Reduced Order Modeling for Parameterized Electromagnetic Simulation Based on Tensor Decomposition

Xiao-Feng He, Liang Li, Stéphane Lanteri, Kun Li

Reduced Order Modeling for Parameterized Electromagnetic Simulation Based on Tensor Decomposition

Xiao-Feng He, Liang Li, Stéphane Lanteri, Kun Li

***Abstract*—We present a data-driven surrogate modeling for parameterized electromagnetic simulation. This method extracts a set of reduced basis (RB) functions from full-order solutions through a two-step proper orthogonal decomposition (POD) method. A mapping from the time/parameter to the principal components of the projection coefficients, extracted by canonical polyadic decomposition (CPD) , is approximated by a cubic spline interpolation (CSI) approach. The reduced-order model (ROM) is trained in the offline phase, while the RB solution of a new time/parameter value is recovered fast during the online phase. We evaluate the performance of the proposed method with numerical tests for the scattering of a plane wave by a 2-D multi-layer dielectric disk and a 3-D multi-layer dielectric sphere.**

***Index Terms*—electromagnetic simulation, model order reduction, proper orthogonal decomposition, canonical polyadic decomposition, cubic spline interpolation**

I. Introduction

ELECTROMAGNETIC scattering problems are generally described as the parameterized time-domain Maxwell equations [16],where the parameters may represent geometric shapes, material properties, system configurations and so on. Solving the full-order model (FOM) repeatedly for varying parameter values by high-fidelity simulations is often required in many applications such as real-time system control, weather forecast and structure design, leading to great CPU time and memory demands. It is urgent to study the parametric surrogate modeling to accelerate the solution of the partial differential equations (PDEs) in an efficient and accurate way. Model order reduction (MOR) techniques have been proposed to solve this problem in the past decades. The main idea of MOR is to replace a high-dimensional FOM with a surrogate model, which has a lower dimensionality and acceptable accuracy loss [20].

Some techniques such as the fast multipole method (FMM) [29], subdomain multilevel approach (SMA) [30] and characteristic basis functions method (CBFM) [25] have been proposed to accelerate the solution of scattering problems. For example, CBFM has been developed for an efficient solution of Method of Moments matrix equations. Compared to conventional RWG basis functions [28] bounded by the $\lambda/20$ domain discretization, this approach is based on high order basis functions over enlarged blocks, thus reducing the number of degrees of freedom (DoFs). A reduced basis (RB) method adopted in this work reduces the dimensionality of FOM in a completely different manner. In general, repeated simulations are usually not necessary because intrinsic similarities among the solutions of different parameters can be used to predict the solutions for new parameter values [11]. The RB method with an offline-online procedure, is one of the most widely used techniques for the surrogate modeling of time-dependent parameterized problems. In the offline phase, a reduced space representing the main dynamics of the FOM is constructed from full-order snapshots of different time/parameter values. There are several methods to extract the reduced basis functions, such as the proper orthogonal decomposition (POD) [14], [21], [22], greedy algorithms [18], [24] and an autoencoder [8], [10]. As the most widely-used technique, POD is used in this paper to generate the RB functions with a singular-value decomposition (SVD) [11]. As an approximation to the full-order solution, the RB solution, also named the reduced-order solution is recovered as a linear combination of the RB functions, where only the combination coefficients, also named projection coefficients will be calculated in the online phase [20]. It is advised that the methods mentioned above can be combined to accelerate the solution of parameterized electromagnetic scattering problems. For example, we could build a data-driven ROM based on snapshots generated by CBFM offline, then recover the RB solution for a new parameter online.

RB methods are divided into intrusive and non-intrusive (or data-driven) ones according to the methodology of calculating the projection coefficients. The former determines the projection coefficients by solving a low dimensional linear system, i.e., a projection of the full-order model onto the reduced space. And the Galerkin procedure is often used for the projection [7], [26]. However, the main drawback is that intrusive ROMs require access to the high dimensional model, and provide limited computational gain especially for complex nonlinear problems [1].The non-intrusive ROMs approximate the projection coefficients over the parameter domain by using a database of reduced order information [5]. The high-fidelity solvers such as FEM, FDM, DGTD are only used to generate snapshots, leading to a full decoupling of the online stage and the high-fidelity scheme. Many excellent non-intrusive ROMs are proposed in the past. For instance, a feed-forward neural

The first two authors are with the School of Mathematical Sciences, University of Electronic Science and Technology of China (hexf@std.uestc.edu.cn, plum.liliang@gmail.com, plum_liliang@uestc.edu.cn). Stéphane Lanteri is with Inria ATLANTIS project-team (Stephane.Lanteri@inria.fr). Kun Li is with the School of Mathematics, Southwestern University of Finance and Economics (rclikun@163.com, likun@swufe.edu.cn). The corresponding author is Liang Li.

Manuscript received March 25, 2023

network (FNN) has been designed for parameterized unsteady flows [15], [31], where a FNN is trained to approximate the projection coefficients. Interpolation and regression methods such as cubic spline interpolation (CSI) [3], [20], radial basis functions (RBF) [33] and Gaussian process regression (GPR) [11], [34] have also been adopted to approximate the projection coefficients. By combining an FNN-based regression model with a physics-informed neural network (PINN) [23], [27], a hybrid surrogate model is devised in [6], which requires less training data.

Data-driven reduced-order modeling based on RB method has been applied to electromagnetic scattering simulations [20], [34] recently. These related works use the POD technique to extract a set of time-/parameter-independent orthogonal basis functions and compute a series of projection coefficient matrices, then perform SVD for each projection matrix to extract the dominating temporal and parametric modes. Interpolation or regression models are built to approximate the temporal and parametric modes respectively. These methods are accurate enough for time-dependent parametric problems, but require to build a large number of interpolation or regression models (determined by the number of basis functions and the number of truncated singular values of each projection coefficient matrix), which increases the test time online. Motivated by the above problem, a ROM composed of two-step POD, canonical polyadic decomposition (CPD), and cubic spline interpolation (CSI) is developed for parameterized electromagnetic simulation. We employ a canonical polyadic decomposition (CPD) [17] to extract the principal components of the projection coefficients tensor, instead of repeated SVD for all projection coefficient matrices, which significantly reduces the number of interpolation or regression models, thus reducing the test time online with an acceptable loss of accuracy. The proposed method, named POD-CPD-CSI ROM is validated to be efficient and accurate for parameterized electromagnetic scattering problems.

The symbols and notations used in this paper are introduced as follows. Notations t and $\boldsymbol{\mu}$ denote time and parameter respectively. $\mathbf{S}_{\mathbf{u}}^{j}$ is the time trajectory matrix and $\mathbf{S}_{\mathbf{u}}$ is the snapshot matrix. $\mathcal{N}_h$ is the geometric DoFs, $\mathcal{N}_t$ and $\mathcal{N}_p$ are the number of time and parameter samples respectively. $\mathbf{u}_h(t,\boldsymbol{\mu})$ and $\mathbf{u}_h^r(t,\boldsymbol{\mu})$ are the full-order solution and reduced-order solution respectively. $\boldsymbol{\alpha}_{\mathbf{u}}(t,\boldsymbol{\mu})$ denotes the projection coefficient vector and $\mathbf{V}$ is the POD basis matrix. $\mathbf{P}_{\mathbf{u}}^{\ell}$ is the projection coefficient matrix and $\mathcal{G}_{\mathbf{u}}$ is the projection coefficient tensor.

