Advanced Parallel Algorithms in Computational Electromagnetics

by

Shu Wang

B.E. Tianjian/Nankai University, 2011
M.E. Beihang University, 2014
M.S. University of New Mexico, 2019

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

June, 2020

Dedication

Dedicated to my wife, daughter and parents for their love and support.

Advanced Parallel Algorithms in Computational Electromagnetics

by

Shu Wang

B.E. Tianjian/Nankai University, 2011

M.E. Beihang University, 2014

M.S. University of New Mexico, 2019

Ph.D., Electrical Engineering, University of New Mexico, 2020

Abstract

The rapid development of high performance computing has pushed the computational electromagnetics(CEM) towards high accuracy, high fidelity and extreme computational scales. There is a great need for existing CEM solvers to have enhanced parallelism and scaling capability. The purpose of this dissertation is to investigate advanced parallel algorithms for both frequency and time domain solvers. In frequency domain, this work first develop the underpinnings of parallel preconditioning technique and high-order transmission condition in the context of multi-solver scheme. The result is a computing resource-aware and implementation wise compact solver. Then this work targeted at developing efficient algorithms for cases where iteration of simulations,e.g. parameter sweep, is necessary. The proposed platform Green's function method can effectively reduce the turn-around time by exploiting reusable matrices.

In time domain, due to the ever-increasing sophistication in EM systems, a typical transient simulation may require many time steps. Most current transient solvers exploit parallelism in spatial domain which it is not trivial to sustain parallel scaling capability in high level. This work,therefore, provided a new perspective in parallelism, parallel-in-time(PIT). The problem is first decomposed based on superposition principle and corresponding effective integration methods are developed. Next, a hybrid parallel scheme, space-time building block method, which is based on reduced order model, is proposed for applications like meta-material simulation. A improved scaling efficiency and 3x speed-up is observed in our work. Finally, PIT is extended to improve scaling efficiency for nonlinear circuit-electromagnetic co-simulations, where 2x better efficiency is achieved by proposed algorithms.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computational electromagnetic(EM) becomes essential for design and analysis in today's highly complicated engineering environments. A variety of numerical methods have been developed to analyze different type of problems. In the time-harmonic regime, surface integral equation(SIE) [1] methods are useful for the solution of large, impenetrable or homogeneous EN scatterings. Some fast algorithms, such as multilevel fast multiple method(MLFMA) can achieve good acceleration and near optimal computational complexity of O(NlogN) [2]. Another widely used method is finite element method(FEM) [3, 4] as it provides accurate results in dealing with complex geometric features and material properties.

More recently, many practical interests have been focused on electrically large object and complicated platform simulations since the ever-growing power of computational capabilities. Among various numerical methods, domain decomposition methods(DDMs) and multi-solver schemes [5, 6] have been introduced to greatly extend the capability by applying the optimal solvers for each individual subdomains and formulate inter-subdomain coupling as incident fields.

Over the past decades, time-domain (TD) computational electromagetics (CEM) methods have also enjoyed considerable success in solving transient and broadband electromagnetic (EM) problems. The TD nature of these methods, compared to its frequency domain(FD) counterpart, offers direct and efficient approaches in certain EM problems where we need to study a transient field effect of an arbitrary time-signal excitation(e.g. EMC/EMI coupling, ultra-wideband antennas), or the nonlinear circuit-EM interactions. Many TD numerical methods, including finite element methods [7–13], integral equation methods [14–18], finite difference methods [19–23], discontinuous Galerkin methods [24–28], pseudospectral method [29–31], finite integration techniques [32, 33], and transmission-line-matrix methods [34] have been developed. These techniques have been thoroughly tested to be robust and efficient in a wide range of problems [35–37].

1.1 Overview

The era of supercomputing has landed as more challenges have come in company [38]. One highlight in the trend of HPC is that the scale of computing has gone massively parallel instead of clock speed going significantly faster. The development of CEM algorithms should also keep up with the pace. As the power of high performance computing(HPC) grows, the demand for efficient and parallel solvers also stands out. We can approach the challenging problems in CEM from the perspective of increasing complexity.

The first scenario is the single frequency simulation which is often raised in antenna RCS evaluation and object scattering analysis. Since the CAD models are built based on the awareness of geometry and materials, divide-and-conquer is mainly considered for parallelism. One of the most successful framework is multi-solver domain decomposition(MSDDM) where each submain is solver by separate solvers and glued together by necessary continuity conditions. The partition of subdomain and choice of solver are often the time manually with prior knowledge. The process should be as modular as possible for the ease of implementation and plug-in-play components should be the ideal. Design optimization and iteration are never rare which putting high demands for simulation-to-result time.

The second scenario is the broad-band response evaluation which show itself in EMC/EMI application and meta-materials design. Transient behaviors are of interest. Long term simulation is inevitable for certain resonating structures and stability is another concern. Lots of work has been done in discretizing PDE such that more concurrency can be exploited but there is always bottle-neck when the number of processes grows large. Time integration is sequential in nature by first look and how to break this nature will be a breakthrough point.

The last scenario comes in when more physics is involved. Electromagnetic circuit

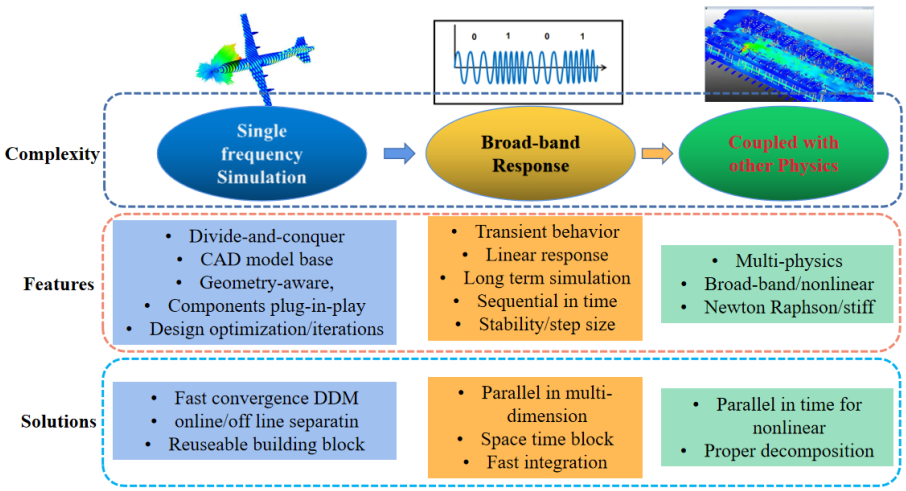


Figure 1.1: Overview of challenging problems in CEM

co-simulation is an typical example where non-linearity features. If consider in a coupled system, different physics may have various scale of evolving rate which could be translated into stiffness in mathematical language. These level up the difficulties towards parallelism. The three scenarios and corresponding features and solution can be found in Figure 1.1.

1.2 Issues to be addressed

The goal for this dissertation is to utilize advanced CEM algorithms to address most challenging problems in CEM. The contributions can be made from the following aspects of needs.

(1) The advance to the next-generation HPC architectures will be achieved by increased concurrency, not increased processor speed. Unfortunately, traditional time-dependent solvers are typically parallel only in the spatial domain, and advance time-step by time-step. For example, discontinuous Galerkin(DG) spatial discretiza-

tion permits us to take advantage of using unstructured high-order finite elements and some OpenMP/MPI parallelizaton. The sequential-in-time nature of the solvers may present good parallel scaling when the number of spatial mesh points per core is large. But the parallel efficiency quickly deteriorates even saturates if the spatial parallelism has been fully exploited. This issue is known as the strong scaling limit, which has surfaced an obstacle to achieve exascale parallel performance.

(2) The successful application of DDM in time-harmonic has no alternative in transient regime. Can we utilize the repetion in both space and time in periodic structures and create some reusable building blocks becomes a critical issue in further reduce simulation-to-solution time.

(3) This time-domain hybrid model for Nonlinear circuit-EM cosimulation also poses some challenges for existing simulation capabilities. (1) Due to the nonlinear nature of some lump circuit components, the global system become more stiff. The previously proposed methods, especially thosed steming from linear superposition principle should fail. (2) Newton-Raphson iterations are required for solving nonlinear system of equation every time the right hand side of ODE system needs to be evaluted. Unlike linear case where several matrix-vector multiplcations are just required to forward one step in time, additional computations are inevitable such as evaluating Jacobian, making tentative searching step and et al. These make forwarding one time step much slower than linear problems. (3) Multirate nature [39] of circuit phenomenon and EM physics.

(4) For multi-solver schemes, it should be noticed that the following limitations. First, the coupling between touching sub-domains only supports Robin-type transmission condition [40,41] because individual subdomain solvers may not be consistent in structure. This barrier prevents the further enhancement of convergence. Second, there lacks of a flexible preconditioning technique to be adaptive to the available parallel computing environment. Existing parallel preconditioners are of Jacobi type

but poor in convergence.

(5) The CEM based in-situ antenna design goes beyond just performing a single simulation. It often needs to perform a number of simulations in order to navigate highly complex design spaces. Each simulation should complete within at most a few minutes even a few seconds in an industrial design environment. In the case where multiple simulations,e.g. parameter sweep, are required, the turn-around time always get to expensive to afford. Does it exist a technique to significantly shorten the simulation-to-engineering process by creating some reusable building blocks? Clearly, fundamental research into innovative mathematics and algorithms are required.

1.3 Outline

This dissertation proposal is organized as follows. Chapter 2 will discuss parallel-in-time method for transient EM problem. Chapter 3 will result in a novel time-evolution schemes with parallelism in both space and time. Chapter 4 will address the extension of parallel in time method in circuit-EM cosimulation. And Chapter 5 will discuss the parallel and fast methods for solving time-harmonic Maxwell equations by proposing hybrid preconditioning technique and platform Green's function. The future direction and conclusions are discussed in Chapter 6.

Chapter 2

Parallel in time method for transient electromagnetic analysis

2.1 Introduction

The question to be answered in this chapter is how to incorporate a new level of parallelism in the temporal dimension, in order to achieve high parallel efficiency and reduce the time-to-solution on massively distributed and parallel computing platforms. There have been various efforts to develop time parallel time integration methods in the applied math community. Examples include waveform relaxation methods [42–45], parareal methods [46–51], space-time multigrid methods [52, 53], overlapping domain decomposition methods [54], and parallel deferred correction framework [55–57]. A recent survey paper can be found in [58]. Nevertheless, the research field is in many ways on the going. The parallel efficiency is far from ideal, and the success has been shown primarily for parabolic equations, instead of hyperbolic equations of this paper's interests.

To break the sequential barrier, the proposed work exploits the direct parallel-in-time methodology with a space-time domain decomposition scheme. A pioneer work, so-called parallel exponential propagation (ParaEXP) was presented [59], and recently extended to EM wave problems [60, 61]. The contributions of this work include the efficient, parallel calculation of time-evolution operator, and the fusion of parallel-in-space and parallel-in-time methods for three-dimensional transient EM problems. The technique ingredients include the discontinuous Galerkin weak formulation in the spatial domain and the rational Arnoldi approximation of the evolution operator in the temporal domain. The work has two major benefits: (**i**) it results in a scalable, parallel time-domain solver with flexible space and time decomposition scheme; and (**ii**) it provides a natural way to integrate with many existing time integration techniques. The capability and performance of the computational algorithms are illustrated and validated through numerical experiments.

2.2 Formulation

2.2.1 Discontinuous Galerkin Finite element in Time Domain

To illustrate, we consider time-domain Maxwell's Equations for heterogeneous linear media in a bounded domain Ω:

$$\epsilon\frac{\partial \mathbf{E}}{\partial t} + \sigma\mathbf{E} - \nabla \times \mathbf{H} = -\mathbf{J} \qquad \text{in } \Omega \times [0,T], \tag{2.1}$$

$$\mu\frac{\partial \mathbf{H}}{\partial t} + \nabla \times \mathbf{E} = \mathbf{M} \qquad \text{in } \Omega \times [0,T], \tag{2.2}$$

$$\mathbf{E}\left(\mathbf{r}, t=0\right) = \mathbf{E}_0, \tag{2.3}$$

$$\mathbf{H}\left(\mathbf{r}, t=0\right) = \mathbf{H}_0. \tag{2.4}$$

For the spatial discretization, we briefly outline the discontinuous Galerkin (DG) finite element method. To formulate the semi-discretized DG for Equation 2.1 and Equation 2.2, the spatial domain Ω is partitioned into N non-overlapping finite elements, $\Omega = \bigcup_{i=1}^{N} K_i$. Let $V_h^k(K_i)$ denote the space of vectorial Nedelec basis over element i, the weak formulation is stated as, to find $(\mathbf{E}, \mathbf{H}) \in V_h^k(K_i) \times V_h^k(K_i)$, such that

$$\begin{aligned}
&\int_{K_i} \mathbf{w} \cdot \left(\nabla \times \mathbf{E} + \mu\frac{\partial \mathbf{H}}{\partial t} - \mathbf{M}\right) d\Omega \\
&- \int_{K_i} \mathbf{v} \cdot \left(\nabla \times \mathbf{H} - \epsilon\frac{\partial \mathbf{E}}{\partial t} - \sigma\mathbf{E} - \mathbf{J}\right) d\Omega \\
&+ \int_{\partial K_i} \{\!\{\mathbf{v}\}\!\} \cdot [\![\mathbf{H}]\!]_\gamma ds - \int_{\partial K_i} \{\!\{\mathbf{w}\}\!\} \cdot [\![\mathbf{E}]\!]_\gamma ds \\
&+ e\int_{\partial K_i} [\![\mathbf{v}]\!]_\pi \cdot [\![\mathbf{E}]\!]_\pi + f\int_{\partial K_i} [\![\mathbf{w}]\!]_\pi \cdot [\![\mathbf{H}]\!]_\pi = 0, i = 1, 2, ...N.
\end{aligned} \tag{2.5}$$

where $(\mathbf{w}, \mathbf{v}) \in V_h^k(K_i) \times V_h^k(K_i)$ are the test functions. The above equation is a sum-up of volume residuals and numerical fluxes on the interfaces (∂K_i) of each

element. The numerical flux is essential to allow for discontinuities between element, and they are defined as $\{\!\{\mathbf{u}\}\!\} = (\pi_\tau(\mathbf{u}_i) + \pi_\tau(\mathbf{u}_j))/2$, $[\![\mathbf{u}]\!]_\gamma = \gamma_\tau(\mathbf{u}_i) + \gamma_\tau(\mathbf{u}_j)$ and $[\![\mathbf{u}]\!]_\pi = \pi_\tau(\mathbf{u}_i) - \pi_\tau(\mathbf{u}_j)$, where $\gamma_\tau(\mathbf{u}_i) = \hat{\mathbf{n}}_i \times \mathbf{u}_i$ and $\pi_\tau(\mathbf{u}_i) = \hat{\mathbf{n}}_i \times \mathbf{u}_i \times \hat{\mathbf{n}}_i$ are twisted and tangential trace operators, respectively. For special choice of coefficients $e = 1/(2Z_\Gamma)$ and $f = 1/(2Y_\Gamma)$ with Z_Γ and Y_Γ being average impedance and admittance over adjacent elements, we arrive at upwind flux [62, 63] with optimal rate of convergence($O(h^3)$) for spatial discretization. [64].

The semi-discrete matrix system can be written as:

$$\mathbb{M}^\epsilon \frac{d\mathbf{E}}{dt} = -\mathbb{C}^{\mathrm{e}}\mathbf{E} + \mathbb{S}^{\mathrm{e}}\mathbf{H} - \mathbf{f}^{\mathrm{e}} \qquad \text{in } \Omega, \tag{2.6}$$

$$\mathbb{M}^\mu \frac{d\mathbf{H}}{dt} = -\mathbb{S}^{\mathrm{h}}\mathbf{E} - \mathbb{C}^{\mathrm{h}}\mathbf{H} + \mathbf{f}^{\mathrm{h}} \qquad \text{in } \Omega, \tag{2.7}$$

where $\mathbb{M}^{\epsilon,\mu}$ and $\mathbb{S}^{\mathrm{e,h}}$ are element-wise mass matrices and stiffness matrices. Matrices $\mathbb{C}^{\mathrm{e,h}}$ are constructed from DG testing from adjacent elements' faces and conductive term. $\mathbf{f}^{\mathrm{e,h}}$ represent electrical and magnetical source terms.

Concerning the time discretization, a variety of time-stepping schemes have been investigated, including LeapFrog scheme [65], Runge-Kutta method [66], implicit schemes [63], the Crank-Nicolson scheme [67], and hybrid implicit-explicit Runge-Kutta (IMEX-RK) scheme [68].

In a distributed memory HPC environment, the DG discretizations are partitioned into a number of spatial subdomains, which are mapped adaptively to available Message Passing Interface (MPI) processors; information exchanges are required only at the boundary between neighboring subdomains. Moreover, individual subdomains are allowed to choose their own time-stepping sizes and schemes. The hybridization of MPI parallelization and local time-stepping strategy has been successfully applied to large, multiscale problems with non-uniform, non-conformal discretizations [26, 69, 70].

In general, the space-parallel DGTD solver can achieve good parallel efficiency in

weak scaling experiments, but often exhibits limited performance for strong scaling tests. Namely, as the number of MPI processors for a given transient EM problem increases, the solution time does not decrease proportionally. The bottleneck is due to the limited inter-core data transfer bandwidth between processors. The increased MPI processors lead to increased number of subdomains, which in turn, increase the inter-core communication. In the following, we will incorporate additional parallelism in temporal dimension to achieve high parallel efficiency and to reduce the time-to-solution on large-scale parallel HPC computers.

2.2.2 Temporal Domain Decomposition

Initial value problem

Attributed to the advantages of DG weak formulation, the mass matrices $\mathbb{M}^{\epsilon}$ and $\mathbb{M}^{\mu}$ are block diagonal and trivially invertible. Equations Equation 2.6 and Equation 2.7 can be combined into an equivalent form [71]:

$$\frac{d\mathbf{u}}{dt} = -\mathbb{A}\mathbf{u} + \tilde{\mathbf{f}}(t), \quad \mathbf{u}(t=0) = \mathbf{v}_0, \tag{2.8}$$

with $\mathbf{u}(t) = [\mathbf{E}(t), \mathbf{H}(t)]^{\mathrm{T}}$, $\mathbb{A} := \mathbb{M}^{-1}\mathbb{S}$, $\tilde{\mathbf{f}}(t) = \mathbb{M}^{-1}\mathbf{f}(t)$, $\mathbf{f}(t) = \left[\mathbf{f}^{\mathrm{e}}, \mathbf{f}^{\mathrm{h}}\right]^{\mathrm{T}}$ is the excitation term, and $\mathbf{v}_0$ is the initial condition. The compact mass and stiffness matrices are defined by:

$$\mathbb{M} := \begin{bmatrix} \mathbb{M}_{\epsilon} & \\ & \mathbb{M}_{\mu} \end{bmatrix}, \quad \mathbb{S} := \begin{bmatrix} \mathbb{C}^{\mathrm{e}} & -\mathbb{S}^{\mathrm{e}} \\ \mathbb{S}^{\mathrm{h}} & \mathbb{C}^{\mathrm{h}} \end{bmatrix}. \tag{2.9}$$

In traditional sequential time-marching algorithms, the solution $\mathbf{u}$ is obtained by advancing time step by time step. To break the sequential barrier, the work extends

the domain decomposition idea to the temporal dimension, resulting in a time-domain parallel scheme for 3D transient EM problems.

Decomposition of sub-problems

After applying the variation-of-constants formula [72] to Equation 2.8, the semi-analytical solution can be written as:

$$\mathbf{u}(T) = \exp\left(-T\mathbb{A}\right)\mathbf{v}_0 + \int_0^T \exp\left(-\left(T-t\right)\mathbb{A}\right)\tilde{\mathbf{f}}(t)dt, \tag{2.10}$$

where $\exp\left(-T\mathbb{A}\right)$ is the exponential, time-evolution operator and $\exp\left(-T\mathbb{A}\right)\mathbf{v}_0$ gives the homogeneous solution determined solely by the initial condition. The convolution integral leads to the particular solution due to the complex source term $\tilde{\mathbf{f}}(t)$.

The key idea of this work is to increase the concurrency in the parallel computing. Since we have the complete knowledge of the source term, $\tilde{\mathbf{f}}(t)$, over the entire time period $[0, T]$, the source-dependent problem can be solved parallel in time [59]. To illustrate, we divide the time period into 3 non-overlapping subintervals $[0, T/3]$, $[T/3, 2T/3]$ and $[2T/3, T]$, Correspondingly, the source term is decomposed into three components, $\tilde{\mathbf{f}} = \tilde{\mathbf{f}}_1 \cup \tilde{\mathbf{f}}_2 \cup \tilde{\mathbf{f}}_3$, which are defined as:

$$\tilde{\mathbf{f}}_i\left(t\right) = \begin{cases} \tilde{\mathbf{f}}\left(t\right) & \frac{(i-1)T}{3} \leq t \leq \frac{iT}{3}. \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

Based on the principle of superposition, the solution to the original problem can be written as:

$$\begin{aligned} \mathbf{u}(T) = \exp\left(-T\mathbb{A}\right)\mathbf{v}_0 + \int_0^T \exp\left(-\left(T-t\right)\mathbb{A}\right)\tilde{\mathbf{f}}_1(t)dt + \\ \int_0^T \exp\left(-\left(T-t\right)\mathbb{A}\right)\tilde{\mathbf{f}}_2(t)dt + \int_0^T \exp\left(-\left(T-t\right)\mathbb{A}\right)\tilde{\mathbf{f}}_3(t)dt. \end{aligned} \tag{2.12}$$

Through straightforward manipulations, we can then rewrite Equation 2.12 as:

$$\begin{aligned}\mathbf{u}(T) =& \exp\left(-T\mathbb{A}\right)\mathbf{v}_0 + \\ & \exp\left(-\frac{2T}{3}\mathbb{A}\right)\int_0^{\frac{T}{3}} \exp\left(-\left(\frac{T}{3}-t\right)\mathbb{A}\right)\tilde{\mathbf{f}}_1(t)dt + \\ & \exp\left(-\frac{T}{3}\mathbb{A}\right)\int_{\frac{T}{3}}^{\frac{2T}{3}} \exp\left(-\left(\frac{2T}{3}-t\right)\mathbb{A}\right)\tilde{\mathbf{f}}_2(t)dt + \\ & \int_{\frac{2T}{3}}^{T} \exp\left(-\left(T-t\right)\mathbb{A}\right)\tilde{\mathbf{f}}_3(t)dt. \end{aligned} \tag{2.13}$$

We observe the three integrals in Equation 2.13 are corresponding to three temporal subproblems resulting from the sources $\tilde{\mathbf{f}}_1$, $\tilde{\mathbf{f}}_2$, and $\tilde{\mathbf{f}}_3$, respectively. They are perfectly parallelizable since they can be evaluated independently. In the proposed work, these temporal subproblems are solved by the space-parallel DGTD method, where the spatial domain is partitioned into nonoverlapping subdomains. Each temporal subproblem is allowed to choose its own time-stepping scheme according to the Fourier spectrum of the source term $\tilde{\mathbf{f}}_i$. The solutions of the subproblems can be expressed as:

$$\mathbf{v}_i = \int_{\frac{(i-1)T}{3}}^{\frac{iT}{3}} \exp\left(-\left(\frac{iT}{3}-t\right)\mathbb{A}\right)\tilde{\mathbf{f}}_i(t)dt, \quad i=1,2,3. \tag{2.14}$$

The final solution can be written as:

$$\mathbf{u}(T) = \left[\exp(-T\mathbb{A}) \;\; \exp\left(-\frac{2T}{3}\mathbb{A}\right) \;\; \exp\left(-\frac{T}{3}\mathbb{A}\right) \;\; \mathbb{I}\right] \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix}. \tag{2.15}$$

The expression in Equation 2.15 involves a sum of multiplications between time-evolution operators and input vectors ($\mathbf{v}_i, i = 0, \cdots, 3$). They can be again evaluated completely independently and concurrently. Clearly, the calculation of Equation 2.15

lies largely upon the efficient and time-parallel means of applying the time-evolution operator, which will be discussed in detail in the next subsection. We remark that we have confined our discussion to linear problems in the present work. The possible extension to nonlinear problems can be found in [73].

Matrix Exponential Propagator

The application of time-evolution operator, $\mathbf{y}(t) = \exp(-t\mathbb{A})\mathbf{v}$, can be understood as the Green's function propagator in time. It involves the action of the exponential of matrix $\mathbb{A}$ on a given source vector $\mathbf{v}$ at a future time t. There are various methods available in the literature to calculate the matrix exponential. Representative examples include the Chebyshev polynomials [74], scaling and squaring with Pade or Taylor approximations and others [75, 76]. These methods are efficient for small, dense matrices, but very expensive for large-scale matrices. Fortunately, attributed to the advantage of DG weak formulation, the mass matrix $\mathbb{M}$ is block diagonal and trivially invertible. The matrix $\mathbb{A} = \mathbb{M}^{-1}\mathbb{S}$ remains sparse and structured after the matrix product. Therefore, we will consider the Krylov subspace methods to construct a parallelizable and reduced order approximation of $\exp(-t\mathbb{A})\mathbf{v}$, instead of forming $\exp(-t\mathbb{A})$ explicitly.

We recall the essential idea of Krylov subspace methods is to construct a low-dimensional subspace $\mathcal{K}_m(\mathbb{A}, \mathbf{v})$=span$\{\mathbf{v}, \mathbb{A}\mathbf{v}, \mathbb{A}^2\mathbf{v}, \cdots, \mathbb{A}^{m-1}\mathbf{v}\}$ [72]. The action of matrix exponential can be projected and evaluated on the reduced Krylov subspace. In the standard polynomial Krylov method, an orthonormal basis $\mathbb{V}_m = [\mathbf{v}_1, \cdots, \mathbf{v}_m] \in \mathbb{R}^{n\times m}$ is built up by the well-known Arnoldi orthogonalization process:

$$\mathbb{A}\mathbb{V}_m = \mathbb{V}_m\mathbb{H}_{m,m} + \mathbf{v}_{m+1}h_{m+1,m}\mathbf{e}_m^T, \tag{2.16}$$

where $\mathbf{e}_m = [0, ...0, 1]^T \in \mathbb{R}^m$ is the last canonical basis vector in $\mathbb{R}^m$, $\mathbb{H}_{m,m}$ is the

upper Hesssenberg matrix which consists of the orthogonalization coefficients.

By the orthogonality of $\mathbb{V}_m$, the Hessenberg matrix $\mathbb{H}_{m,m} = \mathbb{V}_m^T \mathbb{A} \mathbb{V}_m$ represents the projection of $\mathbb{A}$ onto the $\mathcal{K}_m$. Then the polynomial Arnoldi approximation $\mathbf{y}_m(t)$ of $\mathbf{y}(t) = \exp(-t\mathbb{A})\mathbf{v}$ can be described as

$$\mathbf{y}_m(t) = \mathbb{V}_m \exp\left(-t\mathbb{H}_{m,m}\right) \mathbb{V}_m^* \mathbf{v}. \tag{2.17}$$

The efficiency of the polynomial Krylov method rests on the fact $\mathbf{y}_m(t)$ approximates $\mathbf{y}(t)$ even with small number of m. Consequently, the computational cost of the matrix exponential $\exp(-t\mathbb{H}_{m,m})$ can be much cheaper than the direct calculation of $\exp(-t\mathbb{A})$ since $m \ll n$.

Nevertheless, due to the nature of matrix-vector multiplication in the polynomial Krylov subspace, the eigenvalues of Hessenberg matrix $\mathbb{H}_{m,m}$ approximate predominantly the large eigenvalues of $\mathbb{A}$. But in long-time simulations, the large eigenvalues (in modulus) are not as significant as the small ones due to the exponential decay. Thus, the polynomial Krylov method only exhibits super-linear convergence for short time simulations.

To address this shortcoming, we exploit the recent advances in the rational Krylov approximation of matrix functions [77–79]. The orthogonal basis vector in rational Krylov subspace is constructed using the recursion:

$$\mathbf{v}_{j+1} h_{j+1,j} = \left(\mathbb{I} - \frac{\mathbb{A}}{\xi_j}\right)^{-1} \mathbb{A}\mathbf{v}_j - \sum_{i=1}^{j} \mathbf{v}_i h_{i,j}. \tag{2.18}$$

The mth-order rational Arnoldi decomposition can be written in the matrix form as:

$$\mathbb{A}\mathbb{V}_m \left(\mathbb{I}_m + \mathbb{H}_{m,m}\mathbb{X}_m\right) = \mathbb{V}_{m+1}\overline{\mathbb{H}}_{m+1,m}, \tag{2.19}$$

where $\mathbb{X}_m = \mathrm{diag}(\xi_1^{-1}, \xi_2^{-1}, ..., \xi_m^{-1})$, $\mathbb{I}_m$ is the identity matrix.
And $\overline{\mathbb{H}}_{m+1,m} = \left[\mathbb{H}_{m,m}, h_{m+1,m}\mathbf{e}_m^T\right]^T$. To simplify the notation, we introduce $\mathbb{K}_m =$

$(\mathbb{I}_m + \mathbb{H}_{m,m}\mathbb{X}_m)$, the rational Arnoldi decomposition can be expressed as

$$\mathbb{A}\mathbb{V}_m = \mathbb{V}_m\mathbb{H}_{m,m}\mathbb{K}_m^{-1} + \mathbf{v}_{m+1}h_{m+1,m}\mathbf{e}_m^T\mathbb{K}_m^{-1}. \tag{2.20}$$

Using this decomposition we can construct the rational Arnoldi approximation of the matrix exponential as:

$$\mathbf{y}_m(t) = \mathbb{V}_m \exp\left(-t\mathbb{A}_m\right)\mathbb{V}_m^*\mathbf{v}, \tag{2.21}$$

where $\mathbb{A}_m = \mathbb{H}_{m,m}\mathbb{K}_m^{-1}$, which is also known as the matrix Rayleigh quotient [80]. In this work, a residual based error estimation [81] is adopted for the convergence criteria. Let $\delta_m = \|\mathbf{y}_m - \mathbf{y}_{m-1}\|_2 \,/\, \|\mathbf{y}_m\|_2$ be the normalized difference of $\mathbf{y}_m$, the residual-based error ϵ_m is defined as:

$$\epsilon_m = \min(1 + \|\mathbf{y}_m\|_2, \frac{\delta_m}{1-\delta_m}\|\mathbf{y}_m\|_2). \tag{2.22}$$

We would like to highlight a few appealing aspects of the rational Arnoldi approximation:

- The rational Arnoldi approximation is equivalent to the action of a rational function $\mathcal{R}_m$ on a vector. Namely, it interpolates the matrix exponential as: $\mathbf{y}_m(t) = \mathbb{V}_m \exp\left(-t\mathbb{A}_m\right)\mathbb{V}_m^*\mathbf{v} = \mathcal{R}_m\left(\mathbb{A}\right)\mathbf{v}$. The denominator polynomial $q_{m-1} \in \mathbb{P}_{m-1}$ of the rational function satisfies [82]:

  $$q_{m-1}\left(\mathbb{A}\right) := \prod_{j=1}^{m-1}\left(\mathbb{I} - \mathbb{A}/\xi_j\right), \tag{2.23}$$

  where poles $\xi_1, \xi_2, \cdots, \xi_{m-1} \in \overline{\mathbb{C}} := \mathbb{C} \cup \{\infty\}$. In practice, it is efficient to use cyclically repeated poles, where the number of distinct poles is far fewer than the rational Krylov space dimension. Comparing to the direct rational interpolation of the time-evolution operator [83, 84], the rational Krylov approximation incorporates the properties of the starting vector $\mathbf{v}$, thus requires much less expansion terms than the direct method.

- The rational Krylov space of order m can be stated as:

$$\mathcal{Q}_m(\mathbb{A}, \mathbf{v}) := q_{m-1}(\mathbb{A})^{-1}\mathcal{K}_m(\mathbb{A}, \mathbf{v}). \tag{2.24}$$

  It is able to account for the contribution from different eigenvalues by pre-selecting the denominator polynomial q_{m-1}. If all the poles $\xi_1, \xi_2, ..., \xi_{m-1}$ are set to infinity, then $\mathcal{Q}_m(\mathbb{A}, \mathbf{v})$ reduces to a polynomial Krylov space $\mathcal{K}_m(\mathbb{A}, \mathbf{v})$. If let $\xi_i = \sigma < \infty, i = 1, 2, ..., m-1$, then $\mathcal{Q}_m(\mathbb{A}, \mathbf{v})$ reduces to a Krylov subspace $\mathcal{K}_m((\mathbb{A} - \sigma\mathbb{I})^{-1}, \mathbf{v})$ which results in a special case known as the shift-invert method [61, 85].

- The rational Arnoldi approximation can be considered as a nonstandard time integrator comparing to the traditional time-stepping schemes. The method provides high accuracy, superlinear convergence, and good scaling efficiency. Since there is no discretization in the temporal dimension, it does not suffer from small time steps required for highly oscillatory wave problems. Thus, it is particularly suitable for long time EM simulations. Moreover, the iteration number (Krylov subspace dimension) is almost constant when the spatial discretization becomes finer, i.e., the convergence is almost mesh independent. This property inspires a heuristic poles selection strategy in the next section.

2.2.3 Parallel-in-Space-and-Time Simulation

Overview of proposed work

In the proposed work, the parallel-in-space-time simulation consists of two computing stages, as illustrated in Figure 2.1. The first stage is the source-dependent propagation, corresponding to Equation 2.14. The time period $[0, T]$ is partitioned into N time windows $W_i = (t_{i-1}, t_i]$ with $i = 1, 2, ..., N$. Accordingly, the source

term $\tilde{\mathbf{f}}(t)$ is decomposed as shown in Equation 2.11. Within each time window, the spatial domain is decomposed into M sub-domains. The space-parallel DGTD (with M cores/sub-domains) is applied to the temporal subproblem of individual time windows W_i, denoted by Group_1 to Group_N in Figure 2.1. This stage calculates all the initial vectors $\mathbf{v}_i$ as in Equation 2.15. The main computational effort during this stage is the matrix-vector multiplication (explicit scheme) or matrix system solution (implicit scheme) in the DGTD method.

The second stage is the matrix exponential (Green's Function) propagation, which is independent of source terms. The time period $[0, T]$ is divided into N overlapped time-windows with $W_i' = (t_{i-1}, T]$. The matrix exponential propagation tasks on each W_i' are carried out concurrently and independently as illustrated in Equation 2.15.

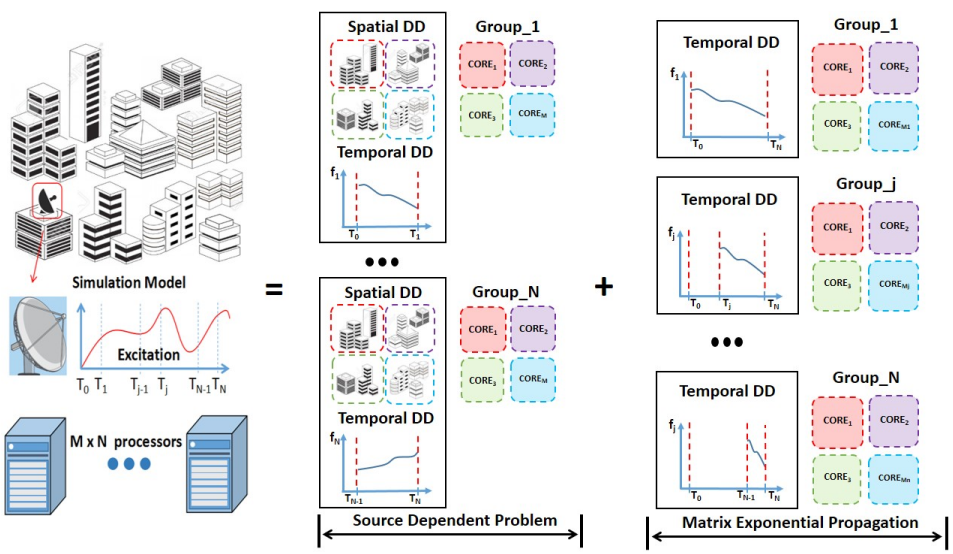


Figure 2.1: Schematic diagram of the proposed work

Parallel rational Krylov approximation

The parallelization of rational Krylov method was first studied for the eigenvalue computations [86, 87]. Recently, a new strategy of parallel rational approximation is

presented in [77], aiming to construct multiple basis vectors in $\mathcal{Q}_m(\mathbb{A}, \mathbf{v})$ concurrently. In this work, the parallelization of rational Arnoldi decomposition consists of two main parts: parallel matrix system solutions for a series of shifted system and parallel matrix vector multiplications for Gram-Schmidt orthogonalization.

