Tensor Train Completion of System Tensors in FFT-Accelerated Integral Equation Simulators

Xiao Cong[(1)], Mingyu Wang[(2)], Xi Wang[(1)], Xiaofan Jia[(1)], Chao-Fu Wang[(3)] and Abdulkadir C. Yucel[(1)]

(1) School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore

(2) Xpeedic Co., LTD. 60 NaXian Road, Building 5, Pudong New Area, 201210, Shanghai, China

(3) National University of Singapore, 5A Engineering Drive 1, 117411, Singapore

***Abstract* — In this study, a tensor train-based tensor completion scheme is proposed to reduce the setup time of fast Fourier transform (FFT) - accelerated integral equation (IE) simulators. The proposed scheme is applied to completion of the system tensors arising in FFT-accelerated IE simulators developed for the analyses from DC to high-frequencies. The preliminary results show that the scheme yields 10.8x, 3x, and 2.6x computational time-saving during the generation of system tensors with 10^{-4} accuracy for electrostatic, quasi-magneto-static, and full-wave analyses, respectively.**

***Index Terms* — Fast Fourier transform (FFT), integral equation, system tensors, tensor train (TT) decompositions.**

I. Introduction

Integral equation (IE)-based electromagnetic (EM) simulators have become the popular choice of engineers/scientists for analysis, design, and characterization in many EM problems from DC to terahertz. Among these simulators, fast Fourier transform (FFT)-accelerated ones [1-5] offer substantially low computational time and memory requirements compared to the others in exchange for requiring a discretization via voxels and/or panels on a structured grid. In FFT-accelerated IE simulators, the matrix-vector multiplication is executed via convolutions performed on circulant tensors derived from system tensors. The generation of these system tensors requires excessive time that dominates the setup time of these simulators and sometimes is even more than iterative solution time for large-scale structures. To this end, a strategy based on Tucker decompositions [6-11] was proposed to accelerate the generation of these system tensors in [2-4]. In this strategy, very large system tensors for voxels with unit edge length are generated, compressed via Tucker decompositions, and stored in hard disk during the installation stage of the simulator. During the setup stage, the compressed tensors are restored and scaled by the actual voxel size; the restored tensors are resized according to the required computational domain size. This strategy yields a significant reduction in the setup time of the simulators, yet it requires pre-computation of the Tucker-compressed tensors during the installation stage.

This study proposes a tensor train (TT) decomposition-based [11] tensor completion scheme to compute system tensors efficiently without any pre-computation. In particular, TT-compressed representations of system tensors are obtained by alternating minimal energy cross algorithm. Then the TT-compressed representations are used to restore the original system tensors. As system tensors are low-rank, the computational time required to obtain TT-compressed representations is much shorter than the computational time required for obtaining full system tensors. The proposed TT-based tensor completion scheme is applied to generate the system tensors of open-source VoxCap [3], SuperVoxHenry/VoxHenry [1, 2], and MARIE [5] simulators developed for electrostatic, magneto-quasi-static, and full-wave analyses. Numerical results show that the TT-based tensor completion scheme yields significant time-saving during the setup stage for increasing computational domain size and TT tolerance. In particular, the scheme yields 10.8x, 3x, and 2.6x less computational time during the setup stages of VoxCap, VoxHenry, and MARIE for generating the system tensors of a computational domain discretized by $600\times600\times600$ voxels for an accuracy of 10^{-4}. It should be noted here that the TT-accelerated IE simulators have been thoroughly studied in recent years [12, 13]. However, a TT-based tensor completion scheme is proposed for the first time in this study to obtain the system tensors in FFT-accelerated IE simulators.

II. Formulation

In this section, the system tensors (and their TT representations) arising in FFT-accelerated IE simulators (in VoxCap, VoxHenry, and MARIE) are expounded after their corresponding integral equations and linear system of equations (LSEs) are explained.

Let S' and V' denote the surface and volume of a structure (a conductor or a dielectric target) residing in a vacuum with permittivity ε_0 and permeability μ_0. The computational domain enclosing the structure is discretized by $N_t = N_x \times N_y \times N_z$ voxels with edge length Δx ,where N_x ,

N_y and N_z denote the number of voxels along *x*-, *y*-, and *z*-directions, respectively. While the structure occupies K non-air voxels, there exist N panels discretizing the surfaces of structures.