II. Two-step POD method

Assume that $\mathcal{P}_h^{tr} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_{\mathcal{N}_p}\}$ is a parameter sampling that we're interested in. For each $\boldsymbol{\mu}_j \in \mathcal{P}_h^{tr}$, we have a time trajectory matrix

$$\mathbf{S}_{\mathbf{u}}^{j} = \left[\mathbf{u}_h(t_1,\boldsymbol{\mu}_j) \mid \mathbf{u}_h(t_2,\boldsymbol{\mu}_j) \mid \cdots \mid \mathbf{u}_h(t_{\mathcal{N}_t},\boldsymbol{\mu}_j)\right] \in \mathbb{R}^{\mathcal{N}_h \times \mathcal{N}_t}, \tag{1}$$

with $\mathbf{u} \in \{\mathbf{E}, \mathbf{H}\}$, which could be obtained either from high-fidelity solutions or from experiments, for a time sampling $\mathcal{T}_h^{tr} = \{t_1, t_2, \cdots, t_{\mathcal{N}_t}\}$. Here, $\mathcal{N}_h$ denotes the number of geometric DoFs. A snapshot matrix for all parameters in $\mathcal{P}_h^{tr}$ is then formed

$$\mathbf{S}_{\mathbf{u}} = \left[\mathbf{S}_{\mathbf{u}}^{1} \mid \mathbf{S}_{\mathbf{u}}^{2} \mid \cdots \mid \mathbf{S}_{\mathbf{u}}^{\mathcal{N}_p}\right] \in \mathbb{R}^{\mathcal{N}_h \times \mathcal{N}_s}, \tag{2}$$

with $\mathcal{N}_s = \mathcal{N}_t \cdot \mathcal{N}_p$. To reduce the dimensionality of FOM, we seek a low-rank decomposition to the snapshot matrix and construct a reduced space $V_{\mathbf{u},rb} = \operatorname{span}\{\mathbf{v}_{\mathbf{u},1}, \mathbf{v}_{\mathbf{u},2}, \cdots, \mathbf{v}_{\mathbf{u},d_{\mathbf{u}}}\}$ of dimension $d_{\mathbf{u}} \ll \min\{\mathcal{N}_h, \mathcal{N}_s\}$, where $\{\mathbf{v}_{\mathbf{u},i}\}_{i=1}^{d_{\mathbf{u}}}$ are time- and parameter-independent orthogonal RB functions [11], [34].

For the large snapshots matrix $\mathbf{S}_{\mathbf{u}}$, i.e. when $\mathcal{N}_s$ is large, it will be very computationally expensive to directly perform POD on $\mathbf{S}_{\mathbf{u}}$. So, we adopt a two-step POD method [20] shown in Algorithm 1 to reduce the demands on computational cost. Details of the computational complexity and error analysis of this method can be found in [20]. The RB solution $\mathbf{u}_h^r(t,\boldsymbol{\mu})$, an approximation to the full-order solution $\mathbf{u}_h(t,\boldsymbol{\mu})$, is recovered as

$$\mathbf{u}_h^r(t,\boldsymbol{\mu}) = \mathbf{V}_{\mathbf{u}}\boldsymbol{\alpha}_{\mathbf{u}}(t,\boldsymbol{\mu}) = \sum_{i=1}^{d_{\mathbf{u}}} \alpha_{\mathbf{u},i}(t,\boldsymbol{\mu})\mathbf{v}_{\mathbf{u},i}, \tag{3}$$

where $\mathbf{V}_{\mathbf{u}} \in \mathbb{R}^{\mathcal{N}_h \times d_{\mathbf{u}}}$ is the POD basis matrix generated by the two-step POD and vector $\alpha_{\mathbf{u}}(t,\boldsymbol{\mu}) = [\alpha_{\mathbf{u},1}(t,\boldsymbol{\mu}), \cdots, \alpha_{\mathbf{u},d_{\mathbf{u}}}(t,\boldsymbol{\mu})]^T$ is called the coefficient vector [34]. After calculating the POD basis functions, we need to build a mapping between the time/parameters and the projection coefficients.

Algorithm 1: Two-step POD approach

input : $\mathbf{S}_{\mathbf{u}}^{1}, \mathbf{S}_{\mathbf{u}}^{2}, \cdots, \mathbf{S}_{\mathbf{u}}^{\mathcal{N}_p}$ and truncation parameters ρ_t and $\rho_{\boldsymbol{\mu}}$

output: POD basis matrix $\mathbf{V}_{\mathbf{u}}$

Calculate matrices $\mathbf{T}_{\mathbf{u}}^{j}$ =POD $(\mathbf{S}_{\mathbf{u}}^{j}, \rho_t)$ for $j = 1, 2, \cdots, \mathcal{N}_p$;

Assemble matrix $\mathbf{T}_{\mathbf{u}} = [\mathbf{T}_{\mathbf{u}}^{1} \mid \mathbf{T}_{\mathbf{u}}^{2} \mid \cdots \mid \mathbf{T}_{\mathbf{u}}^{\mathcal{N}_p}]$;

Compute matrix $\mathbf{V}_{\mathbf{u}}$ =POD $(\mathbf{T}_{\mathbf{u}}, \rho_{\boldsymbol{\mu}})$.

Function $\mathbf{V}$=POD $(\mathbf{A}, \rho)$

  Compute the eigenvalues $\{\lambda_i\}_{i=1}^{r}$ and the eigenvectors $\{\mathbf{u}_i\}_{i=1}^{r}$ of matrix $\mathbf{A}^T\mathbf{A}$ with $r = \operatorname{rank}(\mathbf{A})$;

  Determine the number of basis functions d based on $d = \arg\min\{\pi(d) : \pi(d) \geqslant 1 - \rho\}$ and $\pi(d) = \sum_{i=1}^{d}\lambda_i / \sum_{i=1}^{r}\lambda_i$;

  Calculate the RB functions $\mathbf{v}_i = \frac{1}{\sqrt{\lambda_i}}\mathbf{A}\mathbf{u}_i$ for $i = 1, 2, \cdots, d, d \ll r$;

  Assemble matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_d]$.

end

III. CP-decomposition-based interpolation

Each vector in the snapshot matrix (2) can be approximated by a linear combination of the RB basis vectors (3) [34].

In such a way, we create training data and rewrite all the projection coefficients with a matrix form

$$\begin{aligned}\mathbf{P}_{\mathbf{u}}^{\ell} &= \begin{pmatrix} \alpha_{\mathbf{u},\ell}(t_1,\boldsymbol{\mu}_1) & \alpha_{\mathbf{u},\ell}(t_1,\boldsymbol{\mu}_2) & \cdots & \alpha_{\mathbf{u},\ell}(t_1,\boldsymbol{\mu}_{\mathcal{N}_p}) \\ \alpha_{\mathbf{u},\ell}(t_2,\boldsymbol{\mu}_1) & \alpha_{\mathbf{u},\ell}(t_2,\boldsymbol{\mu}_2) & \cdots & \alpha_{\mathbf{u},\ell}(t_2,\boldsymbol{\mu}_{\mathcal{N}_p}) \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{\mathbf{u},\ell}(t_{\mathcal{N}_t},\boldsymbol{\mu}_1) & \alpha_{\mathbf{u},\ell}(t_{\mathcal{N}_t},\boldsymbol{\mu}_2) & \cdots & \alpha_{\mathbf{u},\ell}(t_{\mathcal{N}_t},\boldsymbol{\mu}_{\mathcal{N}_p}) \end{pmatrix} \\ &= \left[\mathbf{v}_{\mathbf{u},\ell}^T\mathbf{S}_{\mathbf{u}}^1;\ \mathbf{v}_{\mathbf{u},\ell}^T\mathbf{S}_{\mathbf{u}}^2;\ \cdots;\ \mathbf{v}_{\mathbf{u},\ell}^T\mathbf{S}_{\mathbf{u}}^{\mathcal{N}_p}\right]^T \in \mathbb{R}^{\mathcal{N}_t\times\mathcal{N}_p},\end{aligned} \tag{4}$$

with $\ell = 1, 2, \cdots, d_{\mathbf{u}}, \mathbf{u} \in \{\mathbf{E}, \mathbf{H}\}$. Instead of dealing with matrix $\mathbf{P}_{\mathbf{u}}^{\ell}$ directly [20], [34], we concatenate the coefficient matrices to form a coefficient tensor $\mathcal{G}_{\mathbf{u}}$ of dimension $(\mathcal{N}_t, \mathcal{N}_p, d_{\mathbf{u}})$. The CP decomposition [17] illustrated in Fig. 1 factorizes $\mathcal{G}_{\mathbf{u}}$ into a sum of component rank-one tensors

$$\mathcal{G}_{\mathbf{u}} \approx \hat{\mathcal{G}}_{\mathbf{u}} = \sum_{r=1}^{R_{\mathbf{u}}} \boldsymbol{\varphi}_{\mathbf{u},r} \circ \boldsymbol{\psi}_{\mathbf{u},r} \circ \boldsymbol{\zeta}_{\mathbf{u},r}, \tag{5}$$

where $R_{\mathbf{u}}$ is a positive integer and determines the error of CP decomposition. The symbol $\circ$ represents the vector outer product and $\boldsymbol{\varphi}_{\mathbf{u},r} \in \mathbb{R}^{\mathcal{N}_t}, \boldsymbol{\psi}_{\mathbf{u},r} \in \mathbb{R}^{\mathcal{N}_p}$, and $\boldsymbol{\zeta}_{\mathbf{u},r} \in \mathbb{R}^{d_{\mathbf{u}}}$ for $r = 1, \ldots, R_{\mathbf{u}}$ are vectors. Our goal is to compute a CP