There are M processors assigned to each task defined on W_i'. The processors are further distributed according to the number of poles pre-selected such that matrix system solutions are computed concurrently.

The parallel implementation is described in Algorithm Algorithm 1, where $\mathbf{v}, \underline{\mathbf{k}}, \underline{\mathbf{h}}$ are column vectors of matrices $\mathbb{V}_{m+1}, \mathbb{K}_m$ and $\overline{\mathbb{H}}_m$, respectively. The number of matrix factorizations is equal to the number of distinct poles. These matrix factorizations can be precomputed and reused in practice. The number of matrix system solutions, m, is equal to the dimension of rational Krylov space. The complexity for Gram-Schmidt process is of order $O(m^2 n)$, where n is the dimension of matrix $\mathbb{A}$.

Assume a total number of p processors (or groups) are used and pole ξ_i is assigned to each processor (group). The p shifted systems are factorized/solved in parallel. The continuation pairs are chosen to guarantee the newly computed Krylov vectors $\mathbf{w}_{s+l+1}$ being orthogonal to rational Krylov spaces generated previously. Their qualities determine the number of parallel poles.

Next, the newly computed Krylov vectors are made to be orthogonal to each other. The communication between p processors (groups) involves $O(p^2)$ message and the p is no more than 8 in practice. Additional parallelism within each processor (group) can be achieved by parallel linear algebra operations (e.g. matrix-vector multiplication).

Algorithm 1 Compute $\mathbb{A}\mathbb{V}_{m+1}\mathbb{K}_m = \mathbb{V}_{m+1}\overline{\mathbb{H}}_m$

Input:$\mathbb{A}$,$\mathbf{v}$,ξ_i, processors

Assign ξ_i to processors labelled by $l = 1, 2, ..., p$

$\mathbf{v}_1^{[l]} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}$

for $j = 1, ..., [m/p]$ **do** ▷ in parallel

    $s = (j-1)p$

    choose continuation pair$(\eta_{s+l}^{[l]}/\rho_{s+l}^{[l]}, \mathbf{t}_{s+l}^{[l]})$

    $\mathbf{w}_{s+l+1}^{[l]} \leftarrow (\mathbb{A} - \xi_{s+l}I)^{-1}(\rho_{s+l}^{[l]}\mathbb{A} - \eta_{s+l}^{[l]}I)\mathbb{V}_{s+l}^{[l]}\mathbf{t}_{s+l}^{[l]}$

    $\mathbf{w}_{s+l+1}^{[l]} \leftarrow \mathbf{w}_{s+l+1}^{[l]} - \mathbb{V}_{s+l}^{[l]}[(\mathbb{V}_{s+l}^{[l]})^*\mathbf{w}_{s+l+1}^{[l]}]$

    for $i = 1, ..., p$ **do**

        Orthogonalize $\mathbf{w}_{s+l+1}^{[l]}$ to each other

        Update and BroadCast $\underline{\mathbf{k}_{s+i}^{[l]}}, \underline{\mathbf{h}_{s+i}^{[l]}}, \mathbf{t}_{s+i}^{[l]}, \mathbf{v}_{s+i}^{[l]}$

    end for

end for

Parallel efficiency and Load balancing

Let's first review the strong scaling efficiency of parallel-in-space (PIS) method. Assume the time period is $[0, T]$, and the time step is chosen as Δt to meet certain accuracy requirement. The total number of time steps is obtained by: $\frac{T}{\Delta t}$.

Let t_K^{PIS} be the run time that the parallel-in-space method takes for marching one time step on K MPI processors. The parallel scaling efficiency η_K^{PIS} can be expressed

as:

$$\eta_K^{\text{PIS}} = \frac{t_1^{\text{PIS}}}{K \cdot t_K^{\text{PIS}}}, \tag{2.25}$$

where the t_1^{PIS} is the run time taking on a single processor. The total run time on K MPI processors for $\frac{T}{\Delta t}$ time steps can be written as:

$$T^{\text{PIS}} = \frac{T}{\Delta t} \cdot t_K^{\text{PIS}} = \frac{T \cdot t_1^{\text{PIS}}}{\Delta t \cdot K \eta_K^{\text{PIS}}}. \tag{2.26}$$

Due to the increase of communication to computation ratio, the scaling efficiency is expected to decay as: $\eta_1^{\text{PIS}} > \eta_2^{\text{PIS}} > ... > \eta_K^{\text{PIS}}$. Therefore, the run time, T^{PIS}, does not linearly decrease with the increase of MPI processors.

Based on the same calculation, the run time in the 1st stage of the parallel-in-space-time method on K MPI processors can be expressed as (assuming N time windows and M processors per time window):

$$T_1 = \frac{T}{N \cdot \Delta t} \cdot t_M^{\text{PIS}}, \tag{2.27}$$

where $K = M \cdot N$. The run time for the 2nd stage is:

$$T_2 = \max_{1 \le j \le N} t_j^{\text{exp}} = \beta \cdot T_1, \tag{2.28}$$

where β is the ratio of run time between 1st and 2nd stage. Therefore, the speed up over the parallel-in-space only method is written as:

$$S = \frac{T^{\text{PIS}}}{T_1 + T_2} = \frac{\eta_M^{\text{PIS}}}{(1+\beta)\eta_K^{\text{PIS}}}. \tag{2.29}$$

The speed up scales up to $\eta_M^{\text{PIS}}/\eta_K^{\text{PIS}}$ as the run time for the 2nd stage is insignificant comparing to the 1st stage.

It is verified by numerical experiments that the parallel scaling efficiency of PIS method decreases quickly with respect to increasing number of processors, K. As a comparison, the parallel-in-space-time method is allowed to use a fixed number of processors per time window, M, and increases the number of time windows when more processors are available. Therefore, higher speed-up will be achieved on larger scale computers.

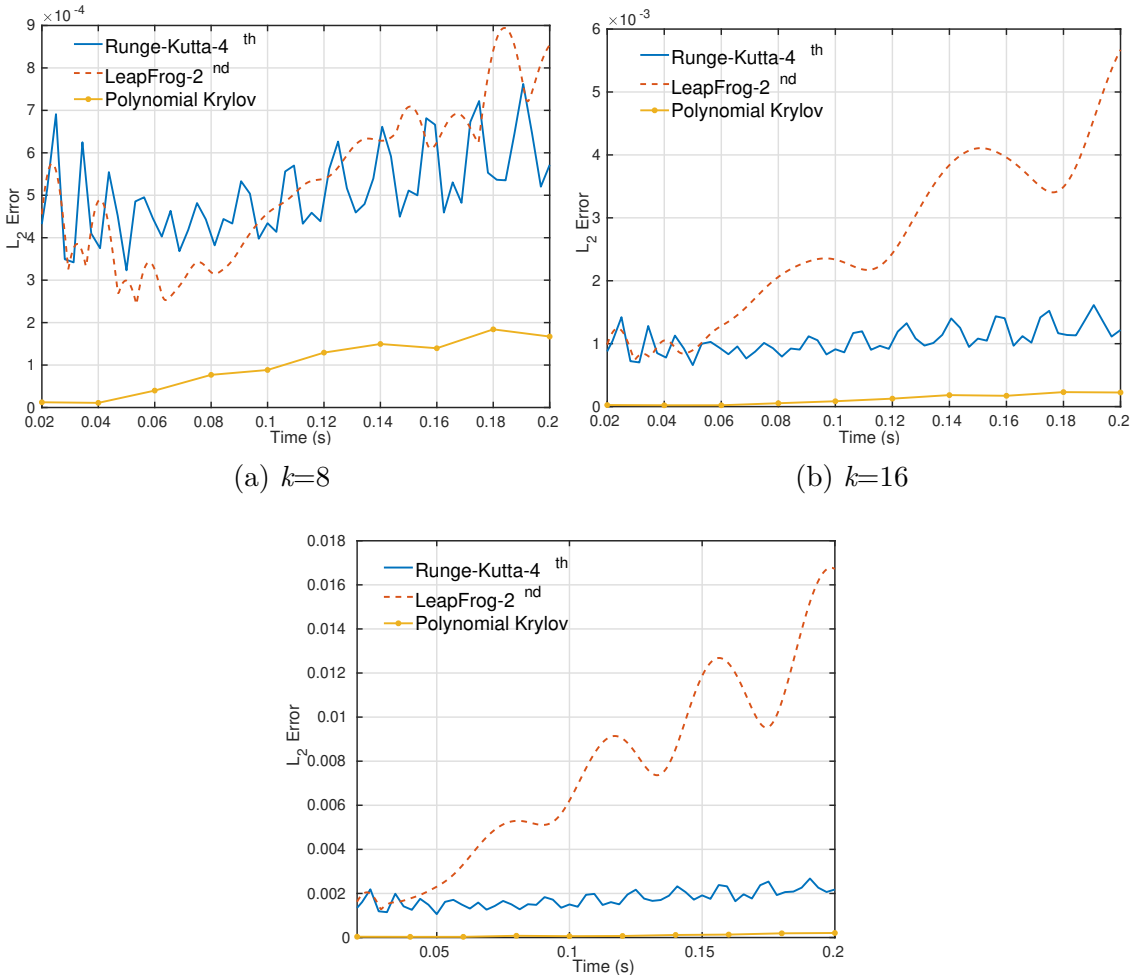
A two-level matrix exponential propagator

As alluded to in the previous section, the polynomial Krylov approximation is suitable for short time simulations. On the other hand, the rational Krylov approximation is very efficient for long time propagations, but is more expensive since it requires solving matrix systems of equations. We propose a two-level matrix exponential propagator to combine the strength of both methods. The rational Krylov approximation is exploited for the propagation over regular, coarse-level time units. Arbitrarily long timespan simulation can be achieved by periodically applying time unit propagators. The polynomial Krylov approximation is then used as a fine level propagator within the time units. The result will be demonstrated in the numerical experiment section.

2.3 Numerical Experiments

2.3.1 Accuracy and Convergency Study

We first examine the accuracy of the Krylov subspace approximation of matrix exponential propagator. The accuracy study is conducted on two examples. For the first one, a one-dimensional Maxwell equations is solved in $x \in [-1, 1]$ with Dirichlet boundary conditions. The boundary values satisfy $E_t(-1) = E_t(1) = 0$. The initial condition is set as $E_0(x) = H_0(x) = \sin(k\pi x), k = 8, 16, 24$. The spatial domain is partitioned into 60 segments of grids and a fourth-order Lagrange polynomial is used as basis function. The evolution of error is illustrated in Figure 2.2, where the reference solution at each time snapshot is achieved by evaluating $\exp(t\mathbb{A})\mathbf{v}_0$ directly. Here the polynomial Krylov method is used in a time stepping way (10 steps) and the average polynomial Krylov dimension is 20. The convergence tolerance for polynomial Krylov method is is set to 1.0×10^{-3} such that the total number of matrix-vector multiplications of all three methods are approximately the same. We can see that the polynomial Krylov method shows great advantage in accuracy over traditional time marching schemes when the solution has high frequency components.

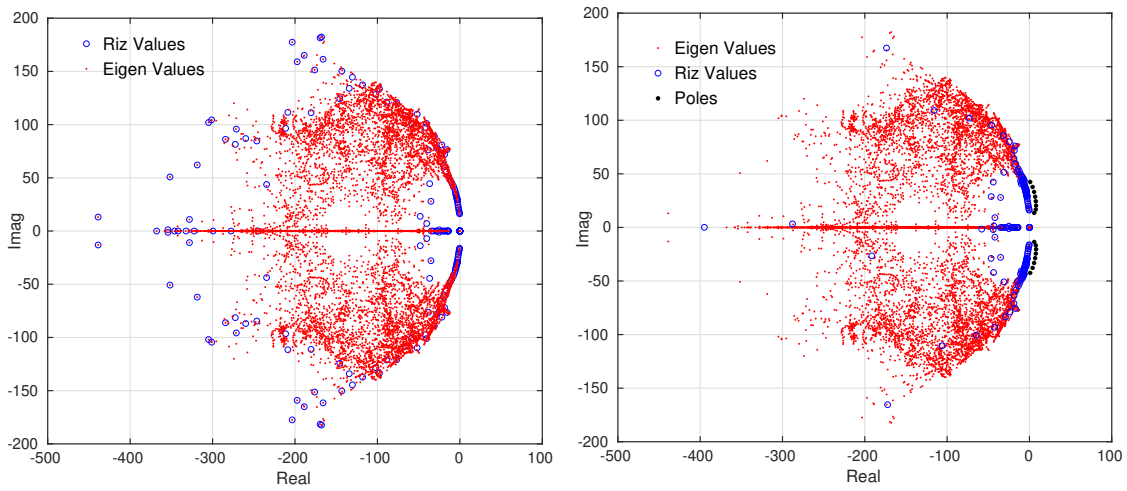


(a) k=8 (b) k=16

(c) k=24

Figure 2.2: Error evolution with time

The second example is a two-port rectangular WG90 waveguide with dimensions 2.286cm×1.016cm×10.8.cm. The total number of unknowns is 7,967.The eigenspectrum of the original matrix $t\mathbb{A}$, and Riz values (eigenvalues of reduced matrix $t\mathbb{A}_m$) of polynomial/rational Krylov methods are shown in Figure 2.3. The time window length $t = 0.4$ns. In the polynomial Krylov method, the Riz values approximate the large eigenvalues of $t\mathbb{A}$. In the rational Krylov method, the Riz values are clustered around the pre-selected, optimized 16 poles (only distinct poles are displayed in the figure).



(a) Polynomial Krylov method (b) Rational Krylov method

Figure 2.3: Eigenspectrum study of original and reduced system

Next, we compare the error in the matrix exponential approximation using these two methods, where the error is defined as: $\|\exp(-t\mathbb{A})\,\mathbf{v}_0 - \mathbb{V}_m \exp(-t\mathbb{A}_m)\,\mathbb{V}_m^*\mathbf{v}_0\|$. The error convergence histories are presented in Figure 2.4. The results show that the rational Krylov method provides much more accurate approximation with smaller Krylov subspace dimension.

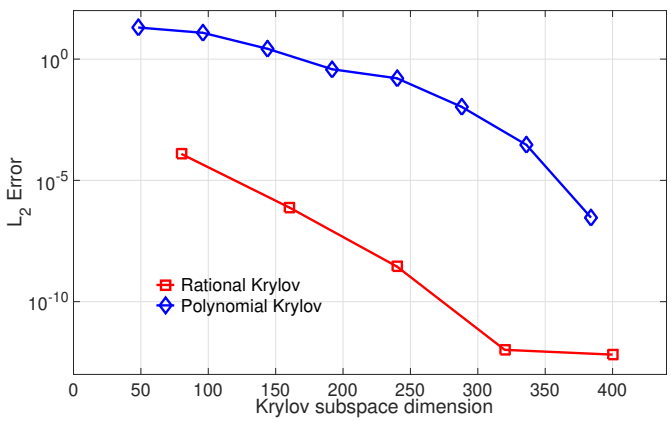
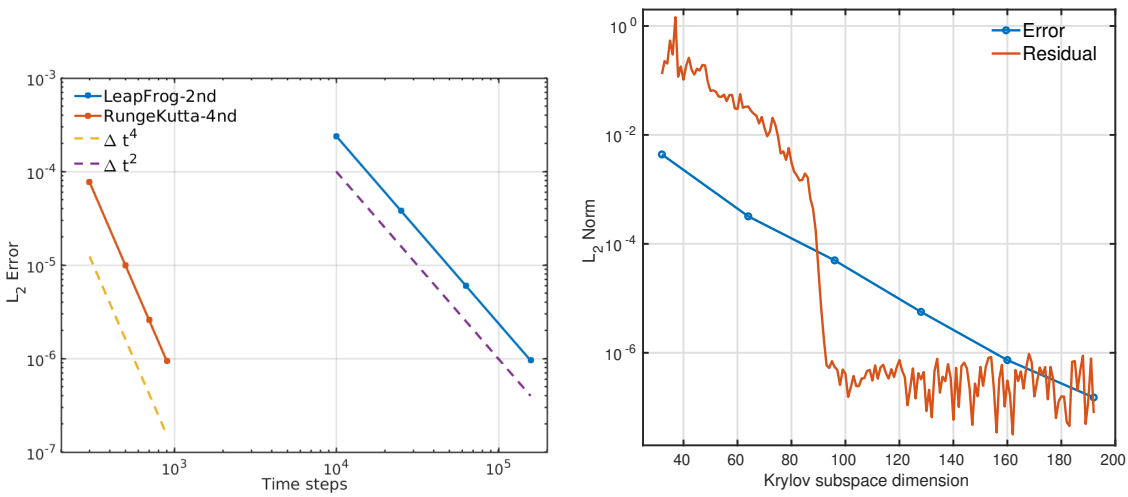


Figure 2.4: Convergence history w.r.t. Krylov subspace dimension.

The rational Krylov approximation can be considered as a nonstandard time in-

tegrator comparing to the standard time-stepping schemes. In this experiment, we compare the efficiency of matrix exponential propagator with two standard time-stepping schemes, Runge-Kutta-4$^{\text{nd}}$ and LeapFrog-2$^{\text{nd}}$ methods. The numerical results are shown in Figure 2.5. The rational Krylov method requires very fews iterations to achieve the same order of accuracy. In turn, it offers a big advantage in the run time. The computational statistics are presented in Table 2.1.



(a) Traditional time-stepping schemes (b) Matrix exponential propagator

Figure 2.5: Convergence histories of traditional and exponential time integrator

Table 2.1: Run Time Statistics

Error Norm	**Methods**	**Steps**	**RunTime(s)**	**Mxv**
1.0×10^{-4}	Runge-Kutta-4$^{\text{nd}}$	300	0.3	1200
	LeapFrog-2$^{\text{nd}}$	15000	6.75	30000
	Rational Krylov	1	0.1	96
1.0×10^{-5}	Runge-Kutta-4$^{\text{nd}}$	500	0.49	2000
	LeapFrog-2$^{\text{nd}}$	50000	23.1	100000
	Rational Krylov	1	0.13	128
1.0×10^{-6}	Runge-Kutta-4$^{\text{nd}}$	900	0.88	3600
	LeapFrog-2$^{\text{nd}}$	158490	69.1	316980
	Rational Krylov	1	0.16	160

2.3.2 Heuristic Pole Selection Strategy

In this section, we present a heuristic pole selection strategy for the rational Krylov method. A careful examination reveals that the eigenspectrum of the time-evolution matrix, $\mathbb{A}$, is determined by the EM topology, boundary conditions, and mesh density of the computational domain. In particular, due to the curl differential operator, the spectrum radius of $\mathbb{A}$ scales as $\mathcal{O}\left(h^{1/d}\right)$, where h,d are the discretization size and spatial dimension respectively.

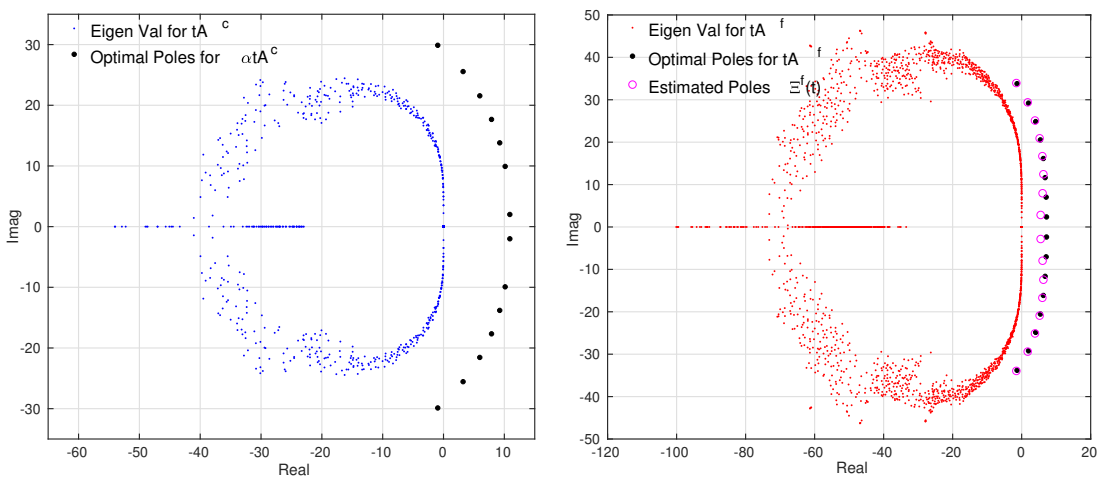
In the proposed work, we start from a coarse discretization of the computational domain, whose average mesh density is h^{c}. The resulting time-evolution matrix is denoted by $\mathbb{A}^{\mathrm{c}}$. An optimized set of poles, $\xi_1^{\mathrm{c}}, \xi_2^{\mathrm{c}}, ..., \xi_{m-1}^{\mathrm{c}}$ can be obtained by performing the following rational least square fitting problem,

$$\min \| e^{t\mathbb{A}^{\mathrm{c}}}\mathbf{v} - r_m(t\mathbb{A}^{\mathrm{c}})\mathbf{v} \|_2^2 \tag{2.30}$$

where $r_m(\cdot) = p_m(\cdot)/q_m(\cdot)$. The optimal $q_m(\cdot)$ is achieved by iteratively conducting the following procedure [77]. With the initial $q_m(\cdot)$, a rational Krylov space $\mathcal{Q}_m(\mathbb{A}^{\mathrm{c}}, \mathbf{v})$ is built by Arnoldi process. In the next step, a vector $\tilde{\mathbf{v}} \in \mathcal{Q}_m(\mathbb{A}^{\mathrm{c}}, \mathbf{v})$ is to be found such that $e^{t\mathbb{A}^{\mathrm{c}}}\tilde{\mathbf{v}}$ has least projection error in $\mathcal{Q}_m(\mathbb{A}^{\mathrm{c}}, \mathbf{v})$. New $q_m(\cdot)$ is updated by $\tilde{q}_m(\cdot)$ such that $\tilde{\mathbf{v}} = \tilde{q}_m(t\mathbb{A}^{\mathrm{c}})q_m(t\mathbb{A}^{\mathrm{c}})^{-1}\mathbf{v}$.

Subsequently, the coarse discretization is refined to a fine discretization with mesh density h^{f}, which is determined by the spatial resolution requirement. Due to the scaling effect from discontinuous Galerkin operator, the eigenspectrum of spatial discretization matrix $\mathbb{A}^{\mathrm{f}}$ (characterized by h^{f}) exhibits the same pattern with $\mathbb{A}^{\mathrm{c}}$ except that the spectrum radius is scaled by $\alpha = (h^{\mathrm{c}}/h^{\mathrm{f}})^{1/d}$. Thus the spectrum pattern of $t\mathbb{A}^{\mathrm{f}}$ can be approximated by that of $(\alpha t)\mathbb{A}^{\mathrm{c}}$. This is equivalent to finding a set of optimal poles $\Xi^{\mathrm{c}}(\alpha t)=\left\{\xi_1^{\mathrm{c}}(\alpha t), \xi_2^{\mathrm{c}}(\alpha t), ..., \xi_{m-1}^{\mathrm{c}}(\alpha t)\right\}$ on the coarse mesh for time period of αt. Finally, $\Xi^{\mathrm{f}}(t) \approx \Xi^{\mathrm{c}}(\alpha t)$ can be regarded as a set of estimated poles for $t\mathbb{A}^{\mathrm{f}}$.

To illustrate, a transverse magnetic (TM) EM field in a two-dimensional PEC cavity is discretized by the discontinuous Galerkin formulation with two different discretization sizes: $h^{\mathrm{c}} = 0.22$ and $h^{\mathrm{f}} = 0.125$. The eigenspecturms of scaled time-evolution matrices $\mathbb{A}^{\mathrm{c/f}}$ are shown in Figure 2.6. The optimal and estimated poles for the fine discretization are also presented in Figure 2.6. We notice that the estimated poles are very close to the optimal poles, which validates the proposed heuristic pole selection strategy. Finally, we present the convergence histories of the rational Krylov method using estimated and optimal poles in Figure 2.7. The initial condition is given as $E_z = sin(\pi x)sin(\pi y)$, $H_x = H_y = 0$. The results indicate the estimated poles are good approximation of optimal ones, where similar convergence behaviors are obtained.



(a) Coarse discretization (b) Fine discretization

Figure 2.6: Eigenspectrum and optimal/estimated poles on unstructured meshes

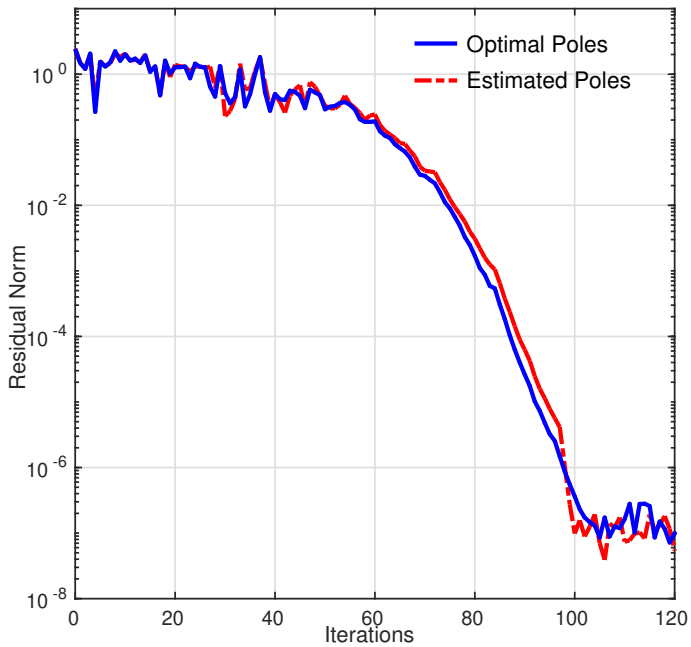
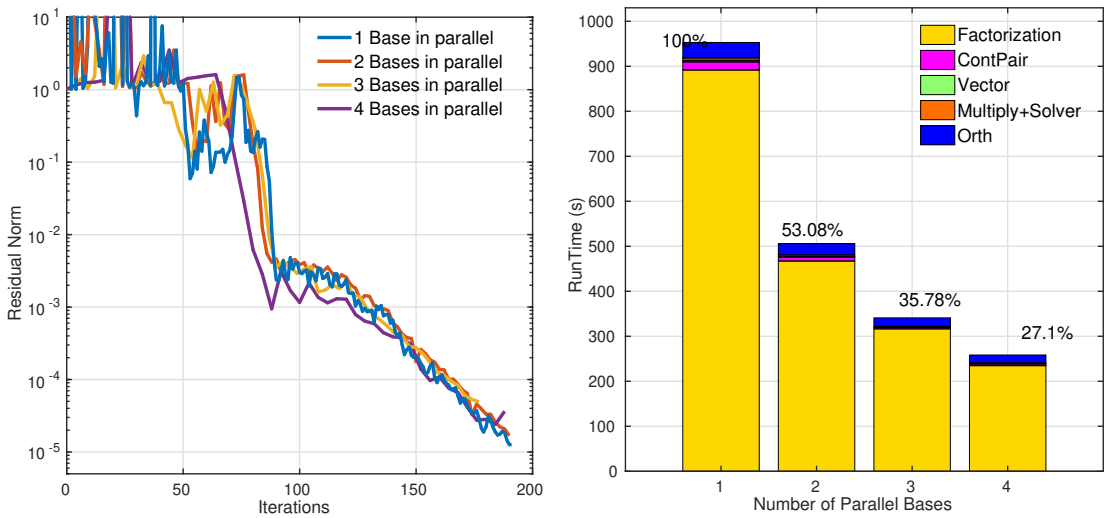


Figure 2.7: Convergence of the rational Krylov Method using optimal/estimated poles

2.3.3 Scaling of Parallel Rational Krylov Method

Rational Krylov approximation is the most computationally intensive part in the source-independent propagation. The construction of rational Krylov spaces can be implemented in parallel. Figure 2.8 shows the convergence history and run time statistics of such parallel construction. The test example is a rectangular WR90 waveguide operating at 8GHz with total number of unknowns of 111,592, where 12 pre-selected poles are used. We observe with more poles computed in parallel, the convergence is slightly changed but the run time decreases almost linearly. Although matrices factorizations take a large portion of run time, the computation can be performed in an off-line scheme. The results are also reusable in the different simulations. We also notice the run time scales well with the increase of number of parallel poles, which validates the re-assignment of parallel poles to processors (groups) can provide flexibilities to exploiting the parallel scaling efficiency.

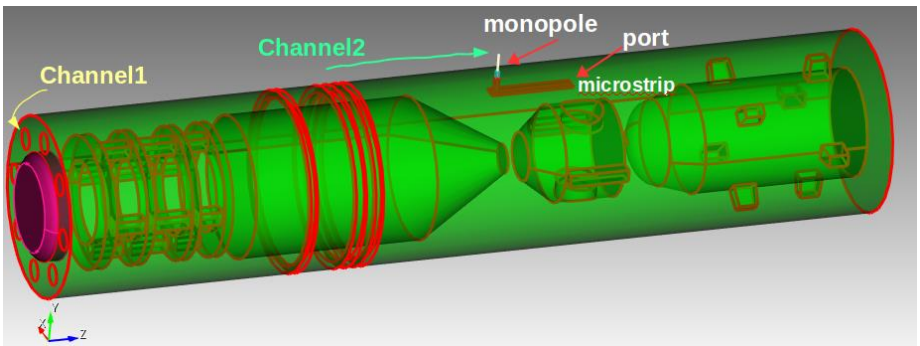


(a) Convergence history (b) Run time statistics

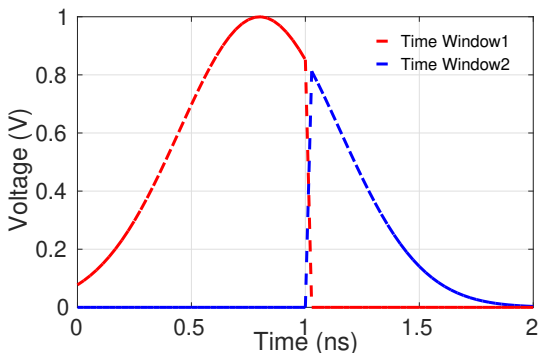
Figure 2.8: Assessment of parallel performance of the rational Krylov method

2.3.4 Performance of Space-Time Parallel Method

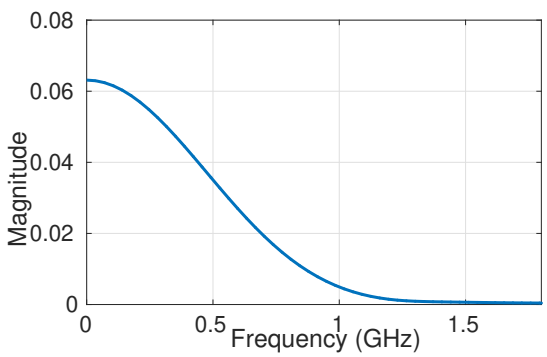
In this subsection, a mockup missile impinged by a Gaussian pulse is simulated. The configuration of the missile model and internal microstrip waveguide are given in Figure 2.9(a). The EM signal has two channels to propagate into the cavity: front door coupling through the missile chamber, and back door coupling through the mounted antenna. The entire missile model is placed inside a cylindrical air box, where the 1st order absorbing boundary condition is used at the exterior surface of the air box. The pulse has the expression of $\mathbf{V} = e^{-(t-t_0)^2/\sigma^2}V$, where $\sigma = 0.5$ns, $t_0 = 0.75$ns. The time and frequency domain information of the input pulse are given by Figure 2.9(b-c). The total number of unknowns of DG discretization is 446,296.



(a) Geometry and internal structures



(b) Excitation in time domain



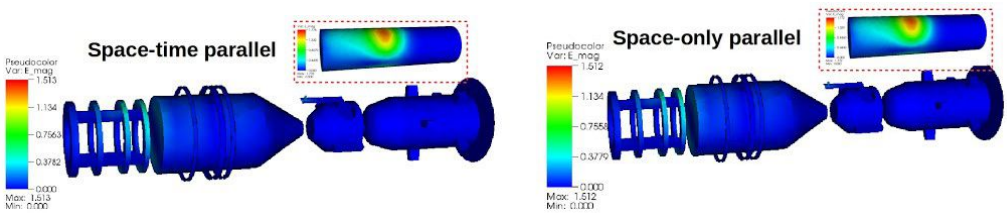
(c) Excitation in frequency domain

Figure 2.9: Setup of missile and excitation

Source dependent period

Since the Gaussian pulse excitation only lasts for 2ns, the source dependent period spans $[0, 2\text{ns}]$. In the first experiment, the source term is divided into 2 even time windows, as shown in Figure 2.9(b). Within each time window, the spatial domain is decomposed according to the number of cores assigned. For instance, in the case of total number of cores being 16 and two time windows division, the spatial domain is divided into 2 subdomains within each time window. Each spatial subdomain will be computed over a period of 1ns by 4 cores. We first apply the space-parallel DGTD to these two temporal subproblems, then employ the rational Krylov method

for the matrix exponential propagation. The obtained electric current distribution is compared with the standard DGTD method, where the time marching scheme is LeapFrog-2$^{\text{nd}}$, without temporal decomposition in Figure 2.10. A good agreement is observed.



(a) Space-time parallel result (b) Reference result

Figure 2.10: Surface current plot at 2ns

In order to show the accuracy of rational Krylov method, the results at 4 time snapshots over time-span [1ns, 2ns] are compared with the LeapFrog-2$^{\text{nd}}$ method. Let $\mathbf{u}_{\Delta t}$ and $\mathbf{u}_{m\Delta t}$ denote the solution under step size Δt and $m\Delta t$, where m is an integer. The reference solution $\mathbf{u}_{ref}$ is approximated by Richardson extrapolation [88], as

$$\mathbf{u}_{ref} = \frac{m^2\mathbf{u}_{\Delta t} - \mathbf{u}_{m\Delta t}}{m^2 - 1}. \tag{2.31}$$

The time step size of LeapFrog-2$^{\text{nd}}$ is chosen to be the maximal under the stability restriction ($dt = 0.122$ps).

We can see from Figure 2.11 that rational Krylov method outperforms LeapFrog-2$^{\text{nd}}$ method in terms of accuracy. The dimension of rational Krylov subspace is 160.

