

# Analysis of periodic anisotropic media by means of split-field FDTD method and GPU computing

J. Francés<sup>a,b\*</sup>, S. Bleda<sup>a,b</sup>, M. L. Álvarez<sup>a,b</sup>, F. J. Martínez<sup>a</sup>, A. Márquez<sup>a,b</sup>, C. Neipp<sup>a,b</sup> and A. Beléndez<sup>a,b</sup>

<sup>a</sup>Dept. of Physics, Systems Engineering and Signal Theory, Alicante Univ./San Vicente del Raspeig Drive, San Vicente del Raspeig, Alicante, España E-3080

<sup>a</sup>University Institute of Physics to Sciences and Technologies, Alicante Univ./San Vicente del Raspeig Drive, San Vicente del Raspeig, Alicante, España E-3080

## ABSTRACT

The implementation of the Split-Field Finite Difference Time-Domain (SP-FDTD) method in Graphics Processing Units is described in this work. This formalism is applied to light wave propagation through periodic media with arbitrary anisotropy. The anisotropic media is modeled by means of a permittivity tensor with non-diagonal elements and absorbing boundary conditions are also considered. The split-field technique and the periodic boundary condition allow to consider a single period of the structure reducing the simulation grid. Nevertheless, the analysis of anisotropic media implies considering all the electromagnetic field components and the use of complex notation. These aspects reduce the computational efficiency of the numerical method compared to the isotropic and non-periodic implementation. With the upcoming of the new generation of General-Purpose Computing on Graphics Units many scientific applications have been accelerated and others are being developed into this new parallel digital computing architecture. Specifically, the implementation of the SP-FDTD in the Fermi family of GPUs of NVIDIA is presented. An analysis of the performance of this implementation is done and several applications have been considered in order to estimate the possibilities provided by both the formalism and the implementation into GPU.

The formalism has been used for analyzing different structures and phenomena: binary phase gratings and twisted-nematic liquid crystal cells. The numerical predictions obtained by means of the FDTD method here implemented are compared with theoretical curves achieving good results, thus validating the accuracy and the potential of the implementation.

**Keywords:** SF-FDTD, GPU computing, anisotropic media, diffraction efficiency, polarization gratings, binary phase gratings

## 1. INTRODUCTION

It has been demonstrated that the Finite-Difference Time-Domain (FDTD) method is an extremely important numerical method for solving Maxwell's equations. Since 1966 when Yee proposed the basis of this numerical method there have been a great number of contributions focused on extending its capabilities for modeling new applications.

Nonetheless, the potential of this method has not been massively exploited till the 80s and 90s when the performance of modern computers began to rise following the well known Moore's Law. The number of contributions related with FDTD has increased dramatically in the last twenty years due to the possibility of modeling complex structures easily with standard computers. Many authors focused their attention in the application of FDTD to periodic media. This kind of arrangements are usual in optical applications such as liquid crystal light valves and photonic wave guides for instance. The scheme proposed for modeling accurately this kind of problems is known as the Split-Field FDTD<sup>1,2</sup> formulation. This theory describes an efficient scheme for modeling lightwave propagation through periodic media by means of FDTD. Recently, SF-FDTD implementation schemes

---

Further author information: (Send correspondence to J. Francés)

J. Francés.: E-mail: jfmonllor@ua.es, Telephone: +34 96 590 3400 1158

for anisotropic media in two and three dimensions have been detailed in<sup>3,4</sup>. The adapting capability of FDTD makes easier to simulate complex structures compared to purely analytic methods, that hardly provide closed expressions for these situations.

However, modeling optical elements with anisotropic dielectric properties, such as waveplates and liquid crystal light valves implies considering a full tensor description of the relative dielectric permittivity. On the other hand, for correctly applying the Periodic Boundary Conditions (PBC) for arbitrary angle of incidence it is necessary to consider the complex notation of the electromagnetic field. In terms of computer resources this issue implies an increasing factor of two in the memory usage. The number of operations is also increased since real and imaginary parts of the fields must be considered. All these aspects reduce the performance of SF-FDTD in terms of computational resources and time simulation.

In the last years, many scientific areas are taking advantage of the general purpose Graphics Processing Units (GPUs) based on the Compute Unified Device Architecture<sup>5,6</sup> (CUDA). The analysis of complex systems in many cases requires massively computations that makes unfordable the processing by means of modern workstations. Many authors are focusing their efforts in how to adapt the implementation of numerical methods under the CUDA compliance. Some examples can be found in literature for different topics such as physics<sup>7,8</sup>, astronomy<sup>9,10</sup>, images<sup>11-13</sup>, biomedical applications<sup>14,15</sup> and computer science<sup>16</sup>. The acceleration of the SF-FDTD by means of GPU computing has been recently introduced by Shahmansouri<sup>17</sup> *et al.* In their work GPU computing is applied to accelerate the SF-FDTD implementation for Drude-Lorentz dispersive media. Nevertheless, the analysis of anisotropic media and GPU computing has not been performed under the best of our knowledge.

