Journal of Computational Physics 238 (2013) 48–70

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Non-conformal and parallel discontinuous Galerkin time domain method for Maxwell's equations: EM analysis of IC packages

Stylianos Dosopoulos *, Bo Zhao, Jin-Fa Lee

ElectroScience Laboratory, The Ohio State University, 1320 Kinnear Road, Columbus, OH 43212, United States

ARTICLE INFO

Article history:
Received 28 May 2012
Received in revised form 24 November 2012
Accepted 26 November 2012
Available online 27 December 2012

Keywords:
Discontinuous Galerkin
Time-domain
MPI parallelization
Local time-stepping
IC packages

ABSTRACT

In this article, we present an Interior Penalty discontinuous Galerkin Time Domain (IPDGTD) method on non-conformal meshes. The motivation for a non-conformal IPDGTD comes from the fact there are applications with very complicated geometries (for example, IC packages) where a conformal mesh may be very difficult to obtain. Therefore, the ability to handle non-conformal meshes really comes in handy. In the proposed approach, we first decompose the computational domain into non-overlapping subdomains. Afterward, each sub-domain is meshed independently resulting in non-conformal domain interfaces, but simultaneously providing great flexibility in the meshing process. The non-conformal triangulations at sub-domain interfaces can be naturally supported within the IPDGTD framework. Moreover, a MPI parallelization together with a local time-stepping strategy is applied to significantly increase the efficiency of the method. Furthermore, a general balancing strategy is described. Through a practical example with multi-scale features, it is shown that the proposed balancing strategy leads to better use of the available computational resources and reduces substantially the total simulation time. Finally, numerical results are included to validate the accuracy and demonstrate the flexibilities of the proposed non-conformal IPDGTD.

1. Introduction

The complexity of modern electromagnetic applications demands for sophisticated and efficient numerical methods. It can be argued that these requirements are even more critical when transient simulations are coveted. Moreover, recent hardware architectures such as multi-core CPUs and/or Graphics Processor Units (GPUs) are widely available for computation. As a consequence, numerical methodologies that exhibit high parallelism and can be mapped effectively to the latest hardware architectures are also highly desirable.

Discontinuous Galerkin (DG) finite element methods are a good candidate for time-domain simulations. DG methods can support various types and shapes of elements, non-conformal meshes and non-uniform orders of approximation. Additionally, a dispersion error of $O\left((hk)^{2p+3}\right)$ [1,2] can be shown (k is the exact wavenumber), where p is the polynomial approximation order and high orders of approximation can be easily realized. Moreover, since the tangential continuity of the fields and boundary conditions can be enforced in the weak sense, significant freedom in the choice of basis functions is available. In this way, a great amount of *flexibility* is available and this is a major strength of DG methods. Furthermore, the resulting

* Corresponding author.
E-mail address: sdosopoulos@gmail.com (S. Dosopoulos).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.11.048

mass matrix is a block diagonal matrix with the block size equal to the number of degrees of freedom in the element. Therefore, the method can lead to a fully explicit time-marching scheme for the solution in time. Finally, information exchange is required only between neighboring elements regardless of approximation order and shape, which then leads to high efficiency in parallelization.

In electromagnetism, DG methods were recently applied for the solution of the time dependent Maxwell's equations [3–7]. In [3], Heasthaven and Warburton developed a low storage, high-order Runge–Kutta DGTD method based on upwind fluxes. The authors showed that upwind fluxes, will result in optimal h-convergence rate $O(h^{p+1})$. Moreover, in [1,2] the authors present a $O\left((hk)^{2p+3}\right)$ error estimate for the dispersion error, and a $O\left((hk)^{2p+2}\right)$ for the dissipation error respectively, again for an upwind flux DGTD. On the other hand, Fezoui et al. [4] formulated an energy conservative DGTD method based on central fluxes and leap-frog discretization in time, but with sub-optimal h-convergence rate $O(h^{p})$. Furthermore, Montseny et al. [5] proposed a DGTD method for hexahedra with the option of either central or upwind flux. The authors used Gauss–Lobatto quadrature formulas to obtain a method with low cost in memory and CPU time. They concluded on a $O(h^{p-1})$ convergence rate for the central flux option and a $O(h^{p-1/2})$ convergence rate for the upwind flux case. Finally, in Fahs et al. [8] the authors studied a high order DGTD based on central fluxes and non-conformal meshes and results were shown for two dimensional problems.