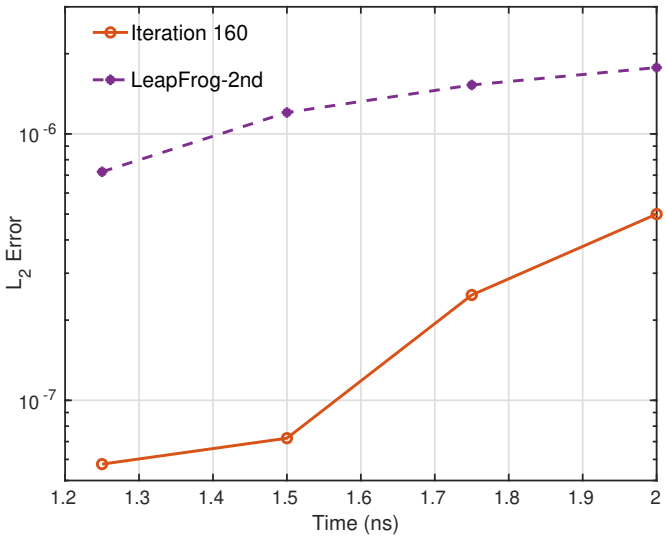
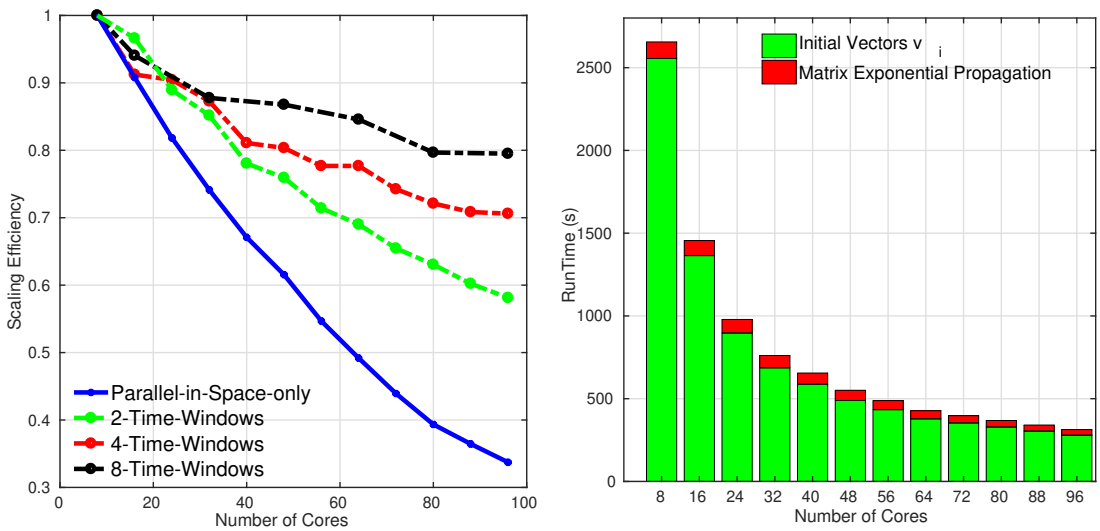


Figure 2.11: A comparison of the error with time evolution



(a) Parallel scaling efficiency (b) Source dependent period

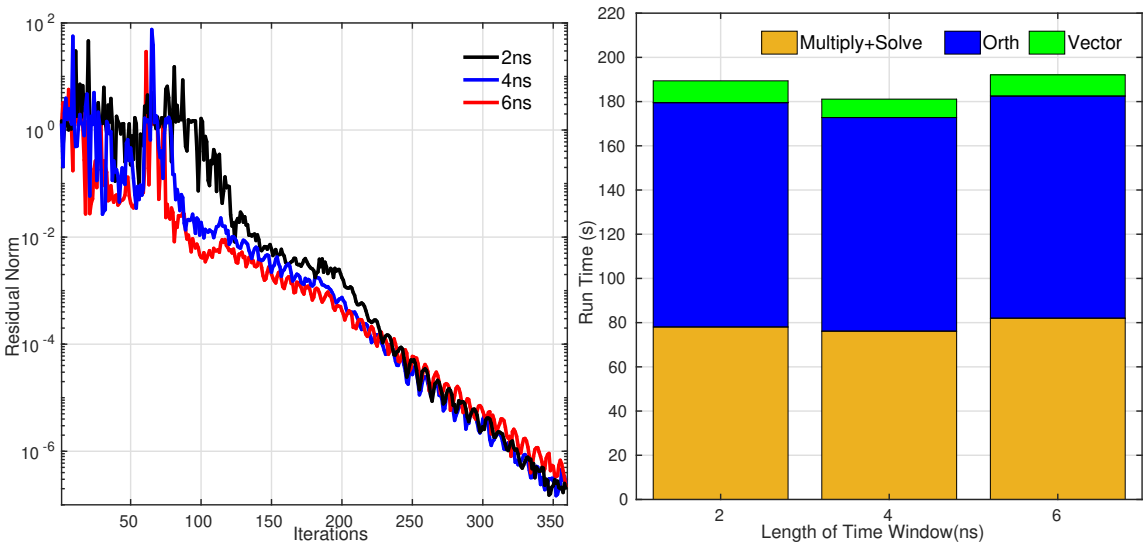
Figure 2.12: Source dependent period

We proceed to study the parallel efficiency with respect to increasing number of processors. As evident in Figure 2.12(a), the parallel-in-space DGTD solver exhibits good parallel scaling at the beginning, but the parallel efficiency quickly deteriorates

as the number of processors increases. As a comparison, the parallel-in-space-time method achieves a much better parallelization efficiency by introducing additional parallelism in the time dimension. The time period $[0, 2\text{ns}]$ is evenly divided into 2, 4, and 8 time windows. Each time window is assigned to equal number of processors. The detailed run time statistics for the 4 time windows scenario are presented in **??**(b).

Source independent period

We proceed to demonstrate the advantage of the parallel rational Krylov method for long time simulations. Due to the high Q-factor of the missile cavity, it takes a lengthy time for the EM energy to decay after being coupled into the cavity. In this experiment, we take the EM fields at 2ns as the initial vector, and continue the simulation for time periods of $2\text{ns}, 4\text{ns}$ and 6ns, respectively. The convergence history and runtime are shown in Figure 2.13. There are 8 pre-selected cyclically repeated poles selected by the heuristic poles selection strategy. For example, the estimated poles for 4ns are $\Xi = \{-2.0014 \pm 15.2575i, 0.7192 \pm 10.2914i, 1.9046 \pm 5.9581i, 2.8733 \pm 1.8710i\}$.



(a) Convergence history (b) Run time statistics

Figure 2.13: Source independent period.

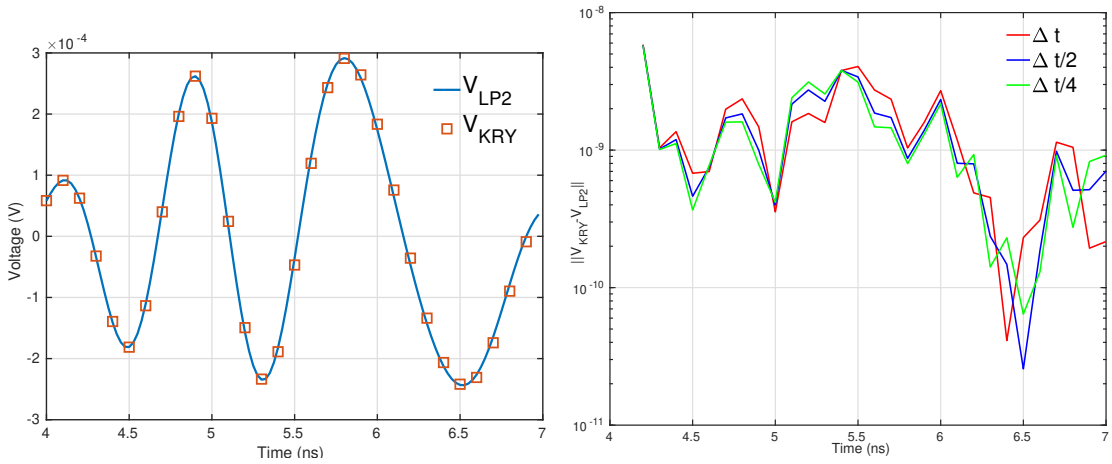
We notice that the convergence of the rational Krylov method only changes slightly with different time periods. The scale convergence results in apparent advantages in long time simulations. The computational statistics are compared to LeapFrog time stepping methods in Table 2.2.

Table 2.2: Computational Statistics

Rational Krylov method		**LeapFrog time stepping**	
Period	2/4/6ns	Period	2/4/6ns
Multiply+Solve	87/84/91s	Time Steps	16371/32742/49913
Orth	101/96/100s	Δt	0.12216ps
Total Time	189/181/192s	Total Time	1164/2328/3492s

Two-level matrix exponential propagator

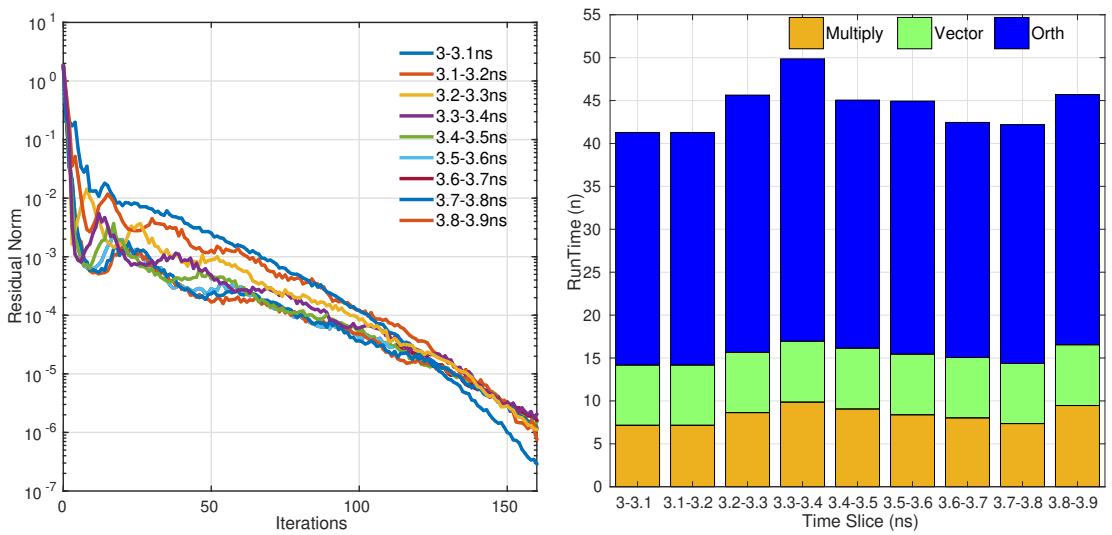
Finally, we demonstrate the two-level matrix exponential propagator combining the rational and polynomial Krylov methods. In this study, the rational Krylov method is used for the coarse-level propagation over 1ns time unit. Within each time unit, the polynomial Krylov method is applied as a fine-level propagator. The two-level propagator scheme is applied to analyze the transient induced voltage at the microstrip waveguide inside the cavity. The voltages are shown in Figure 2.14, together with a comparison between the voltages obtained by LeapFrong-2nd method with different sizes of time step. The discrepancy remains at a low level.



(a) Transient voltage by two methods (b) Difference in the calculated voltages

Figure 2.14: Transient voltage on the microstrip port

The voltages at time snapshot 4ns, 5ns and 6ns are calculated by the rational Krylov method, the rest are from the polynomial method. The convergence histories and runtime statistics of the polynomial Krylov method in the time unit of $[3\text{ns}, 4\text{ns}]$ are showed in Figure 2.15.



(a) Convergence history (b) Run time statistics

Figure 2.15: Polynomial Krylov method as a fine level propagator

Chapter 3

Space-time building block method for transient analysis of metamaterials

Over the past two decades, metamaterials and metasurfaces have received significant attentions in the scientific and engineering community due to their unique electromagnetic (EM) and optical properties. They offer a range of advantageous and exceptional responses that are not readily available in nature. The time-domain full-wave analysis is an essential tool to understand spectral and temporal properties of those advanced materials/structures. One advantage is the ability to directly analyze broadband responses and short-pulse waveforms of metamaterials. Moreover, it is considered as the most natural way to interface with nonlinear lumped circuit elements. But due to increasingly complex structures and design, it is very time-consuming to generate the computer-aided-design (CAD) model and to discretize the entire microscale structure. One significant property of meta-materials is that there exists many repetition units in the geometry. This chapter proposed a new space-time domain decomposition and building block methodology by leveraging the principles of linear superposition and space-time causality in EM physics. Given the geometry and material of the problem to be simulated, periodicity is invoked and a finite number of 4D space-time building blocks are identified. Each building block is constructed upon 3D spatial unit cell and 1D time unit. Within each time unit, the entire domain solution is obtained by superimposing solutions from building block subproblems, all of which are solved independently and in parallel. To address spatial, temporal multi-scale challenges, the time-evolution in space-time subproblems is computed by the Krylov-subspace based reduced-order model. The work results in novel time integration schemes, which exhibit high-order accuracy and are excitation-aware.

3.1 Space-Time Domain Decomposition

Evidently, for metamaterial problems of interest, there exhibits certain symmetry, periodicity, and repetition in the 4D (3D space, 1D time) computational domain. To exploit the symmetry in the physical domain, we first introduce a space-time domain decomposition. The spatial domain Ω is decomposed into subdomains $\Omega_m, m = 1, \cdots, M$. For typical metamaterial configurations, the spatial subdomain Ω_m may be a single unit cell or span a few unit cells. Next, we define a time unit, T_{u}, which corresponds to the time it takes for the wave traveling from Ω_m to its surrounding subdomains. A sample space-time partition is illustrated in Figure 3.1.

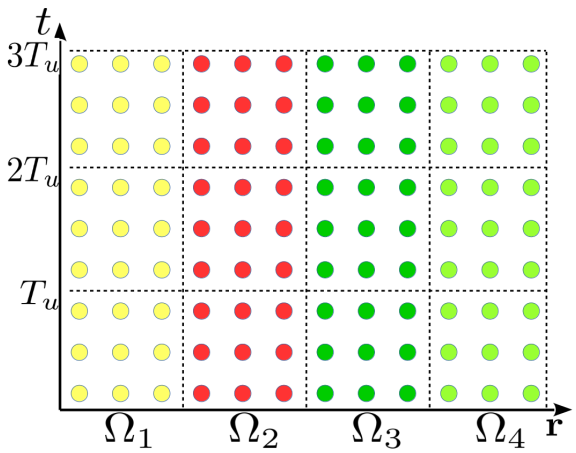


Figure 3.1: A space-time partition

We now consider a space-time subproblem between time-interval $[(n-1)T_{\mathrm{u}}, nT_{\mathrm{u}}]$ as shown in Figure 3.2:

$$\frac{d\mathbf{v}_m(t,\mathbf{r})}{dt} = -\mathbb{A}_m\mathbf{v}_m(t,\mathbf{r}) + \tilde{\mathbf{f}}_m(t,\mathbf{r}) \tag{3.1}$$

with initial condition at time $t = (n-1)T_{\mathrm{u}}$ given as:

$$\mathbf{v}_m\left((n-1)\,T_{\mathrm{u}},\mathbf{r}\right) = \begin{cases} \mathbf{u}\left((n-1)\,T_{\mathrm{u}},\mathbf{r}\right) & \mathbf{r}\in\Omega_m \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

and local source term $\tilde{\mathbf{f}}_m$ in Ω_m.

Based on the space-time causality, $\mathbb{A}_m$ is a truncated submatrix of $\mathbb{A}$, which is localized at the subdomain Ω_m and its surrounding subdomains.

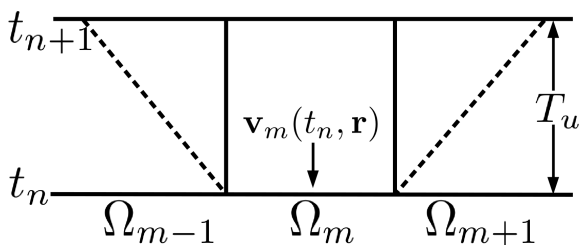


Figure 3.2: Space-time subproblem

Afterwards, the entire domain solution at $t = nT_\mathrm{u}$ can be obtained by the principle of linear superposition:

$$\mathbf{u}\left(nT_\mathrm{u}, \mathbf{r}\right) = \underbrace{\mathbf{v}_1(nT_\mathrm{u}, \mathbf{r}) + \mathbf{v}_2(nT_\mathrm{u}, \mathbf{r}) + \cdots \mathbf{v}_M(nT_\mathrm{u}, \mathbf{r})}_{\text{can be evaluated concurrently}} \tag{3.3}$$

We note that individual subproblem solutions in Equation 3.3 can be computed trivially in parallel. The repetition and periodicity in the spatial domain Ω reflects to the similarity in solving subproblems, Equation 3.1. Those observations motivate us to develop a space-time building block propagator, which will be discussed in detail in the next subsection.

3.1.1 Space-Time Building Block Propagator

We consider the space-time subproblem in Equation 3.1 defined on time interval $[t_n, t_{n+1}]$. The solution can be decomposed into the particular solution, $\tilde{\mathbf{v}}_m$, due to the source term $\tilde{\mathbf{f}}_m$, and the homogeneous solution, $\bar{\mathbf{v}}_m$, determined solely by the initial condition. Namely, $\mathbf{v}_m = \tilde{\mathbf{v}}_m + \bar{\mathbf{v}}_m$, such that:

$$\frac{d\tilde{\mathbf{v}}_m(t, \mathbf{r})}{dt} = -\mathbb{A}_m \tilde{\mathbf{v}}_m(t, \mathbf{r}) + \tilde{\mathbf{f}}_m(t, \mathbf{r}), \quad \tilde{\mathbf{v}}_m(t_n, \mathbf{r}) = 0 \tag{3.4}$$

$$\frac{d\bar{\mathbf{v}}_m(t, \mathbf{r})}{dt} = -\mathbb{A}_m \bar{\mathbf{v}}_m(t, \mathbf{r}), \quad \bar{\mathbf{v}}_m(t_n, \mathbf{r}) = \mathbf{v}_m(t_n, \mathbf{r}) \tag{3.5}$$

Due to the way the problem is decomposed, source-dependent propagation as in Equation 3.4 is perfectly parallel in both temporal and spatial dimension while the

source-independent propagation as in Equation 3.5 relies on $\tilde{\mathbf{v}}_m(t_n, \mathbf{r})$ and its past values $\bar{\mathbf{v}}_m(t, \mathbf{r}),\ t < t_n$. Therefore, source-dependent problems defined on all space-time blocks are evaluated in the first stage. In the second stage, source-independent propagation is carried on by taking in all necessary initial values yield from last stage.

Source-dependent propagation

Regarding to the source-dependent propagation in Equation 3.4, the solution $\tilde{\mathbf{v}}_m(t, \mathbf{r})$ can be obtained by:

$$\tilde{\mathbf{v}}_m(t, \mathbf{r}) = \int_0^t \exp((\tau - t)\mathbb{A}_m)\tilde{\mathbf{f}}_m(\tau, \mathbf{r})d\tau \tag{3.6}$$

In general, Equation 3.6 does not have analytical solutions. Nevertheless, we recognize that, in many metamaterial applications the source is implemented by applying a time modulation function, $\tilde{\mathbf{g}}_m$, on the eigenmode field at port boundary or the plane wave field at domain boundary. Namely, the source term is denoted by:

$$\tilde{\mathbf{f}}_m(t, \mathbf{r}) = \mathbf{b}_m(\mathbf{r})\tilde{\mathbf{g}}_m^T(t) \tag{3.7}$$

where $\mathbf{b}_m(\mathbf{r})$ is the eigenmode or plane wave field distribution, which only exhibits spatial dependence. A pictorial illustration can be found in Figure 3.3.

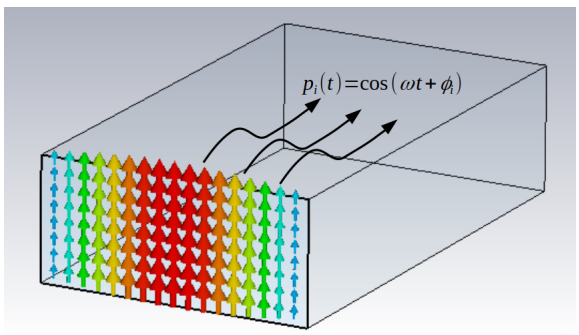


Figure 3.3: Field distribution and time modulation

Therefore, we first construct a low-dimensional Krylov subspace $\mathcal{K}_k(\mathbb{A}_m, \mathbf{b}_m) = \text{span}\{\mathbf{b}_m, \mathbb{A}_m\mathbf{b}_m, \cdots, \mathbb{A}_m^{k-1}\mathbf{b}_m\}$. An orthonormal basis set $\mathbb{W}_k = [\mathbf{w}_1, \cdots, \mathbf{w}_k] \in$

$\mathbb{R}^{n\times k}$ ($n = \dim(\mathbb{A}_m)$) is then built up by the well-known Arnoldi orthogonalization process:

$$\mathbb{A}_m\mathbb{W}_k = \mathbb{W}_k\mathbb{H}_{k,k} + \mathbf{w}_{k+1}h_{k+1,k}\mathbf{e}_k^T \tag{3.8}$$

where $\mathbf{e}_k = [0, \cdots, 0, 1]^T$, $\mathbb{H}_{k,k}$ is the upper Hesssenberg matrix which consists of the orthogonalization coefficients. By the orthogonality of $\mathbb{W}_k$, the Hessenberg matrix $\mathbb{H}_{k,k} = \mathbb{W}_k^T\mathbb{A}_m\mathbb{W}_k$ represents the excitation-aware projection of $\mathbb{A}_m$ onto the Krylov subspace $\mathcal{K}_k$. Note that the dimension $k \ll \dim(\mathbb{A}_m)$.

The solution to Equation 3.6 can now be rewritten as:

$$\begin{aligned}\tilde{\mathbf{v}}_m(t,\mathbf{r}) &= \int_0^t \exp((\tau-t)\mathbb{A}_m)\tilde{\mathbf{f}}_m(\tau,\mathbf{r})d\tau \\ &\approx \mathbb{W}_k\int_0^t \exp((\tau-t)\mathbb{H}_{k,k})\tilde{\mathbf{g}}_m(\tau)d\tau\mathbb{W}_k^{\mathrm{T}}\mathbf{b}_m(\mathbf{r}) \\ &= \beta\mathbb{W}_k\underbrace{\int_0^t \exp((\tau-t)\mathbb{H}_{k,k})\mathbf{e}_1\tilde{\mathbf{g}}_m(\tau)d\tau}_{\tilde{\mathbf{v}}_m^k}\end{aligned} \tag{3.9}$$

where $\beta = \|\mathbf{b}\|$ and $\mathbf{e}_1 = [1, \cdots, 0, 0]^T$. The expression in Equation 3.9 inspires us to obtain the solution to the original problem in Equation 3.4 in the following two steps.

We begin with the computation of time-evolution problem for a reduced subspace matrix $\mathbb{H}_{k,k}$:

$$\frac{d\tilde{\mathbf{v}}_m^k}{dt} = -\mathbb{H}_{k,k}\tilde{\mathbf{v}}_m^k + \mathbf{e}_1\tilde{\mathbf{g}}_m^T(t), \quad \tilde{\mathbf{v}}_m^k(t_n,\mathbf{r}) = 0 \tag{3.10}$$

As the dimension of matrix $\mathbb{H}_{k,k}$ is much smaller than $\mathbb{A}_m$, Equation 3.10 can be solved much faster than the original problem in Equation 3.4.

After $\tilde{\mathbf{v}}_m^k$ is computed, the solution to Equation 3.4 is then sought by $\tilde{\mathbf{v}}_m \approx \beta\mathbb{W}_k\tilde{\mathbf{v}}_m^k$ according to Equation 3.9.

We remark that the reduced matrix $\mathbb{H}_{k,k}$ is constructed once for all repetitive spatial subdomains with the same port excitation. Furthermore, as the space-time relation is decoupled in Eq. 3.7, the same reduced matrix can be reused for different time domain waveforms. The saving of simulation time can be very significant when waveform parameters (e.g. rising time, pulse rate) need to be iterated during design stage.

Source-independent propagation

For the homogeneous, source-independent problem in Equation 3.5, it has the semi-analytical solution: we may follow a similar procedure and construct a Krylov subspace $\mathcal{K}_k(\mathbb{A}_m, \bar{\mathbf{v}}_m^0) = \text{span}\{\bar{\mathbf{v}}_m^0, \mathbb{A}_m\bar{\mathbf{v}}_m^0, \cdots, \mathbb{A}_m^{k-1}\bar{\mathbf{v}}_m^0\}$, with $\bar{\mathbf{v}}_m^0 = \bar{\mathbf{v}}_m(t_n, \mathbf{r})$ represents the initial condition.

After the construction of the projection matrix $\mathbb{W}_k$ and the upper-Hessenberg matrix $\mathbb{H}_{k,k}$, we solve the homogeneous time-evolution problem on the reduced order system:

$$\frac{d\bar{\mathbf{v}}_m^k}{dt} = -\mathbb{H}_{k,k}\bar{\mathbf{v}}_m^k, \quad \bar{\mathbf{v}}_m^k(t_n) = \left\|\bar{\mathbf{v}}_m^0\right\| \mathbf{e}_1 \tag{3.11}$$

Note that Equation 3.11 is solved by $\bar{\mathbf{v}}_m^k(t_{n+1}) = \exp\left(-T_\text{u}\mathbb{H}_{k,k}\right)\bar{\mathbf{v}}_m^k(t_n)$. The solution to the original problem in Equation 3.5 can then be obtained by:

$$\bar{\mathbf{v}}_m(t_{n+1}, \mathbf{r}) \approx \mathbb{W}_k\bar{\mathbf{v}}_m^k(t_{n+1}) = \left\|\bar{\mathbf{v}}_m^0\right\| \mathbb{W}_k \exp\left(-T_\text{u}\mathbb{H}_{k,k}\right)\mathbf{e}_1 \tag{3.12}$$

In summary, given the geometry and material of the problem to be simulated, we first invoke periodicity and identify a finite number of 4D space-time building block. Within each time unit, the entire domain solution is obtained by superimposing solutions from building block subproblems, all of which are solved independently and in parallel. Arbitrarily long timespan simulation is achieved by repetitively applying

building block propagators. To address spatial, temporal multi-scale challenges, the time-evolution in space-time subproblems is performed on the Krylov-subspace based reduced-order model. The work leads to much-desired robustness, accuracy and parallel efficiency for transient EM analysis.

3.1.2 Recursive correction integration scheme

Matrix exponential based integration schemes embrace both good stability and accuracy properties. But in order to out-perform explicit time stepping schemes, the matrix exponential integrators need to deal with memory cost among other implementation issues. Solving for the reduced matrix system in Equation 3.10 and building up the Krylov subspace $\mathcal{K}_k(\mathbb{A}_m, \mathbf{b}_m)$ are two major computationally intensive steps in source-dependent propagation. For time unit T_u, the dimension of Krylov subspace could be growing drastically depending on the accuracy required. To tackle this problem, a residual based restarting method is proposed to strike a balance between accuracy and computational cost. The source term $\tilde{\mathbf{f}}_m(t, \mathbf{r}) = \mathbf{b}_m(\mathbf{r})\tilde{\mathbf{g}}_m^T(t)$ is actually a rank-1 matrix if $\tilde{\mathbf{g}}_m(t)$ takes samples at $T_L = \{t_1, t_2, ..., t_L\}$. Simply choose the initial guess solution $\tilde{\mathbf{v}}^0(t) = 0$ and the dependency on $\mathbf{r}$ and subscript m are omitted for brevity of notations. After construction of Krylov subspace of dimension k and solving for Equation 3.10, the residual of Equation 3.10 corresponding to approximated solution $\tilde{\mathbf{v}}^1(t)$ can be written as,

$$r^1(t) = -\mathbb{A}\tilde{\mathbf{v}}^1(t) - \tilde{\mathbf{v}}^1(t)' + \mathbf{b}\tilde{\mathbf{g}}^T(t) \tag{3.13}$$

The error of solution which is measured by difference $\tilde{\mathbf{e}}(t) = \tilde{\mathbf{v}}^1(t) - \tilde{\mathbf{v}}(t)$, forms a correction equation as,

$$\frac{d\tilde{\mathbf{e}}(t)}{dt} = -\mathbb{A}\tilde{\mathbf{e}}(t) - r^1(t),\ \tilde{\mathbf{e}}(0) = 0 \tag{3.14}$$

,where $r^1(t)$ is another rank-1 matrix evalutated at samples T_L. To see this, just combine Equation 3.8 and Equation 3.10, then re-write $r^1(t)$ as,

$$\begin{aligned} r^1(t) &= \mathbf{b}\tilde{\mathbf{g}}^T(t) - \beta(\mathbb{W}_k\mathbb{H}_{k,k} + \mathbf{w}_{k+1}h_{k+1,k}\mathbf{e}_k)\tilde{\mathbf{v}}^k(t) - \beta\mathbb{W}\tilde{\mathbf{v}}^k(t)' \\ &= \mathbf{b}\tilde{\mathbf{g}}^T(t) - \beta\mathbb{W}_k(\mathbf{e}_1\tilde{\mathbf{g}}^T(t)) - \mathbf{w}_{k+1}h_{k+1,k}\mathbf{e}_k\tilde{\mathbf{v}}^k(t) \\ &= \mathbf{b}\tilde{\mathbf{g}}^T(t) - \mathbf{b}\tilde{\mathbf{g}}^T(t) - \mathbf{w}_{k+1}h_{k+1,k}\mathbf{e}_k\tilde{\mathbf{v}}^k(t) \\ &= -\mathbf{w}_{k+1}h_{k+1,k}\mathbf{e}_k\tilde{\mathbf{v}}^k(t) \end{aligned} \tag{3.15}$$

Here $\mathbf{w}_{k+1}$ is the last column in $\mathbb{W}_k$ and identity $\mathbf{e}_1\tilde{\mathbf{f}}(t) = \mathbf{V}_k^{\mathrm{T}}r_0(t)$ is used. Set the initial approximated solution $\mathbf{u}_0(t) = 0$ then compare Equation 3.14 with original problem Equation 3.4, we can find the analogy such as

$$\mathbf{w}_{k+1} \to \mathbf{b} \quad , -h_{k+1,k}\mathbf{e}_k\tilde{\mathbf{v}}^k(t) \to \tilde{\mathbf{g}}(t) \tag{3.16}$$

We can solve for a corresponding correction equation like Equation 3.14 by replacing $\mathbf{b}$ by $\mathbf{w}_{k+1}$ and $\tilde{\mathbf{g}}(t)$ by $h_{k+1,k}\mathbf{e}_k\tilde{\mathbf{v}}^k(t)$. The original solution is sought after by $\tilde{\mathbf{v}}(t) = \tilde{\mathbf{v}}^1(t) - \tilde{\mathbf{e}}(t)$. This process can be conducted recursively as described below.

Algorithm 2 Recursive correction integration scheme

Input: $\mathbb{A}, \mathbf{b}, \tilde{\mathbf{g}}(t)$, toler

Output: $\mathbf{y}(t)$, solution to Equation 3.4

$l \leftarrow 0, \tilde{\mathbf{v}}(t) \leftarrow 0, \mathbf{b}_l \leftarrow \mathbf{b}, \tilde{\mathbf{g}}_l(t) \leftarrow \tilde{\mathbf{g}}(t)$

function $\mathbf{y}(t) = \text{RecursiveCorrection}(\tilde{\mathbf{v}}, \mathbf{b}_l, \tilde{\mathbf{g}}_l(t))$

Solve for correction equation with Krylov subspace dimension k : $\tilde{\mathbf{e}}_l(t)' =$

$-\mathbb{A}\tilde{\mathbf{e}}_l(t) + \mathbf{b}_l\tilde{\mathbf{g}}_l(t),\ \tilde{\mathbf{e}}_l(0) = 0$

$r^l(t) = \|\mathbf{w}_{k+1}h_{k+1,k}\mathbf{e}_k\tilde{\mathbf{v}}^k(t)\|$

if $r^k <$ toler **then**

$\mathbf{y}(t) \leftarrow \tilde{\mathbf{v}}(t) - \tilde{\mathbf{e}}_l(t)$

else

$l \leftarrow l + 1,\ \mathbf{b}_l \leftarrow \mathbf{w}_{k+1},\ \tilde{\mathbf{g}}_l(t) \leftarrow -h_{k+1,k}\mathbf{e}_k\tilde{\mathbf{v}}^k(t)$

$\mathbf{y}(t) \leftarrow \tilde{\mathbf{v}}(t) - \text{RecursiveCorrection}(\tilde{\mathbf{e}}_l(t), \mathbf{b}_l, \tilde{\mathbf{g}}_l(t))$

Here the norm can be understood as either taking several samples in time or maximal in T_L.

Adaptive-recycling Krylov subspace based integration scheme

Since the second stage is still of sequential nature in time, we proposed a adaptive-recycling exponential integration scheme to reduce the computational cost.

The solution to Equation 3.5 is

$$\begin{aligned}\bar{\mathbf{v}}(t) &= \exp(-t\mathbb{A})\bar{\mathbf{v}}(0) \\ &= \bar{\mathbf{v}}(0) + t\varphi(-t,\mathbb{A})\mathbb{A}\bar{\mathbf{v}}(0)\end{aligned} \tag{3.17}$$

,where $\varphi(-t,A) = (I - \exp(-t\mathbb{A}))/(t\mathbb{A})$. Actually $t\varphi(-t,\mathbb{A})\mathbb{A}\bar{\mathbf{v}}(0)$ solves the following equation.

$$\mathbf{p}'(t) = -\mathbb{A}\mathbf{p}(t) + \mathbb{A}\bar{\mathbf{v}}(0),\ \mathbf{p}(0) = 0 \tag{3.18}$$

Let $\mathbf{p}_k(t)$ be the Krylov subspace solution to Equation 3.18, the residual can be written as,

$$\begin{aligned}r(t) &= -\mathbf{p}_k(t)' - \mathbb{A}\mathbf{p}_k(t) + \mathbb{A}\bar{\mathbf{v}}(0) \\ &= -h_{k+1,k}\left\|\mathbb{A}\bar{\mathbf{v}}(0)\right\|\mathbf{w}_{k+1}\mathbf{e}_k^T\varphi(-t,\mathbb{H}_{k,k})\mathbf{e}_1\end{aligned} \tag{3.19}$$

By inspection of Equation 3.19, $r(t)$ depends on t the residual can be evaluated on the fly as Krylov subspace k grows. Literature something shows link of residual and error. A proper t corresponding to preset accuracy level can be found by evaluating $\varphi(-t,\mathbb{H}_{k,k})\mathbf{e}_1$. [89] reveals that residual grows as t increases. The time step estimator algorithm is designed based on bi-section search and described in Algorithm 3.

Algorithm 3 Time Step Estimator

Input : $\mathbf{H}_m, h_{m+1,m}$, toler

Output : Δt

$t \leftarrow t_{\text{guess}}$(Large enough)

$r \leftarrow h_{m+1,m}\mathbf{e}_m^{\mathrm{T}}(e^{-t\mathbf{H}_m}\mathbf{e}_1)$

$k_l = 0,\ k_r = 1.0$

while $\|k_r - k_l\| >= 1e^{-6}$ **do**

$k_m \leftarrow \frac{k_l+k_r}{2}$

$r \leftarrow h_{m+1,m}\mathbf{e}_m^{\mathrm{T}}(\varphi(-k_m t, \mathbf{H}_m)\mathbf{e}_1)$

if $r <$ toler **then**

$k_l \leftarrow k_m$

else

$k_r \leftarrow k_m$

$\Delta t = k_m t_{\text{guess}}$

The Krylov subspace $\mathcal{K}_k(\mathbb{A}, \mathbb{A}\bar{\mathbf{v}}(0))$ can be recycled to advance a few steps(N_{recycle}) forward. Here a approximation is employed such that $\mathcal{K}_k(\mathbb{A}, \mathbb{A}\bar{\mathbf{v}}(t)) \approx \mathcal{K}_k(\mathbb{A}, \mathbb{A}\bar{\mathbf{v}}(0))$. The schematic of this integration scheme is shown in Figure 3.4 as well as algorithmic description in Algorithm 4.