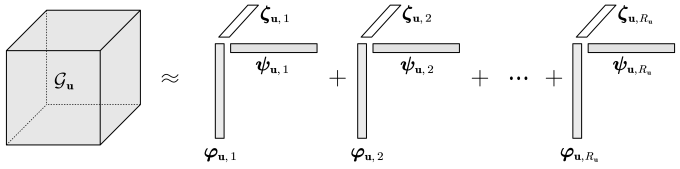


Fig. 1. CP decomposition of a three-order tensor $\mathcal{G}_{\mathbf{u}}$.

decomposition with $R_{\mathbf{u}}$ components that best approximates $\mathcal{G}_{\mathbf{u}}$, i.e., to find

$$\min_{\hat{\mathcal{G}}_{\mathbf{u}}} \|\mathcal{G}_{\mathbf{u}} - \hat{\mathcal{G}}_{\mathbf{u}}\| \quad \text{with} \quad \hat{\mathcal{G}}_{\mathbf{u}} = \sum_{r=1}^{R_{\mathbf{u}}} \boldsymbol{\varphi}_{\mathbf{u},r} \circ \boldsymbol{\psi}_{\mathbf{u},r} \circ \boldsymbol{\zeta}_{\mathbf{u},r}. \tag{6}$$

We adopt an alternating least squares (ALS) method [4], [12] restated in Algorithm 2 to find the CP decomposition in this work, where $\mathcal{G}_{\mathbf{u}(n)}$ is the mode-n unfolding of tensor $\mathcal{G}$, and notation $\odot$ and $\circledast$ denote the Khatri-Rao product and Hadamard product respectively. $\mathbf{A}^{\dagger}$ denotes the Moore-Penrose pseudoinverse of $\mathbf{A}$. Readers are referred to [17] for more details.

With the discrete data set, a cubic spline interpolation-based model $\widehat{\varpi}_{\mathbf{u}}$ is trained to approximate the continuous modes as

$$\begin{aligned} &t \longmapsto \widehat{\boldsymbol{\varphi}}_{\mathbf{u},r}(t) \text{ with } \{(t_i, (\boldsymbol{\varphi}_{\mathbf{u},r})_i), i = 1, 2, \cdots, \mathcal{N}_t\}, \\ &\boldsymbol{\mu} \longmapsto \widehat{\boldsymbol{\psi}}_{\mathbf{u},r}(\boldsymbol{\mu}) \text{ with } \{(\boldsymbol{\mu}_j, (\boldsymbol{\psi}_{\mathbf{u},r})_j), j = 1, 2, \cdots, \mathcal{N}_p\}, \end{aligned} \tag{7}$$

for $r = 1, 2, \cdots, R_{\mathbf{u}}$. Elementwise, we have

$$(\mathcal{G}_{\mathbf{u}})_{ijk} \approx (\hat{\mathcal{G}}_{\mathbf{u}})_{ijk} \approx \sum_{r=1}^{R_{\mathbf{u}}} \widehat{\boldsymbol{\varphi}}_{\mathbf{u},r}(t_i)\widehat{\boldsymbol{\psi}}_{\mathbf{u},r}(\boldsymbol{\mu}_j)(\boldsymbol{\zeta}_{\mathbf{u},r})_k, \tag{8}$$

with $1 \le i \le \mathcal{N}_t, 1 \le j \le \mathcal{N}_p, 1 \le k \le d_{\mathbf{u}}$. For a new time/parameter value $(t^*, \boldsymbol{\mu}^*)$, the RB solution is recovered as

$$\mathbf{u}_h^r(t^*, \boldsymbol{\mu}^*) = \mathbf{V}_{\mathbf{u}}\widehat{\boldsymbol{\alpha}}_{\mathbf{u}}(t^*, \boldsymbol{\mu}^*) = \mathbf{V}_{\mathbf{u}} \sum_{r=1}^{R_{\mathbf{u}}} \widehat{\boldsymbol{\varphi}}_{\mathbf{u},r}(t^*)\widehat{\boldsymbol{\psi}}_{\mathbf{u},r}(\boldsymbol{\mu}^*)\boldsymbol{\zeta}_{\mathbf{u},r}. \tag{9}$$

Algorithm 2: ALS method for CP decomposition of the projection coefficient tensor $\mathcal{G}_{\mathbf{u}}$

input : projection coefficient tensor $\mathcal{G}_{\mathbf{u}} \in \mathbb{R}^{\mathcal{N}_t\times\mathcal{N}_p\times d_{\mathbf{u}}}$ and a truncation parameter $R_{\mathbf{u}}$.
output: factor matrices $\Phi_{\mathbf{u}} = [\boldsymbol{\varphi}_{\mathbf{u},1}, \cdots, \boldsymbol{\varphi}_{\mathbf{u},R_{\mathbf{u}}}]$, $\Psi_{\mathbf{u}} = [\boldsymbol{\psi}_{\mathbf{u},1}, \cdots, \boldsymbol{\psi}_{\mathbf{u},R_{\mathbf{u}}}]$ and $\Sigma_{\mathbf{u}} = [\boldsymbol{\zeta}_{\mathbf{u},1}, \cdots, \boldsymbol{\zeta}_{\mathbf{u},R_{\mathbf{u}}}]$.

Initialize $\Phi_{\mathbf{u}} \in \mathbb{R}^{\mathcal{N}_t\times R_{\mathbf{u}}}$, $\Psi_{\mathbf{u}} \in \mathbb{R}^{\mathcal{N}_p\times R_{\mathbf{u}}}$ and $\Sigma_{\mathbf{u}} \in \mathbb{R}^{d_{\mathbf{u}}\times R_{\mathbf{u}}}$;
repeat
  $\Phi_{\mathbf{u}} = \mathcal{G}_{\mathbf{u}(1)}(\Sigma_{\mathbf{u}} \odot \Psi_{\mathbf{u}})(\Sigma_{\mathbf{u}}^T\Sigma_{\mathbf{u}} \circledast \Psi_{\mathbf{u}}^T\Psi_{\mathbf{u}})^{\dagger}$
  $\Psi_{\mathbf{u}} = \mathcal{G}_{\mathbf{u}(2)}(\Phi_{\mathbf{u}} \odot \Sigma_{\mathbf{u}})(\Phi_{\mathbf{u}}^T\Phi_{\mathbf{u}} \circledast \Sigma_{\mathbf{u}}^T\Sigma_{\mathbf{u}})^{\dagger}$
  $\Sigma_{\mathbf{u}} = \mathcal{G}_{\mathbf{u}(3)}(\Psi_{\mathbf{u}} \odot \Phi_{\mathbf{u}})(\Psi_{\mathbf{u}}^T\Psi_{\mathbf{u}} \circledast \Phi_{\mathbf{u}}^T\Phi_{\mathbf{u}})^{\dagger}$
until *fit cases to improve or maximum iterations exhausted*;

The procedure of POD-CP ROM is decoupled into offline/online phases, which is shown in Algorithm3, Fig. 2 and Fig. 3. Compared to previous works [20], [34],fewer interpolation or regression models are built in the offline stage, thus reducing the test time to evaluate the RB solution online. Specifically, for each EM field component, $2d_{\mathbf{u}}\sum_{\ell=1}^{d_{\mathbf{u}}} Q_{\mathbf{u},\ell}$ interpolation-based models are built in [20], where $Q_{\mathbf{u},\ell}$ denotes the number of singular values truncated of coefficient matrix $\mathbf{P}_{\mathbf{u}}^{\ell}$, while only $2R_{\mathbf{u}}$ interpolation-based models are built in POD-CP ROM.