In the electrostatic analysis of the structure, VoxCap solves a surface integral equation, which reads as

$$\Phi(\mathbf{r}) = \int_{S'} \frac{\rho(\mathbf{r}')}{4\pi\varepsilon_0 |\mathbf{r}-\mathbf{r}'|} dS' , \qquad (1)$$

where $\mathbf{r}$ and $\mathbf{r}'$ denote the observer and source points, respectively. $\Phi(\mathbf{r})$ and $\rho(\mathbf{r}')$ denote the potential and charge density on the surface of the structure (conductor), respectively. After discretizing $\rho(\mathbf{r}')$ by piecewise constant basis function $w_l(\mathbf{r}')$ defined on panels and applying Galerkin testing, an $N \times N$ system matrix $\bar{\mathbf{P}}$ is obtained and grouped with respect to the normal direction of panels as

$$\bar{\mathbf{P}} = \begin{bmatrix} \bar{\mathbf{P}}^{x,x} & \bar{\mathbf{P}}^{x,y} & \bar{\mathbf{P}}^{x,z} \\ \bar{\mathbf{P}}^{y,x} & \bar{\mathbf{P}}^{y,y} & \bar{\mathbf{P}}^{y,z} \\ \bar{\mathbf{P}}^{z,x} & \bar{\mathbf{P}}^{z,y} & \bar{\mathbf{P}}^{z,z} \end{bmatrix} . \qquad (2)$$

In VoxCap, system Toeplitz tensors $\mathcal{P}^{\alpha,\beta}$, $\alpha,\beta = \{x,y,z\}$, corresponding to the blocks $\bar{\mathbf{P}}^{\alpha,\beta}$ in $\bar{\mathbf{P}}$ are obtained via [3]

$$\mathcal{P}^{\alpha,\beta}_{i,j,k} = \int_{S_l}\int_{S_1} \frac{w_l(\mathbf{r}) w_1(\mathbf{r}')}{4\pi\varepsilon_0 |\mathbf{r}-\mathbf{r}'|} dS' dS , \qquad (3)$$

where $i, j,$ and k indices vary from 1 to N_x+1, N_y+1, and N_z+1, respectively, while $l = i + (N_x+1)(j-1) + (N_x+1)(N_y+1)(k-1)$.

In the magneto-quasi-static analysis, VoxHenry solves the current continuity law $\nabla \cdot \mathbf{J}(\mathbf{r}) = 0$ and a volume integral equation, which is [1]

$$\frac{\mathbf{J}(\mathbf{r})}{\sigma} + j\omega\mu_0 \int_{V'} \frac{\mathbf{J}(\mathbf{r}')}{4\pi|\mathbf{r}-\mathbf{r}'|} dV' = -\nabla\Phi(\mathbf{r}) , \qquad (4)$$

where σ and ω denote the conductivity of the structure and the angular frequency, respectively. After discretizing the current density $\mathbf{J}(\mathbf{r}')$ by piecewise constant and linear basis functions, $\mathbf{f}_l^{\beta}, \beta = \{x,y,z,2D,3D\}$, and applying Galerkin testing, a $5K \times 5K$ system matrix $\bar{\mathbf{Z}}$ is obtained as [1]

$$\bar{\mathbf{Z}} = \begin{bmatrix} \bar{\mathbf{Z}}^{x,x} & 0 & 0 & \bar{\mathbf{Z}}^{x,2D} & \bar{\mathbf{Z}}^{x,3D} \\ 0 & \bar{\mathbf{Z}}^{y,y} & 0 & \bar{\mathbf{Z}}^{y,2D} & \bar{\mathbf{Z}}^{y,3D} \\ 0 & 0 & \bar{\mathbf{Z}}^{z,z} & 0 & \bar{\mathbf{Z}}^{z,3D} \\ \bar{\mathbf{Z}}^{2D,x} & \bar{\mathbf{Z}}^{2D,y} & 0 & \bar{\mathbf{Z}}^{2D,2D} & \bar{\mathbf{Z}}^{2D,3D} \\ \bar{\mathbf{Z}}^{3D,x} & \bar{\mathbf{Z}}^{3D,y} & \bar{\mathbf{Z}}^{3D,z} & \bar{\mathbf{Z}}^{3D,2D} & \bar{\mathbf{Z}}^{3D,3D} \end{bmatrix} . \qquad (5)$$