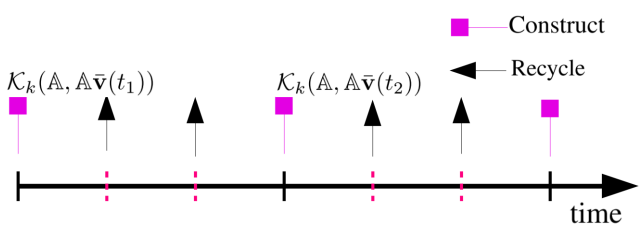


Figure 3.4: Schematic of Adaptive-Recycling φ integrator

Algorithm 4 Adaptive-Recycling φ integrator

Input : $T_{\text{end}}, \mathbf{u}(0), \mathbb{A}, m, N_{\text{recycle}}, \text{toler}$

Output : $\mathbf{u}(t_k), k = 1, ...,$

$\mathbf{v} \leftarrow \mathbf{u}(0)$, t=$t_1$

while $t < T_{\text{end}}$ **do**

$\mathbf{w}_1 \leftarrow \mathbb{A} * \mathbf{v}$

$\mathbb{AW}_m \leftarrow \mathbb{W}_m \mathbb{H}_{m,m} + h_{m+1,m} \mathbf{w}_{m+1} \mathbf{e}_m^{\mathrm{T}}$

$\Delta t \leftarrow \text{TimeStepEstimator}(\mathbb{H}_{m,m}, h_{m+1,m}, \text{toler})$

$\mathbf{y}_m \leftarrow \Delta t \varphi(-\Delta t, \mathbb{H}_{m,m}) \mathbf{e}_1$

$\mathbf{v} \leftarrow \mathbf{v} + \mathbb{W}_m * \mathbf{y}_m$

for $i = 1, N_{\text{recycle}}$ **do**

$\mathbf{x} \leftarrow \mathbb{A} * \mathbf{v}$

$\mathbf{g} \leftarrow \mathbb{W}_m^{\mathrm{T}} \mathbf{x}$

$\mathbf{y}_m \leftarrow \Delta t \varphi(-\Delta t, \mathbb{H}_m) \mathbf{g}$

$\mathbf{v} \leftarrow \mathbf{v} + \mathbb{W}_m \mathbf{y}_m$

$\mathbf{u}(t_k) \leftarrow \mathbf{v}, \ k = k + 1$

$t = t + \Delta t$

3.2 Numerical Performance

3.2.1 Accuracy and Convergence

Recursive correction integration

To verify the algorithm proposed in Algorithm 2, first we define a problem as

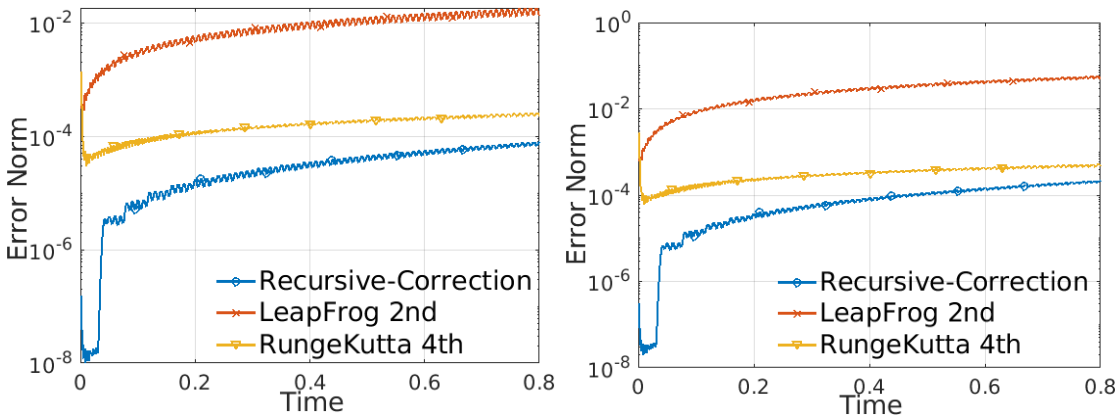
$$\frac{d\mathbf{y}(t)}{dt} = -\mathbb{A}\mathbf{y}(t) - \mathbb{A}\mathbf{u},\ \mathbf{y}(0) = 0 \tag{3.20}$$

where $\mathbb{A} \in \mathbb{R}^{2000\times2000}$ comes from spatial discretization with upwind flux and Lagrange polynomial basis function of a one-dimensional Maxwell problem defined in $x \in [-1, 1]$. The boundary conditions at $x = 1, 1$ are set to be PEC. Equation 3.20 admits a analytic solution $\mathbf{y}(t) = \exp(-t\mathbb{A})\mathbf{u} - \mathbf{u}$, which can be yielded by high accuracy matrix exponential algorithms. The norm of absolute error can be then examined by $\|e_{\text{abs}}\| = \|\mathbf{y}_{\text{analytic}} - \mathbf{y}\|$. Here $\mathbf{u}$ can be understood as the initial condition for homogeneous IVP $\mathbf{z}(t)' = -\mathbb{A}\mathbf{z}(t),\ \mathbf{z}(0) = \mathbf{u}$. Initial conditions $\mathbf{u} = [\mathbf{E}(t_0), \mathbf{H}(t_0)]^T;\ \mathbf{E}(t_0) = sin(k\pi x),\ k = 8, 16, 24$ with different frequency components are tested respectively. Time stepping schemes like LeapFrog 2nd order(LP2) and Runge-Kutta 4th order(RK4) are also compared in Table 3.1.

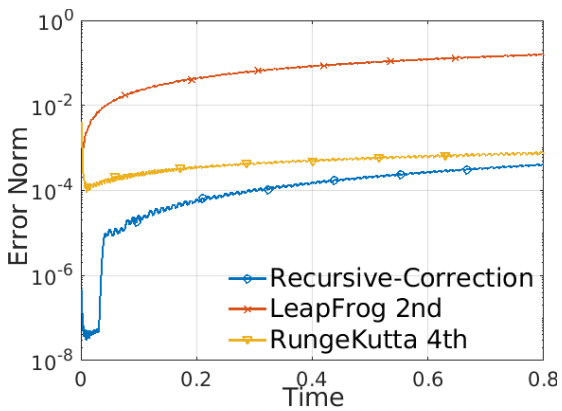
Table 3.1: Runtime Statistics

Methods	**MxV +ODE solve MxV**	**Runtime(k=8/16/24)**
Proposed	1920(24×80)+8000×80×24	5.28/5.28/5.29
LP2	4400(2200×2)	5.57/5.68/5.37
RK4	4400(1100×4)	5.63/5.49/5.6

The corresponding error plots versus time for different frequency components are shown in Figure 3.5.



(a) k=8 (b) k=16



(c) k=24

Figure 3.5: Error evolution with various frequency components

Within the same amount of time, recursive correction integration owns the best accuracy. The convergence history($k = 8$ case) of residual with respect to error norm is illustrated in Figure 3.6. Different choice of Krylov subspace dimension are made. Here total dimension indicates the counts of Krylov basis combined for each level of correction as algorithm goes.

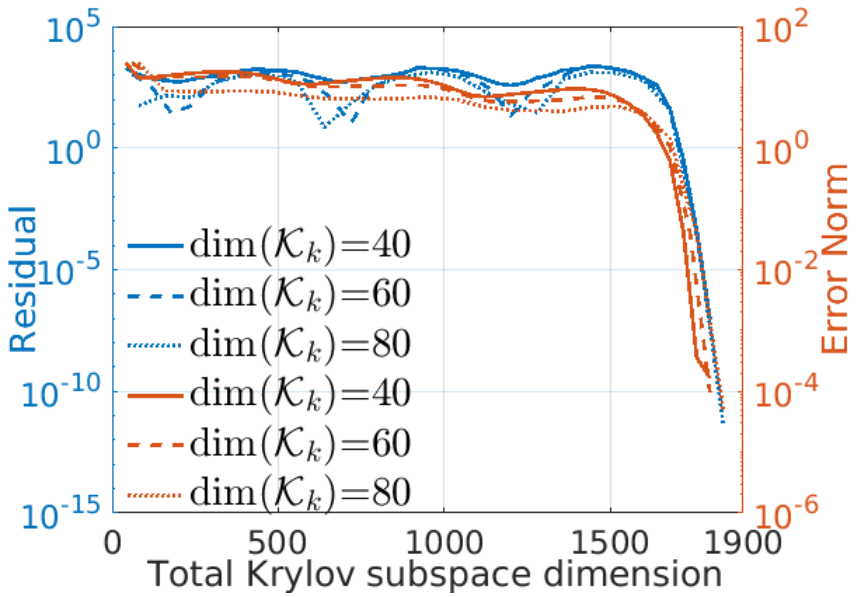


Figure 3.6: Convergence history of initial condition k=8

It is easy to tell that residual actually a good indicator for norm of error. Under different choices of Krylov subspace dimension, the total number basis which is directly associated with matrix-vector multiplication(MxV) operations ,keeps stable with the level of correction just being different. The smaller $\dim(\mathcal{K}_k)$ is, the deeper level of correction is required. So the dimension can be determined on the fly depending on the memory available. After certain level of correction, it's only takes a few more computations to reach higher level of accuracy without notably additional cost of storage since a Krylov subspace of dimension at most k is only required all the time. Next 4 levels of accuracy$(10^{-3}, 10^{-4}, 10^{-5}, 10^{-6})$ are tested with runtime statistics shown in Table 3.2.

Table 3.2: Runtime Statistics (in sec)

Method	**MxV + ODE solve(MxV)**	**Runtime**
Proposed	(240*8)+(1.2k/3.7k/12k/36k)*240*8	3.96/4.93/7.89/16.69
LP2	(N.A.)*2	20.42/64.98/207.03/N.A.
RK4	(0.8k/1.35k/2.41k/4.2k)*4	4.07/7.04/12.5/20.26

Proposed method outperforms the other two schemes in each level of accuracy.

3.2.2 Application to source dependent problems

We consider three examples to illustrate the advantages of proposed method in solving source dependent problem here. First S-parameters are evaluated in analysis of a via pair transmission line, shown in Figure 3.7, between two parallel plates, which acts as transition between the top layer and bottom layer transmission lines. The characteristic impedance of each port is set to be 50Ω. The mesh is partitioned into 12 subdomains with total number of degree of freedoms 1.38 million. A T=0.4ns simulation period is conducted.

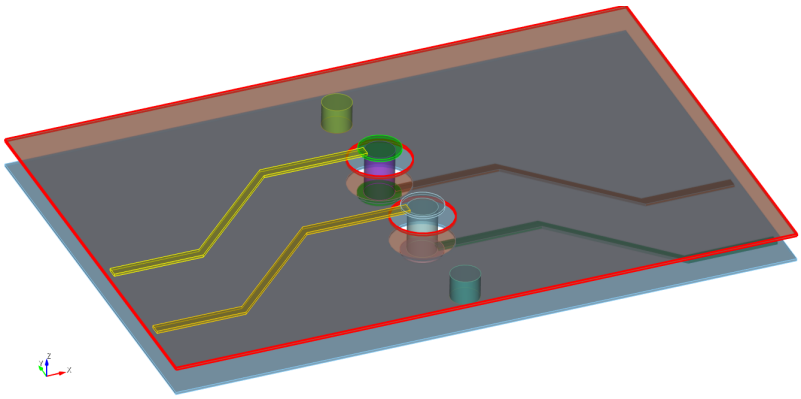
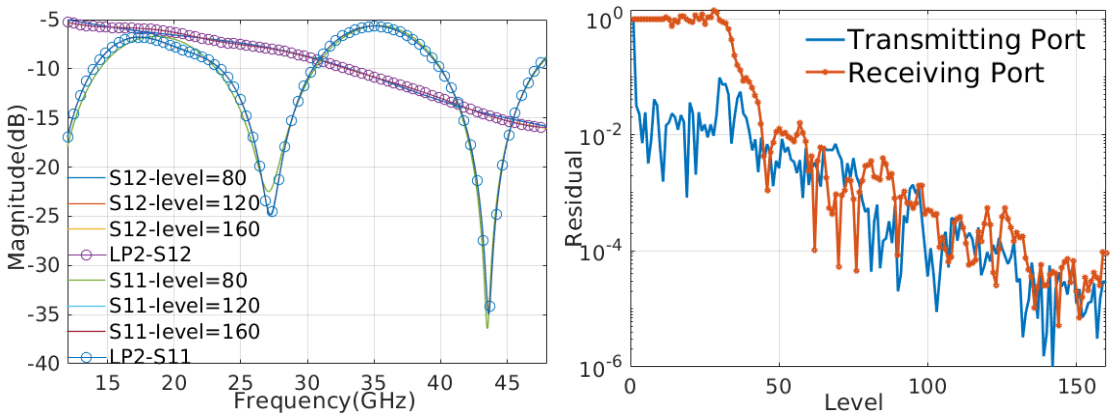


Figure 3.7: Schematic of via pair transmission line

The S-parameters and residual vector $r(t)$ projected to both ports are also monitored in Figure 3.8.



(a) Validation of S-parameters (b) Convergence of residual on ports

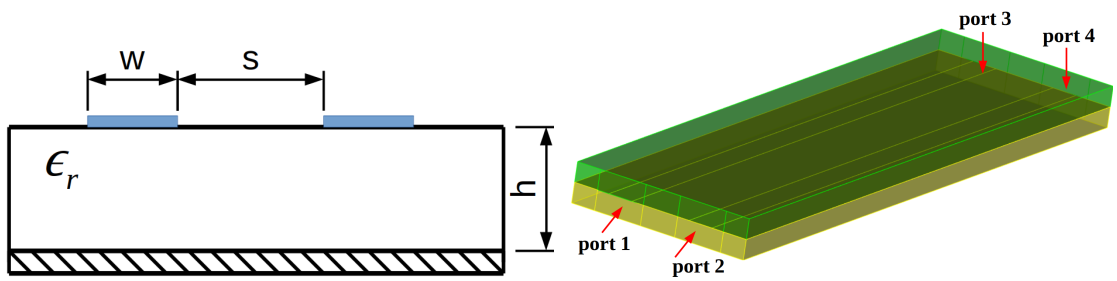
Figure 3.8: S-parameters and convergence history of residual on ports

The run-time statistics of proposed method compared with LP2 is compared in Table 3.3.

Table 3.3: Computational statistics

Methods	**MxV**	**Coupling**	**Ode solve**	**Orth**	**Total**
Proposed	8.5min	60.5s	159s	263.6s	16.5min
LP2	40min	–	–	–	40min

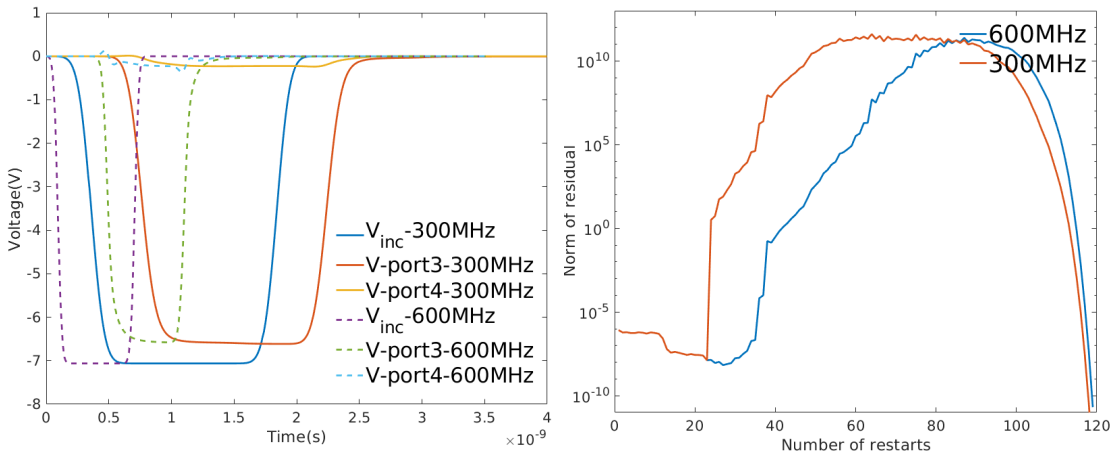
The second example is a parallel microstrip line excited by rectangular pulses and the goal is to evalute the crosstalk w.r.t parameters of waveform. The geometric settings of this model is shown in Figure 3.9



(a) Side view (b) 3D view

Figure 3.9: Geometry of parallel microstrip line

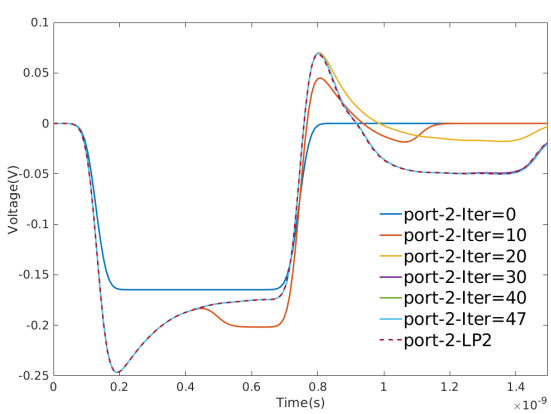
,where w=2mm , s=5mm, h=1.6mm, ϵ_r=4.3 and the length of microstrip line is 50mm. One bit of a single rectangular pulse is used to excite port 1. The widths of the two pulses are set to half of the 300 MHz and 600 MHz signals. The corresponding data rates for each frequency are 600 Mbit/s and 1.2 Gbit/s, respectively. Smoothing is necessarily applied to remove undesirable high-frequency components from the signal. The voltages are monitored as shown in Figure 3.10 at through port(port 3) and coupled port(port 4). The convergence history for each case are also illustrated. The time domain response of the 1.2 Gbit/s signal is slightly distorted in the beginning at throught port while that of the 600 Mbit/s signal seems to remain undistorted. The time domain response at port 4 shows that the higher data rate signal causes the stronger crosstalk on another signal path.



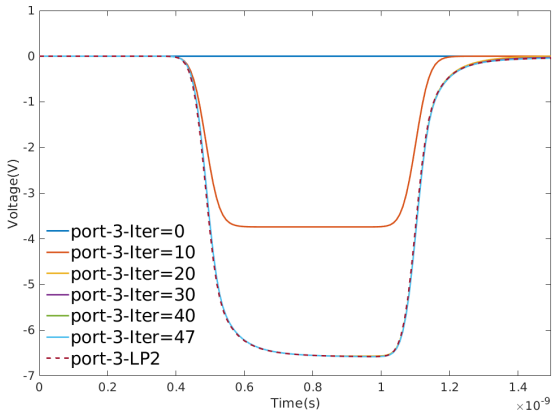
(a) Voltages at ports (b) History of restarting

Figure 3.10: Ports voltages and convergence of residual

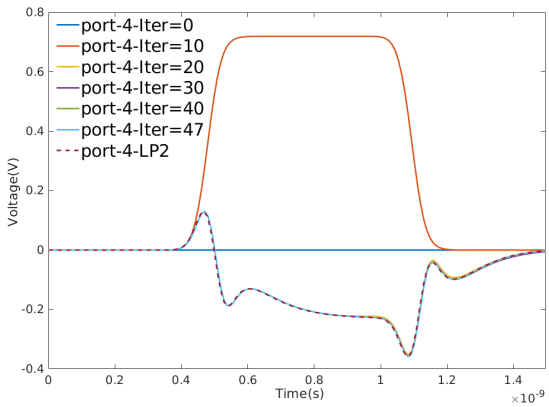
To verify the accuracy, the results yield by 2nd-order LeapFrog time marching scheme is compared with proposed method w.r.t the number of restarts in Figure 3.11. The voltages shown corresponds to 1.2Gbit/s scenario.



(a) Voltages at port 2



(b) Voltages at port 3



(c) Voltage at port 4

Figure 3.11: Ports voltages of parallel microstrip line

The run-time statistics of proposed method compared with LeapFrog scheme is compared in Table 3.4. 6 subdomains are partitioned and T = 4ns.

Table 3.4: Computational statistics

Methods	**Matvec**	**Ode solve**	**Orth**	**Total**
Proposed work	445.9s	191.1s	474.98s	18.9min
LeapFrog	2× 41.5min	–	–	83min

Note that, for traditional time stepping method, 2 separate simulations have to

be conducted corresponding to different excitation. As a compare, the proposed method just need to run once. The speed-up becomes more significant.

The third application is for modeling high accuracy modeling of microstrip discontinuities using time domain reflectometry(TDR). Several factors affect a TDR system's ablility to resolve discontinuities, like rise time, settling time and pulse aberrations of stimuls signal. Two neighboring discontinuities may be indistinguishable to the measurement instrument if the distance between them amounts to less than half the system rise time. In order to illustrate the proposed method performance, a step-like discontinuity of microstrip line with mentioned design parameters is simulated. The settings are listed in Figure 3.12, where $W_{sub} = 12mm, L_{sub} = 18mm, W_f = 1.5mm, W_s = 4, L_s = 2mm, h = 0.8mm$. 5 pulses with different rise time are launched into port 1.

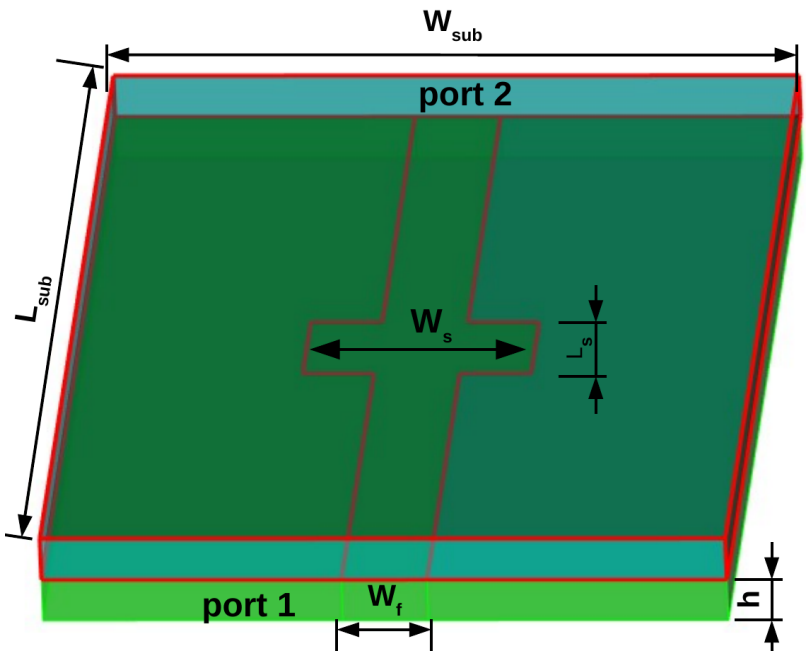


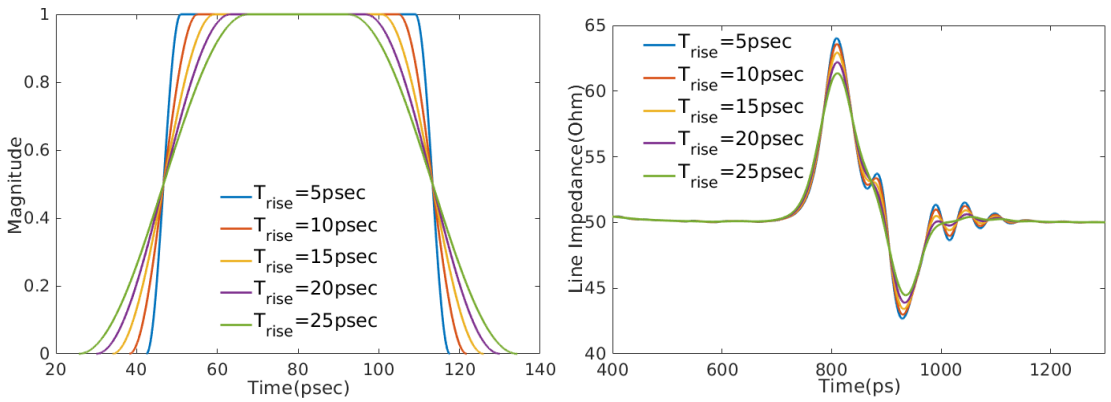
Figure 3.12: Geometry of discontinuous microstrip line

As shown in Figure 3.13, as the rise time of excitation signal increases from 5psec to 25psec, the line impedance at the center local of step discontinuity is varied from 64.04 to 61.4Ω. So the line impedance is controllable by the rise time. The time

dependent line impedance is expressed as,

$$\rho(t) = \frac{V_{ref}(t)}{V_{inc}(t)}, \;\; Z_L(t) = Z_0 \frac{1+\rho(t)}{1-\rho(t)}, \tag{3.21}$$

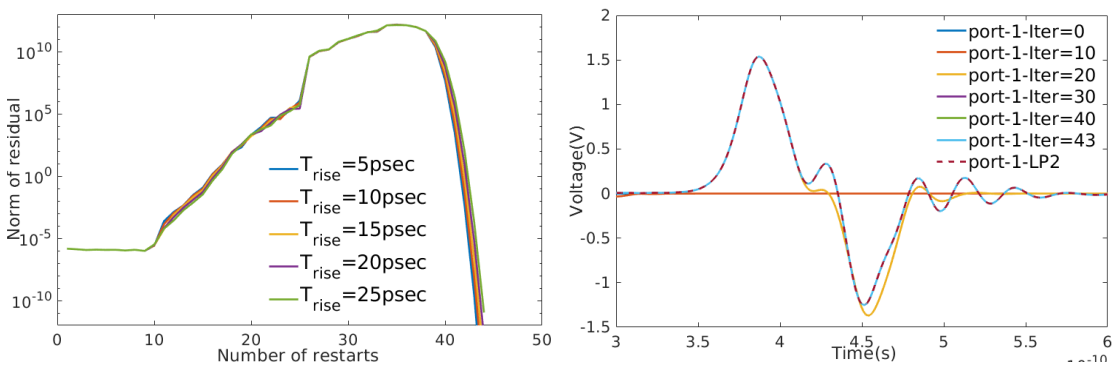
where Z_0 and ρ are characteristic impedance and reflection coefficient.



(a) 5 pulses with different rising time (b) Resulting line impedance

Figure 3.13: Time domain results of discontinuous microstrip line

The convergence history for each rise time case are shown in Figure 3.14 as well as the voltage compared with time marching scheme for T_{rise}=5psec case.



(a) Convergence of residual (b) Convergence of voltage of port 1

Figure 3.14: Convergence history of discontinuous microstrip line

The run-time statistics are listed in table Table 3.5

Table 3.5: Computational statistics

Methods	**Matvec**	**Ode solve**	**Orth**	**Couple**	**Total**
Proposed work	152.6s	70.9s	47.8s	7.6s	4.6min
LeapFrog	5× 6.1min	–	–	–	30.5min

Adaptive-Recycling φ integrator

Figure 3.15 gives the residual of Equation 3.20 evolving with time. Clearly, the monotonically increasing residual curve guarantees the bi-section search works to find a time step such that the norm of is small enough.

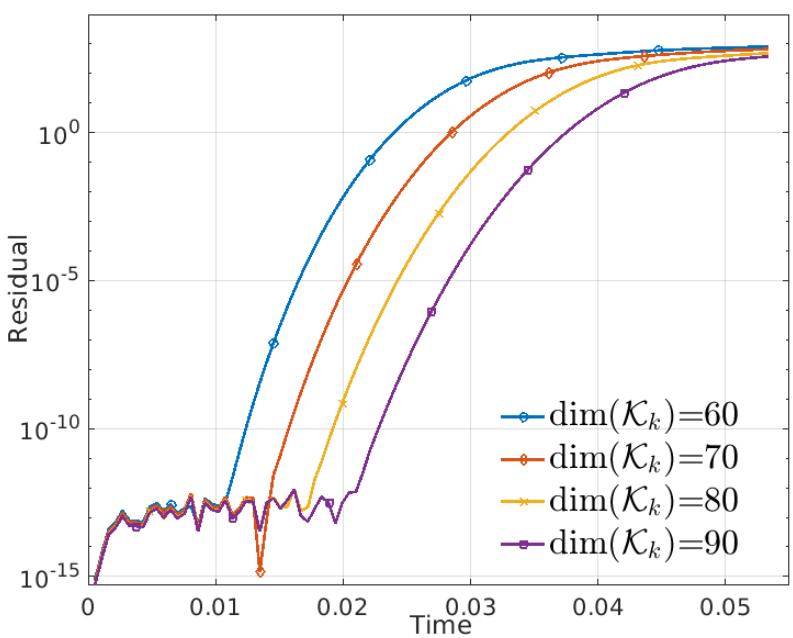


Figure 3.15: Residual evolving with time

In the first example, the number of recycle steps is fixed to 4 and different dimensions of Krylov subspace are tested to observe the convergence of error. Here the time step size is also fixed.

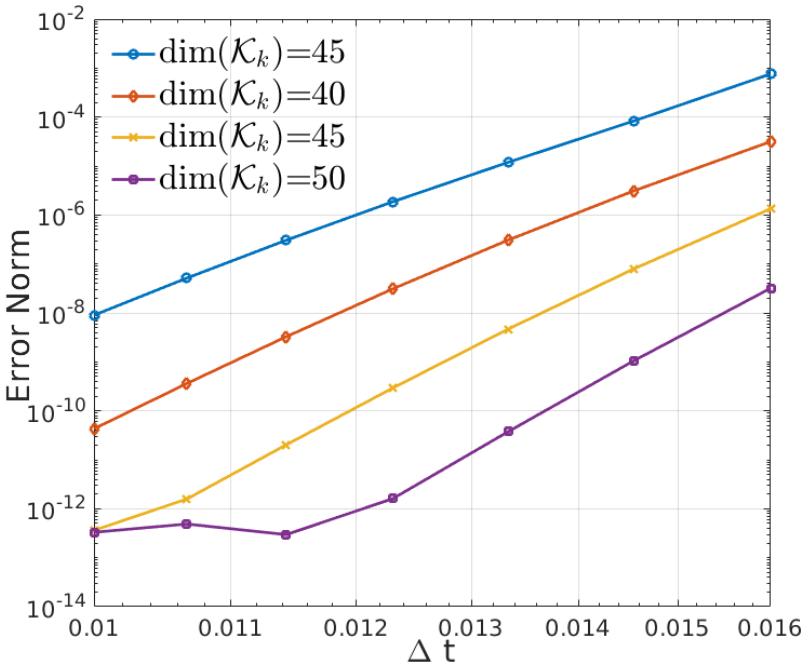


Figure 3.16: Convergence of error with fixed time step

Concluded from Figure 3.16, for each case of Krylov subspace dimension, adaptive-recycling φ integrator shows fast convergence with respect to step size. And large dimension of Krylov subspace helps it converge fast. The roll-off at $\dim(\mathcal{K}_k) = 50$ case indicates that proposed algorithm reaches the same level of accuracy with the matrix exponential algorithm. Next, different number of recycle steps is tested. The convergence of error is shown in Figure 3.17. Here the step size is produced by the time step estimator.

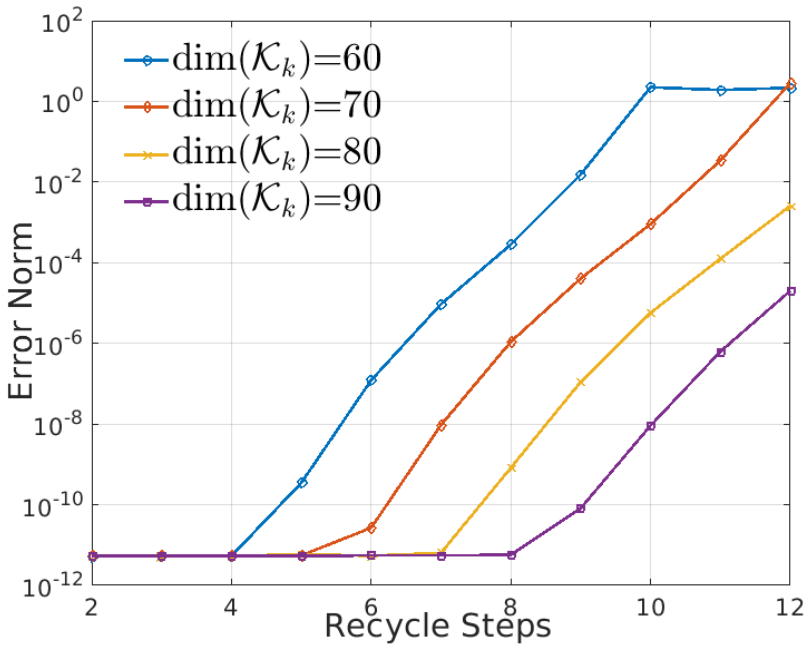


Figure 3.17: Convergence of error with various recycle steps

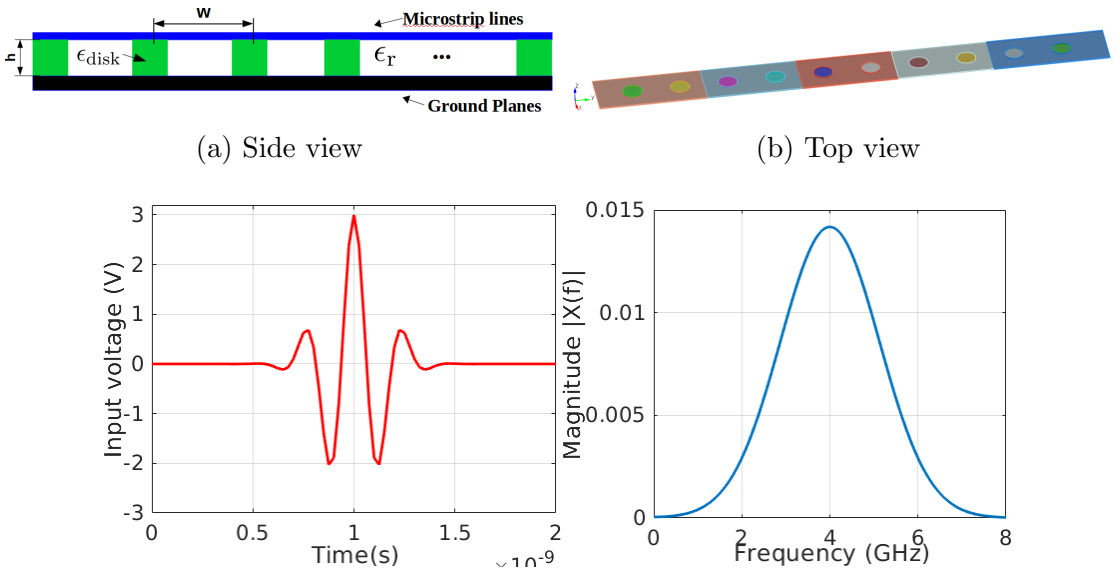
With the increasing of recycle steps, the algorithm losses accuracy although large dimension of Krylov subspace helps. So in practice, trade-off needs to be made between accuracy and efficiency.

3.3 Numerical Examples

3.3.1 Repetition unit in 1D: Photonic-Bandgap Microstrip

We consider the time domain simulation of a microstrip photonic-bandgap (PBG) structure [90]. The geometry of the PBG structure is shown in Figure 3.18. The thickness h and width w of microstrip are 0.5mm and 19.625mm, respectively. There are 10 dielectric disks ($\epsilon_{\text{disk}} = 9$) between the microstrip line and the ground plane. The background permittivity $\epsilon_{\text{r}} = 1$. The disk spacing is equal to the microstrip width and the disk radius is 3.925mm. A broadband (2-8 GHz) , Gaussian modulated

pulse is launched at the input port. The waveform is described by $\tilde{\mathbf{f}}(t) = \cos(\omega(t - t_0))\exp(-\frac{(t-t_0)^2}{\tau^2})$, $\omega = 4$GHz, t_0=1ns and τ =0.2ns and also shown in Figure 3.18.



(a) Side view

(b) Top view

(c) Waveform of excitation

Figure 3.18: Problem setup for photonic bandgap example

The spatial domain of the PBG microstrip is decomposed into $M = 5$ subdomains. The time period [0, 2ns] is divided into $N = 20$ time windows with time unit of 0.1ns. To illustrate the performance of proposed work, we also consider a standard time-stepping scheme, Leap Frog 2nd order method. The maximum time stepping size is chosen as 5.8×10^{-5} ns. Since T_u is set to be 0.1ns, only the first subdomain block is needed in the source dependent problem. So only 1 process is used to run the job. To glue the 20 snapshots of field between 0.1ns to 2ns, a integration with adaptive-recycling φ algorithm is carried out on 5 processes and each process spans a task over 2 or 3 blocks. We have the total field distribution for all 5 sub-domains at the end of 2ns shown. Here for the adaptive-recycling φ integrator, a fixed Krylov subspace dimension is used instead of time step estimator. The runtime statistics of this two stages for 0-2ns are summarized in Table 3.6. As a compare, it takes 2440s for LeapFrog scheme, where a nearly 10x speed-up is achieved. The electric

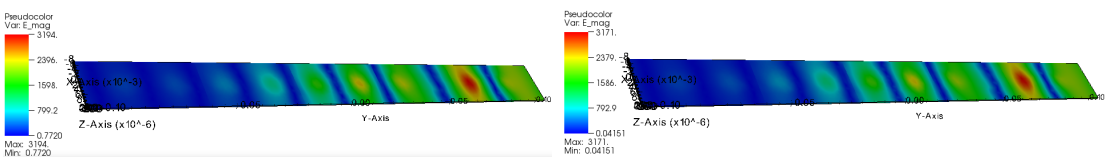
field plots obtained by using both LeapFrog scheme and proposed work are shown in Figure 3.19. All experiments are conducted on a workstation with Intel Core i7-6700HQ.

