvers on graphic processing units," *IEEE Trans. Magn.*, vol. 47, no. 5, pp. 1162–1165, May 2011.

[21] A. F. P. de Camargos, V. C. Silva, J. Guichon, and G. Munier, "Efficient parallel preconditioned conjugate gradient solver on GPU for FE modeling of electromagnetic fields in highly dissipative media," *IEEE Trans. Magn.*, vol. 50, no. 2, pp. 569–572, Feb. 2014.

[22] A. Dziekonski and M. Mrozowski, "Block conjugate-gradient method with multilevel preconditioning and GPU acceleration for FEM problems in electromagnetics," *IEEE Antennas Wireless Propag. Lett.*, vol. 17, no. 6, pp. 1039–1042, Jun. 2018.

[23] P. B. Johns and M. O'Brien, "Use of the transmission-line modelling (t.l.m.) method to solve non-linear lumped networks," *Radio Electron. Engineer*, vol. 50, no. 1.2, pp. 59–70, Jan. 1980.

[24] A. Joseph, A. Joseph, and Administer, *Theory and Problems of Electric Circuits*. New York, NY, USA: McGraw Hill, 1994.

[25] 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.

[26] J. Li, P. Liu, and V. Dinavahi, "Matrix-free edge-domain decomposition method for massively parallel 3-D finite element simulation with field-circuit coupling," *IEEE Trans. Magn.*, vol. 56, no. 10, pp. 1–9, Oct. 2020.

[27] T. Cheng, T. Duan, and V. Dinavahi, "Parallel-in-time object-oriented electromagnetic transient simulation of power systems," *IEEE Open Access J. Power Energy*, vol. 7, pp. 296–306, Jul.2020.

[28] S. Schops, I. Niyonzima, and M. Clemens, "Parallel-in-time simulation of eddy current problems using parareal," *IEEE Trans. Magn.*, vol. 54, no. 3, pp. 1–4, Mar. 2018.

JIACONG LI (Student Member, IEEE) received the B.Sc. degree in electrical and computer engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2015. He is currently working toward the M.Sc. degree with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include computational electromagnetics, finite element analysis, and parallel and distributed processing.

PENG LIU (Student Member, IEEE) received the B.Sc. and M.Eng. degrees in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2013 and 2015, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include computational electromagnetics, high-performance finite element analysis, and parallel and distributed processing.

VENKATA DINAVAHI (Fellow, IEEE) received the B.Eng. degree in electrical engineering from the Visvesvaraya National Institute of Technology, Nagpur, India, in 1993, the M.Tech. degree in electrical engineering from the Indian Institute of Technology Kanpur, Kanpur, India, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2000. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include real-time simulation of power systems and power electronic systems, electromagnetic transients, device-level modeling, large-scale systems, and parallel and distributed computing.