This article is organized as follows. In Section 3, we describe the IPDGTD formulation and discuss the corresponding stability condition. Next, in Section 4, we present the current approach on the support for non-conformal meshes. We begin by partitioning the computational domain into non-overlapping sub-domains. Subsequently, each of the sub-domains is meshed independently resulting in non-conformal domain interfaces but simultaneously providing great flexibility in the meshing process. In this way a final non-conformal mesh can be obtained for computation. In Section 5, a MPI parallelization together with a local time-stepping (LTS) is proposed to significantly increase the efficiency of the IPDGTD method. The proposed MPI design follows the one presented in [9]. However, an additional and general balancing strategy is included herein, which provides a more efficient usage of the available computational resources and improves substantially the solution time. To the best of our knowledge, a parallel and non-conformal (both geometrically and mesh-wise) DGTD method for unstructured meshes in three dimensions has not been reported yet in the literature and this is the main contribution of this paper. Finally, some interesting numerical examples demonstrate the capabilities of the proposed approach to the analysis of engineering applications.

2. Original initial value problem (IVP)

We consider the time-dependent Maxwell's equations in three dimensions. The electric permittivity $\epsilon(\mathbf{r})$ and the magnetic permeability $\mu(\mathbf{r})$ are positive and varying in space.

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} \quad \text{in } \Omega \times [0, T] \tag{1}$$

$$\nabla \times \mathbf{H} = \epsilon \frac{\partial \mathbf{E}}{\partial t} \quad \text{in } \Omega \times [0, T] \tag{2}$$

$$\mathbf{E}(\mathbf{r}, t = 0) = \mathbf{E}_0, \quad \mathbf{H}(\mathbf{r}, t = 0) = \mathbf{H}_0 \tag{3}$$

The above equations are solved in a bounded domain $\Omega \subset R^3$. Moreover, we apply boundary conditions that are:

$$\hat{\mathbf{n}} \times \mathbf{E} = 0 \quad \text{on } \Gamma_{PEC}$$
$$\hat{\mathbf{n}} \times \mathbf{H} = 0 \quad \text{on } \Gamma_{PMC}$$

and/or some kind of absorbing boundary condition on the boundary $\partial\Omega$ of the computational domain, applied for the truncation of the computational domain.

3. IPDGTD formulation

In this section we briefly outline our DGTD method since a more detailed description of the formulation can be found in [6].

3.1. Trace operators and notations

Let Ω be the computational domain of interest and $\mathcal{T}_h$ the discretization of Ω into polyhedral elements, namely, $\mathcal{T}_h = \{K_i\}$. We denote by $\mathcal{F}_h^I$ the set of all interior faces, $\partial K_i \cap \partial K_j$, with K_i and K_j be two adjacent elements of $\mathcal{T}_h$. Moreover, by $\mathcal{F}_h^B$ we delineate the set of all boundary faces $\partial K_i \cap \partial\Omega$, such that $\mathcal{F}_h = \mathcal{F}_h^I \cup \mathcal{F}_h^B$. Next, we introduce the notations for the trace operators employed in our analysis. Define the tangential trace and projection ("components trace") operators, $\gamma_\tau(\cdot)$ and $\pi_\tau(\cdot)$ respectively, as $\gamma_\tau(\mathbf{u}_i) = \hat{\mathbf{n}}_i \times \mathbf{u}_i|_{\partial K_i}$ and $\pi_\tau(\mathbf{u}_i) = \hat{\mathbf{n}}_i \times (\mathbf{u}_i \times \hat{\mathbf{n}}_i)|_{\partial K_i}$ where $\hat{\mathbf{n}}_i$ is the boundary normal pointing out of the element K_i. Additionally, we have also adopted the following notations:

$$\text{on} \quad \mathcal{F}_h^I \begin{cases} \{\mathbf{u}\} = (\pi_\tau(\mathbf{u}_i) + \pi_\tau(\mathbf{u}_j))/2 \\ [\![\mathbf{u}]\!]_\gamma = \gamma_\tau(\mathbf{u}_i) + \gamma_\tau(\mathbf{u}_j) \\ [\![\mathbf{u}]\!]_\pi = \pi_\tau(\mathbf{u}_i) - \pi_\tau(\mathbf{u}_j) \end{cases} \tag{4}$$