The recovery error the POD-CPD-CSI ROM at $(t^*, \boldsymbol{\mu}^*)$ will be written as

$$\begin{aligned} \varepsilon &= \|\mathbf{u}_h(t^*, \mu^*) - \mathbf{u}_h^r(t^*, \mu^*)\|^2 \\ &= \|\mathbf{u}_h(t^*, \mu^*) - \mathbf{V}_{\mathbf{u}}\widehat{\alpha}_{\mathbf{u}}(t^*, \mu^*)\|^2 \\ &= ||\mathbf{u}_h(t^*, \mu^*) - \mathbf{V}_{\mathbf{u}}\mathbf{V}_{\mathbf{u}}^T\mathbf{u}_h(t^*, \mu^*) + \mathbf{V}_{\mathbf{u}}\mathbf{V}_{\mathbf{u}}^T\mathbf{u}_h(t^*, \mu^*) \\ &\qquad - \mathbf{V}_{\mathbf{u}}\widehat{\alpha}_{\mathbf{u}}(t^*, \mu^*)||^2 \\ &\leqslant \varepsilon_{\mathrm{POD}}^2 + \|\mathbf{V}_{\mathbf{u}}\| \left\|\mathbf{V}_{\mathbf{u}}^T\mathbf{u}_h(t^*, \mu^*) - \widehat{\alpha}_{\mathbf{u}}(t^*, \mu^*)\right\|^2 \\ &= \varepsilon_{\mathrm{POD}}^2 + \sum_{i=1}^{\mathcal{N}} \left(\alpha_{\mathbf{u},i}(t^*, \mu^*) - \widehat{\alpha}_{\mathbf{u},i}(t^*, \mu^*)\right)^2 \\ &= \varepsilon_{\mathrm{POD}}^2 + \varepsilon_{\mathrm{coe}}^2, \end{aligned} \tag{10}$$

where $\varepsilon_{\mathrm{POD}}^2$ is the truncation error of the two-step POD. And $\varepsilon_{\mathrm{coe}}^2$ denotes the error of predicting the projection coefficients, which includes the error of CP decomposition and the CSI error of $\widehat{\boldsymbol{\varphi}}_{\mathbf{u},r}(t^*)$ and $\widehat{\boldsymbol{\psi}}_{\mathbf{u},r}(\boldsymbol{\mu}^*)$.

IV. Numerical results

In this section, we present four numerical tests to evaluate the performance of the proposed POD-CPD-CSI ROM. Moreover we compare the proposed ROM with the POD-ANN ROM [31] and POD-CSI ROM [20] both in time-efficiency and errors. We consider non-magnetic materials ($\nu_r = 1$) in a vacuum for all numerical tests. The norm of a tensor $\mathcal{G}$ is defined as

$$\|\mathcal{G}\| = \sqrt{\sum_{i}^{\mathcal{N}_t}\sum_{j}^{\mathcal{N}_p}\sum_{k}^{d_{\mathbf{u}}}[(\mathcal{G})_{ijk}]^2}. \tag{11}$$

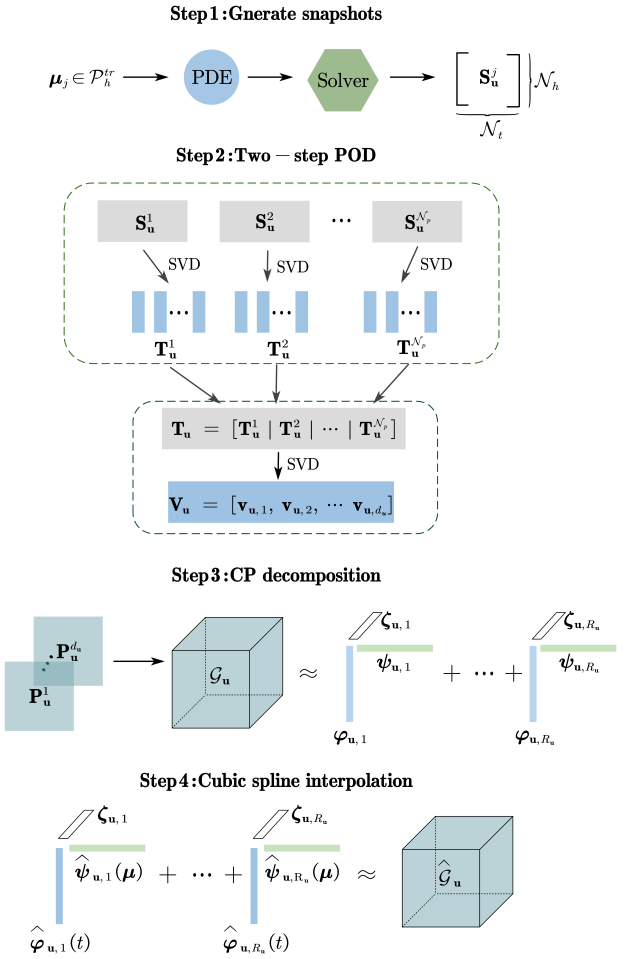


Fig. 2. Offline satge of the proposed POD-CPD-CSI ROM.

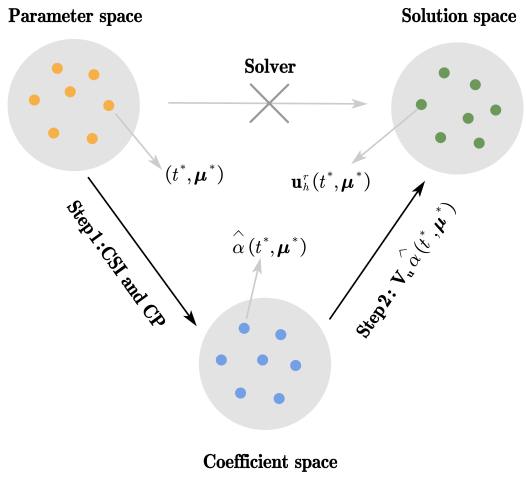


Fig. 3. Online stage of the proposed POD-CPD-CSI ROM.

Algorithm 3: POD-CPD-CSI ROM

Function $[\mathbf{V}_{\mathbf{u}}, \widehat{\varpi}_{\mathbf{u}}]$=`Offline`$(\mathcal{P}, \mathcal{T})$
  Prepare a training parameter sampling $\mathcal{P}_h^{tr} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_{\mathcal{N}_p}\} \subset \mathcal{P}$;
  Compute the high-fidelity solutions $\mathbf{u}_h(t_i, \boldsymbol{\mu}_j)$ in the time domain $\mathcal{T}_h \subset \mathcal{T}$ for each physical parameter $\boldsymbol{\mu}_j$, $j = 1, 2, \cdots, \mathcal{N}_p$;
  Prepare a training time sampling $\mathcal{T}_h^{tr} = \{t_1, t_2, \cdots, t_{\mathcal{N}_t}\} \subseteq \mathcal{T}_h \subset \mathcal{T}$ and assemble a snapshot matrix $\mathbf{S}_{\mathbf{u}}$;
  Compute the POD basis matrix $\mathbf{V}_{\mathbf{u}}$ via the two-step POD method in Algorithm 1;
  Perform CP decomposition of the projection coefficient tensor $\mathcal{G}_{\mathbf{u}}$ via the ALS method in Algorithm 2;
  Train the CSI-based model $\widehat{\varpi}_{\mathbf{u}}$ based on 7.
end
Function $[\mathbf{u}_h^r(t^*, \boldsymbol{\mu}^*)]$ =`Online` $(\mathbf{V}_{\mathbf{u}}, \widehat{\varpi}_{\mathbf{u}}, (t^*, \boldsymbol{\mu}^*))$
  Calculate the projection coefficient $\widehat{\alpha}_{\mathbf{u}}(t^*, \boldsymbol{\mu}^*)$ through CSI-based model $\widehat{\varpi}_{\mathbf{u}}$;
  Evaluate the RB solution $\mathbf{u}_h^r(t^*, \boldsymbol{\mu}^*)$ via formula 9.
end

We determine the truncation parameter $R_{\mathbf{u}}$ of the CP decomposition by the relative error indicator $e_{\mathbf{u},\mathrm{CP}} = \|\mathcal{G}_{\mathbf{u}} - \hat{\mathcal{G}}_{\mathbf{u}}\| / \|\mathcal{G}_{\mathbf{u}}\|$. In the online phase, for a new time value $t \in \mathcal{T}_h^{te}$ and physical parameter value $\boldsymbol{\mu} \in \mathcal{P}_h^{te}$, the relative projection error

$$e_{\mathbf{u},\mathrm{Pro}}(t, \boldsymbol{\mu}) = \frac{\left\|\mathbf{u}_h(t, \boldsymbol{\mu}) - \mathbf{V}_{\mathbf{u}}\mathbf{V}_{\mathbf{u}}^T\mathbf{u}_h(t, \boldsymbol{\mu})\right\|_2}{\|\mathbf{u}_h(t, \boldsymbol{\mu})\|_2}, \tag{12}$$

provides a lower bound for the relative error of the POD-CPD-CSI ROM

$$e_{\mathbf{u},\mathrm{POD-CPD-CSI}}(t, \boldsymbol{\mu}) = \frac{\|\mathbf{u}_h(t, \boldsymbol{\mu}) - \mathbf{V}_{\mathbf{u}}\widehat{\alpha}_{\mathbf{u}}(t, \boldsymbol{\mu})\|_2}{\|\mathbf{u}_h(t, \boldsymbol{\mu})\|_2}. \tag{13}$$