The system Toeplitz tensors $\mathcal{Z}^{\alpha,\beta}$, $\alpha,\beta = \{x,y,z,2D,3D\}$, corresponding to blocks $\bar{\mathbf{Z}}^{\alpha,\beta}$ in $\bar{\mathbf{Z}}$, are obtained via [1]

$$\mathcal{Z}^{\alpha,\beta}_{i,j,k} = j\omega\mu_0 \int_{V_l}\int_{V_1} \frac{\mathbf{f}_l^{\alpha}(\mathbf{r}) \cdot \mathbf{f}_1^{\beta}(\mathbf{r}')}{4\pi|\mathbf{r}-\mathbf{r}'|} dV' dV , \qquad (6)$$

where $i, j,$ and k indices vary from 1 to N_x, N_y, and N_z, respectively, while $l = i + N_x(j-1) + N_x N_y (k-1)$.

In the full-wave analysis of a structure, MARIE solves the electric field volume integral equation, which reads as [5]

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{\text{inc}}(\mathbf{r}) + \frac{1}{j\omega\varepsilon_0}(k_0^2 + \nabla\nabla\cdot)\int_{V'} \frac{e^{-jk_0|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \mathbf{J}(\mathbf{r}') dV' , \quad (7)$$

where $\mathbf{E}(\mathbf{r})$ and $\mathbf{E}^{\text{inc}}(\mathbf{r})$ denote total electric field in the structure (dielectric target) and incident electric field, respectively, while $k_0 = \omega(\mu_0\varepsilon_0)^{0.5}$. After discretizing $\mathbf{J}(\mathbf{r}')$ with pulse basis functions $\mathbf{f}_l^{\beta}$, $\beta = \{x,y,z\}$, [5], and applying Galerkin testing, a $3K \times 3K$ system matrix $\bar{\mathbf{G}}$ is obtained as

$$\bar{\mathbf{G}} = \begin{bmatrix} \bar{\mathbf{G}}^{x,x} & \bar{\mathbf{G}}^{x,y} & \bar{\mathbf{G}}^{x,z} \\ \bar{\mathbf{G}}^{y,x} & \bar{\mathbf{G}}^{y,y} & \bar{\mathbf{G}}^{y,z} \\ \bar{\mathbf{G}}^{z,x} & \bar{\mathbf{G}}^{z,y} & \bar{\mathbf{G}}^{z,z} \end{bmatrix} . \qquad (8)$$

The system Toeplitz tensors $\mathcal{G}^{\alpha,\beta}$, $\alpha,\beta = \{x,y,z\}$, corresponding to the blocks $\bar{\mathbf{G}}^{\alpha,\beta}$, in $\bar{\mathbf{G}}$ are obtained via

$$\mathcal{G}^{\alpha,\beta}_{i,j,k} = \sum_m \sum_n (\hat{\mathbf{n}}_m \times \hat{\boldsymbol{\alpha}}) \cdot (\hat{\mathbf{n}}_n \times \hat{\boldsymbol{\beta}}) \int_{S_m}\int_{S'_n} \frac{e^{-jk_0|\mathbf{r}-\mathbf{r}'+\mathbf{r}_l-\mathbf{r}'_1|}}{4\pi|\mathbf{r}-\mathbf{r}'+\mathbf{r}_l-\mathbf{r}'_1|} dS' dS , \quad (9)$$

where $i = 1,2,...,N_x$, $j = 1,2,...,N_y$, $k = 1,2,...,N_z$, $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}} = \{\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}\}$, while $l = i + N_x(j-1) + N_x N_y (k-1)$. The summation on m and n are performed on each face of voxels and the surface integration is performed on S_m and S'_n with unit normals $\hat{\mathbf{n}}_m$ and $\hat{\mathbf{n}}_n$, respectively.

