GPU accelerated time domain DGA method for wave propagation problems on tetrahedral grids

Matteo Cicuttin\(^1\), Lorenzo Codecasa\(^2\), Bernard Kapidani\(^3\), Ruben Specogna\(^3\), Francesco Trevisan\(^3\)

\(^1\)Université Paris-Est, Cermics (ENPC), F-77455 Marne-la-Vallée, France
\(^2\)Politecnico di Milano, Dip. di Elettronica, Informazione e Bioingegneria, I-20133, Milano, Italy
\(^3\)Polytechnic Department of Engineering and Architecture (DPIA), Università di Udine, I-33100 Udine, Italy

kapidani.bernard@spes.uniiud.it

The classical Finite Difference Time Domain method due to Yee is known to be massively parallelizable and efficiently deployable on Graphical Processing Units (GPUs). Classical FDTD however requires a Cartesian discretization of the computational domain, which limits the ability to model arbitrary geometries. The recently introduced explicit, consistent and conditionally stable DGATD method, which employs tetrahedral grids, enables full modeling flexibility while maintaining great computational efficiency. In this paper we present an evaluation of the DGATD method implemented on GPUs using NVIDIA CUDA libraries.

Index Terms—Time Domain methods, wave propagation, Parallel computing, Tetrahedral meshes

I. INTRODUCTION

The classical Finite Difference Time Domain (FDTD) method, devised by Yee \(^1\) on two interlocked Cartesian orthogonal grids, has been extensively studied and used for its simplicity and computational efficiency. This method is used to numerically solve Maxwell’s equations in a bounded region \(\Omega \subset \mathbb{R}^3\). We recall that the equations for the continuous case are

\[
\begin{aligned}
\frac{\partial d}{\partial t} &= \nabla \times h, \\
e &= \varepsilon^{-1} d, \\
\frac{\partial b}{\partial t} &= -\nabla \times e, \\
h &= \mu^{-1} b.
\end{aligned}
\]

(1a) \hspace{1cm} (1b) \hspace{1cm} (1c) \hspace{1cm} (1d)

The generic FDTD algorithm has been reinterpreted in the FIT framework \(^2\), which is also based on a discretization employing two interlocked grids, a primal grid \(\mathcal{G}\) and a dual grid \(\hat{\mathcal{G}}\) obtained by barycentric subdivision of \(\mathcal{G}\). This yields the discrete system of equations

\[
\begin{aligned}
\tilde{\Psi}^n - \tilde{\Psi}^{n-1} &= C\tilde{F}^{n-\frac{1}{2}}, \\
U^n &= M_{\varepsilon^{-1}} \tilde{\Psi}^n, \\
\Phi^{n+\frac{1}{2}} - \Phi^{n-\frac{1}{2}} &= -CU^n, \\
\tilde{F}^{n+\frac{1}{2}} &= M_{\mu^{-1}} \Phi^{n+\frac{1}{2}},
\end{aligned}
\]

(2a) \hspace{1cm} (2b) \hspace{1cm} (2c) \hspace{1cm} (2d)

in which \(\tilde{\Psi}^n, \tilde{\Psi}^{n-1}\) are the vectors of the fluxes of the electric displacement across the faces of \(\hat{\mathcal{G}}\) at time instant \(n\Delta t\) and \((n-1)\Delta t\) respectively, \(U^n\) is the vector of the line integrals of the electric field along the edges of \(\hat{\mathcal{G}}\) at time instant \(n\Delta t\); \(\Phi^{n+\frac{1}{2}}, \Phi^{n-\frac{1}{2}}\) are the vectors of the fluxes of the magnetic induction across the faces of \(\mathcal{G}\) at times \((n + \frac{1}{2})\Delta t\) and \((n - \frac{1}{2})\Delta t\); and \(\tilde{F}^{n+\frac{1}{2}}\) is the vector of the line integrals of the magnetic field along the edges of \(\hat{\mathcal{G}}\) at time instant \((n + \frac{1}{2})\Delta t\). Matrices \(C\) and \(\tilde{C}\) are the face-edge incidence matrices of \(\mathcal{G}\) and \(\hat{\mathcal{G}}\), respectively. Equations (2a) and (2c) are discrete equivalents of Ampère–Maxwell’s law and Faraday’s law respectively, and are exact equations. Equations (2b) and (2d) are discrete counterparts of the electric and magnetic constitutive relations and are instead approximate equations:

in which the performance is bound by how efficiently we can perform matrix-vector products. As with the original Cartesian orthogonal scheme \(^6\), we can show that the algorithm is amenable to massive parallelization: since each unknown in column vector \(U^n\), computed in (3a), depends only on the quantities computed in (3b) at time step \((n - \frac{1}{2})\Delta t\) in the two dual volumes that intersect the primal edge to which the unknown corresponds (see Fig. [1]), we can compute the contributions of each dual volume to \(U^n\) concurrently. Similarly, each
unknown in column vector $\mathbf{F}^{n+\frac{1}{2}}$, computed in (3b), depends only on the quantities computed in (3a) at time step $n\Delta t$ in the two primal volumes that intersect the dual edge to which the unknown corresponds (see Fig. 2), and again we can compute the contributions of each primal volume to $\mathbf{F}^{n+\frac{1}{2}}$ concurrently.

II. PRELIMINARY NUMERICAL RESULTS

The DGATD method was implemented inside our own C++ electromagnetic framework (EMT) using the NVIDIA cuSparse library, which allows to perform sparse matrix-vector product (SpMv) on GPU without having to write our own CUDA kernels. We used a TESLA C2075 accelerator for all tests. The accelerator features the double-precision Fermi microarchitecture and compute capability 2.0; the underlying processor supports 448 threads and the available memory is 6 GB. For all meshes, we set $\Delta t = 1$ ps to ensure stability on the finest mesh used. To add comparisons to a Cartesian orthogonal FDTD scheme, we used the free FDTD package MEEP [7]. It is worth noting that, as with Finite Elements, the DGATD method requires a smaller time step with respect to the Cartesian orthogonal FDTD scheme.

The accuracy and performance of the various competing methods is assessed on a problem for which a very accurate solution is available on Cartesian grids. We simulated a rectangular waveguide of size $5 \times 2.5$ cm and length 10 cm in the $z$ direction. At $z = 0$, a TE$_{10}$ electric field is applied while at the other end ($z = 10$ cm) a Perfect Electric Conductor (PEC) termination is applied.

III. CONCLUSIONS

Despite the GPU implementation of DGATD being preliminary and not fully optimized, we observe important speedups with respect to the CPU implementation. This suggests that the explicit nature of DGATD combined with the processing power of modern GPU accelerators can open the way for new and effective tools in computational electromagnetism.

REFERENCES