Table 3.6: Computational statistics(in sec)

Stage	**Matvec**	**Ode solve**	**Mat-expo**	**Total**
Source dependent	9.42	0.52	0.27	10.21

Table 3.7: Computational statistics(in min)

Stage	**Proc 0**	**Proc 1**	**Proc 2**	**Proc 3**	**Proc 4**	**Maximum**
Source independent	2.5	4.24	3.08	3.05	2.11	4.24



(a) LF2 (b) Proposed

Figure 3.19: Fields of PBG simulation at 2ns

The rest of the simulation from 2ns to 12ns for calculating S-parameters is conducted on a cluster(Intel(R) Xeon(R) CPU E5-2640,2.60GHz) and higher order of basis function is used. With the refinement of mesh, construction of Krylov subspace consumes increasing memory. Various Krylov subspace dimensions are compared. The run time is also shown in Table 3.8.

Table 3.8: Computational statistics(in sec)

Methods(Krylov dim)	**Matvec+Orth**	**Mat-expo**	**Total**	**Memory**/Gb
Proposed(no restart)	583.8	13.2	597	1.92
Proposed(50)	502.96	7.3	510.26	0.64
Proposed(100)	500.2	8.72	509.2	0.96
LeapFrog	5430	–	5430	0.16

The calculated S-parameters are plotted in Figure 3.20, and compared to those from the Rayleigh multipole method [90]. We observe a very good agreement and also clearly the bandgap effect of this structure.

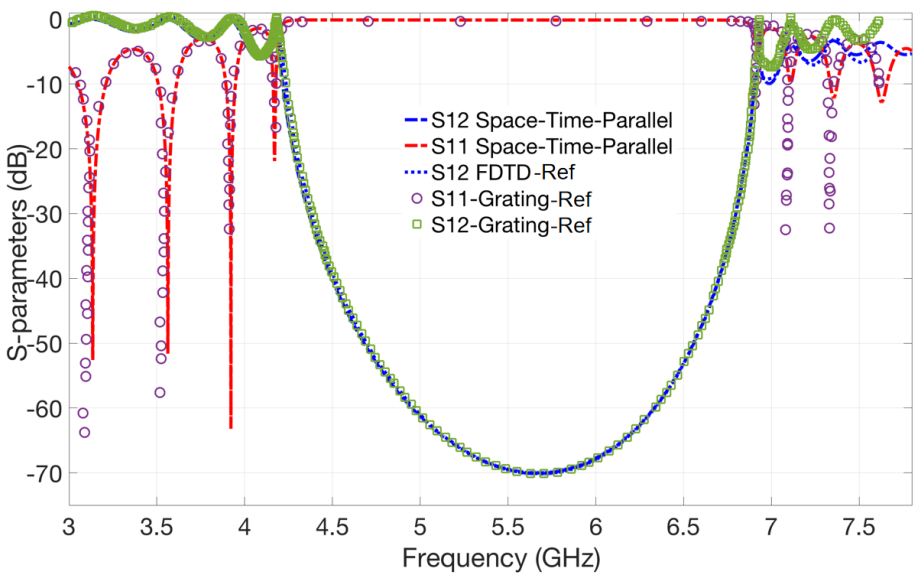
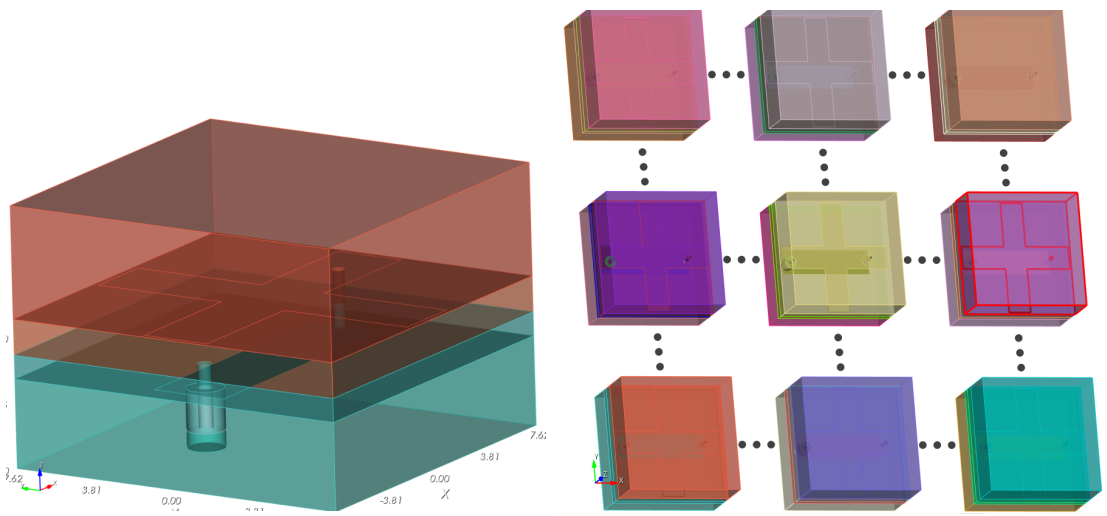


Figure 3.20: S-parameters of PBG microstrip

3.3.2 Repetition unit in 2D: metasurface antenna

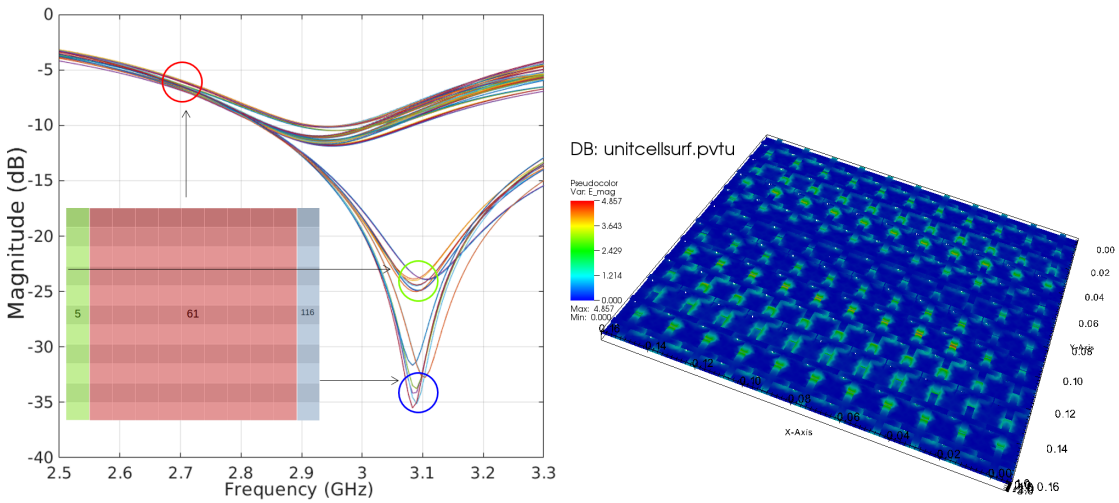
Next we simulate a metasurface antenna array [91], which has repetitions in both x and y direction. A modulated Gaussian pulse(ω=4GHz, t_0 = 1ns, τ=0.25ns), similar to 1D case above, is launched at port of each unit cell. The setup for array can be found in Figure 3.21.



(a) Unit cell (b) Arrangement of array

Figure 3.21: Setup of meta-surface antenna array

9 distinct unit cells are necessary to incorporate all possible boundary conditions. Each unit cell has 135,574 DoFs. In the setup of space-time building block, a typical sub-problem spans a 3x3 block array(2x2 or 2x3 if located on the boundaries). Thus there will be 25 distinct types of scenarios in total. Accordingly, the time unit is set to be 0.05ns which leads to 40 parallel source dependent sub-problems. Simulation periods are chosen to be 2ns for both first and second stage. The voltages on each port of the 11x11 array and surface electric fields are shown in Figure 3.22.



(a) S_{11} of 11x11 metasurface antenna array (b) Surface electric field

Figure 3.22: Simulation results of meta-surface antenna array

We can clearly see that antennas in different locations display various radiation properties. This reveals the necessity of full-wave modeling for antenna array simulation. For the first stage, the run time statistics is illustrated by Figure 3.23 and the stopping criterion of residual is set to be 1×10^{-12}.

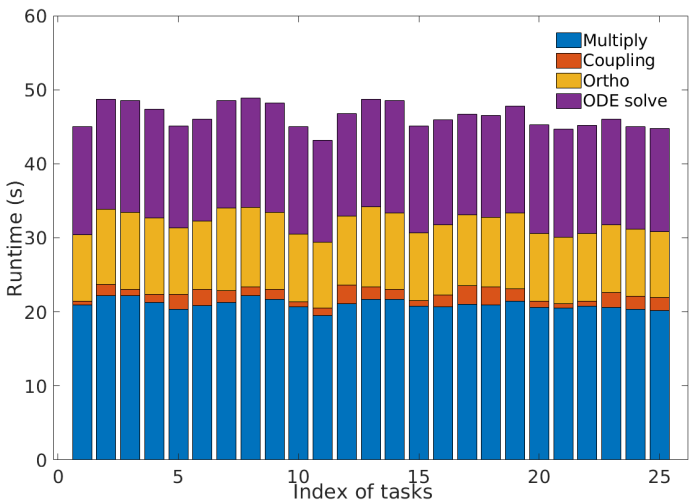


Figure 3.23: Stage 1 run time statistics

Each task spans on 3x3(or 2x3, 2x2) blocks. Run time is 50s on average and a good balance among them is also observed. To run this, a total number of 169

cores on 11 computing nodes with Intel(R) Xeon(R) CPU E5-2640, 2.6GHz CPU are used. To study the scaling performance and robustness of proposed work, we assemble various arrays of size 5x5, 8x8 and 11x11 using the metasurface antenna unit cell and the state vectors computed from the first stage. The voltages of some representing unit cells(indexed 5, 61 and 116) are shown in Figure 3.24.

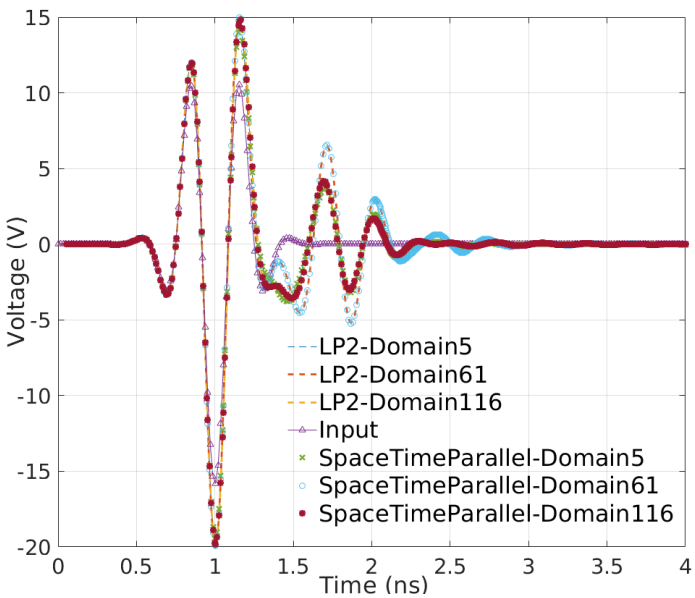


Figure 3.24: Voltages computed from proposed work

A good agreement is observed between proposed work and LeapFrog 2nd scheme. To consider both the accuracy and efficiency, the number recycle steps is set to be 2. The dimension of Krylov subspace is 120 and stopping criterion for residual is 1×10^{-12}. Figure 3.25 lists the step sized computed by the time step estimator algorithm.

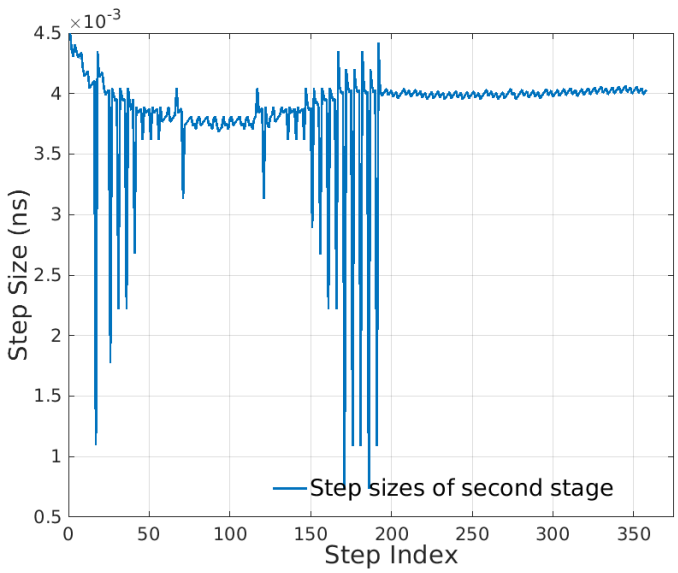


Figure 3.25: Step sizes for stage 2

The drastic change of step size in the first half indicates the in-take of state vectors from stage 1 at interfaces of time windows. Lastly, the run time statistics reveals a roughly 3x speed-up over LeapFrog scheme as in Table 3.9.

Table 3.9: Computational statistics (min)

Array	**MxV**	**Coupling**	**Orth**	**ODE solve**	**Total**	**LP2**
5x5	19.96	1.7	31.4	0.4	53.5	225.4
8x8	20.6	1.4	45.7	0.4	68.1	231.1
11x11	20.2	1.1	48.2	0.4	69.9	220.9

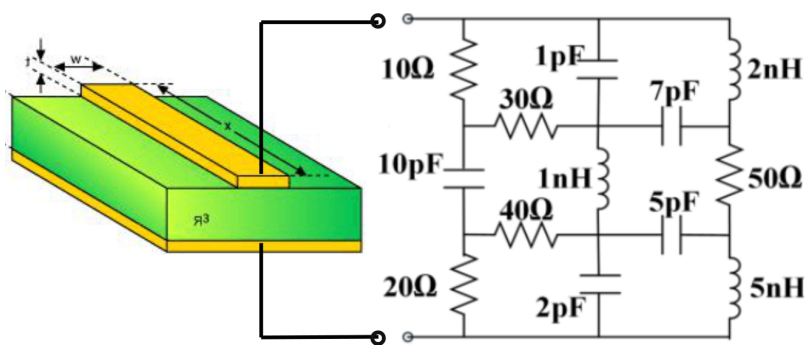
Chapter 4

Parallel-in-time algorithm for electromagnetics circuit co-simulation

In this chapter, we extend the parallel in time method for EM-circuit co-simulation. Modern integrated circuits(IC) have been used in high integration density and very fast operating speed. In order to accurately simulation both EM fields and circuit devices, a efficient tool like cosimulation is inevitable. E.g. SPICE model based general simulators are considered here for conducting circuit simulation and transient solvers for EM full-wave simulation. The EM and circuit solvers are coupled via surface ports under a self-consistent coupling scheme. One challenge is that these two domains are dominated by two sets of physics and always come in significantly different scales in terms of spatial unknowns and size of time step. This chapter develops a parallel in time(PIT) integration scheme for cosimulation and a macro-modeling approach is also proposed under the PIT framework to further accerate the cosimulation.

4.1 EM-Circuit Co-simulation Formulation

We use a microstrip line waveguide terminated with a device or circuit network as a example in Figure 4.1. There are two major parts of this model, EM modeled by full-wave Maxwell equation and circuit modeled by modified nodal analysis(MNA).



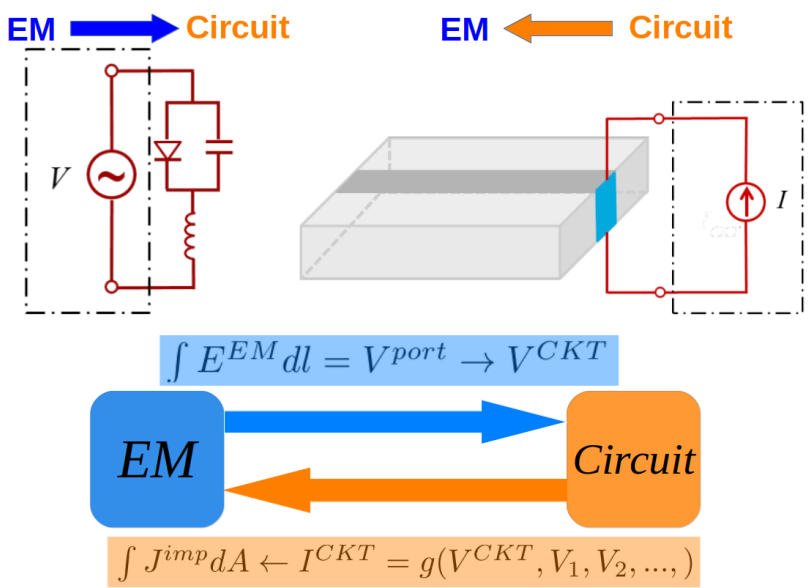
(a)

Figure 4.1: EM-circuit co-simulation model

The first thing needs to be addressed is the coupling between these two domains.

4.1.1 EM-circuit coupling scheme

Here a self-consistent scheme to couple two parts is considered which ensures that the voltages and electric currents on the coupling port are equivalent. $I^{CKT} = I^{EM}$, $V^{EM} = V^{CKT}$. The information flows in the way that: a) EM model acts as voltages source to circuit; (b) circuit model feeds back impressed electric current to EM model.



(a)

Figure 4.2: Information flow for coupling

Here $g(V^{CKT}, V_1, V_2, ...)$ is a nonlinear function taking source voltage and voltages at non-reference nodes of circuit and returning a current flowing through the source port. J^{imp} is the corresponding impressed current density which drives EM model.For the other direction, V^{CKT} is equvalent to line integral of electric field at the surface port. The unknowns for EM model are $\mathbf{u} = [\mathbf{E}, \mathbf{e}^{port}, \mathbf{H}, \mathbf{h}^{port}]^T$ where $\mathbf{E}, \mathbf{H}$ are dofs to the interior of simulation domain and $\mathbf{e}^{port}, \mathbf{h}^{port}$ are associated with the port. For

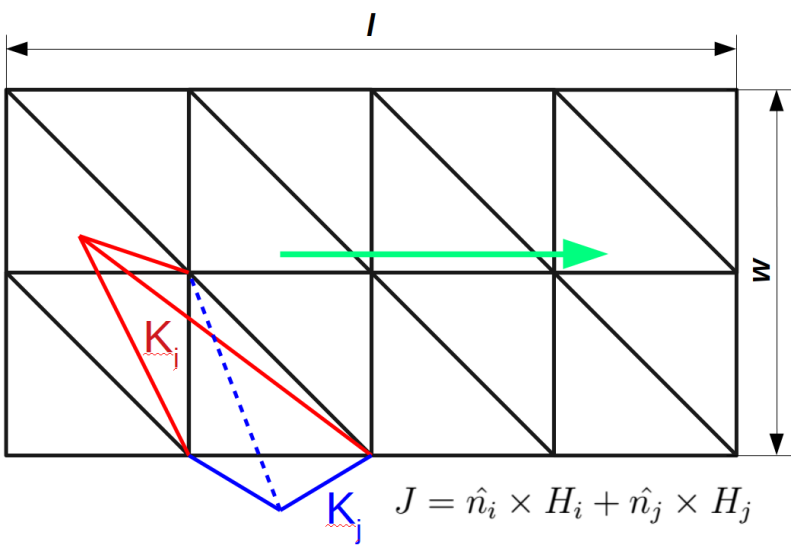
the circuit side, $\mathbf{u} = [V_1, V_2, ..., V_k]$ are non-reference nodes' voltage. Starting from a steady state at time stamp n, assuming $V_n^{CKT} = V_n^{EM}$, we want to find V_{n+1}^{EM}. March Δt in time, doing the following:

$$V_{n+1}^{CKT} \xrightarrow{\text{g}} I_{n+1}^{CKT} = I_{n+1}^{EM} \xrightarrow{\text{f}} V_{n+1}^{EM} \tag{4.1}$$

where f, a linear operator, denotes any time integration method for the EM model.

4.1.2 From circuit to EM

Followed by the same time-domain Maxwell equations and discontinusous Galerkin discretization described in Chapter 2, a surface port boundary condition plays such a role in the vicinity of the circuit with current source. The surface port is modeled as planer impedance surface, illustrated in Figure 4.3.



(a)

Figure 4.3: Illustration of surface port

Here we assume two elements K_i and K_j sharing a face $\partial K_i \cap \partial K_j$. Starting from the boundary conditions of the electric and magnetic fields on the port, we can

obtain the current density on the surface Γ_{CKT} as,

$$\begin{aligned} J_{CKT} = \frac{I_{CKT}}{w} &= \hat{n_i} \times \mathbf{H}_i + \hat{n_j} \times \mathbf{H}_j \\ \mathbf{0} &= \hat{n_i} \times \mathbf{E}_i + \hat{n_j} \times \mathbf{E}_j \end{aligned} \tag{4.2}$$

The electric field is tangenially continuous across the surface port and magnetic field has a jump. In weak form, the relationship is implemented as residuals in Equation 4.3,

$$\begin{aligned} \mathbf{R}^{\mathbf{H}}_{\Gamma_{CKT}} &= [\![\mathbf{H}]\!]_\gamma - J_{CKT} \in \mathbf{H}(div_\tau, \Gamma_{CKT}) \\ \mathbf{R}^{\mathbf{E}}_{\Gamma_{CKT}} &= [\![\mathbf{E}]\!]_\gamma \in \mathbf{H}(div_\tau, \Gamma_{CKT}) \end{aligned} \tag{4.3}$$

Due to the dual pairing principle, the term $\mathbf{R}^{\mathbf{H}}_{\Gamma_{CKT}}$ and $\mathbf{R}^{\mathbf{E}}_{\Gamma_{CKT}}$ are tested with curl-conforming functions $\pi_{tau}(\mathbf{w})$ and $\pi_{tau}(\mathbf{v})$ to form energy density. Then the weak form of the surface port related elements is written as **??**,

$$\begin{aligned} &\int_\Omega \mathbf{w} \cdot \left(\nabla \times \mathbf{E} + \mu \frac{\partial \mathbf{H}}{\partial t} - \mathbf{M} \right) d\Omega \\ &- \int_\Omega \mathbf{v} \cdot \left(\nabla \times \mathbf{H} - \epsilon \frac{\partial \mathbf{E}}{\partial t} - \sigma \mathbf{E} - \mathbf{J} \right) d\Omega \\ &+ \int_{\partial \mathcal{F}_h} \{\!\{\mathbf{v}\}\!\} \cdot ([\![\mathbf{H}]\!]_\gamma - J_{CKT}) ds - \int_{\partial \mathcal{F}_h} \{\!\{\mathbf{w}\}\!\} \cdot [\![\mathbf{E}]\!]_\gamma ds = 0 \end{aligned}$$

4.1.3 Circuit modeling with State Variables Form

Modified Nodal Analysis(MNA),based on Kirchhoff's circuit laws, is a typical formulation for SPICE simulations. MNA augments standard nodal analysis equations with eqution for each voltages source. To model nonlinear circuits, KCL is enforced at all nonreference nodes and KVL is applied to independent loops. As listed in Equation 4.4

$$\begin{bmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_{sp} \\ \dot{\mathbf{i}}_v \end{bmatrix} + \begin{bmatrix} \mathbf{Y} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{sp} \\ \mathbf{i}_v \end{bmatrix} + \begin{bmatrix} \mathbf{i}^{nl}(\mathbf{x}_{sp}) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} -\mathbf{i}_s \\ \mathbf{v}_s \end{bmatrix} \tag{4.4}$$

, the unknowns are voltages at each nonreference node($\mathbf{x}_{sp}$) and currents ($\mathbf{i}_v$) flow through independent voltage sources and inductors. Here, the admittance matrix $\mathbf{X}, \mathbf{Y}$ are determined by interconnections between circuit elements. $\mathbf{B}$ is determined by the connection of supplied voltage sources with only 0, ± 1 entries. $\mathbf{i}^{nl}(\mathbf{x}_{sp})$ represents currents through branches containing nonlinear elements. $\mathbf{i}_s, \mathbf{v}_s$ denote independent current/voltage sources. From Equation 4.4 we can easily see that the rank-deficiency may leads to a nonlinear differential algebraic equation(DAE). When coupled with EM equations, special tricks are needed to solve DAE system. In order to maintain the global system as a ODE, state variable (SV) formulation is considered to describe the behavior of circuit. In SV form, the circuit equation is written in the general form:

$$\frac{d\mathbf{x}_{SV}}{dt} = C\mathbf{x}_{SV} + D\mathbf{u}, \tag{4.5}$$

where $\mathbf{x}_{SV} = [V_c, I_l]^T$ is the state variables and $\mathbf{u}$ being the input, e.g. the independent voltage source. The corresponding unknowns are then voltages across each capacitor and currents flowing through each inductor. The discretized system of equation for **??** is then formulated as Equation 4.6,

$$\begin{aligned}
M_\epsilon \frac{\partial e_i^{port}}{\partial t} &= S_e h_i^{port} - F_e^{ii} h_i^{port} + B_e^J g(\mathbb{P}e_i^{port}, V_1, ..., V_k, I_1, ..., I_m) - F_e^{ij} h_j^{port} \\
M_\mu \frac{\partial h_i^{port}}{\partial t} &= -S_h e_i^{port} + F_h^{ii} e_i^{port} + F_h^{ij} e_j^{port} \\
\frac{\partial V_i}{\partial t} &= g_i(\mathbb{P}\mathbf{e}^{port}, V_1, ..., V_k, I_1, ..., I_m) \\
\frac{\partial I_i}{\partial t} &= g_i(\mathbb{P}\mathbf{e}^{port}, V_1, ..., V_k, I_1, ..., I_m)
\end{aligned} \tag{4.6}$$

, where $B_e^J = \frac{1}{w} \cdot \int_{\partial \mathcal{F}_h} \pi_\tau(\mathbf{v}) \cdot \bar{\mathbf{l}}$ is the matrix projecting I^{EM} to current density and $\mathbb{P}$ is to project $\mathbf{e}^{port}$ to voltage across the surface port. $\bar{\mathbf{l}}$ denotes the unit vector along the length direction of the port. This coupling scheme being self-consistant lies in that all the unknowns from EM and circuit are described in one set of equations. So the equality sign is enforced.

To solve Equation 4.5 with available ODE integration schemes, evaluating the right-hand-side is a necessary step which is illustrated in the following section.

4.1.4 Right-hand-side evaluation

The physical meaning for right-hand-side of Equation 4.5 is the scaled current through capacitor(I_c) and voltage across inductor(V_l) when V_c, I_l is acknowledged. So the external behavior of capacitor and inductor is totally determined by state variables. At time instance t_n, a equivalent circuit can be constructed by replacing all ith capacitor with a voltage source $VC_i = V_c$ and jth inductor with a current source $IL_j = I_l$. A simple illustration can be found in Figure 4.4.

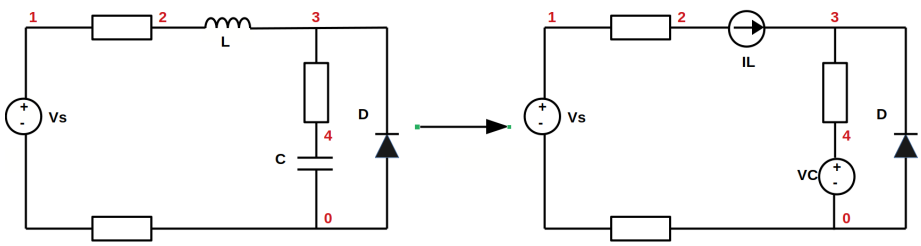


Figure 4.4: Equivalent circuit in right-hand-side evaluation with SPICE op analysis

For general cases, one operating point analysis is preformed to obtain the right-hand-side I_c and V_l. But for some nonlinear cases, the nonlinear components can also display dynamic properties. For instance, the SPICE representation of diode includes transit time(diffusion capacitance) and junction capacitance as shown in Figure 4.5 [92]

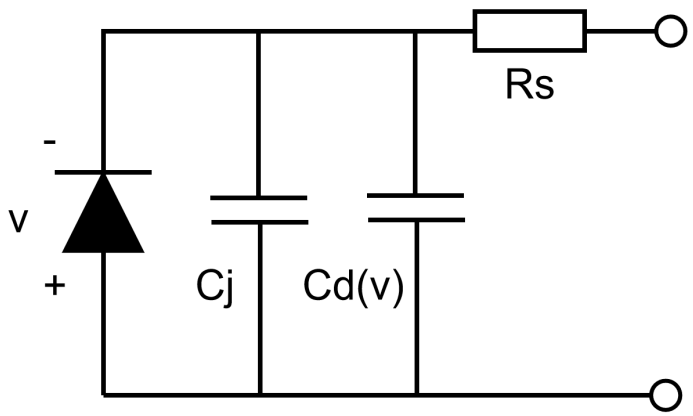
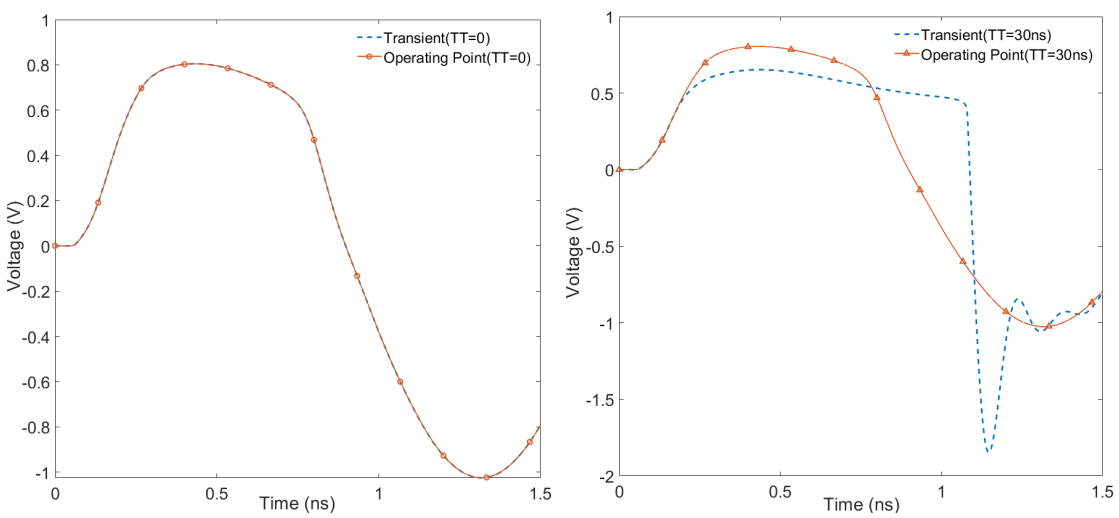


Figure 4.5: SPICE diode model representation

,where diffusion capacitance Cd is a nonlinear. The impact of such component could be significant especially in higher frequencies. To see this, Figure 4.6 shows the difference from the transit time(TT) parameter in a co-simulation of a microstrip line and circuit in Figure 4.10 with L=1.2nH, C=0.35pF.



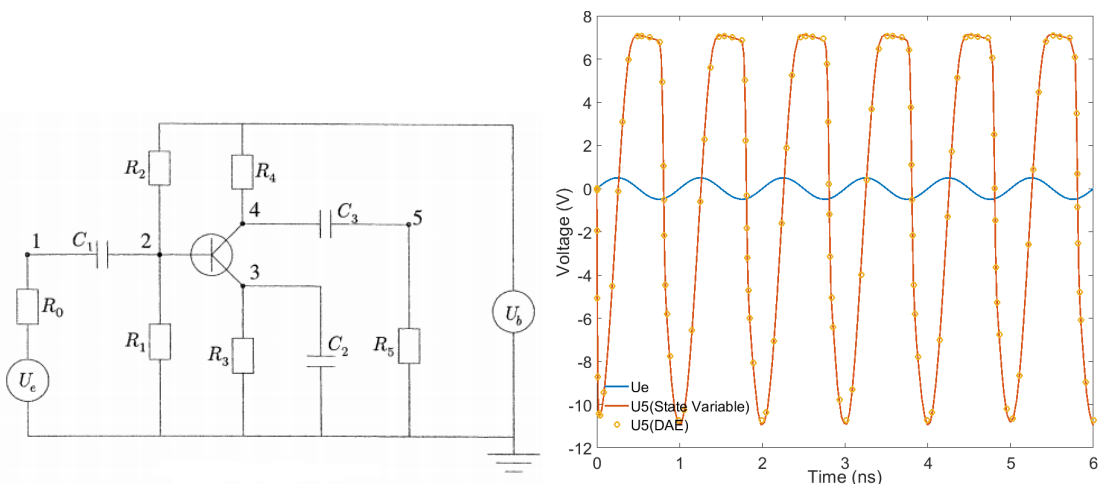
(a) With diffusion capacitor (b) Without diffusion capacitor

Figure 4.6: Impact of nonlinear passive component

So for simplicity, the paper deals with TT=0ns(no diffusion capacitance) but keeps all other nonlinear components.

4.1.5 State Variable Form in Circuit/Co-simulation

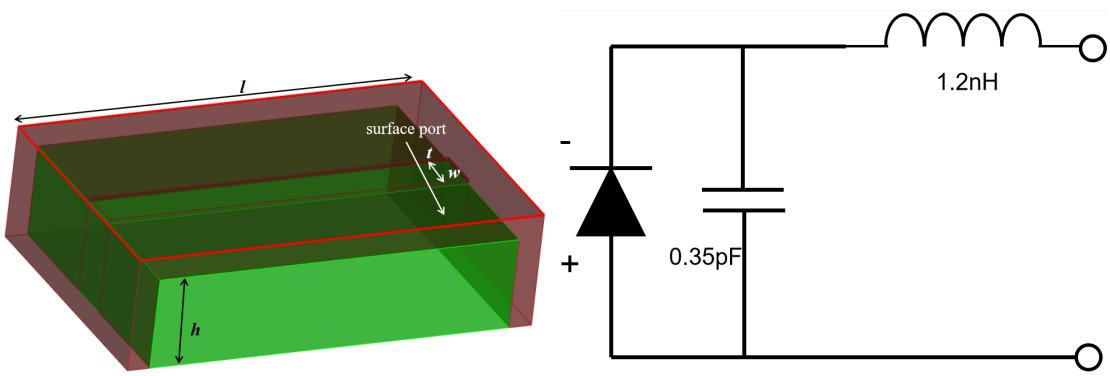
As a demonstration of how state variable approach applied in circuit simulation, first simply consider a transistor amplifier illustrated in Figure 4.7. The amplifier is modeled by Ebers-Moll model [93].