All system tensors can be efficiently obtained by TT completion. To this end, let $\mathcal{T} = \{\mathcal{P}, \mathcal{Z}, \mathcal{G}\}$ represent a real/complex four-dimensional array, i.e., $\mathcal{T} \in \{\mathbb{R}^{n_1 \times \ldots \times n_4} / \mathbb{C}^{n_1 \times \ldots \times n_4}\}$. Its TT decomposition reads as [14]:

$$\mathcal{T}(i_1,...,i_4) = \sum_{\alpha_0,...,\alpha_4} \mathcal{Y}_1(\alpha_0, i_1, \alpha_1) ... \mathcal{Y}_4(\alpha_3, i_4, \alpha_4) . \quad (10)$$

Here $\mathcal{Y}_k \in \mathbb{C}^{r_{k-1} \times n_k \times r_k}, k = 1,...,4$, represents TT core tensor and r_k is the TT rank ($r_0 = r_4 = 1$). i_k and α_k stand for spatial and auxiliary indices, respectively. An alternating minimal energy cross approximation [15] can be used to obtain the TT cores, which then can be used to obtain the full tensor $\mathcal{T}$. While obtaining TT cores, small numbers of rows, columns, and

fibers of these tensors are obtained by calling the original tensor fill routines in VoxCap, VoxHenry, and MARIE, since these tensors are low-rank. After obtaining TT cores, the rest of the entries are obtained by TT completion, which is performed much faster than obtaining the tensor entries from original routines employing quadrature or analytical expressions.

III. Numerical Results

This section provides the preliminary results demonstrating the efficiency and accuracy achieved by TT completion scheme. All numerical tests are conducted on a Linux server with Intel® Xeon® Gold 6142 CPU 2.60GHz and MATLAB R2019a.

First, the system tensors generated in VoxCap simulator are obtained by the proposed scheme. In particular, a perfectly electric conducting sphere with a diameter of 0.5m is discretized by voxels of size 0.005, 0.0025, 0.00167, 0.00125, 0.001, and 0.00083 m for the test. This gives rise to $N_x = N_y = N_z = 100,\ldots,600$ with an increase of 100 voxels along each principal axis for each test. Initially, we compare the setup time to obtain $\mathcal{P}^{\alpha,\beta}$, $\alpha,\beta = \{x,y,z\}$ via original tensor fill routines in VoxCap and TT-completion scheme with tolerance 10^{-4} and 10^{-6} [Fig. 1]. In Fig 1, the data points corresponding to computational domain sizes $100\times100\times100$, ..., $600\times600\times600$ are the ones with tensor sizes of 6×10^6, ..., 1.296×10^9. For the largest computational domain realized by $600\times600\times600$ voxels, the TT-completion scheme yields 10.8x and 6.08x computational saving for TT tolerance 10^{-4} and 10^{-6} compared to original routines in VoxCap employing analytical expression and quadrature. The maximum TT rank is obtained as 13 and 20 while the memory requirement of the TT core tensors is 3.95 MB and 8.5 MB for TT tolerance of 10^{-4} and 10^{-6}, respectively. Note that the memory requirement of the original tensor is 9888 MB. Furthermore, to evaluate the accuracy of the proposed scheme, the system tensors obtained by the TT-completion scheme are used while obtaining the self-capacitance of the sphere. Fig. 2 compares the self-capacitance obtained via analytical formula and TT completion-augmented VoxCap scheme with TT tolerance 10^{-4} and 10^{-6}. Clearly, the self-capacitance values match up to two digits with increasing $1/\Delta x$ (or increasing system tensor size and resolution). To obtain the self-capacitance via direct VoxCap and TT completion-augmented VoxCap, both schemes require the same number of iterations during the iteration solution, indicating that the accuracy obtained by the TT-completion scheme with 10^{-4} accuracy is sufficient for accurate analysis.