Furthermore, the average relative errors on a test set $\mathcal{T}_h^{te} \times \mathcal{P}_h^{te}$ of size $\mathcal{N}_{te}$ are calculated to measure the accuracy of the POD-CPD-CSI ROM, which are defined as

$$\bar{e}_{\mathbf{u},\mathrm{POD-CPD-CSI}} = \frac{\sum_{(t,\boldsymbol{\mu}) \in \mathcal{T}_h^{te} \times \mathcal{P}_h^{te}} e_{\mathbf{u},\mathrm{POD-CPD-CSI}}(t, \boldsymbol{\mu})}{\mathcal{N}_{te}}, \tag{14}$$

$$\bar{e}_{\mathbf{u},\mathrm{Pro}} = \frac{\sum_{(t,\boldsymbol{\mu}) \in \mathcal{T}_h^{te} \times \mathcal{P}_h^{te}} e_{\mathbf{u},\mathrm{Pro}}(t, \boldsymbol{\mu})}{\mathcal{N}_{te}}. \tag{15}$$

We use the function `cp_als` in the MATLAB library `Tensor Toolbox` [2] to decompose the coefficient tensor $\mathcal{G}_{\mathbf{u}}$. The CSI-based approximation of the projection coefficients is built via the MATLAB functions `spline` and `griddedInterpolant` for single-layer cases and multi-layer cases respectively. We refer to [13], [32] for more details of the multivariate CSI method. The architecture of ANN used in the POD-ANN ROM is listed in Table VI. All simulations are run on a desktop with an Intel Core i9 10-core 2.8 GHz $\times$ 20 CPU and 64 GB RAM.

TABLE I
CONFIGURATION OF THE NUMERICAL TESTS.

Case	Relative permittivity $\boldsymbol{\mu}_i$ of each layer				Radius of each layer (m)			
	1	2	3	4	1	2	3	4
disk	[1,5]	–	–	–	0.6	–	–	–
multi-layer disk	[5.0, 5.6]	[3.25, 3.75]	[2.0, 2.5]	[1.25,1.75]	0.15	0.3	0.45	0.6
sphere	[2,6]	–	–	–	1.0	–	–	–
multi-layer sphere	[4.0,4.5]	[8.0, 8.5]	[2.0, 2.5]	–	0.6	1.0	1.5	–

TABLE II
PARAMETERS SETTING IN THE TWO-STEP POD METHOD.

case	ρ_t	$\rho_{\boldsymbol{\mu}}$	d_{H_x}	d_{H_y}	d_{H_z}	d_{E_x}	d_{E_y}	d_{E_z}
disk	0.1	1e-5	20	35	–	–	–	22
multi-layer disk	0.1	1e-5	17	15	–	–	–	15
sphere	0.1	1e-6	45	44	39	43	37	36
multi-layer sphere	0.1	1e-6	48	47	43	45	44	40

TABLE III
PARAMETERS SETTING IN THE CP DECOMPOSITION OF POD-CPD-CSI ROM.

case	R_{H_x}	R_{H_y}	R_{H_z}	R_{E_x}	R_{E_y}	R_{E_z}
disk	50	50	–	–	–	50
multi-layer disk	60	60	–	–	–	60
sphere	60	60	60	60	60	60
multi-layer sphere	50	50	50	50	50	50

TABLE IV
NUMBER OF CSI MODELS BUILT IN THREE ROMS.

case	POD-ANN	POD-CSI	POD-CPD-CSI
disk	-	1330	300
multi-layer disk	-	819	360
sphere	-	2196	720
multi-layer sphere	-	2423	600

TABLE V
SIZE OF TRAINING AND TEST DATA SETS.

Case	Training data set		Test data set	
	$\mathcal{N}_p$	$\mathcal{N}_t$	$\mathcal{N}_p$	$\mathcal{N}_t$
disk	81	263	4	263
multi-layer disk	3^4	253	4	253
sphere	81	100	4	100
multi-layer sphere	5^3	100	4	100

TABLE VI
ARCHITECTURE OF THE ANN USED FOR THE POD-ANN ROM.

Case	Layer						
	i	h_1	h_2	h_3	h_4	h_5	o
disk	2	20	40	60	-	-	77
multi-layer disk	5	10	20	30	-	-	47
sphere	2	50	55	60	65	70	242
multi-layer sphere	4	50	55	60	65	70	267

A. Scattering of a plane wave by a dielectric disk

First, we consider the scattering problem described by the 2D time-domain TM waves in the normalized form

$$\begin{cases} \nu_r \dfrac{\partial \mathbf{H}}{\partial t} + \operatorname{curl} E = 0, \\ \varepsilon_r \dfrac{\partial E}{\partial t} - \operatorname{curl} \mathbf{H} = 0, \end{cases} \tag{16}$$

where the magnetic field $\mathbf{H} = (H_x, H_y)^T$ and the electric field $E = E_z$. An incident plane wave takes the form $H_x^{\text{inc}}(x,y,t) = 0$, $H_y^{\text{inc}}(x,y,t) = -\cos(\omega t - kx)$ and $E_z^{\text{inc}}(x,y,t) = \cos(\omega t - kx)$ with a frequency $f = 300$ MHz [34]. The computational domain is a square $\Omega = [-2.6\lambda_0, 2.6\lambda_0]^2$ and the radius of the inner disk is 0.6 m (background wavelength $\lambda_0 = 1$m). The range of the relative permittivity ε_r is presented in TableI, i.e., the parameter domain we are interested in is $\mathcal{P} = \{\boldsymbol{\mu} = \varepsilon_r \in [1,5]\}$. To determine the number of sampling parameter points in $\mathcal{P}$, we create three training parameter sets uniformly with different sampling intervals $\Delta p = 1, 0.5$ and 0.05, resulting in $\mathcal{N}_p = 5, 9$ and 81 points respectively. A DGTD solver [19] is used to generate a collection of high-fidelity solutions of the three training parameter sets with $\mathcal{N}_t = 263$ samples in the last period $\mathcal{T}_h^{tr} = \{49.0024, 49.006, \cdots, 49.966\}$. We evaluate the proposed method on a test parameter set $\mathcal{P}_h^{te} = \{1.215, 2.215, 3.215, 4.215\}$ and a test time set $\mathcal{T}_h^{te} = \mathcal{T}_h^{tr}$. Note that the truncation parameter $R_{\mathbf{u}}$ is selected carefully such that the projection coefficient tensor $\mathcal{G}_{\mathbf{u}}$ lose little information in the CP decomposition. Fig. 4 shows the average relative errors $\bar{e}_{\mathbf{u},\text{Pro}}$ and $\bar{e}_{\mathbf{u},\text{POD}-\text{CPD}-\text{CSI}}$ on the training set $\mathcal{P}_h^{tr} \times \mathcal{T}_h^{tr}$ with a fixed truncation parameter $\rho_t = 0.1$. The performance of the POD-CP ROM mainly depends on the size of training set as well as the truncation parameter $\rho_{\boldsymbol{\mu}}$ used. So the training parameter set $\mathcal{P}_h^{tr} = \{1, 1.05, \cdots, 5\}$ is adopted in the following discussion.

The convergence curves of the average relative errors $\bar{e}_{\mathbf{H},\text{Pro}}$ and $\bar{e}_{\mathbf{E},\text{Pro}}$ are plotted in Fig. 5, from which we can see that the tolerance ρ_t has little impact on the performance of the two-step POD method. By decreasing the tolerance $\rho_{\boldsymbol{\mu}}$ from 1×10^{-1} to 1×10^{-5}, there is a significant improvement of the two-step POD performance. Therefore, we set $\rho_t = 1\times10^{-1}$ and $\rho_{\boldsymbol{\mu}} = 1\times10^{-5}$. The number of POD basis functions $d_{\mathbf{u}}$ generated by the two-step POD method are listed in Table**??**. The relative errors $e_{\mathbf{u},\text{CP}}$ for different truncation parameters $R_{\mathbf{u}}$ adopted in the CP decomposition are plotted in Fig. 6.

TABLE VII
AVERAGE RELATIVE ERRORS OF THREE ROMS.