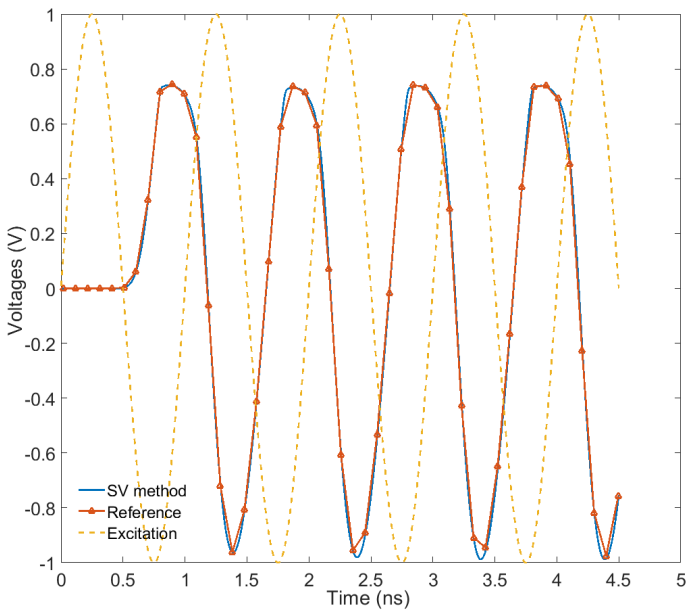
(a) Transistor Amplifier (b) Voltage at Node 5

Figure 4.7: Transistor Amplifier circuit and voltages

The SV approach with 3 unknowns is able to reproduce the result by MATLAB's ode23t integration scheme. The second example is a co-simulation of 3D microstrip line connected to circuit shown in as in Figure 4.8.



(a) Geometry of microstrip line (b) Circuit connected to surface port

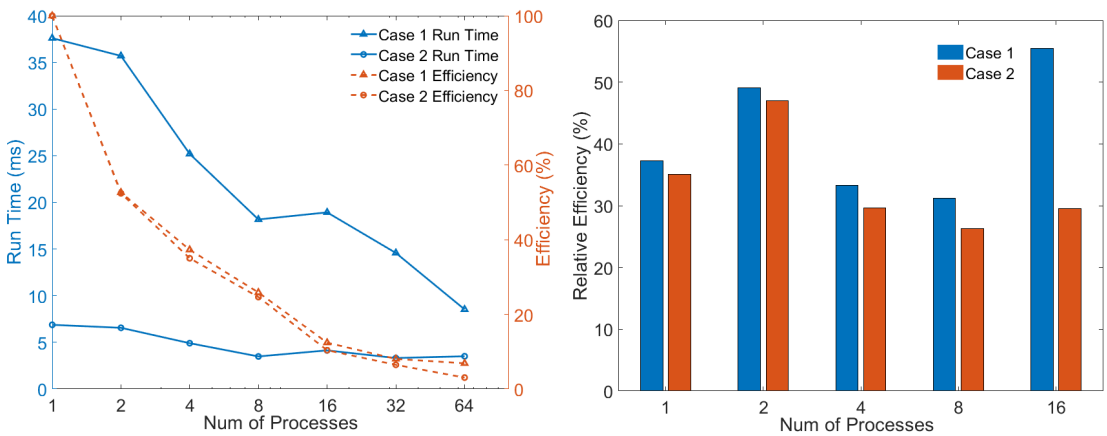


(c) Voltages by SV method

Figure 4.8: 3D microstrip line connected to model circuit

With the right-hand-side being evaluated for the circuit side, the system of ODE in Equation 4.6 can be solved by appropriate integration schemes. Ideally, V_{n+1}^{EM} and V_{n+1}^{CKT} should be consistent. But due to the highly non-linear function g, the difference could be large enough to result nonphysical solutions. We can see that so long as Δt is necessarily small, g can be regarded as linear. But this approach would consume many computational resources. This is another cause of multi-rate nature of hybrid problem. Traditional space-parallelism is prune to have decreasing parallel efficiency as the number of parallel processes increases. To show this, two different density of meshing are used to model this example resulting DOF of 208707(case 1)

and 82750(case 2) respectively for the EM part. The running time for performing one integration step and parallel efficiency are recorded in Figure 4.9(a). The efficiency decreases to very low level with the increase of processes.



(a) Run time and Efficiency (b) Relative efficiency

Figure 4.9: Parallel efficiency for space-only parallel method

The relative efficiency(<50%) when adding 4 times more processes is also illustrated in Figure 4.9(b). The DOF per process at 32 processes scenario is 6522 and 2586 respectively. Going more parallel can no longer maintain parallel efficiency especially when the number of processes is already high, or DOF per process is relatively small. So there is urge need to develop parallelism in the dimension of time.

4.2 Parallel-in-time integration for nonlinear EM-Circuit cosimulation

4.2.1 Parallel-in-time algorithm

The system of Equation 4.6 can be viewed as a nonlinear ODE as,

$$\frac{d\mathbf{u}}{dt} = -\mathbb{A}\mathbf{u} + \mathbb{B}(\mathbf{u})\mathbf{u} + \tilde{\mathbf{f}}(t), \quad \mathbf{u}(t_0) = \mathbf{u}_0. \tag{4.7}$$

Now split the variable as $\mathbf{u} = \mathbf{v} + \mathbf{w}$, here $\mathbf{v}$ and $\mathbf{w}$ solve the following system of equation,

$$\begin{aligned} \frac{d\mathbf{v}}{dt} &= -\mathbb{A}\mathbf{v} + \mathbb{B}(\mathbf{v}+\mathbf{w})\mathbf{v} + \tilde{\mathbf{f}}(t), \quad \mathbf{v}(0) = 0, t \in [0, T] \\ \frac{d\mathbf{w}}{dt} &= -\mathbb{A}\mathbf{w}, \quad \mathbf{w}(0) = \mathbf{u}_0, t \in [0, T] \end{aligned} \tag{4.8}$$

Clearly, $\mathbf{w}$ solves a linear ODE but with non-zero initial condition, while $\mathbf{v}$ solves for nonlinear, homogeneous ODE. Partition time interval $[0, T]$ into N time windows $[T_{n-1}, T_n]$, a parallel in time method can be directly applied as performing k = 1,2,...

$$\begin{aligned} \text{Stage\#1}: \frac{d\mathbf{w}_n^k}{dt} &= -\mathbb{A}\mathbf{w}_n^k(t), \ , \mathbf{w}_n^k(T_{n-1}) = \mathbf{v}_{n-1}^{k-1}(T_{n-1}), \ t \in [T_{n-1}, T] \\ \text{Stage\#2}: \frac{d\mathbf{v}_n^k}{dt} &= -\mathbb{A}\mathbf{v}_n^k + \mathbb{B}(\mathbf{v}_n^k + \sum_{j=1}^{n} \mathbf{w}_j^k)\mathbf{v} + \tilde{\mathbf{f}}(t), \quad \mathbf{v}_n^k(T_{n-1}) = 0, t \in [T_{n-1}, T_n] \end{aligned} \tag{4.9}$$

The algorithm can be described by Figure 4.10

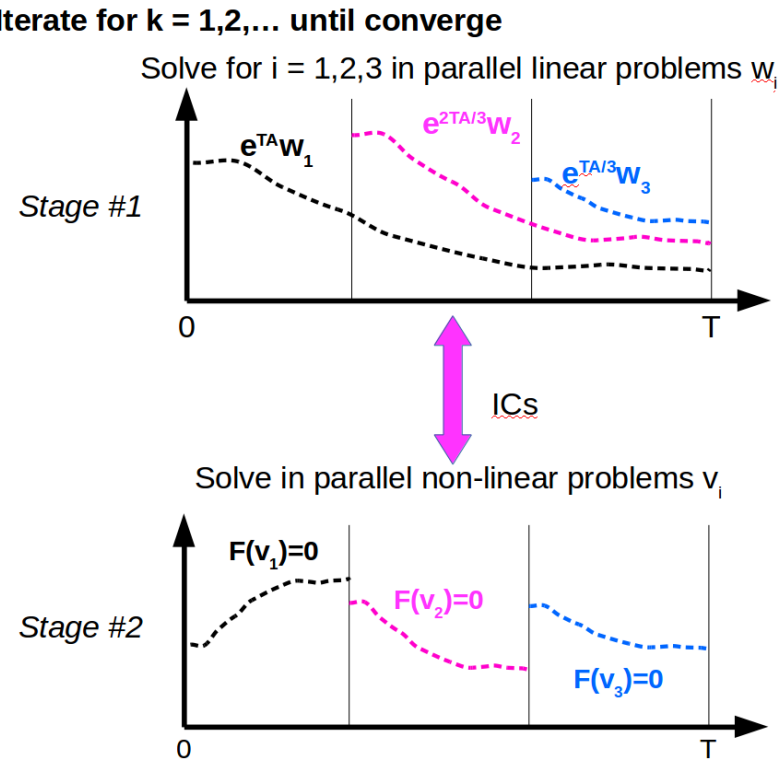


Figure 4.10: Information flow for coupling

Realize that $\mathbf{u}^k(t) = \mathbf{v}_n^k + \sum_{j=1}^{n} + \mathbf{w}_j^k$ at $t \in [T_{n-1}, T_n]$, substitute $\mathbf{v}^k$ with $\mathbf{u}^k$, we have

$$\text{Stage\#1}: \frac{d\mathbf{w}_n^k}{dt} = -\mathbb{A}\mathbf{w}_n^k(t),\ \mathbf{w}_n^k(T_{n-1}) = \mathbf{u}_n^k(T_{n-1}) - \sum_{j-1}^{n-1} \mathbf{w}_j^{k-1}(T_{n-1}),\ t \in [T_{n-1}, T] \quad (1)$$

$$\text{Stage\#2}: \frac{d\mathbf{u}_n^k}{dt} = -\mathbb{A}\mathbf{u}_n^k + \mathbb{B}(\mathbf{u}_n^k)\mathbf{u}_n^k + \tilde{\mathbf{f}}(t),\ \ \mathbf{u}_n^k(T_{n-1}) = \sum_{j=1}^{n} \mathbf{w}_j^k(T_{n-1}), t \in [T_{n-1}, T_n] \quad (2)$$

(4.10)

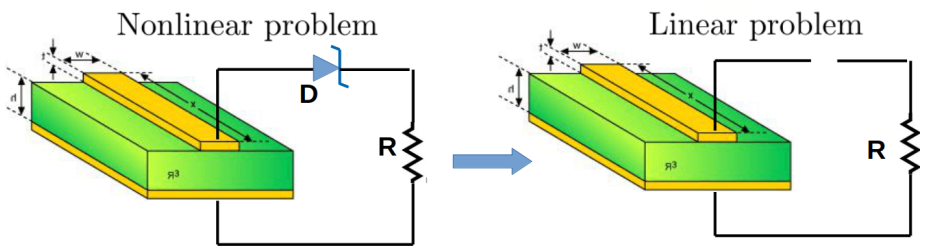
The nonlinear part provides initial conditions for linear part. The benefit for this decomposition lies in that $\mathbf{u}$ can be solved totally in parallel on time window $[T_{n-1}, T_n]$. The success of this algorithm is determined by the number of iterations

required to converge and how effective linear part $\mathbf{w}_n^k$ can be solved on overlapped windows. It should be mentioned that this framework only exploits parallelism in time. For solving both (1)(2) in Equation 4.10 decomposition in space is still applicable. For this chapter, we only discuss the performance of parallel in time.

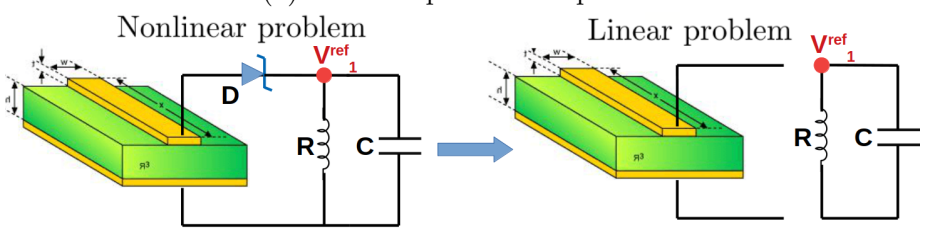
Here provides two simple examples, see Figure 4.11, to illustrate how to extract linear part $\mathbf{w}$ from $\mathbf{u}$. The first case is a microstrip line terminated with a diode-resistor circuit. Since the reference voltages in the circuit can be directly yielded from EM, the complete nonlinear system of equation writes as,

$$\begin{aligned} M_\epsilon \frac{\partial e_i^{port}}{\partial t} &= S_e h_i^{port} - F_e^{ii} h_i^{port} + B_e^J g(\mathbb{P} e_i^{port}) - F_e^{ij} h_j^{port} \\ M_\mu \frac{\partial h_i^{port}}{\partial t} &= -S_h e_i^{port} + F_h^{ii} e_i^{port} + F_h^{ij} e_j^{port} \end{aligned} \tag{4.11}$$

Once the nonlinear term $B_e^J g(\mathbb{P} e_i^{port})$ is removed, the remaining part behaviors just like open-circuit at the coupling port.



(a) Without passive component



(b) With passive component

Figure 4.11: How to extract linear problem

For the second case where one reference voltage in the circuit is inscribed in the

system, the system is written as,

$$\begin{aligned} M_\epsilon \frac{\partial e_i^{port}}{\partial t} &= S_e h_i^{port} - F_e^{ii} h_i^{port} + B_e^J g(\mathbb{P} e_i^{port} - V_1) - F_e^{ij} h_j^{port} \\ M_\mu \frac{\partial h_i^{port}}{\partial t} &= -S_h e_i^{port} + F_h^{ii} e_i^{port} + F_h^{ij} e_j^{port} \\ C \frac{\partial V_1}{\partial t} &= g(\mathbb{P}\mathbf{e}^{port} - V_1) - \frac{V_1}{R} \end{aligned} \tag{4.12}$$

Correspondingly, the linear system includes two separate parts, one is the microstrip line with a open coupling port, and the other is a RC loop.

The typical communication pattern for space-time grid is illustrated by Figure 4.12. One significant features is that the space and time dimension are independent. This gives us a great flexibility in attribution of computing resources. If the DOF per space process is large enough, we would still prefer more computing resources in the space dimension. When the decrease of efficiency is observed, time dimension would come in to rescue.

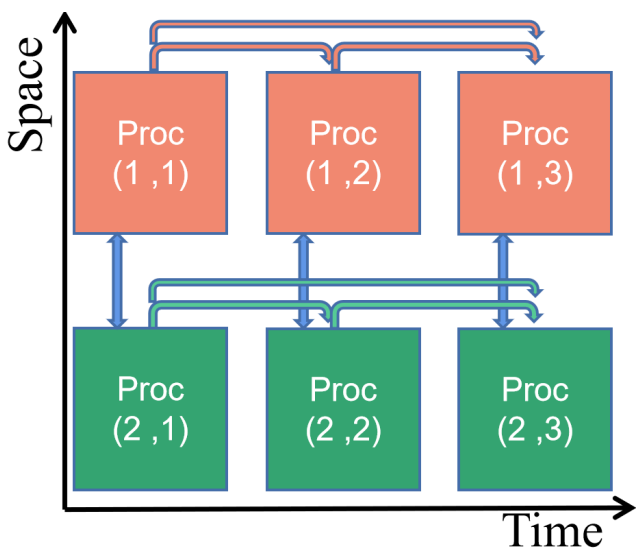


Figure 4.12: Communication pattern for space-time grid

4.2.2 Effective Integration Technique for Linear Problem

Stage 1 in Equation 4.10 indicates that the solution to N linear non-homogeneous problems defined on overlapped time intervals are required. This imposes great im-

balance in terms of parallelizaion. Also note that only the solutions at time interfaces are needed. Therefore the choice of integrator determines the over all parallel performance. Numerical study shows that only coarse integrator(high truncation error) is a good candidate here. To achieve this, recycling ϕ integrator is considered here. The work-flow is demonstrated by Figure 4.13 with N=4 assumed and algorithm description in Algorithm 5.

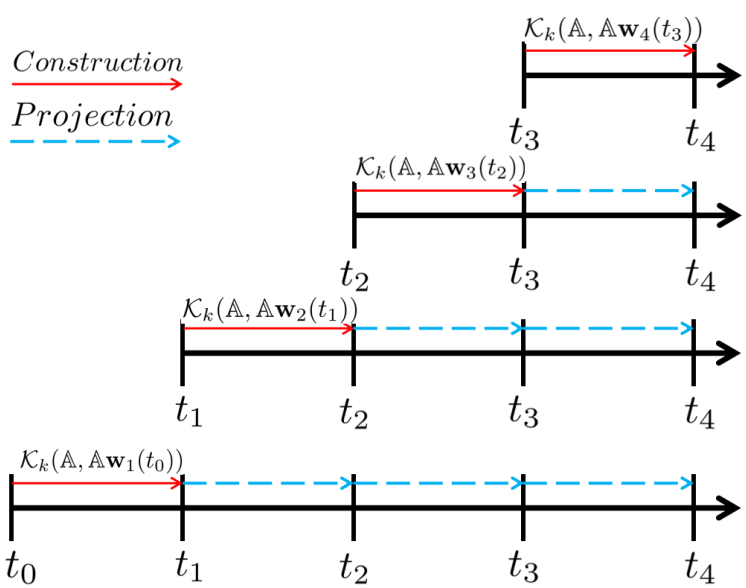


Figure 4.13: Work-flow of effective integration for linear part

Algorithm 5 Coarse φ integrator at n-th Time Window

Input: $\mathbf{w}_n(t_{n-1}), \mathbb{A}, k, N$

Output: $\mathbf{w}(t_m), m = n, ..., N-1$

$\Delta t \leftarrow t_n - t_{n-1}$

$\mathbf{v} \leftarrow \mathbf{w}_n(t_{n-1})$

$\mathbf{w}_1 \leftarrow \mathbb{A} * \mathbf{v}$

$\mathbb{A}\mathbb{W}_k \leftarrow \mathbb{W}_k \mathbb{H}_{k,k} + h_{k+1,k}\mathbf{w}_{k+1}\mathbf{e}_k^{\mathrm{T}}$

$\mathbf{y}_k \leftarrow \Delta t \varphi(-\Delta t, \mathbb{H}_{k,k})\mathbf{e}_1$

$\mathbf{v} \leftarrow \mathbf{v} + \mathbb{W}_k * \mathbf{y}_k$

$\mathbf{w}_n(t_n) = \mathbf{v}$

for $i = n+1, N$ **do**

$\mathbf{x} \leftarrow \mathbb{A} * \mathbf{v}$

$\mathbf{g} \leftarrow \mathbb{W}_k^{\mathrm{T}}\mathbf{x}$

$\mathbf{y}_k \leftarrow \Delta t \varphi(-\Delta t, \mathbb{H}_k)\mathbf{g}$

$\mathbf{v} \leftarrow \mathbf{v} + \mathbb{W}_k \mathbf{y}_k$

$\mathbf{w}_n(t_i) \leftarrow \mathbf{v}$

The benefit of Algorithm 5 is that each process constructs the Krylov subspace once and for the following several steps, only projection is required. Construction of Krylov subspace is known to be more expansive than projection.

4.3 Numerical examples

4.3.1 Convergence and Parallel Efficiency study

Since PIT algorithms performs several iterations to converge to true solution, the convergence rate is studied. The example used here is a 1D transmission line connected to a half-wave rectifier shown in Figure 4.14. The transmission line is modeled by discontinuous Galerkin with nodal basis function. To also evaluate the performance of PIT applied to linear problem, the diode can be replaced by a resistor.

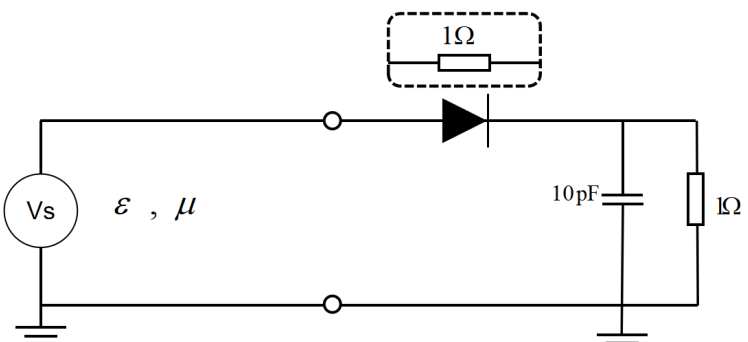
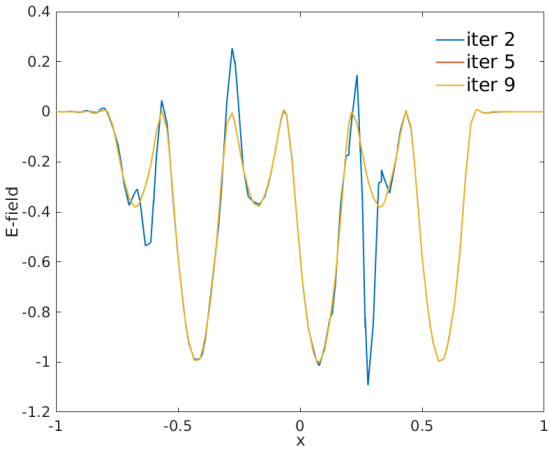
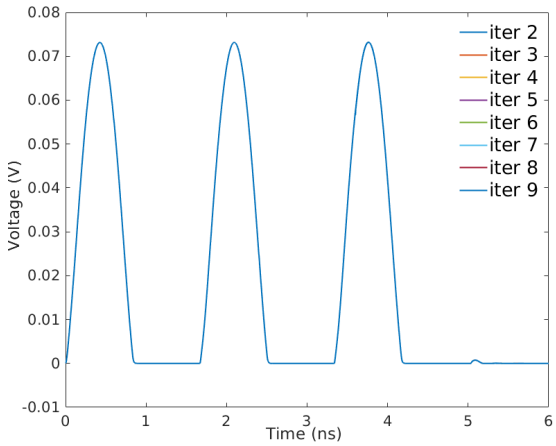


Figure 4.14: 1D Transmission Line example

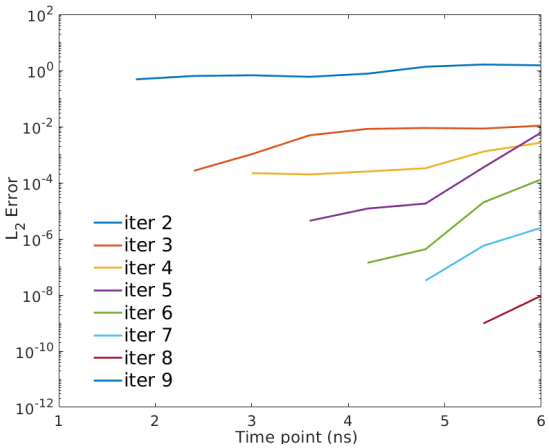
In both cases, T_{end}=6ns is divided into 10 time windows. The electrical field, voltage across capacitor are monitored as well as the convergence VS. iterations in Figure 4.15 and Figure 4.16.



(a) Electric field at transmission line

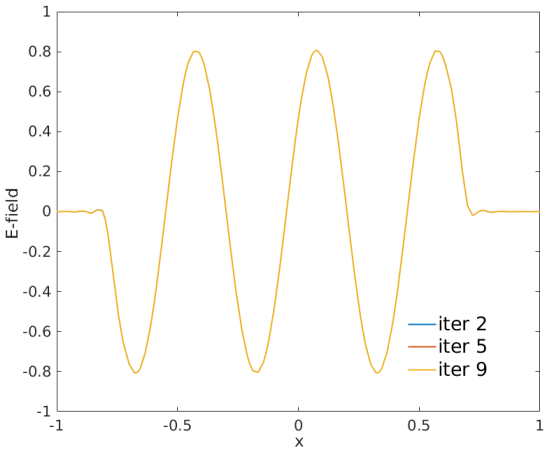


(b) Voltage across capacitor

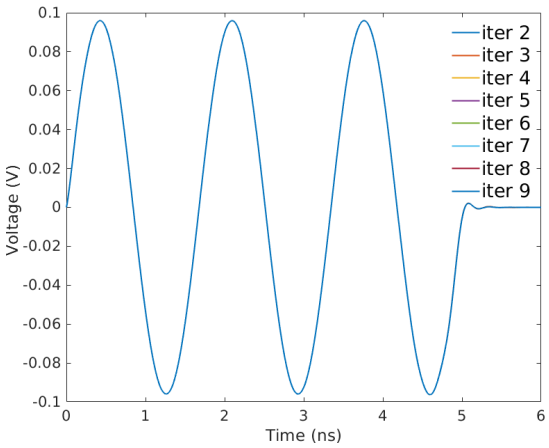


(c) Convergence of error

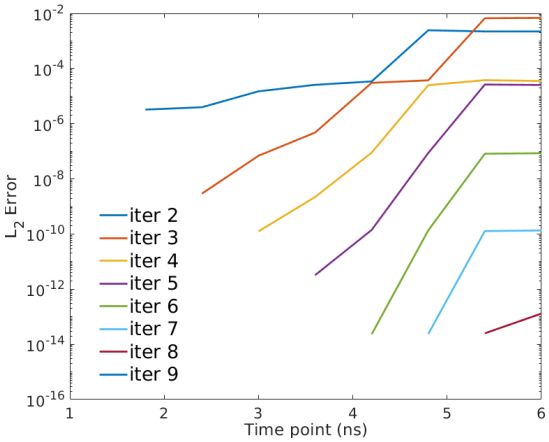
Figure 4.15: 1D transmission line connected with nonlinear load



(a) Electric field at transmission line



(b) Voltage across capacitor



(c) Convergence of error

Figure 4.16: 1D transmission line connected with linear load

Error decreases with more iterations carried out for both cases, while the linear one converges faster. After the N-th iteration applied, PIT result converges exactly to the serial time stepping result. Here it needs to assure that time step for stage 2 is identical to serial time stepping method's. Not a full set of iteration is necessary once the preset error bound is met. This property leaves a good room for performance. Next the parallel efficiency is studied by refining the mesh creating different DOFs for EM model. The parallel efficiency is shown below.

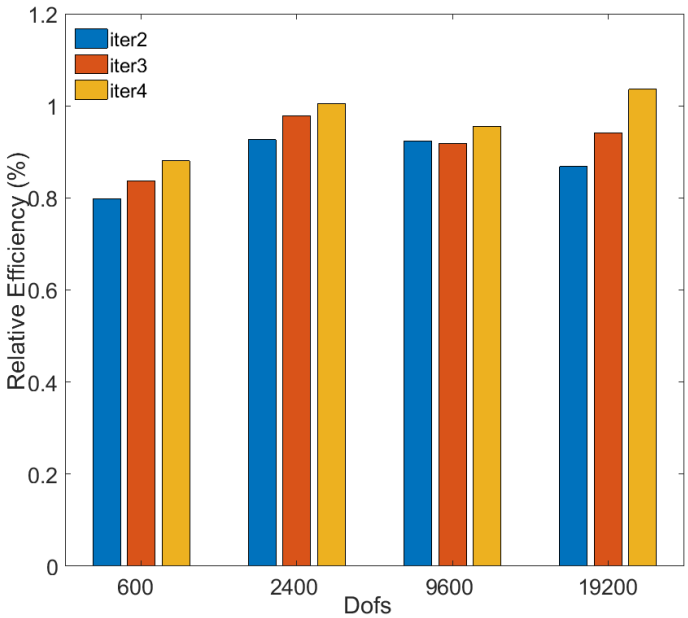


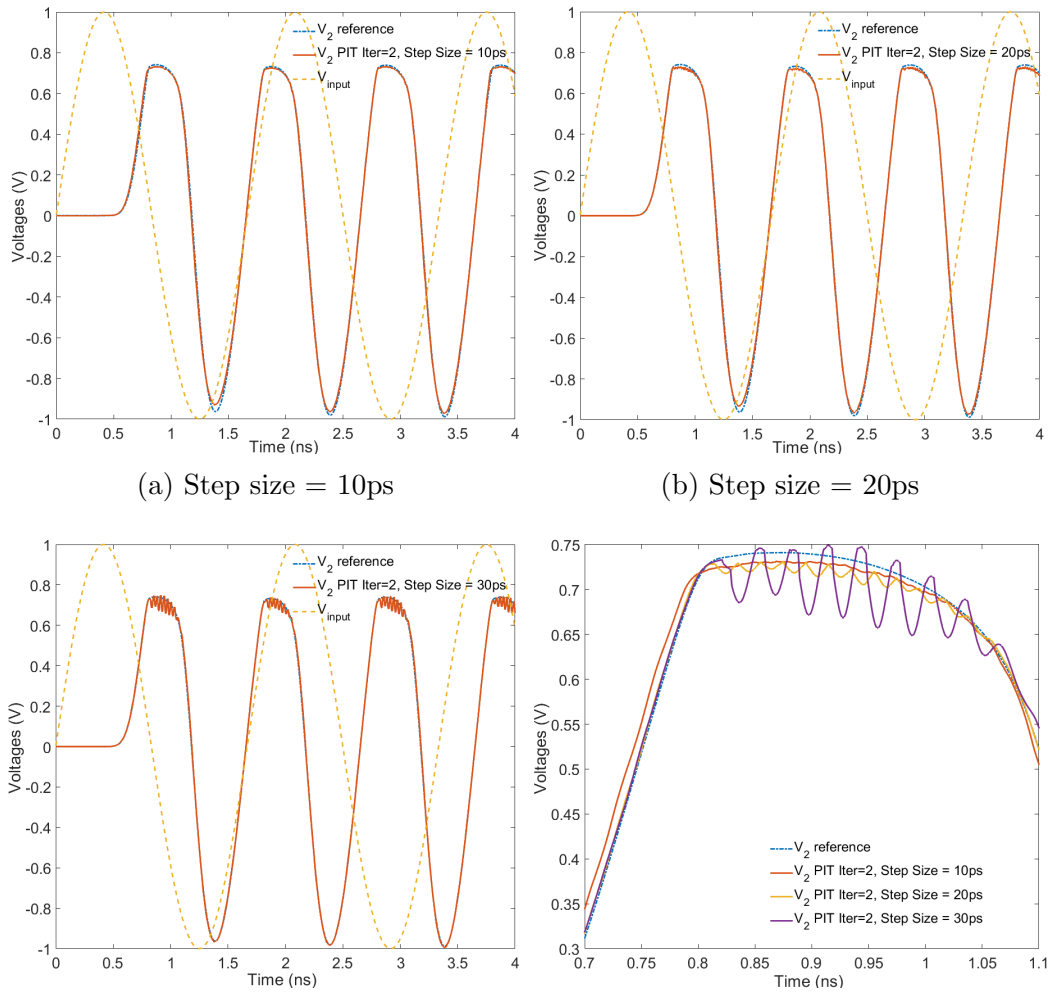
Figure 4.17: Parallel Efficiency of 1D transmission line-half wave rectifier example

The comparison is conducted between running PIT algorithm with 1 and 4 processes. Concluded from Figure 4.17 that increase of DOFs per process barely has impact on the efficiency and it can be maintained at very high level($> 80\%$). Parallelism in time dimension is capable of achieving good efficiency compared to parallelism in space(PIS) only. But it also worth to point out that, PIT algorithm brings in additional computational efforts and sacrifice of precision to some degree. A good combination of parallelism in space and time should be adopted. General guideline is that PIT should come on top of PIS especially when the DOFs per process is already small.

4.3.2 Time Stepping Strategy and Balance of Parallelization

The parallel in time algorithm involves running a coarse integrator on the system of similar dimension with the original problem. As our choice, Algorithm 5 is con-

sidered. This class of method shows super convergence when the integration time period is relatively small. Therefore, the strategy for running PIT algorithm is still time stepping based which means that PIT is applied within a coarse time step. To demonstrate, the same example in Figure 4.8 is also used. Different step sizes are employed and shown in Figure 4.18.



(a) Step size = 10ps

(b) Step size = 20ps

(c) Step size = 30ps

(d) Differences at nonlinear period

Figure 4.18: Impact of step size

In this study, T_{end}=4ns is divided into time step size of 10,20 and 30ps. Within each time step, a PIT is run with 2 iterations. With smaller step size, the result approximates reference better especially at the time period when diode is forward

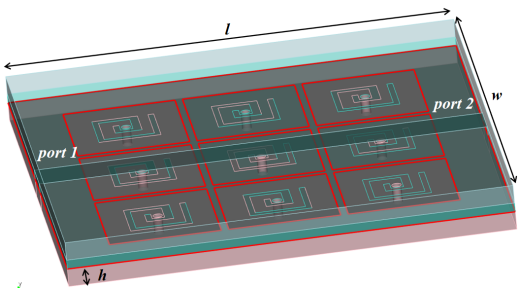
biased. The run time for each case on 4 processes are 70.72s, 63.75s and 49.64s. The more subtle time step is, the more communication over-heads there are. So there will be a trade-off between performance and accuracy.Algorithm 5 is able to maintain a good balance among processes considering each sub-problem is defined on overlapped time period. The runtime for linear subproblem on each process is listed in Table 4.1. The construction of Krylov subspace takes the majority of time and a good balance is observed among 4 processes.

Table 4.1: Runtime Statistics of linear subproblem for each process

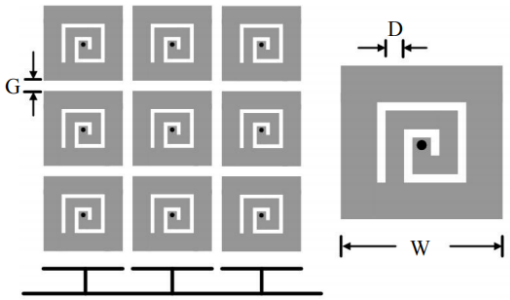
Proc	0	1	2	3
Construction(s)	36.9	36.6	37.0	36.8
Projection(s)	4.2	2.1	0.002	0.002

4.3.3 Applications

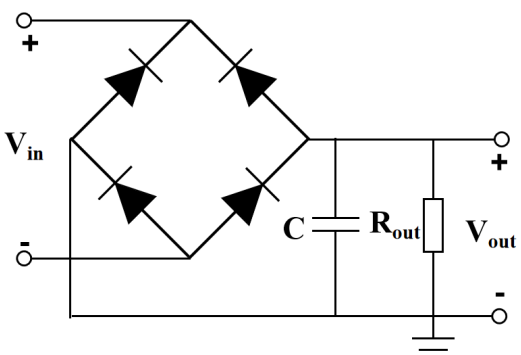
The first application is a full-wave rectifier circuit connected to a spiral electromagnetic bandgap structure(EBG) [94]. Each element of this EBG lattice consists of a square metal patch with a a spiral branch inserted inside, as shown in Figure 4.19(a). The dimensions are G = 4mm, D = 6mm, W = 60mm, l = 240mm, h = 10mm and w = 24mm. The simulated S_{12} is shown in Figure 4.19(d) where it is below -10dB beyond 700MHz. Next a 500MHz and 1GHz continuous wave are launched into port 1 and rectifier is connected to port 2. From (e)(f) we can see that a 1.4V voltage difference is observed in 500MHz case which agrees with physical principle of silicon diodes whose forward voltage is about 0.7V.



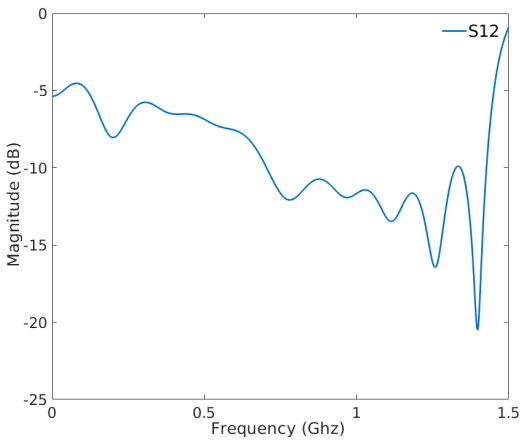
(a) Geometry of spiral EBG



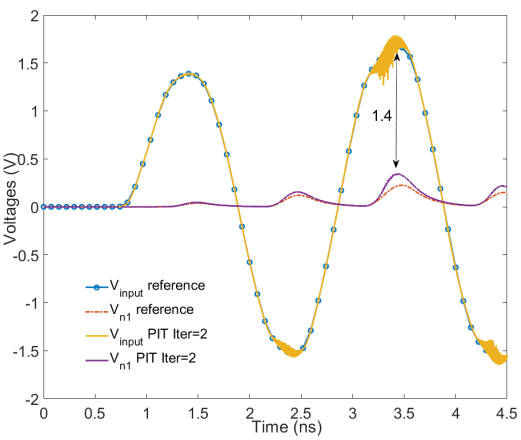
(b) Full-wave bridge rectifier(R_{out}=200Ω,C=1pF)



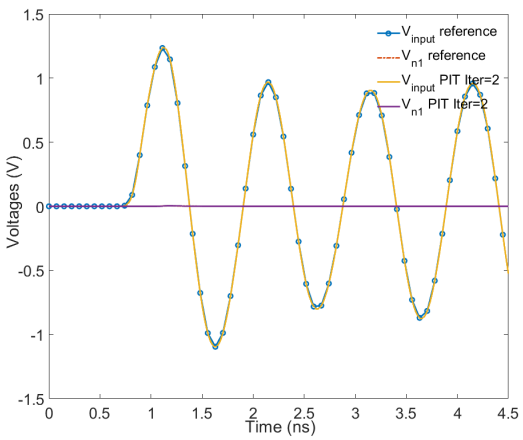
(c) Voltages at port 1,2



(d) Magnitude of S_{12}



(e) Port 2 voltage at 500MHz



(f) Port 2 voltage at 1GHz

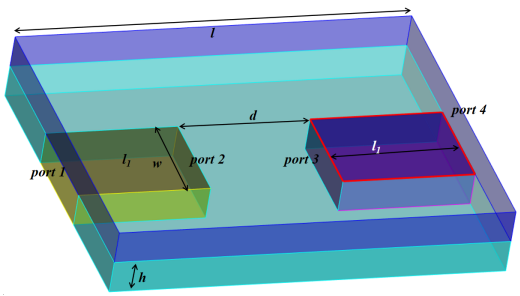
Figure 4.19: Spiral EBG connected to full-wave rectifier

The run time statistics are listed in Table 4.2. A two times improvement of parallel efficiency is observed for different number of iterations with 4 time windows divided.