Next, the proposed scheme is applied to the completion of system tensors in VoxHenry. Again, the computational domain size is increased from $100\times100\times100$ to $600\times600\times600$ with 100 voxel increment along each principle axis for each test, while voxel size is decreased from 0.05 to 0.0083 m. The setup time to obtain $\mathcal{Z}^{\alpha,\beta}, \alpha,\beta = \{x,y,z,2D,3D\}$, via original tensor-fill routines in VoxHenry and TT-completion scheme with TT tolerance 10^{-4} and 10^{-6} is compared in Fig.3. In Fig.3, the data points corresponding to computational domain sizes $100\times100\times100$, ..., $600\times600\times600$ are the ones with tensor sizes of 1×10^7, …, 2.16×10^9. For the largest computational domain $600\times600\times600$, the proposed scheme yields 3x and 1.7x computational saving for TT tolerance 10^{-4} and 10^{-6}. Again, maximum TT rank is 14 and 22, while the memory requirements of TT cores are 2.3 and 3.96 MB, while the memory requirement of the original tensor is 32,959 MB.

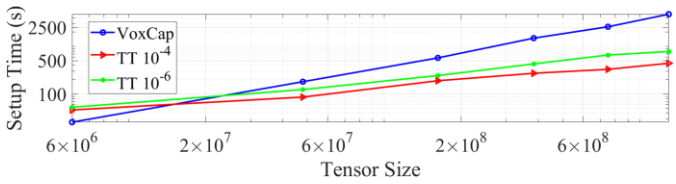

Fig. 1. Setup time required to obtain VoxCap's system tensors via its tensor-fill routines and TT-completion scheme.

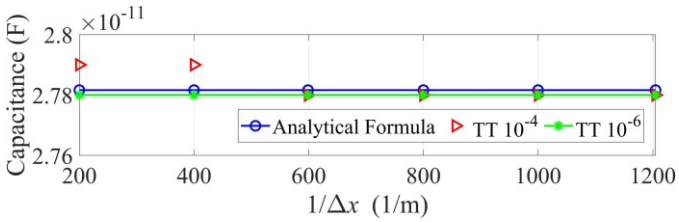

Fig. 2. The self-capacitance of a sphere obtained by analytical formula and TT completion-augmented VoxCap.

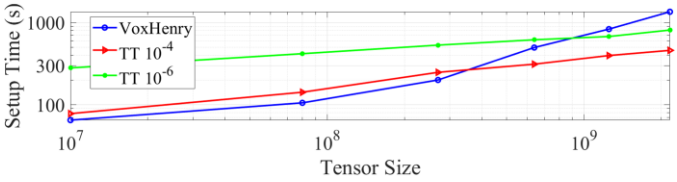

Fig. 3. Setup time required to obtain VoxHenry's system tensors via its tensor-fill routines and TT-completion scheme

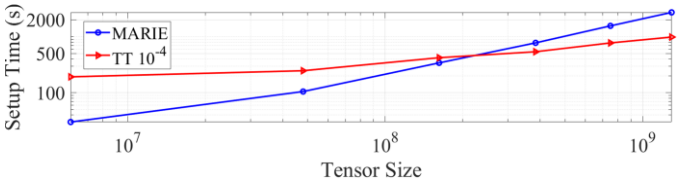

Fig. 4. Setup time required to obtain MARIE's system tensors via its tensor-fill routines and TT-completion scheme

Finally, the proposed scheme is used to efficiently obtain the system tensors in MARIE. For this purpose, the computational domain is discretized by voxels of size 0.05, 0.025, 0.0167, 0.0125, 0.01, and 0.0083 m for the test. This gives rise to $N_x = N_y = N_z = 100,\ldots,600$ with an increase of

100 voxels along each principal axis. The angular frequency ω is set to 1.885 MHz. The setup time to obtain $\mathcal{G}^{\alpha,\beta}$, $\alpha,\beta=\{x,y,z\}$ via original tensor-fill routines in MARIE and the TT-completion scheme with TT tolerance of 10^{-4} is provided in Fig.4. In Fig.4, the data points corresponding to computational domain sizes $100\times100\times100$, ..., $600\times600\times600$ are the ones with tensor sizes of 6×10^6, ..., 1.296×10^9. Clearly, after certain tensor size, for the computational domains with $N_x=N_y=N_z=\{400,500,600\}$, TT-completion scheme requires far less computational time. For the largest computational domain $600\times600\times600$, the proposed scheme yields 2.6x computational saving and the maximum TT rank is 13. While the memory requirements of TT cores are 1.2 MB, that of the original tensor is 19,874 MB.