Case	Projection		POD-ANN		POD-CSI		POD-CPD-CSI	
	$\bar{e}_{\mathbf{H}}$	$\bar{e}_{\mathbf{E}}$	$\bar{e}_{\mathbf{H}}$	$\bar{e}_{\mathbf{E}}$	$\bar{e}_{\mathbf{H}}$	$\bar{e}_{\mathbf{E}}$	$\bar{e}_{\mathbf{H}}$	$\bar{e}_{\mathbf{E}}$
disk	1.11e-2	1.36e-2	3.24e-2	4.27e-2	1.16e-2	1.36e-2	1.19e-2	1.36e-2
multi-layer disk	7.37e-3	8.23e-3	2.60e-2	3.03e-2	8.01e-3	8.26e-3	8.03e-3	8.86e-3
sphere	6.64e-4	1.44e-3	3.97e-2	3.21e-2	6.73e-4	1.49e-3	6.89e-4	1.50e-3
multi-layer sphere	7.34e-4	4.02e-3	3.10e-2	4.03e-2	7.50e-4	4.12e-3	7.58e-4	4.15e-3

TABLE VIII
COMPARISON OF THE CPU TIME OF FOM AND THREE ROMS. THE UNIT OF TIME IS SECOND.

Case	FOM		POD-ANN		POD-CSI		POD-CPD-CSI	
	DoFs	CPU time	training	test	training	test	training	test
disk	30264	258	32450	0.02	9	0.25	19	0.04
multi-layer disk	37236	305	31784	0.03	7	0.19	16	0.05
sphere	91125	343	43287	0.03	23	0.34	35	0.07
multi-layer sphere	91125	3710	39749	0.03	27	0.39	38	0.06

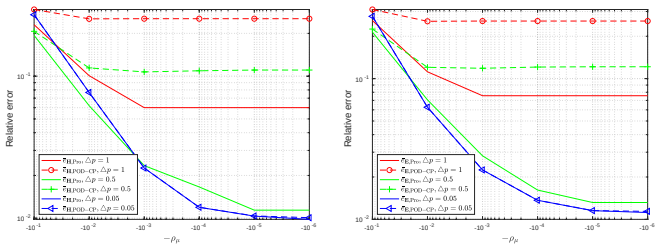


Fig. 4. The average relative errors of $\bar{e}_{\mathbf{H},\mathrm{Pro}}$ and $\bar{e}_{\mathbf{H},\mathrm{POD-CP}}$ (top), $\bar{e}_{\mathbf{E},\mathrm{Pro}}$ and $\bar{e}_{\mathbf{E},\mathrm{POD-CP}}$(bottom) on the training set with different sampling intervals Δp and truncation parameters $\rho_{\boldsymbol{\mu}}$.

We take $R_{H_x} = R_{H_y} = R_{E_z} = 50$ to minimize the information lost in CP decomposition while constructing as few interpolation models as possible.

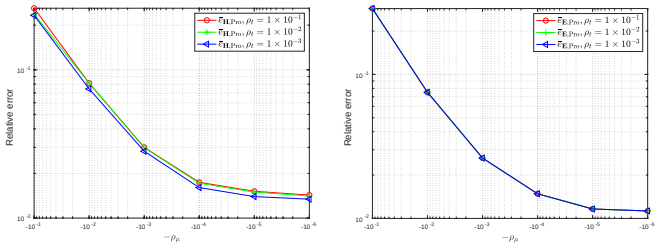


Fig. 5. The average relative errors of $\bar{e}_{\mathbf{H},\mathrm{Pro}}$ (top) and $\bar{e}_{\mathbf{E},\mathrm{Pro}}$ (bottom) with different truncation parameters ρ_t and $\rho_{\boldsymbol{\mu}}$.

For the parameter instance $\boldsymbol{\mu}_3 = 3.215$, comparison of the time evolution of the field H_y and E_z at a fixed point is displayed in Fig. 7. Moreover, the total field of $|H_x|$ calculated by DGTD and the absolute errors between POD-ANN, POD-CSI, POD-CPD-CSI and DGTD respectivily are shown in Fig. 8. The time evolution of the relative error $e_{\mathbf{u},\mathrm{Pro}}$ and $e_{\mathbf{u},\mathrm{POD-ANN}}$, $e_{\mathbf{u},\mathrm{POD-CSI}}$ and $e_{\mathbf{u},\mathrm{POD-CPD-CSI}}$ is plotted in

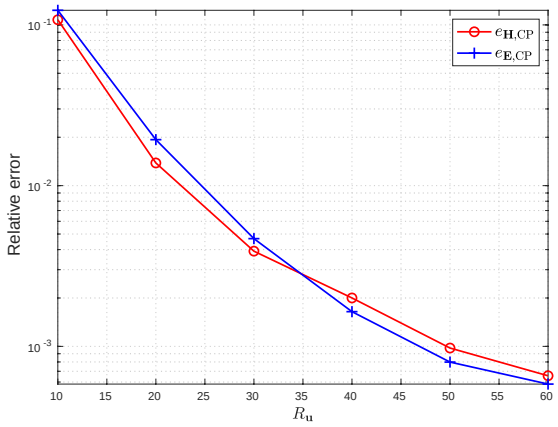


Fig. 6. The relative errors $\bar{e}_{\mathbf{u},\mathrm{CP}}$ with different truncation parameters $R_{\mathbf{u}}$.

Fig. 9. The average projection error $\bar{e}_{\mathbf{u},\mathrm{Pro}}$ and three ROMs are listed in Table VII. It can be seen that the accuracy of POD-CSI and POD-CP-CSI are similar, and both are better than POD-ANN.

In addition, Table VIII shows the average computing time of a time trajectory matrix (1) between DGTD and three ROMs. The online test time of the ROM achieves 5520 times faster than that of the DGTD solver, which demonstrates the high efficiency of the proposed method. The non-intrusive ROMs features an offline-online framework, i.e., offline training is completely decoupled from online testing. Given a new parameter, the RB solution is rapidly recovered without an access to the high-dimensional FOM, which greatly improves the efficiency of the ROM. In addition, POD-CSI and POD-CPD-CSI take less training time offline compared with POD-ANN [31]. Finally, as shown in Table IV, POD-CPD-CSI generates

less CSI models and takes less test time compared to POD-CSI.

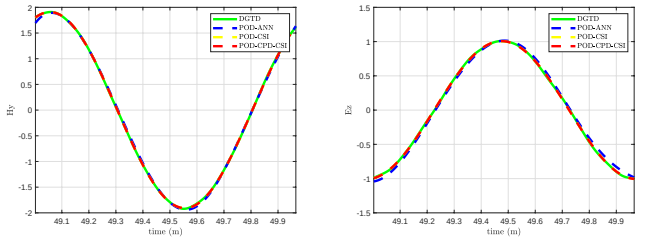


Fig. 7. Comparison of the time evolution of the field H_y (top) and E_z (bottom) at a fixed point.

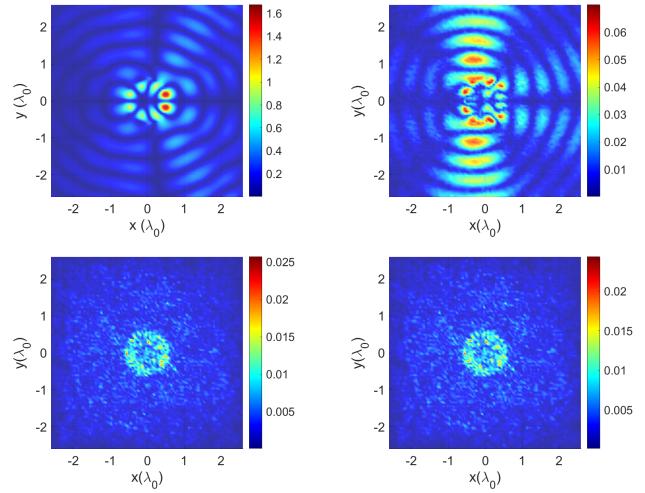


Fig. 8. The DGTD solution of $|H_x|$ (a) at $t = 49.966$ m and the absolute errors between POD-ANN (b), POD-CSI (c), POD-CPD-CSI (d) and DGTD respectively.

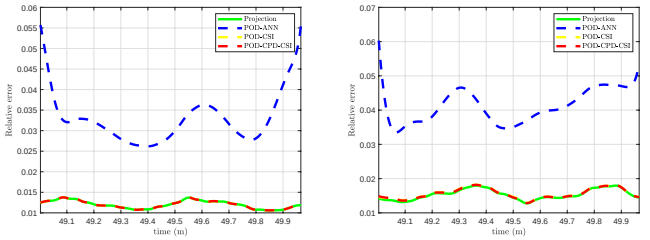


Fig. 9. Comparison of the time evolution of the relative errors $e_{\mathbf{u},\mathrm{Pro}}$, $e_{\mathbf{u},\mathrm{POD-ANN}}$, $e_{\mathbf{u},\mathrm{POD-CSI}}$ and $e_{\mathbf{u},\mathrm{POD-CPD-CSI}}$ for the field $\mathbf{H}$ and $\mathbf{E}$ with the testing-parameter instances $\varepsilon_3 = 3.215$.