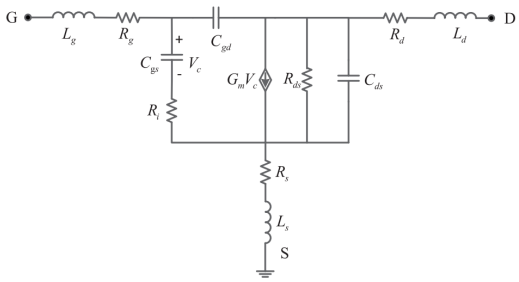
Table 4.2: Runtime Statistics of spiral EBG example

# of Iterations	2	3	4
serial	777.2s	1289.3s	1625.6s
PIT(Time/Efficiency)	271.4s/72%	474.8s/68%	645.7/63%
PIS(Time/Efficiency)	537.3s/36%	859.3s/38%	1122.7s/36%

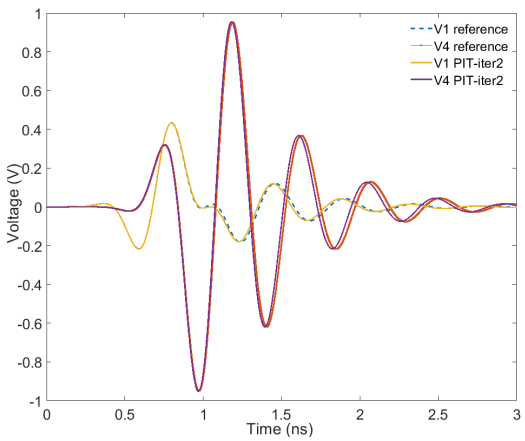
The second application is a microstrip line matching network. A JS8851-AS FET amplifier is mounted over a gap between port 2(gate) and 3(drain). Four ports have the same size. The dimension of microstrip line is w = 7.9mm $l_1 = l_2 = 10$mm, d = 5mm, l = 25mm and h = 2.54mm and the dielectric constant of substrate is $2.17\epsilon_0$. Here amplifier is modeled by small-signal equivalent circuit shown in **??**(b). The total DOF is 237573 for EM and 6 for circuit. To study the broadband characteristics, a modulated Gaussian source is applied at Port 1. A 25dB gain is observed at 2.2GHz. Note that this is a linear circuit, so 2 iterations of PIT is good enough to approximate the result. For runtime statistics in Table **??**, two times improvement of parallel efficiency is guaranteed for various iterations when 4 time windows divided.



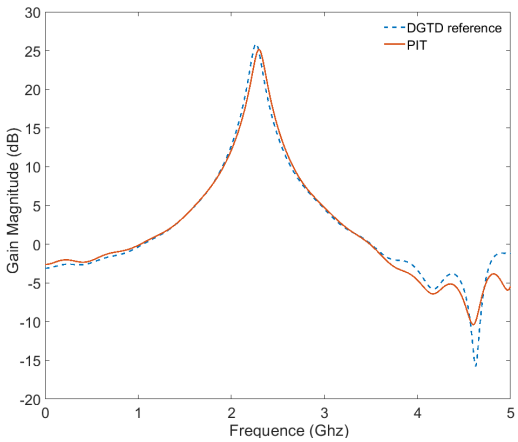
(a) Geometry of matching network



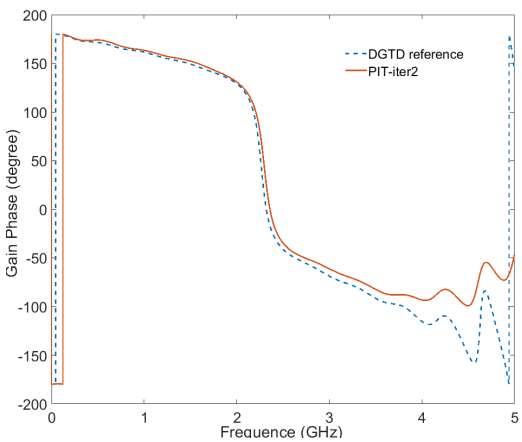
(b) Small-signal equivalent circuit model for the JS8851-AS FET amplifier



(c) Voltages at port 1,4



(d) Magnitude of gain



(e) Phase of gain

Figure 4.20: Matching network with JS8851-AS FET amplifier

# of Iterations	2	3	4
serial	957.3s	1464.7s	2004.5s
PIT(Time/Efficiency)	376.5s /64%	565.8s/65%	761.4s/66%
PIS(Time/Efficiency)	641.6s/37%	983s/37%	1336.3s/38%

Table 4.3: Runtime Statistics of microstrip line matching network

Chapter 5

Parallel and fast methods for solving time harmonic Maxwell equations

5.1 Introduction

In modeling the complex antenna/array radiations, high computational cost and inflexibility cause great trouble for conventional CEM solvers. The multi-solver domain decomposition(MSDDM) scheme is one of the efficient tools tackle those challenges. In this scheme, an object is decomposed into separate parts according to material property and geometry. The major advantages of MSDDM can be summarized into 3 folds: 1) It's very easy to modify the existing CEM solvers and integrate into MSDDM's framework; 2) Individual subdomains can be solved concurrently so that it can fit into the parallel computing environment,e.g. message passing interface(MPI); 3) Krylov iteration scheme is employed to enhance global convergence. In this paper, a optimized transmission condition based on Schwartz domain decomposition is proposed in the framework of MSDDM. From the implementation point of view, the subdomain solvers are treated as black boxes. Numerical example showed that the optimized transmission condition is facilitated as long as necessary information is provided by subdomain solvers. Besides, a hierachical parallel preconditioning techniques which combines of two classical stationary preconditioners is developed to best exploit the parallel computing resources available.

5.2 Technical Approach

5.2.1 Boundary Value Problem

Here we consider the domain decomposition(DD) method to solve time harmonic Maxwell equations. Figure 5.1 illustrates the decomposition for two domains.

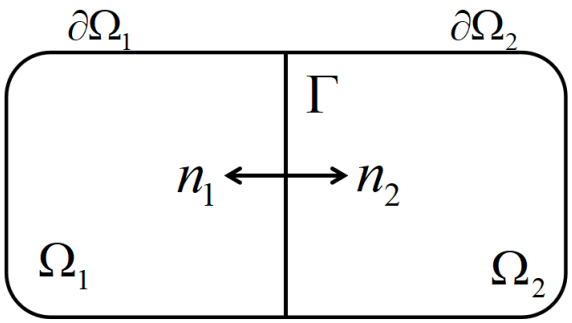


Figure 5.1: Decomposition of domains

Specifically, the DD method is Interior Penalty discontinuous Galerkin method mentioned in [95]. First define the tangential operators $\pi_\tau(\cdot) := \mathbf{n} \times \cdot \times \mathbf{n}$ and $\gamma_\tau(\cdot) := \mathbf{n} \times \cdot$, assuming the outward point normal vector $\mathbf{n}$ is along with $+z$ direction. The boundary value problem is written as

$$\begin{aligned}
\nabla \times \mu_{r1}^{-1} \nabla \times \mathbf{E}_1 - k_0^2 \epsilon_{r1} \mathbf{E}_1 &= -jk_0 \eta_0 \mathbf{J}_1^{\text{imp}} && \text{in } \Omega_1 \\
\nabla \times \mu_{r2}^{-1} \nabla \times \mathbf{E}_2 - k_0^2 \epsilon_{r2} \mathbf{E}_2 &= -jk_0 \eta_0 \mathbf{J}_2^{\text{imp}} && \text{in } \Omega_2 \\
\pi_\tau(\mathbf{E}_1) &= \pi_\tau(\mathbf{E}_2) && \text{on } \Gamma \\
\gamma_\tau(\mu_{r1}^{-1} \nabla \times \mathbf{E}_1) &= -\gamma_\tau(\mu_{r2}^{-1} \nabla \times \mathbf{E}_2) && \text{on } \Gamma \\
\gamma_\tau(\mu_{r1}^{-1} \nabla \times \mathbf{E}_1) &= jk_1 \mu_{r1}^{-1} \pi_\tau(\mathbf{E}_1) && \text{on } \partial\Omega_1 \\
\gamma_\tau(\mu_{r2}^{-1} \nabla \times \mathbf{E}_2) &= jk_2 \mu_{r2}^{-1} \pi_\tau(\mathbf{E}_2) && \text{on } \partial\Omega_2
\end{aligned} \tag{5.1}$$

where $\mathbf{E}_i \in \mathbf{H}_0(\text{curl}, \Omega_i)$ is the electric field in sub-domain and simple first order absorbing boundary condition(ABC) on $\partial\Omega$ is assumed. The impressed electric current is given such that $\mathbf{J}_i^{\text{imp}} \in \mathbf{H}_0(\text{div}, \Omega_i)$. By Galerkin testing procedure, the test functions reside in the dual space of each residual in above equations. On the interface Γ, there are two sets of fields and constitutive parameters. Particularly, we choose $\tilde{k} = (k_1 + k_2)/2$ and $\tilde{\mu_r} = (\mu_{r1} + \mu_{r1})/2$ at Γ, then the middle two relations of Equation 5.1 is

$$\begin{aligned}
\gamma_\tau(\mu_{r1}^{-1} \nabla \times \mathbf{E}_1) - j\tilde{k}\tilde{\mu_r}^{-1} \pi_\tau(\mathbf{E}_1) &= -\gamma_\tau(\mu_{r2}^{-1} \nabla \times \mathbf{E}_2) - j\tilde{k}\tilde{\mu_r}^{-1} \pi_\tau(\mathbf{E}_2) \\
\gamma_\tau(\mu_{r2}^{-1} \nabla \times \mathbf{E}_2) - j\tilde{k}\tilde{\mu_r}^{-1} \pi_\tau(\mathbf{E}_2) &= -\gamma_\tau(\mu_{r2}^{-1} \nabla \times \mathbf{E}_1) - j\tilde{k}\tilde{\mu_r}^{-1} \pi_\tau(\mathbf{E}_1).
\end{aligned} \tag{5.2}$$

Equation 5.2 is denoted as Robin Type transmission condition(TC). It resembles first order ABC. Note that TC allows $\mathbf{E}_i$ to be discontinuous on Γ and here only tangential continuity is enforced for $\mathbf{E}_i$ and normal continuity for $\mu_{ri}^{-1}\nabla \times \mathbf{E}_i$. The details of weak formulation is listed in [95].

In the context of non-overlapping DD methods, convergence is directly related to the TC. By including higher order derivatives in the transverse direction, the convergence can be improved.

5.2.2 Optimized Transmission Condition for MSDDM

The transverse electric transmission condition(TETC) is formulated as [96],

$$\begin{aligned} &\gamma_\tau(\mu_{r1}^{-1}\nabla \times \mathbf{E}_1) + \alpha_1\pi_\tau(\mathbf{E}_1) + \beta\mathcal{S}_{TE}(\pi_\tau(\mathbf{E}_1)) \\ =& -\gamma_\tau(\mu_{r2}^{-1}\nabla \times \mathbf{E}_2) + \alpha_2\pi_\tau(\mathbf{E}_2) + \beta\mathcal{S}_{TE}(\pi_\tau(\mathbf{E}_2)) \\ &\gamma_\tau(\mu_{r2}^{-1}\nabla \times \mathbf{E}_2) + \alpha_2\pi_\tau(\mathbf{E}_2) + \beta\mathcal{S}_{TE}(\pi_\tau(\mathbf{E}_2)) \\ =& -\gamma_\tau(\mu_{r1}^{-1}\nabla \times \mathbf{E}_1) + \alpha_1\pi_\tau(\mathbf{E}_1) + \beta\mathcal{S}_{TE}(\pi_\tau(\mathbf{E}_1)) \end{aligned} \tag{5.3}$$

with complex coefficients α and β to be determined. $\mathcal{S}_{TE}(\cdot) := \nabla_\tau \times \nabla_\tau \times$ denotes the transverse electric(TE) operator on tangential direction. Note that the form of TCs in Equation 5.3 is taken from an approximation to the TE transparent condition, the exact relation satisfied by a TE-polarized plane wave incident upon an infinite plane [97].

When the TE term $\mathcal{S}_{TE}(\pi_\tau(\mathbf{E}_i))$ is removed from Equation 5.3, the transmission condition degenerates to Robin-type. To deal with non-planar interfaces and non-conformal meshes, a auxiliary unknown is then defined as $\mathbf{j}_i = \frac{1}{k_0}\gamma_\tau(\mu_{ri}^{-1}\nabla \times \mathbf{E}_i) \in \mathbf{L}_0^2(\Gamma_{ij})$ or $\mathbf{H}_0(\mathrm{div}, \Gamma_{ij})$ depending on the boundary type. The TETC can be written

as,

$$\begin{aligned} k_0\mathbf{j}_1 + \alpha_1\pi_\tau(\mathbf{E}_1) + \beta\mathcal{S}_{TE}(\pi_\tau(\mathbf{E}_1)) &= -k_0\mathbf{j}_2 + \alpha_2\pi_\tau(\mathbf{E}_2) + \beta\mathcal{S}_{TE}(\pi_\tau(\mathbf{E}_2)) \\ k_0\mathbf{j}_2 + \alpha_2\pi_\tau(\mathbf{E}_2) + \beta\mathcal{S}_{TE}(\pi_\tau(\mathbf{E}_2)) &= -k_0\mathbf{j}_1 + \alpha_1\pi_\tau(\mathbf{E}_1) + \beta\mathcal{S}_{TE}(\pi_\tau(\mathbf{E}_1)). \end{aligned} \tag{5.4}$$

Equation 5.4 is tested with $\mathbf{u}_1$ *in* $\mathbf{L}_0^2(\Gamma_{12})$ resulting,

$$\begin{aligned} & k_0 \langle \mathbf{u}_1, \mathbf{j}_1 \rangle_{\Gamma_{12}} + \alpha_1 \langle \mathbf{u}_1, \mathbf{e}_1 \rangle_{\Gamma_{12}} + \beta \langle \mathbf{u}_1, \mathcal{S}_{TE}(\mathbf{e}_1) \rangle_{\Gamma_{12}} \\ = & -k_0 \langle \mathbf{u}_1, \mathbf{j}_2 \rangle_{\Gamma_{12}} + \alpha_2 \langle \mathbf{u}_1, \mathbf{e}_2 \rangle_{\Gamma_{12}} + \beta \langle \mathbf{u}_1, \mathcal{S}_{TE}(\mathbf{e}_2) \rangle_{\Gamma_{12}}, \end{aligned} \tag{5.5}$$

here $\mathbf{e}_i := \pi_\tau(\mathbf{E}_i)$.

In the left hand side, integration by parts can simply TE term as

The testing term on $\mathcal{S}_{TE}$ in Equation 5.5 gives,

$$\langle \mathbf{u}_1, \mathcal{S}_{TE}(\mathbf{e}_1) \rangle_{\Gamma_{12}} = \langle \nabla_\tau \times \mathbf{u}_1, \nabla_\tau \times \mathbf{e}_1 \rangle_{\Gamma_{12}} \tag{5.6}$$

From a implementation perspective, any solver with basis function $\mathbf{e}$ can form such a inner product term. Under the framework of MSDDM where each subdomain may be taken care of by different solvers with various basis function or even unknown settings. For the right hand side, such a relation may not hold if the other solver's basis function is in a different space. In other word, the neighboring subdomain may not provide $\nabla_\tau \times \mathbf{e}_2$. The solution here is to rewrite the TE term. First recall that for a 3D vector $\mathbf{u} = (u_1, u_2, u_3)^T$, the curl is defined by: $\nabla \times \mathbf{u} = (\partial_2 u_3 - \partial_3 u_2, \partial_3 u_1 - \partial_1 u_3, \partial_1 u_2 - \partial_2 u_1)^T$. Define the trace operator on the interface Γ_{ij} as ∇_τ, we have

$$\nabla_\tau \times \pi_\tau(\mathbf{u}_i) = \mathbf{n} \cdot (\partial_1 u_2 - \partial_2 u_1) = \mathbf{n}(\mathbf{n} \cdot \nabla \times \mathbf{u}) \tag{5.7}$$

. The TE term can be rewritten as,

$$\begin{aligned} \mathcal{S}_{TE}(\pi_\tau(\mathbf{E}_i)) &= \nabla_\tau \times (\mathbf{n}(\mathbf{n} \cdot \nabla \times \mathbf{E}_i)) \\ &= -j\omega\mu_i \nabla_\tau \times \mathbf{n}(\mathbf{n} \cdot \mathbf{H}_i) \\ &= -j\omega\mu \nabla_\tau (\mathbf{H}_i \cdot \mathbf{n}) \times \mathbf{n} \end{aligned} \tag{5.8}$$

The right hand side is available as long as the normal component of magnetic field $\mathbf{H}_i$ is available from neighboring subdomain which is indicated by Equation 5.8. With such reformulation of TETC, only tangential electric field , normal magnetic field and current are needed in the implementation.

5.2.3 Hierarchical parallel preconditioning

In this section, we present a adaptive hierarchical parallel preconditioning technique which takes advantage of the division of the subdomains on computing nodes. Let's assume there are p nodes with N_j sub-domains on each node. On the first level, the matrix is divided based on the partition of subdomains reside in separate nodes. Consider the case where there are $p = 2$ parallel computing nodes, the matrix $\mathbf{A}$ is written as:$\mathbf{A} = [\mathbf{K}_{1,1}, \mathbf{D}_{1,2}; \mathbf{D}_{2,1}, \mathbf{K}_{2,2}]$, where the subdomain matrices $\mathbf{K}_{i,i}$ and coupling matrices $\mathbf{C}_{i,j}$ are defined as:

$$\mathbf{K}_{1,1} = \begin{bmatrix} \mathbf{A}_1 & \dots & \mathbf{C}_{1,N_1} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{N_1,1} & \dots & \mathbf{A}_{N_1} \end{bmatrix} \mathbf{K}_{2,2} = \begin{bmatrix} \mathbf{B}_1 & \dots & \mathbf{C}_{1,N_2} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{N_2,1} & \dots & \mathbf{B}_{N_2} \end{bmatrix}$$

The off-diagonal blocks $\mathbf{D}_{i,j}$ indicates the coupling matrices between subdomains on separated nodes.

$$\mathbf{D}_{1,2} = \begin{bmatrix} \mathbf{D}_{1,1} & \dots & \mathbf{D}_{1,N_2} \\ \vdots & \ddots & \vdots \\ \mathbf{D}_{N_1,1} & \dots & \mathbf{D}_{N_1,N_2} \end{bmatrix} \mathbf{D}_{2,1} = \begin{bmatrix} \mathbf{D}_{1,1} & \dots & \mathbf{D}_{1,N_1} \\ \vdots & \ddots & \vdots \\ \mathbf{D}_{N_2,1} & \dots & \mathbf{D}_{N_2,N_1} \end{bmatrix}$$

Matrices $\mathbf{A}_i(\mathbf{B}_i)$ are subdomains distributed to the same computing node. The first level preconditioner M^J is of Jacobi type which is applied to the subblock $\mathbf{K}_{i,i}$ and accounts for the coupling between subdomains in different nodes. The application of M^J can be written in a matrix-vector multiplication formulation as illustrated by

line 9 in Algorithm 6. The second level preconditioner M^{GS} is of Gauss-Seidel type which only accounts for the coupling between subdomains within same computing node. This algorithm can be easily extended to $p > 2$ cases.

Algorithm 6 Hierarchical parallel matrix-vector multiplication

Input: $A_i, C_{i,j}, M$

Output: $R = M^{-1}AX$

$X^0 \leftarrow X$

for Node $j = 1, p$ **do** ▷ in parallel

  for $i = 1, N_j$ **do**

    $t_i = \mathbf{0}$

    for $k \in$ neighbour(i) **do**

      if $k \notin$ Node(j) **then**

        $t_i \leftarrow t_i + D_{i,k}X_k^0$

      else

        $t_i \leftarrow t_i + C_{i,k}X_k$

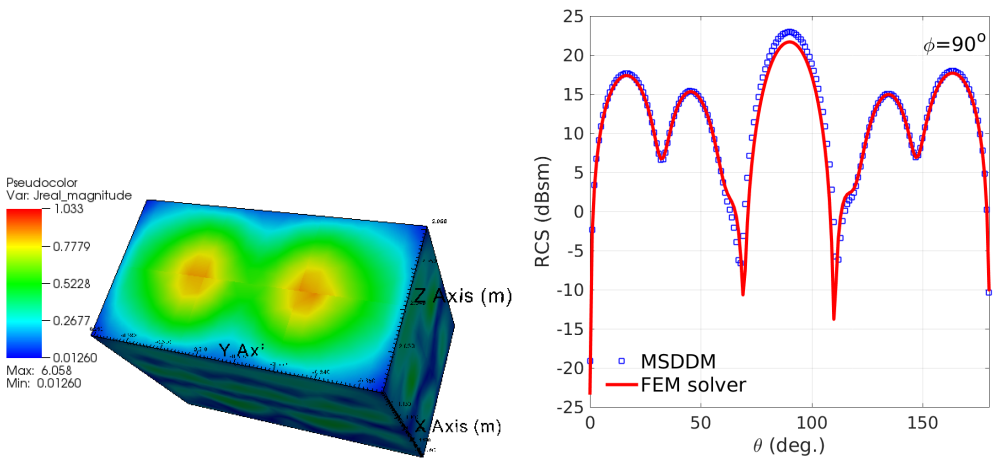
    $X_i \leftarrow A_i^{-1}t_i$

$R_i \leftarrow X_i^0 - X_i$

5.3 Numerical Experiments

5.3.1 Accuracy and Far Field RCS

A Vivaldi antenna array with 6 elements opearting at 9 GHz is simulated. The surface current distribution and the far field radar cross section (RCS) are shown in Figure 5.2.



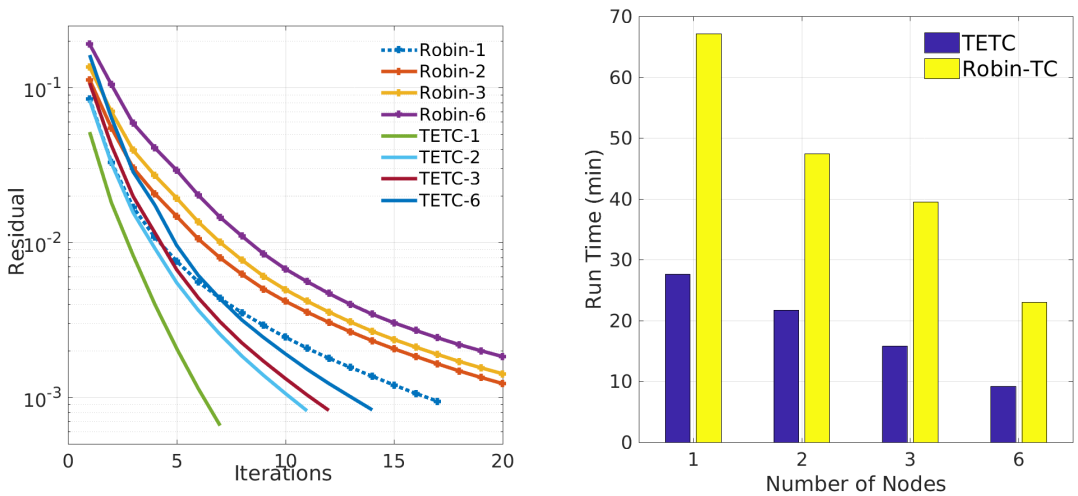
(a) Surface Current (b) RCS

Figure 5.2: Currents and RCS of Vivaldi array

The RCS of our method shows good consistency with the one from conventional FEM solver.

5.3.2 Convergence Performance of hierarchical preconditioner

The convergence history and run time statistics are illustrated in Figure 5.3. The curve "Robin-6"(or "TETC-6") denotes MSDDM with Robin-type(or TETC) transmission condition on 6 computing nodes. In this case, 6 antenna elements are distributed to each node and the hybrid preconditioner degenerates to pure Jacobi type.



(a) Convergence of residual

(b) Run time statistics

Figure 5.3: Performance of hierarchical preconditioner

With the increasing of computing nodes, the hierachical preconditioner improve the convergence performance. Meanwhile, TETC decreases the iterations required by half compared with Robin-type transmission condition.

So far we have addressed the need for a fast and parallel multi-solvers in solving complex antenna simulation problems. The work results in optimized transmission condition under MSDDM framework which demands least requirement for subdomain solvers. Besides, a hierarchical preconditioning technique is developed with great adaptability to parallel computing environments. Moreover, this work has a great potential for multi-scale, multi-physics problems.

5.4 Platform-Aware In-Situ Antenna and Metamaterial Analysis and Design

Modern military and commercial EM systems are routinely equipped with multiple antennas serving for radar and wireless communications. The computational electromagnetics (CEM) has emerged as a powerful and indispensable tool to evaluate the in-situ performance and co-site interference. These simulations are enabled by fast and rigorous numerical solutions of Maxwell's Equations as well as rapid advances in high-performance computing (HPC) systems. On present-day HPC clusters, the in-situ antenna performance on high-fidelity platforms of electrical length of the order of 1000 wavelengths can be analyzed within a couple of hours.

The aim of this section is to address this pressing and challenging engineering need. For many in-situ antenna design and parameter studies, it is typical that those antennas are only allowed to be mounted on certain parts of the platform. Namely, the computational domain can be decomposed into large fixed parts and small portions involving the antenna design. The observation inspires us to investigate a highly efficient, reduced order in-situ antenna analysis framework.

Key ingredients are summarized as follows: **(1)** The discontinuous Galerkin (DG) boundary element (BE) method [98] and geometry-aware domain decomposition (DD) method [99] are employed to facilitate a modular design-oriented decomposition. **(2)** A novel platform Green's function (PGF) is introduced on the outer surface of those antennas. The PGF is calculated once in the offline phase to characterize the coupling with large, fixed platform. It can be reused for all future in-situ computation. **(3)** In the online phase, rapid solution for multi-query antenna design needs is achieved by a Schwarz DD solver of the reduced order system. The computational costs are the same as the free-space radiation.

5.4.1 Problem decomposition

Consider the in-situ analysis of two antennas on the high-definition platform, as illustrated in Figure 5.4. The problem can be decomposed into three sub-regions: sub-regions Ω_1 and Ω_2 contain two antennas, sub-region Ω_3 is electrically large PEC platform. We have used cubical surfaces, Γ_{31} and Γ_{32}, to facilitate the decomposition. The size of the surfaces is determined by the prescribed surface area where the antennas are allowed to be mounted.

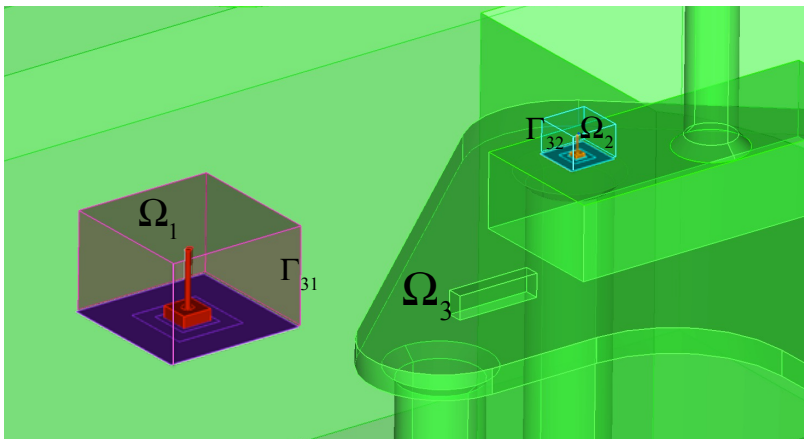


Figure 5.4: DD based model reduction for in-situ antenna analysis.

The finite element (FE) method is used to discretized the volume domain of Ω_1 and Ω_2, and the DG boundary element method is applied to the exterior surface of Ω_1 and Ω_2 and platform Ω_3. The resulting system matrix can be written as:

$$\begin{bmatrix} \begin{bmatrix} \mathcal{A}_1^{\mathrm{FE}} & \mathcal{N}_1 \\ \mathcal{N}_1^T & \mathbf{A}_1^{\mathrm{BE}} \end{bmatrix} & & \mathbf{C}_{12}^{\mathrm{BE}} & \mathbf{C}_{13}^{\mathrm{BE}} \\ & \begin{bmatrix} \mathcal{A}_2^{\mathrm{FE}} & \mathcal{N}_2 \\ \mathcal{N}_2^T & \mathbf{A}_2^{\mathrm{BE}} \end{bmatrix} & & \mathbf{C}_{23}^{\mathrm{BE}} \\ \mathbf{C}_{21}^{\mathrm{BE}} & & & \\ \mathbf{C}_{31}^{\mathrm{BE}} & & \mathbf{C}_{32}^{\mathrm{BE}} & \mathbf{A}_3^{\mathrm{BI}} \end{bmatrix}. \tag{5.9}$$

The resulting sub-domain FE matrices, $\mathcal{A}_m^{\mathrm{FE}}$, m =1,2, have complex nonlinear dependency on a few design parameters (ϵ_m, μ_m, $\mathbf{r}_m$, etc.). The sub-domain BE matrices $\mathrm{A}_m^{\mathrm{BE}}$, m =1,2,3, and coupling matrices $\mathbf{C}_{mn}^{\mathrm{BE}}$ are resulting from free-space Green's function, and has no parametric dependency.

Evidently, the above non-overlapping and non-conforming DG and DD methods lead to a modular design-oriented decomposition. The modification of antenna types, parameters, locations in the design stage is reflected in local and sparse FE matrices, which are decoupled from exterior BE matrices. Nevertheless, every time antenna design is modified, we still need to solve the entire system matrix. That is where the platform Green's function comes into play.

5.4.2 Platform Green's function

The PGF is evaluated on the artificial surfaces, $\Gamma_3 = \Gamma_{31} \cup \Gamma_{32}$, in the offline computing phase. Once calculated, the large fixed platform is rigorously represented by the PGF. The direct calculation of the PGF matrix requires the solution of the large platform Ω_3 with respect to individual unit source currents on Γ_3. Recognizing the coupling between antennas and platform is considerably low-rank, we proposed a novel alternating and random interpolative decomposition (AR-ID) to select the skeleton source currents from the original ones on the exterior surface Γ_3. The AR-ID calculation can be achieved locally per antenna sub-system and embarrassingly in parallel. The reduced model of in-situ antenna system can be written as:

$$\begin{bmatrix} \begin{bmatrix} \mathcal{A}_1^{\mathrm{FE}} & \mathcal{N}_1 \\ \mathcal{N}_1^T & \tilde{\mathbf{A}}_1^{\mathrm{BE}} \end{bmatrix} & & \tilde{\mathbf{C}}_{12}^{\mathrm{BE}} \\ & \begin{bmatrix} \mathcal{A}_2^{\mathrm{FE}} & \mathcal{N}_2 \\ \mathcal{N}_2^T & \tilde{\mathbf{A}}_2^{\mathrm{BE}} \end{bmatrix} & \\ \tilde{\mathbf{C}}_{21}^{\mathrm{BE}} & & \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^{\mathrm{FE}} \\ \mathbf{x}_1^{\mathrm{BE}} \\ \mathbf{x}_2^{\mathrm{FE}} \\ \mathbf{x}_2^{\mathrm{BE}} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1^{\mathrm{FE}} \\ 0 \\ \mathbf{b}_2^{\mathrm{FE}} \\ 0 \end{bmatrix}. \tag{5.10}$$

The updated BE matrices consist of the free-space GF matrices and the AR-ID representation of PGF matrices:

$$\tilde{\mathbf{A}}_1^{\mathrm{BE}} = \mathbf{A}_1^{\mathrm{BE}} - \mathbf{R}_{13}^{\mathrm{n_1 \times k_1}} \cdot \mathbf{S}_{11}^{\mathrm{k_1 \times k_1}} \cdot \mathbf{R}_{31}^{\mathrm{k_1 \times n_1}}, \tag{5.11}$$

$$\tilde{\mathbf{C}}_{12}^{\mathrm{BE}} = \mathbf{C}_{12}^{\mathrm{BE}} - \underbrace{\mathbf{R}_{13}^{\mathrm{n_1 \times k_1}} \cdot \mathbf{S}_{12}^{\mathrm{k_1 \times k_2}} \cdot \mathbf{R}_{32}^{\mathrm{k_2 \times n_2}}}_{\text{Platform GF matrix}}, \tag{5.12}$$

where $\mathbf{S}_{\mathrm{mn}}^{\mathrm{k_m \times k_n}} = \mathbf{B}_{\mathrm{m3}}^{\mathrm{k_m \times n_3}} \cdot \left[\mathbf{A}_3^{\mathrm{BE}}\right]^{-1} \cdot \mathbf{B}_{\mathrm{3n}}^{\mathrm{n_3 \times k_n}}$. The $\mathrm{n_1}$, $\mathrm{n_2}$, $\mathrm{n_3}$ are the number of BE unknowns on antennas and platform. The $\mathrm{k_1}$ and $\mathrm{k_2}$ are skeleton BE unknowns.

During the design stage (online computing), Equation 5.10 is solved by Krylov iterative methods with an additive Schwarz preconditioner [100]. We remark that the online computing complexity does not depend on the size of the in-situ platform. As a result, in-situ design and optimization of multi-antenna systems can be performed at the same cost as the free-space radiation.

5.4.3 Illustration

We consider a metasurface antenna [101] mounted on the top surface of a PEC platform, as shown in Figure 5.5. In the offline calculation, we first generate separate BE meshes on the exterior surface of antenna and the platform. The PGF matrix is then constructed and assembled in the AR-ID representation as in Equation 5.11. We note that the PGF matrix is introduced to characterize the coupling with the platform only, and separately compressed with the free-space GF matrix. Thereby, the rank of the PGF matrix is extremely low, as the skeleton BE unknowns in Figure 5.6(b), and the free-space GF matrix can still be compressed with the fast multipole method. In the online phase, we generate the volume meshes for different antennas independently. The FE matrices are combined with the same GF matrices in Equation 5.11. The in-situ radiation patterns are shown in Figure 5.7.