IV. Conclusion

In this work, a TT-based tensor completion scheme was proposed to reduce the setup time of FFT-accelerated IE solvers. The scheme was applied to the generation of block Toeplitz system tensors of open-source FFT-accelerated IE simulators, including VoxCap, VoxHenry, and MARIE. Compared with existing tensor-fill routines inside these simulators, TT-based completion scheme requires far less computational time and resources. In the talk, further numerical results showing the accuracy and efficiency of the proposed scheme will be provided for the magneto-quasi-static and full-wave analyses.

References

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

[2] 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," *IEEE Trans. Appl. Supercond.,* vol. 31, no. 7, pp. 1-11, Oct. 2021.

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

[4] M. Wang, Y. Liu, P. Ghysels, and Abdulkadir C. Yucel, "VoxImp: impedance extraction simulator for voxelized structures," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.,* Early Access

[5] A. G. Polimeridis, J. F. Villena, L. Daniel, and J. K. White, "Stable FFT-JVIE solvers for fast analysis of highly inhomogeneous dielectric objects," *J. Comp. Phys.,* vol. 269, pp. 280-296, 2014.

[6] I. V. Oseledets, D. V. Savostianov, and E. E. Tyrtyshnikov, "Tucker dimensionality reduction of three-dimensional arrays in linear time," *SIAM J. Matrix Anal. Appl.,* vol. 30, no. 3, pp. 939–956, 2008.

[7] C. Qian and A. C. Yucel, "On the compression of translation operator tensors in FMM-FFT-accelerated SIE simulators via tensor decompositions," *IEEE Trans. Antennas Propag.,* vol. 69, no. 6, pp. 3359-3370, Jan. 2021.

[8] C. Qian, M. Wang, and A. C. Yucel, "Compression of far-fields in the fast multipole method via Tucker decomposition," *IEEE Trans. Antennas Propag.,* vol. 69, no. 10, pp. 6660-6668, Oct. 2021.

[9] I. I. Giannakopoulos, G. D. Guryev, J. E. C. Serralles, I. P. Georgakis, L. Daniel, and J. K. W. Lattanzi, "Compression of volume-surface integral equation matrices via Tucker decomposition for magnetic resonance applications," *IEEE Trans. Antennas Propag.,* vol. 70, no. 1, pp. 459-471, Jan. 2022.

[10] I. I. Giannakopoulos, M. S. Litsarev, and A. G. Polimeridis, "3D cross-Tucker approximation in FFT-based volume integral equation methods," presented at the Proc IEEE Int. Symp. Antennas Propagat. & CNC-USNC/URSI National Radio Sci. Meet., 2018.

[11] A. C. Yucel, L. J. Gomez, and E. Michielssen, "Compression of translation operator tensors in FMM-FFT accelerated SIE solvers via Tucker decomposition," *IEEE Antennas Wireless Propag. Lett.,* vol. 16, pp. 2667-2670, 2017.

[12] Z. Chen, L. J. Gomez, S. Zheng, A. C. Yucel, Z. Zhang, and V. I. Okhmatovski, "Sparsity-aware precorrected tensor train algorithm for fast solution of 2-D Scattering problems and current flow modeling on unstructured meshes," *IEEE Trans. Microw. Theory Tech.,* vol. 67, no. 12, pp. 4833-4847, Dec 2019.

[13] Z. Chen, S. Zheng, and V. I. Okhmatovski, "Tensor train accelerated solution of volume integral equation for 2-D scattering problems and magneto-quasi-static characterization of multiconductor transmission lines," *IEEE Trans. Microw. Theory Tech.,* vol. 67, no. 6, pp. 2181-2196, June 2019.

[14] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.,* vol. 33, no. 5, pp. 2295–2317, 2011.

[15] I. Oseledets and E. Tyrtyshnikov, "TT-cross approximation for multidimensional arrays," *Linear Algebra Appl.,* vol. 432, no. 1, pp. 70-88, Jan. 2010.