B. Scattering of a plane wave by a multi-layer disk

Next, we consider the case of a multi-layer dielectric disk as shown in Fig. 11. A square $\Omega = [-3.2\lambda_0, 3.2\lambda_0]^2$ is the computational domain, and the radius and the range of material parameters of each layer are presented in Table I. The excitation is the same as the case in IV-A.

To determine the number of sampling parameter points in $\mathcal{P} = \mathcal{P}^1 \times \mathcal{P}^2 \times \mathcal{P}^3 \times \mathcal{P}^4$ with $\boldsymbol{\mu} = (\varepsilon_{r,1}, \varepsilon_{r,2}, \varepsilon_{r,3}, \varepsilon_{r,4})$ and $\varepsilon_{r,i} \in \mathcal{P}^i$, we create two training parameter sets uniformly with $\mathcal{N}_{\Delta p} = 2$ and 3 in each layer, resulting in $\mathcal{N}_p = 16$ and 81 points in total. For each parameter point, $\mathcal{N}_t = 253$ snapshots are calculated in time at the last period ($\triangle t = 0.0038$ m and $\mathcal{T}_h^{tr} = \{49.0002, 49.0041, \cdots, 49.9669\}$). We evaluate the proposed method on a test set $\mathcal{P}_h^{te} \times \mathcal{T}_h^{te}$ where $\mathcal{T}_h^{te} = \mathcal{T}_h^{tr}$ and $\mathcal{P}_h^{te} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \boldsymbol{\mu}_4\}$ with $\boldsymbol{\mu}_1 = \{(5.1, 3.3, 2.2, 1.3)\}$, $\boldsymbol{\mu}_2 = \{(5.3, 3.5, 2.2, 1.5)\}$, $\boldsymbol{\mu}_3 = \{(5.5, 3.4, 2.5, 1.7)\}$ and $\boldsymbol{\mu}_4 = \{(5.6, 3.7, 2.4, 1.6)\}$.

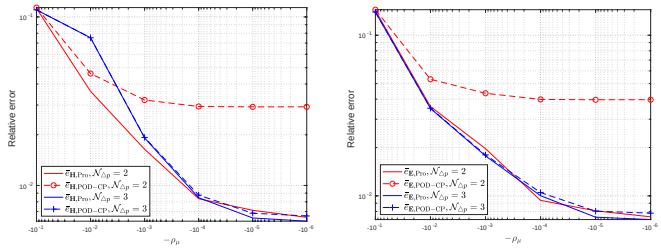


Fig. 10. The average relative errors of $\bar{e}_{\mathbf{H},\mathrm{Pro}}$ and $\bar{e}_{\mathbf{H},\mathrm{POD-CPD-CSI}}$ (top), $\bar{e}_{\mathbf{E},\mathrm{Pro}}$ and $\bar{e}_{\mathbf{E},\mathrm{POD-CPd-csi}}$(bottom) on the training set with different number of sampling points $\mathcal{N}_{\Delta p}$ and truncation parameters $\rho_{\boldsymbol{\mu}}$.

The average relative errors $\bar{e}_{\mathbf{u},\mathrm{Pro}}$ and $\bar{e}_{\mathbf{u},\mathrm{POD-CPD-CSI}}$ on the training set $\mathcal{P}_h^{tr} \times \mathcal{T}_h^{tr}$ with a fixed truncation parameter $\rho_t = 0.1$ are shown in Fig. 10, therefore we select $\mathcal{N}_{\Delta p} = 3$ parameter points uniformly in each layer. The parameter setting in the POD-CPD-CSI ROM is listed in TableIII.

The average computing time of a time trajectory matrix and average relative errors are presented in TableVIII and TableVII respectively.

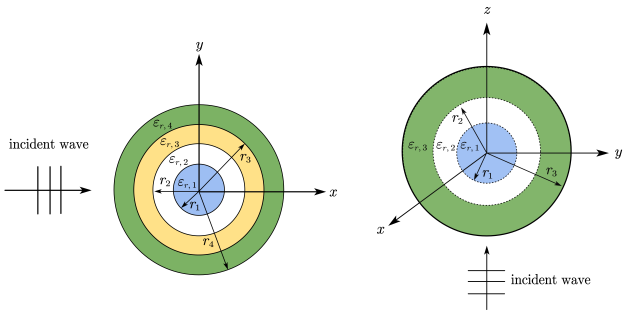


Fig. 11. Configuration and scattering of a plane wave by a multi-layer disk (top) and multi-layer sphere (bottom).

C. Scattering of a plane wave by a dielectric sphere

We consider a plane wave that is polarized in the x direction, propagates in the $+z$ direction, and is scattered by a dielectric sphere [9]. The plane wave is expressed as

$$\mathbf{E}^{\mathrm{inc}} = e^{-jkz}\boldsymbol{e}_x, \ \mathbf{H}^{\mathrm{inc}} = \frac{E_x^{\mathrm{inc}}}{\eta} e^{-jkz}\boldsymbol{e}_y, \tag{17}$$

where the magnitude of the electric part is 1 V/m with a frequency $f = 300$ MHz. The computational domain is a cube $\Omega = [-0.2\lambda_0, 2.2\lambda_0]^3$ with $\lambda_0 = 1$ m, and the range of material parameters is listed in Table I. During the offline phase, high-fidelity solutions are calculated by the Mie series method [9] on $\mathcal{N}_h = 91125$ nodes at $\mathcal{N}_p = 81$ uniformly distributed parameter points ($\mathcal{P}_h^{tr} = \{2, 2.05, \cdots, 5.95, 6\}$) with

$\mathcal{N}_t = 100$ points in a period. We evaluate the proposed method on a test parameter set $\mathcal{P}_h^{te} = \{2.215, 3.215, 4.215, 5.215\}$ and test time set $\mathcal{T}_h^{te} = \mathcal{T}_h^{tr} = \{0, 0.0101, \cdots, 1\}$. The number of basis functions and truncation tolerances of the two-step POD and CP decomposition are presented in TableIII.

The average computing time and average relative errors are presented in TableVIII and TableVII respectively.

D. Scattering of a plane wave by a multi-layer sphere

Finally, we consider a plane wave scattered by a 3-layer dielectric sphere [9] as shown in Fig. 11. The computational domain is $\Omega = [-0.2\lambda_0, 2.2\lambda_0]^3$, and the radius and range of material parameters of each layer are listed in Table I. The excitation is the same as the case in IV-C. In the offline phase, full-order solutions are calculated by the Mie series method on $\mathcal{N}_h = 91125$ nodes at $\mathcal{N}_p = 125$ uniformly distributed parameter points. A test parameter set $\mathcal{P}_h^{te} = \{(2.1, 8.4, 4.2), (2.2, 8.3, 4.4), (2.3, 8.2, 4.1), (2.4, 8.1, 4.3)\}$ and test time set $\mathcal{T}_h^{te} = \mathcal{T}_h^{tr} = \{0, 0.0101, \cdots, 1\}$ are used to evaluate the proposed method. The number of basis functions and truncation tolerances of two-step POD and CP decomposition are presented in TableIII.

The computing time and average relative errors are presented in TableVIII and TableVII respectively.

V. Conclusion

A data-driven RB method based on POD, CP decomposition and CSI is proposed to accelerate the solution of parameterized time-domain Maxwell equations. RB solutions can be obtained at a very low cost from the online phase. We verify the performance of the proposed POD-CP ROM with scattering by a medium disk and a dielectric sphere. As for scattering problems with complex geometric objects, more POD basis functions generated by the two-step POD might be required to capture the main dynamics of the full-order system.

We remark that although data-driven ROMs are effective and easy to implement when sufficient training data (experimental or computational data) is available, it may be still expensive to generate relatively large data in many application fields. In addition, data-driven ROMs fit observed data well, but generally have poor generalization performance for unseen data. A possible approach to address these issues is to take physical information into account. In fact, the governing partial differential equations (PDEs) behind the training data are clearly known, but often ignored by data-driven ROMs. Future research direction is to combine the data and physical information to improve the generalization capacity of ROMs when training data is limited.

Acknowledgement

The last author is supported by NSFC, 12101511 and Fundamental Research Funds for the Central Universities (Grant No. JBK22YJ37).