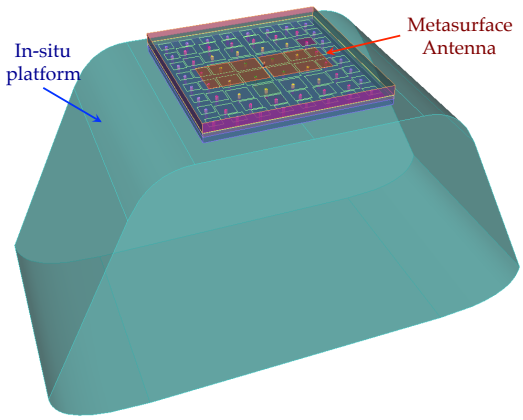
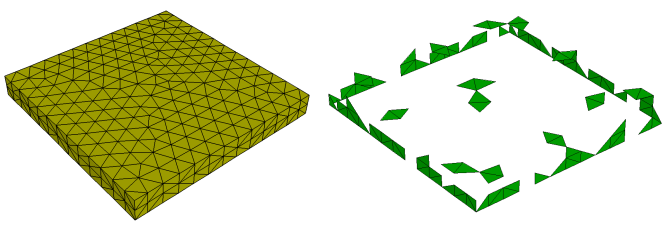
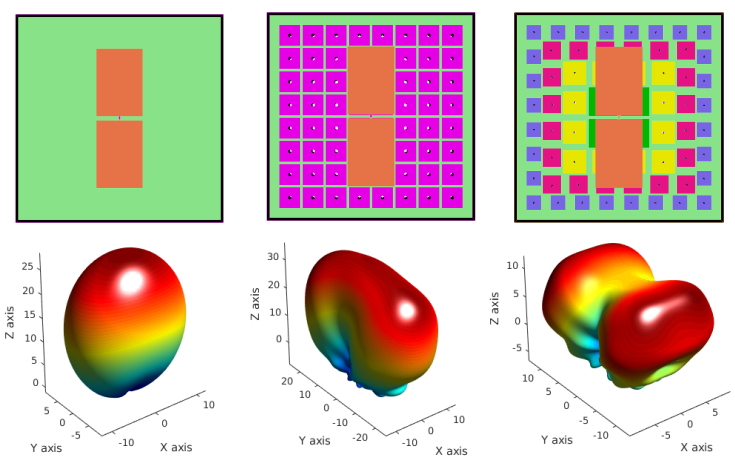


Figure 5.5: Example of a metasurface antenna on PEC platform.



(a) BE unknowns: n = 859 (b) skeleton BE unknowns: k = 42

Figure 5.6: Illustration of a data-sparse representation of PGF.



(a) Patch antenna (b) Patch antenna with uniform EBG (c) Patch antenna with NUA EBG

Figure 5.7: Radiation pattern of in-situ antennas at 550MHz.

The complete process can be summaried by the flow chart in Figure 5.8.

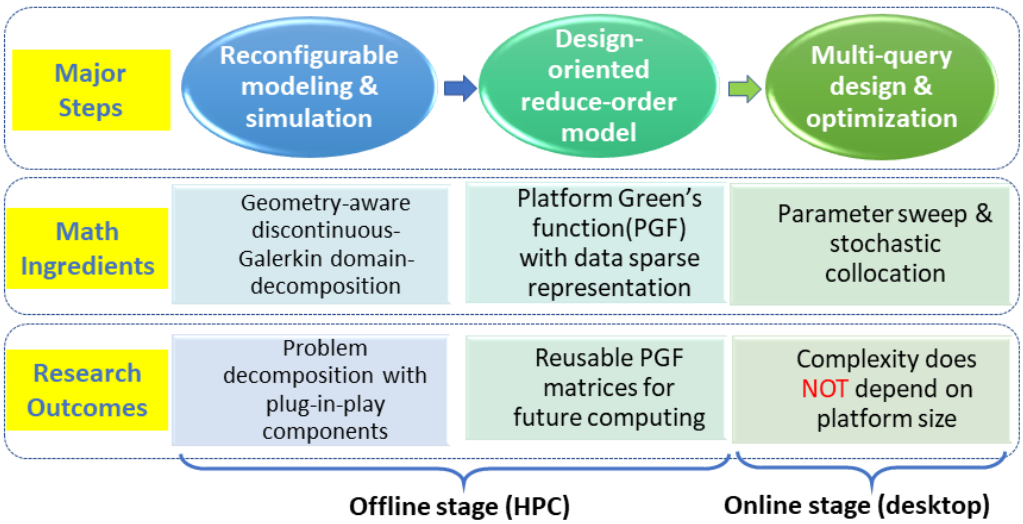
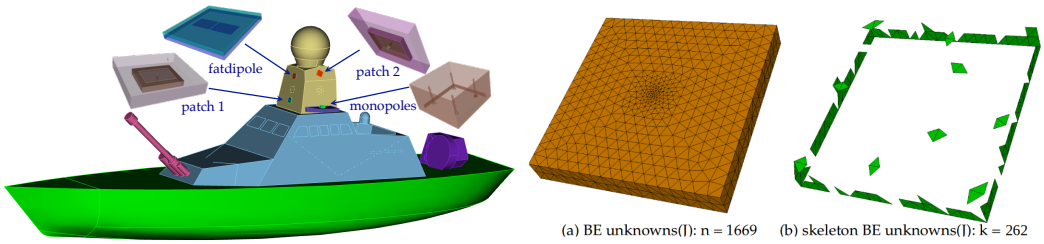


Figure 5.8: Flow chart of platform Green's function method

5.5 Numerical Examples

Rapid analysis for multiple antennas on ship mast

In this example, interference among 4 mounted antennas at frequency 450MHz is studied. There are 1 fat dipole antenna, 1 monopole antenna array with 4 elements and 2 patched antennas mounting on separate locations on a ship mast as shown in Figure 5.9. The ship is modeled by boundary element method and is partitioned into 12 subdomains. The first step is to obtain the platform Green's function. The DOFs and skeletons unknowns for each components are listed in the Table 5.1.



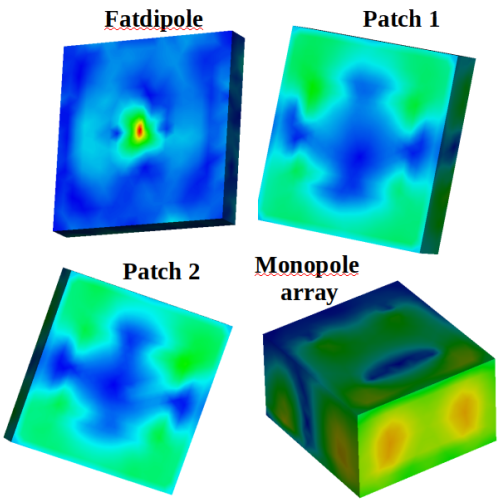
(a) Schemeatic of ship and antennas (b) Skeleton unknowns of fat dipole's air-box

Figure 5.9: Setup of ship and antennas problem

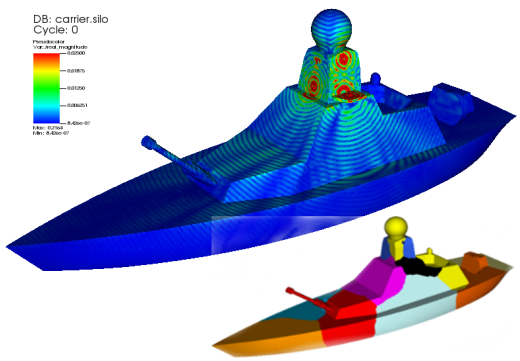
Table 5.1: Object dimensions

Object	fatdipole	patch1	patch2	monopole	ship body
FE DOFs	31563	118672	118672	54043	NA
BE DOFs	1669	2953	2953	1803	164352
skeletons(J/M)	262/254	252/243	243/239	93/97	NA

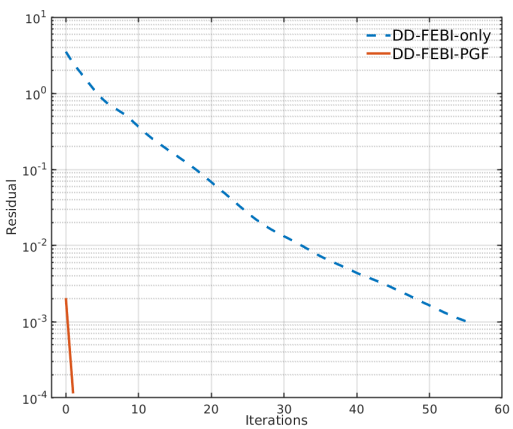
In the online stage, four antennas are turned on one at a time and S_{11} are recorded. The simulation and reference results are listed in Figure 5.10 and Table 5.2. Approximately 240 times speed-up is achieved.



(a) Surface current on antennas



(b) Surface current on ship body



(c) Convergence history of residual

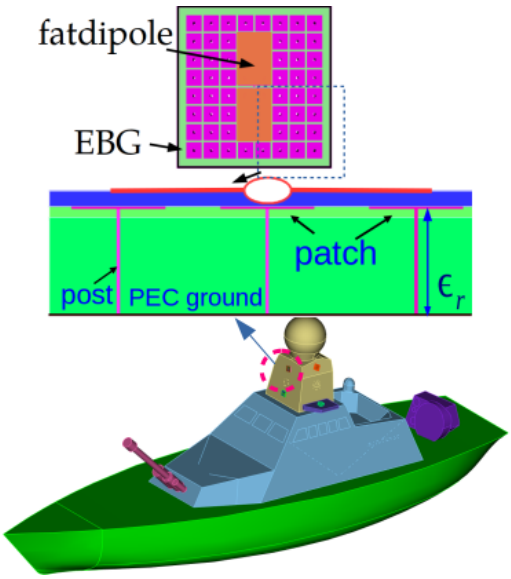
Figure 5.10: Simulation results of antennas and ship

Table 5.2: Simulation results (reference / this work)

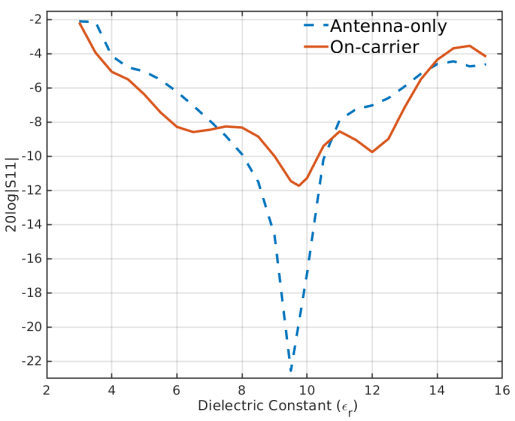
Excitation	fatdipole	patch1	patch2	monopole
S-parameter	0.81 / 0.81	0.44 / 0.42	0.45 / 0.45	0.56 / 0.56
DD iterations	80 / 2	59 / 2	52 / 2	41 / 1
Runtime (s)	1712 / 4	1297 / 5	1061 / 4	852 / 3

Platform-aware design parameter sweep

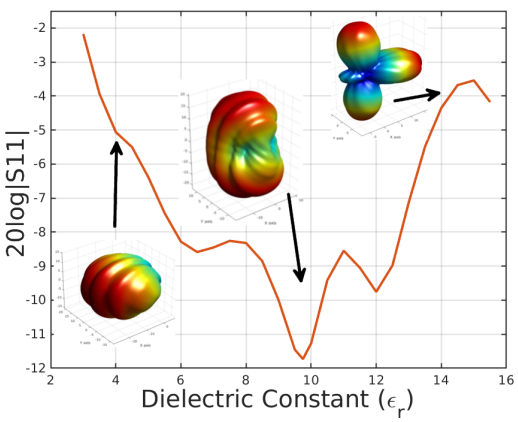
In this part we show a example for design parameter sweep with platform Green's functions available. Similary to the last example, the PEC ground for fat dipole antenna is replaced by electric bandgap metasurface as in Figure 5.11. The parameter for sweep for sweep is substrate's dielectric constant(ϵ_r) and 60 points within range $[2, 16]$ are studied. For single simulation, the runtime is 6s as compare to 2500s in non-PGF case.



(a) Fat dipole antenna on EBG



(b) $S11$



(c) Field patterns

Figure 5.11: Simulation results of parameter sweep

Chapter 6

Conclusion

In this dissertation, we have developed several parallel algorithms for computational electromagnetics in both time and frequency domain based on the idea of domain decomposition, reduce-order modeling, parallel in time and on-line/off-line separation. The goal is to make better use of computing resources, either reduce simulation-to-result-time or improve scaling efficiency. Several applications are also investigated to verify the effectiveness of these algorithms.

In Chapter 2, a space-time parallel domain decomposition algorithm for transient analysis is developed. To assist the decomposition of problem, Krylov subspace based matrix exponential integrators are investigated. Also we proposed a heuristic strategy for pole selection. The results indicate a improvement of parallel efficiency over parallel-in-space-only method through representative numerical examples. The efficiency can reach more than 80% on 96 cores .

In Chapter 3, we developed a space-time domain decomposition and building block methodology to increase the concurrency and efficiency for time domain EM simulations by taking advantage of the repetition in both spatial and temporal domains. Based on the decomposition of the original problem, various integration methods are proposed respectively. The proposed method is particularly suitable for multi-scale, locally periodic EM problems with millions of time steps in temporal dimension. The advancements are verified in simulation of photonic bandgap microstrip structure and meta-surface antenna. Approximately 10 times speed-up is observed in 1D repetition units and 3 times speed-up in 2D repetition units.

In Chapter 4, parallel-in-time algorithm is extended to electromagnetic circuit co-simulation problems by linear/nonlinear decomposition. First, a state-variable form of co-simulation is developed to formulate a system of nonlinear ODE. Next, a iterative refinement scheme is proposed to perform parallel-in-time. Some practices in implementation are also addressed. Finally, a effective coarse φ integrator is designed for linear sub-problem. Numerical examples verify that parallel-in-time

algorithm applied in co-simulation displays twice better parallel efficiency over traditional parallel-in-space-only method.

In Chapter 5, an optimized transmission condition under MSDDM framework which demands least information exchange from neighboring subdomain solvers is first proposed. A hierarchical parallel preconditioning technique is also developed to utilize computing resource available in a more flexible way. To demonstrate the accuracy and advantages, a Vivaldi array is simulated. The RCS result shows that the accuracy of proposed method is guaranteed. The convergence and run time behave exactly like expected. Next, to address the pressing challenging in in-situ antenna design, a novel platform Green's function is proposed. With this work, complex problem can be decomposed into plug-in-play components and PGF matrices are reusable for computing. A large ship mounted with several antennas is simulated. Results show that the on-line computation can achieve 240 times speed-up without compromising accuracy. Parameter sweep becomes efficient due to the reusability of most computationally intensive coupling part.

The future directions of this work include incorporation of higher order transmission condition(transverse magnetic term) under the MSDDM framework. For the platform Green's function, how to speed-up the off-line stage computations would be the future work. For the co-simulation problems, the next step would be the intelligent space-time grid attribution.

References

[1] W. C. Chew, M. S. Tong, and B. Hu, "Integral equation methods for electromagnetic and elastic waves," *Synthesis Lectures on Computational Electromagnetics*, vol. 3, no. 1, pp. 1–241, 2008.

[2] W. C. Chew, E. Michielssen, J. Song, and J.-M. Jin, *Fast and efficient algorithms in computational electromagnetics*. Artech House, Inc., 2001.

[3] J.-M. Jin, *The finite element method in electromagnetics*. John Wiley & Sons, 2015.

[4] P. Monk *et al.*, *Finite element methods for Maxwell's equations*. Oxford University Press, 2003.

[5] X. Wang, Z. Peng, K.-H. Lim, and J.-F. Lee, "Multisolver domain decomposition method for modeling emc effects of multiple antennas on a large air platform," *IEEE Transactions on Electromagnetic Compatibility*, vol. 54, no. 2, pp. 375–388, 2011.

[6] Y. Shao, Z. Peng, K. H. Lim, and J.-F. Lee, "Non-conformal domain decomposition methods for time-harmonic maxwell equations," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 468, no. 2145, pp. 2433–2460, 2012.

[7] V. Shankar, A. H. Mohammadian, and W. F. Hall, "A time-domain, finite-volume treatment for the maxwell equations," *Electromagnetics*, vol. 10, no. 1-2, pp. 127–145, 1990.

[8] S. D. Gedney and U. Navsariwala, "An unconditionally stable finite element time-domain solution of the vector wave equation," *IEEE Microwave and Guided wave letters*, vol. 5, no. 10, pp. 332–334, 1995.

[9] J.-F. Lee and Z. Sacks, "Whitney elements time domain (WETD) methods," *IEEE Transactions on Magnetics*, vol. 31, pp. 1325–1329, May 1995.

[10] J.-F. Lee, R. Lee, and A. Cangellaris, "Time-domain finite-element methods," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 430–442, Mar 1997.

[11] D. Jiao, J.-M. Jin, E. Michielssen, and D. J. Riley, "Time-domain finite-element simulation of three-dimensional scattering and radiation problems using perfectly matched layers," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 2, pp. 296–305, Feb 2003.

[12] Z. Lou and J.-M. Jin, "A novel dual-field time-domain finite-element domain-decomposition method for computational electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 6, pp. 1850–1862, June 2006.

[13] N. Marais and D. B. Davidson, "Numerical evaluation of high-order finite element time domain formulations in electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 12, pp. 3743–3751, Dec 2008.

[14] W. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and efficient algorithms in computational electromagnetics*. Boston: MA: Artech House, 2001.

[15] D. S. Weile, G. Pisharody, N.-W. Chen, B. Shanker, and E. Michielssen, "A novel scheme for the solution of the time-domain integral equations of electromagnetics," *IEEE transactions on Antennas and Propagation*, vol. 52, no. 1, pp. 283–295, 2004.

[16] A. E. Yilmaz, J.-M. Jin, and E. Michielssen, "Time domain adaptive integral method for surface integral equations," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 10, pp. 2692–2708, Oct 2004.

[17] A. Yilmaz, J.-M. Jin, and E. Michielssen, "A TDIE-based asynchronous electromagnetic-circuit simulator," *IEEE Microwave Wireless Compon. Lett.*, vol. 16, no. 3, pp. 122–124, 2006.

[18] B. Shanker, M. Lu, J. Yuan, and E. Michielssen, "Time domain integral equation analysis of scattering from composite bodies via exact evaluation of radiation fields," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 5, pp. 1506–1520, May 2009.

[19] K. Yee, "Numerical solution of initial boundary value problems involving Maxwell's Equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, vol. 14, no. 3, pp. 302–307, May 1966.

[20] T. Weiland, "Time domain electromagnetic field computation with finite difference methods," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 9, no. 4, pp. 295–319, 1996. [Online]. Available: http://dx.doi.org/10.1002/(SICI)1099-1204(199607)9:4⟨295::AID-JNM240⟩3.0.CO;2-8

[21] A. Taflove and S. C. Hagness, *Computational electrodynamics: the finite-difference time-domain method*. Artech house, 2005.

[22] N. V. Venkatarayalu, R. Lee, Y. Gan, and L. Li, "A stable fdtd subgridding method based on finite element formulation with hanging variables," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 907–915, March 2007.

[23] A. Elsherbeni and V. Demir, *The Finite-Difference Time-Domain Method for Electromagntetics with MATLAB Simulations*. Raleigh, NC: SciTech, 2009.

[24] L. Fezoui, S. Lanteri, S. Lohrengel, and S. Piperno, "Convergence and stability of a discontinuous Galerkin time-domain method for the 3D heterogeneous Maxwell equations on unstructured meshes," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 39, no. 6, pp. 1149–1176, 2005.

[25] J. S. Hesthaven and T. Warburton, "Nodal high-order methods on unstructured grids: I. time-domain solution of maxwell's equations," *Journal of Computational Physics*, vol. 181, no. 1, pp. 186–221, 2002.

[26] H. Fahs and S. Lanteri, "A high-order non-conforming discontinuous Galerkin method for time-domain electromagnetics," *Journal of Computational and Applied Mathematics*, vol. 234, no. 4, pp. 1088 – 1096, 2010, proceedings of the Thirteenth International Congress on Computational and Applied Mathematics (ICCAM-2008), Ghent, Belgium, 7â11 July, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042709003070

[27] S. Dosopoulos and J. F. Lee, "Interior penalty discontinuous galerkin finite element method for the time-dependent first order maxwell's equations," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 12, pp. 4085–4090, Dec 2010.

[28] P. Li, Y. Shi, L. J. Jiang, and H. Bagci, "A hybrid time-domain discontinuous Galerkin-boundary integral method for electromagnetic scattering analysis," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 5, pp. 2841–2846, May 2014.

[29] Q. H. Liu, "Large-scale simulations of electromagnetic and acoustic measurements using the pseudospectral time-domain (pstd) algorithm," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 917–926, March 1999.

[30] B. Yang and J. Hesthaven, "Multidomain pseudospectral computation of maxwell's equations in 3-d general curvilinear coordinates," *Applied Numerical Mathematics*, vol. 33, no. 1, pp. 281 – 289, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016892749900094X

[31] Q. Huo Liu and G. Zhao, "Review of pstd methods for transient electromagnetics," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 17, no. 3, pp. 299–323. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jnm.544

[32] T. Weiland, "A discretization model for the solution of Maxwell's equations for six-component fields," *Archiv Elektronik und Uebertragungstechnik*, vol. 31, pp. 116–120, Mar. 1977.

[33] T. Weiland, "Time domain electromagnetic field computation with finite difference methods," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 9, no. 4, pp. 295–319, 1996. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291099-1204%28199607%299%3A4%3C295%3A%3AAID-JNM240%3E3.0.CO%3B2-8

[34] P. B. Johns, "A symmetrical condensed node for the TLM method," *IEEE Transactions on Microwave Theory and Techniques*, vol. 35, no. 4, pp. 370–377, Apr 1987.

[35] J. Alvarez, L. D. Angulo, M. F. Pantoja, A. R. Bretones, and S. G. Garcia, "Source and boundary implementation in vector and scalar dgtd," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 6, pp. 1997–2003, 2010.

[36] J. Alvarez, J. Alonso-Rodriguez, H. Carbajosa-Cobaleda, M. R. Cabello, L. D. Angulo, R. G. Martin, and S. G. Garcia, "Dgtd for a class of low-observable targets: A comparison with mom and (2, 2) fdtd," *IEEE Antennas and Wireless Propagation Letters*, vol. 13, pp. 241–244, 2014.

[37] L. Diaz Angulo, J. Alvarez Gonzalez, F. Teixeira, M. Fernandez Pantoja, S. Gonzalez Garcia *et al.*, "A nodal hybrid continuous-discontinuous galerkin time domain method for maxwell's curl equations," 2015.

[38] R. Lucas, J. Ang, K. Bergman, S. Borkar, W. Carlson, L. Carrington, G. Chiu, R. Colwell, W. Dally, J. Dongarra *et al.*, "Doe advanced scientific computing

advisory subcommittee (ascac) report: top ten exascale research challenges," USDOE Office of Science (SC)(United States), Tech. Rep., 2014.

[39] K. Bittner and H. Brachtendorf, "Multi-rate coupled circuit-device simulation," in *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*. IEEE, 2016, pp. 42–47.

[40] K. Zhao, V. Rawat, and J.-F. Lee, "A domain decomposition method for electromagnetic radiation and scattering analysis of multi-target problems," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 8, pp. 2211–2221, 2008.

[41] A. Toselli and O. Widlund, *Domain decomposition methods-algorithms and theory*. Springer Science & Business Media, 2006, vol. 34.

[42] E. Lelarasmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelli, "The waveform relaxation method for time-domain analysis of large scale integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 1, no. 3, pp. 131–145, July 1982.

[43] J. White, A. Sangiovanni-Vincentelli, F. Odeh, and A. Ruehli, "Waveform relaxation: theory and practice," *Transactions of the Society for Computer Simulation*, vol. 2, pp. 95–133, 1985.

[44] M. J. Gander, L. Halpern, and F. Nataf, "Optimal convergence for overlapping and nonoverlapping Schwarz waveform relaxation," *Eleventh International Conference on Domain Decomposition Methods*, pp. 27–36, 1999.

[45] S. Vandewalle, *Parallel Multigrid Waveform Relaxation for Parabolic Problems*. Stuttgart: Vieweg+Teubner Verlag, 1993.

[46] J.-L. Lions, Y. Maday, and G. Turinici, "Resolution d'EDP par un schema en temps parareal," *Comptes Rendus de l'AcadÃ©mie des Sciences - Series I - Mathematics*, vol. 332, no. 7, pp. 661 – 668, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0764444200017936

[47] G. Bal and Y. Maday, *A "Parareal" Time Discretization for Non-Linear PDE's with Application to the Pricing of an American Put*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 189–202. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-56118-4_12

[48] Y. Maday and G. Turinici, "A parareal in time procedure for the control of partial differential equations," *Comptes Rendus Mathematique*, vol. 335, no. 4, pp. 387 – 392, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1631073X02024676

[49] ——, *The Parareal in Time Iterative Solver: a Further Direction to Parallel Implementation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 441–448. [Online]. Available: http://dx.doi.org/10.1007/3-540-26825-1_45

[50] M. J. Gander and S. Vandewalle, "Analysis of the parareal time parallel time integration method," *SIAM Journal on Scientific Computing*, vol. 29, no. 2, pp. 556–578, 2007.

[51] C. Farhat and M. Chandesris, "Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications," *International Journal for Numerical Methods in Engineering*, vol. 58, no. 9, pp. 1397–1434, 2003. [Online]. Available: http://dx.doi.org/10.1002/nme.860

[52] G. Horton, "The time-parallel multigrid method," *Communications in Applied Numerical Methods*, vol. 8, no. 9, pp. 585–595, 1992. [Online]. Available: http://dx.doi.org/10.1002/cnm.1630080906

[53] G. Horton and S. Vandewalle, "A space-time multigrid method for parabolic partial differential equations," *SIAM Journal on Scientific Computing*, vol. 16, no. 4, pp. 848–864, 1995.

[54] C. Cong, X.-C. Cai, and K. Gustafson, "Implicit space-time domain decomposition methods for stochastic parabolic partial differential equations," *SIAM Journal on Scientific Computing*, vol. 36, no. 1, pp. C1–C24, 2014. [Online]. Available: http://dx.doi.org/10.1137/12090410X

[55] M. Emmett and M. L. Minion, "Toward an efficient parallel in time method for partial differential equations," *Comm.App.Math. and Comp. Sci*, vol. 7, pp. 105–132, 2012.

[56] A. J. Christlieb, C. B. Macdonald, and B. W. Ong, "Parallel high-order integrators," *SIAM Journal on Scientific Computing*, vol. 32, no. 2, pp. 818–835, 2010.

[57] R. Speck, D. Ruprecht, M. Emmett, M. Bolten, and R. Krause, "A space-time parallel solver for the three-dimensional heat equation," *Advances in Parallel Computing*, vol. 25, pp. 263–272, 2014.

[58] M. J. Gander, *50 Years of Time Parallel Time Integration*. Cham: Springer International Publishing, 2015, pp. 69–113. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-23321-5_3

[59] M. J. Gander and S. GuÌttel, "Paraexp: A parallel integrator for linear initial-value problems," *SIAM Journal on Scientific Computing*, vol. 35, no. 2, pp. C123–C142, 2013.

[60] M. Merkel, I. Niyonzima, and S. Schöps, "An application of paraexp to electromagnetic wave problems," in *Electromagnetic Theory (EMTS), 2016 URSI International Symposium on*. IEEE, 2016, pp. 121–124.

[61] M. A. Botchev, "Krylov subspace exponential time domain solution of maxwellâs equations in photonic crystal modeling," *Journal of Computational and Applied Mathematics*, vol. 293, pp. 20 – 34, 2016, efficient Numerical Methods for Large-scale Scientific Computations. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037704271500240X

[62] J. Chen and Q. H. Liu, "Discontinuous Galerkin time-domain methods for multiscale electromagnetic simulations: A review," *Proceedings of the IEEE*, vol. 101, no. 2, pp. 242–254, Feb 2013.

[63] V. Dolean, H. Fahs, L. Fezoui, and S. Lanteri, "Locally implicit discontinuous Galerkin method for time domain electromagnetics," *Journal of Computational Physics*, vol. 229, no. 2, pp. 512 – 526, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999109005300

[64] S. Lanteri and C. Scheid, "Convergence of a discontinuous Galerkin scheme for the mixed time-domain Maxwell's equations in dispersive media," *IMA Journal of Numerical Analysis*, vol. 33, no. 2, pp. 432–459, 2013. [Online]. Available: +http://dx.doi.org/10.1093/imanum/drs008

[65] J. Young, "A full finite difference time domain implementation for radio wave propagation in a plasma," *Radio science*, vol. 29, no. 6, pp. 1513–1522, 1994.

[66] C. A. Kennedy and M. H. Carpenter, "Additive Runge-Kutta schemes for convection-diffusion-reaction equations," *Applied Numerical Mathematics*, vol. 44, no. 1, pp. 139 – 181, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168927402001381

[67] C. Sun and C. Trueman, "Unconditionally stable Crank-Nicolson scheme for solving two-dimensional Maxwell's equations," *Electronics Letters*, vol. 39, no. 7, pp. 595–597, 2003.

[68] A. Kanevsky, M. H. Carpenter, D. Gottlieb, and J. S. Hesthaven, "Application of implicit-explicit high order Runge-Kutta methods to discontinuous-Galerkin schemes," *Journal of Computational Physics*, vol. 225, no. 2, pp. 1753 – 1781,

2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999107000861

[69] S. Dosopoulos, B. Zhao, and J.-F. Lee, "Non-conformal and parallel discontinuous Galerkin time domain method for maxwellâs equations: Em analysis of ic packages," *Journal of Computational Physics*, vol. 238, pp. 48–70, 2013.

[70] J. Chen and Q. H. Liu, "Discontinuous Galerkin time-domain methods for multiscale electromagnetic simulations: A review," *Proceedings of the IEEE*, vol. 101, no. 2, pp. 242–254, 2013.

[71] S. D. Gedney, T. Kramer, C. Luo, J. A. Roden, R. Crawford, B. Guernsey, J. Beggs, and J. Miller, "The discontinuous Galerkin finite element time domain method (DGFETD)," in *Electromagnetic Compatibility, 2008. EMC 2008. IEEE International Symposium on.* IEEE, 2008, pp. 1–4.

[72] Y. Saad, "Analysis of some krylov subspace approximations to the matrix exponential operator," *SIAM Journal on Numerical Analysis*, vol. 29, no. 1, pp. 209–228, 1992.

[73] G. L. Kooij, M. A. Botchev, and B. J. Geurts, "A block krylov subspace implementation of the time-parallel paraexp method and its extension for nonlinear partial differential equations," *Journal of computational and applied mathematics*, vol. 316, pp. 229–246, 2017.

[74] H. D. Raedt, K. Michielsen, J. S. Kole, and M. T. Figge, "Solving the maxwell equations by the chebyshev method: a one-step finite-difference time-domain algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 11, pp. 3155–3160, Nov 2003.

[75] C. Moler and C. V. Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM Review*, vol. 45, no. 1, pp. 3–49, 2003. [Online]. Available: https://doi.org/10.1137/S00361445024180

[76] J. Chen, J. H. Lee, and Q. H. Liu, "A high-precision integration scheme for the spectral-element time-domain method in electromagnetic simulation," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 10, pp. 3223–3231, Oct 2009.

[77] M. Berljafa, "Rational krylov decompositions: Theory and applications," Ph.D. dissertation, The University of Manchester, 2017.

[78] M. Berljafa and S. Guttel, "Generalized rational krylov decompositions with an application to rational approximation," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 2, pp. 894–916, 2015.

[79] V. Druskin, L. Knizhnerman, and M. Zaslavsky, "Solution of large scale evolutionary problems using rational krylov subspaces with optimized shifts," *SIAM Journal on Scientific Computing*, vol. 31, no. 5, pp. 3760–3780, 2009.

[80] J. H. Wilkinson, *The algebraic eigenvalue problem*. Clarendon Press Oxford, 1965, vol. 87.

[81] J. van den Eshof and M. Hochbruck, "Preconditioning lanczos approximations to the matrix exponential," *SIAM Journal on Scientific Computing*, vol. 27, no. 4, pp. 1438–1457, 2006.

[82] J. L. Blue and H. K. Gummel, "Rational approximations to matrix exponential for systems of stiff differential equations," *Journal of Computational Physics*, vol. 5, no. 1, pp. 70–83, 1970.

[83] M. Schreiber, P. S. Peixoto, T. Haut, and B. Wingate, "Beyond spatial scalability limitations with a massively parallel method for linear oscillatory problems," *The International Journal of High Performance Computing Applications*, p. 1094342016687625, 2017.

[84] H. Wang, L. Xu, B. Li, S. Descombes, and S. Lantéri, "A new family of exponential-based high order dgtd methods for modelling 3d transient multiscale electromagnetic problems," 2017.

[85] B. N. Parlett and Y. Saad, "Complex shift and invert strategies for real matrices," *Linear algebra and its applications*, vol. 88, pp. 575–595, 1987.

[86] A. Ruhe, "Rational krylov: A practical algorithm for large sparse nonsymmetric matrix pencils," *SIAM Journal on Scientific Computing*, vol. 19, no. 5, pp. 1535–1551, 1998.

[87] D. Skoogh, "A parallel rational krylov algorithm for eigenvalue computations," in *International Workshop on Applied Parallel Computing*. Springer, 1998, pp. 521–526.

[88] P. Kaps, S. W. Poon, and T. Bui, "Rosenbrock methods for stiff odes: A comparison of richardson extrapolation and embedding technique," *Computing*, vol. 34, no. 1, pp. 17–40, 1985.

[89] M. A. Botchev, V. Grimm, and M. Hochbruck, "Residual, restarting, and richardson iteration for the matrix exponential," *SIAM journal on scientific computing*, vol. 35, no. 3, pp. A1376–A1397, 2013.

[90] D. Schaub and D. Oliver, "Rapid simulation of linear pbg microstrip structures using the rayleigh multipole method," *IEEE transactions on microwave theory and techniques*, vol. 56, no. 1, pp. 49–55, 2008.

[91] M. El Badawe, T. S. Almoneef, and O. M. Ramahi, "A true metasurface antenna," *Scientific reports*, vol. 6, p. 19268, 2016.

[92] R. M. Kielkowski, *SPICE: practical device modeling.* McGraw-Hill, 1995.

[93] M. Lundstrom, "An ebers-moll model for the heterostructure bipolar transistor," *Solid-state electronics*, vol. 29, no. 11, pp. 1173–1179, 1986.

[94] L. Yang, M. Fan, and Z. Feng, "A spiral electromagnetic bandgap (ebg) structure and its application in microstrip antenna arrays," in *2005 Asia-Pacific Microwave Conference Proceedings*, vol. 3. IEEE, 2005, pp. 4–pp.

[95] K. Zhao, V. Rawat, S.-C. Lee, and J.-F. Lee, "A domain decomposition method with nonconformal meshes for finite periodic and semi-periodic structures," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 9, pp. 2559–2570, 2007.

[96] V. Rawat, "Finite element domain decomposition with second order transmission conditions for time-harmonic electromagnetic problems," Ph.D. dissertation, The Ohio State University, 2009.

[97] R. A. Renaut, "Higher order absorbing boundary conditions," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 10, p. 1215, 1992.

[98] Z. Peng, R. Hiptmair, Y. Shao, and B. MacKie-Mason, "Domain decomposition preconditioning for surface integral equations in solving challenging electromagnetic scattering problems," *IEEE Trans. Antennas and Propagation*, vol. 64, no. 1, pp. 210–223, Jan 2016.

[99] X.-Q. Sheng, J.-M. Jin, J. Song, C.-C. Lu, and W. C. Chew, "On the formulation of hybrid finite-element and boundary-integral methods for 3-D scattering," *IEEE Trans. Antennas and Propagation*, vol. 46, no. 3, pp. 303–311, Mar. 1998.

[100] B. MacKie-Mason, A. Greenwood, and Z. Peng, "Adaptive and parallel surface integral equation solvers for very large-scale electromagnetic modeling and simulation (invited paper)," *Progress In Electromagnetics Research, Invited paper for the Commemorative Collection on the 150-Year Anniversary of Maxwell's Equations*, vol. 154, pp. 143–162, 2015.

[101] N. Chamok, M. Ali, T. Anthony, and S. J. Weiss, "Ultra-thin uhf broadband antenna on a non-uniform aperiodic (nua) metasurface," *IEEE Antennas and Propagation Magazine*, vol. 57, no. 2, pp. 167–180, 2015.