References

[1] Francesco Ballarin, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Supremizer stabilization of pod–galerkin approximation of parametrized steady incompressible navier–stokes equations. *International Journal for Numerical Methods in Engineering*, 102(5):1136–1161, 2015.

[2] Brett W. Bader, Tamara G. Kolda and others. Tensor Toolbox for MATLAB. www.tensortoolbox.org, September 21, 2022.

[3] T Bui-Thanh, Murali Damodaran, and Karen Willcox. Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics. In *21st AIAA Applied Aerodynamics Conference*, page 4213, 2003.

[4] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

[5] Fabien Casenave, Alexandre Ern, and Tony Lelievre. A nonintrusive reduced basis method applied to aeroacoustic simulations. *Advances in Computational Mathematics*, 41(5):961–986, 2015.

[6] Wenqian Chen, Qian Wang, Jan S Hesthaven, and Chuhua Zhang. Physics-informed machine learning for reduced-order modeling of nonlinear problems. *Journal of Computational Physics*, 446:110666, 2021.

[7] M Couplet, C Basdevant, and P Sagaut. Calibrated reduced-order pod-galerkin system for fluid flow modelling. *Journal of Computational Physics*, 207(1):192–220, 2005.

[8] Stefania Fresca, Luca Dede', and Andrea Manzoni. A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized pdes. *Journal of Scientific Computing*, 87:1–36, 2021.

[9] G. Kevin Zhu. Sphere scattering. https://www.mathworks.com/matlabcentral/fileexchange/31119-sphere-scattering, 2022.

[10] Francisco J Gonzalez and Maciej Balajewicz. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. *arXiv preprint arXiv:1808.01346*, 2018.

[11] Mengwu Guo and Jan S Hesthaven. Data-driven reduced order modeling for time-dependent problems. *Computer methods in applied mechanics and engineering*, 345:75–99, 2019.

[12] Richard A Harshman et al. Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis. 1970.

[13] Md Sakib Hasan, Syed K Islam, and Benjamin J Blalock. Modeling of soi four-gate transistor (g4fet) using multidimensional spline interpolation method. *Microelectronics journal*, 76:33–42, 2018.

[14] Jan S Hesthaven, Gianluigi Rozza, Benjamin Stamm, et al. *Certified reduced basis methods for parametrized partial differential equations*, volume 590. Springer, 2016.

[15] Jan S Hesthaven and Stefano Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.

[16] Jian-Ming Jin. *Theory and computation of electromagnetic fields*. John Wiley & Sons, 2011.

[17] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[18] Ettore Lappano, Frank Naets, Wim Desmet, Domenico Mundo, and Eugene Nijman. A greedy sampling approach for the projection basis construction in parametric model order reduction for structural dynamics models. In *Proceedings of the 27th international conference on noise and vibration engineering.*, pages 3563–3571. KU Leuven; Leuven, Belgium, 2016.

[19] Kun Li, Ting-Zhu Huang, Liang Li, and Stéphane Lanteri. Pod-based model order reduction with an adaptive snapshot selection for a discontinuous galerkin approximation of the time-domain maxwell's equations. *Journal of Computational Physics*, 396:106–128, 2019.

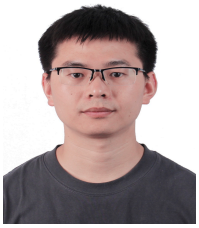
[20] Kun Li, Ting-Zhu Huang, Liang Li, and Stéphane Lanteri. Non-intrusive reduced-order modeling of parameterized electromagnetic scattering problems using cubic spline interpolation. *J. Sci. Comput.*, 87(2):1–29, 2021.

[21] Kun Li, Ting-Zhu Huang, Liang Li, Stéphane Lanteri, Li Xu, and Bin Li. A reduced-order discontinuous galerkin method based on pod for electromagnetic simulation. *IEEE Transactions on Antennas and Propagation*, 66(1):242–254, 2017.

[22] YC Liang, HP Lee, SP Lim, WZ Lin, KH Lee, and CG1237 Wu. Proper orthogonal decomposition and its applications—part i: Theory. *Journal of Sound and vibration*, 252(3):527–544, 2002.

[23] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

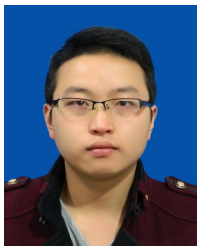
[24] Takaaki Murata, Kai Fukami, and Koji Fukagata. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *Journal of Fluid Mechanics*, 882:A13, 2020.
[25] VVS Prakash, , and Raj Mittra. Characteristic basis function method: A new technique for efficient solution of method of moments matrix equations. *Microwave and optical technology letters*, 36(2):95–100, 2003.
[26] Alfio Quarteroni, Gianluigi Rozza, and Andrea Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1:1–49, 2011.
[27] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
[28] Sadasiva Rao, Donald Wilton, and Allen Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Transactions on antennas and propagation*, 30(3):409–418, 1982.
[29] Jiming Song, Cai-Cheng Lu, and Weng Cho Chew. Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects. *IEEE transactions on antennas and propagation*, 45(10):1488–1493, 1997.
[30] Eric Suter and Juan R Mosig. A subdomain multilevel approach for the efficient mom analysis of large planar antennas. *Microwave and Optical Technology Letters*, 26(4):270–277, 2000.
[31] Qian Wang, Jan S Hesthaven, and Deep Ray. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. *Journal of computational physics*, 384:289–307, 2019.
[32] Ren-Hong Wang. *Multivariate spline functions and their applications*, volume 529. Springer Science & Business Media, 2013.
[33] D Xiao, F Fang, CC Pain, and IM Navon. A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. *Computer Methods in Applied Mechanics and Engineering*, 317:868–889, 2017.
[34] Ying Zhao, Liang Li, and Kun Li. Non-intrusive reduced order modeling of parametric electromagnetic scattering problems through gaussian process regression. *arXiv preprint arXiv:2103.12472*, 2021.



Xiao-Feng He received the B.S. degree in information and computational science from Sichuan Agricultural University, Yaan, China, in 2020, and the M.S. degree in computational mathematics from University of Electronic Science and Technology of China, Chengdu, China, in 2023. His research interests include data-driven reduced-order modeling and numerical solvers in computational electromagnetics.

Liang Li received the B.S., M.Sc., and Ph.D. degrees in computational mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China, in 2004, 2007, and 2009, respectively. During 2010, he was a Visiting Scholar with the Université Libre de Bruxelles, Bruxelles, Belgium. From 2010 to 2012, he was a Postdoctoral Fellow with Inria Sophia Antipolis-Méditerranée research center, France. He is currently a full professor with the School of Mathematical Sciences, UESTC. His current research interests include iterative methods and preconditioning techniques for solving large linear systems, hybridizable discontinuous Galerkin methods, domain decomposition, model order reduction techniques, (algebraic) multigrid methods, and with applications to electromagnetics and nanophotonics.

Stéphane Lanteri is a senior research scientist at Inria center of Université Côte d'Azur. Sophia He holds a PhD in Engineering Sciences from University of Nice-Sophia Antipolis (Year of defense: 1991) and a HDR (habilitation to advise doctoral theses) from the University of Nice/Sophia Antipolis (Year of defense: 2003 - Title: High-performance numerical methods on unstructured meshes with applications to compressible fluid dynamics). From May 1992 to October 1993, he was a postdoctoral fellow at the Center for Aerospace Structure, University of Colorado at Boulder, under the supervision of Charbel Farhat. His current research interests are concerned with numerical modeling of physical problems related to computational electromagnetics and computational nanophotonics: design of high order discontinuous Galerkin type approximation methods on unstructured meshes and of parallel numerical algorithms for solving differential systems modeling electromagnetic wave interaction with complex media with a focus of nanoscale light-matter interaction. Since February 2020, he is the scientific head of the Atlantis project-team at the Inria Research Center of Université Côte d'Azur. He is also the scientific coordinator of the development of the DIOGENeS software suite, which is dedicated to computational nanophotonics. Stéphane Lanteri has authored or co-authored more that 80 publications in international journals in applied mathematics, scientific computing, computational physics and more recently, optics and photonics. Web: https://slanteri.github.io/



Kun Li received the Ph.D. degree in computational mathematics from the University of Electronic Science and Technology of China, Chengdu, China, in 2020. He is currently an assistant professor with the School of Mathematics, Southwestern University of Finance and Economics, Chengdu, China. His current research interests include the discontinuous Galerkin time-domain methods and model order reduction techniques for solving the electromagnetic, and local/nonlocal optical response problems.