Here, a parallel implementation of SF-FDTD for anisotropic media is detailed and compared with the sequential version that runs under a single CPU. Considering anisotropy and a full tensor formulation for the dielectric permittivity implies more computer resources. Firstly, the isotropic version for both CPU and GPU are implemented and validated by means of the diffraction efficiency analysis of dielectric binary gratings. Secondly, the anisotropic version of the SF-FDTD is implemented for GPU computing and also validated by means of the analysis of a polarization grating. In both cases the normal and oblique incidence is considered for the validation of our implementation. The model of this system gives us the opportunity of simulating a wide range of devices such as those based on a Twisted-Nematic Liquid Crystal (TN-LC), Parallel Aligned LCoS or optical activity cavities. The degree of improvement of the GPU code is compared with the sequential version in terms of the relative SpeedUp that is defined as the difference between the sequential and parallel time simulations. This analysis has been performed for both scenarios isotropic and anisotropic media in order to demonstrate the possibilities offered by GPU computing in cases that imply massively computations.

## 2. SF-FDTD THEORY

In this work the Maxwell's equations for nonconducting and nonmagnetic anisotropic medium in phasor form are considered:

$$j\omega\epsilon_0\mathbf{E} = \kappa\nabla \times \mathbf{H}, \quad (1)$$

$$j\omega\mu_0\mathbf{H} = -\nabla \times \mathbf{E}, \quad (2)$$

where  $\omega$  is the angular frequency,  $\epsilon_0$  and  $\mu_0$  are the permittivity and the permeability of free space, respectively. For convenience the impermeability tensor  $\epsilon = \kappa^{-1}$  is introduced instead of  $\kappa$ , which is defined as

$$\epsilon = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} & \epsilon_{xz} \\ \epsilon_{yx} & \epsilon_{yy} & \epsilon_{yz} \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{bmatrix} = \mathbf{T}^{-1}(\alpha, \beta, \gamma) \begin{bmatrix} \epsilon_1 & 0 & 0 \\ 0 & \epsilon_2 & 0 \\ 0 & 0 & \epsilon_3 \end{bmatrix} \mathbf{T}(\alpha, \beta, \gamma), \quad (3)$$

where  $\epsilon_{1,2,3}$  are the relative dielectric constants corresponding to the principal axes,  $\mathbf{T}$  is the transformation matrix fully detailed in<sup>3</sup>, and  $\alpha$ ,  $\beta$  and  $\gamma$  are the Euler angles<sup>18</sup>.

In the Split-Field formulation plane waves propagate along a periodic structure with an angle of incidence  $\theta_0$ . The scheme of simulation is illustrated in Fig.1.

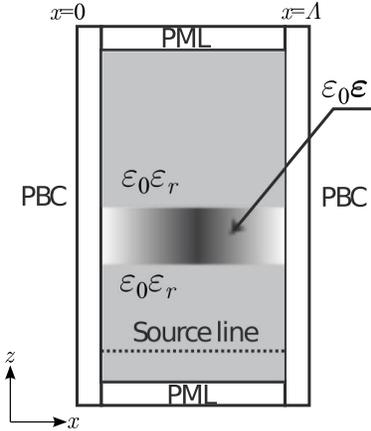


Figure 1. General problem space for FDTD simulations used in this work. The boundaries of the periodic media are established at  $x = 0$  and  $x = \Lambda$ , where  $\Lambda$  is the period of the structure.

By means of the wave-vector  $\mathbf{k}$  the new field variables  $P$  and  $Q$  which implicitly contain the oblique field propagation are defined:

$$\mathbf{P} = \mathbf{E} \exp(jk_x x), \quad (4)$$

$$\mathbf{Q} = c\mu_0 \mathbf{H} \exp(jk_x x), \quad (5)$$

where  $c$  is the speed of light for free space and  $k_x = (\omega/c) \sin \theta_0$ . Substituting Eqs. (4)-(5) into Eqs. (1)-(2) the expressions of each component can be obtained and discretized following the center difference expressions for both time and spatial derivatives<sup>3,4</sup>.

### 3. GPU IMPLEMENTATION OF THE SF-FDTD

Here a GTX470 GPU is considered for the parallel implementation of the SF-FDTD for anisotropic media. This GPU belongs to the Fermi architecture, released by NVIDIA in the spring of 2010. The memory hierarchy is one of the most distinguished features of the NVIDIA GPUs. The addition of a cache hierarchy, consisting of a global L2 cache, as well as a per-Streaming Multiprocessors (SM) L1 cache gives more flexibility in non-uniform memory accesses. The Fermi GTX470 features 448 Scalar Processors (SPs) organized in 14 SMs. Each SM has 32 Single Precision (SPs), 16 Double Precision (DPs), and 4 Special Functions Units (SFUs). Each SM also has a block of local memory called shared memory visible to all threads within a thread block, and a scheduling unit used to schedule warps, the basic computing unit, that consists of 32 threads. The GPU is capable of swapping warps into and out of context without any performance overhead. This functionality provides an important method of hiding memory and instruction latency on the GPU hardware.

To ease the mapping of data to threads, the threads identifiers may be multidimensional and, since a very high number of threads run in parallel, CUDA groups threads in blocks and grids. One of the crucial requirements to achieve a good performance on the NVIDIA GPU is to hide the high latency of the global memory ensuring coalesced memory accesses.

Therefore, concerning the architecture of the GPU is mandatory to be successful in GPU computing. Regarding the SF-FDTD implementation in the GPU it must be said that, a number of blocks related with the number of rows and columns are invoked by means of the kernel functions and an array of  $192 \times 2$  threads are launched per block. This scheme is illustrated in Fig. 2a in which a detail of the threads and their arrangement inside the grid of blocks is shown. Besides the potential of the CUDA kernel, it is necessary to divide the whole computation process in several kernels focused on computing each component of the electromagnetic field. Fig. 2b summarizes the invocation path of the kernels related with the SF-FDTD implementation. In this flow chart the post-process is omitted, but mandatory downloads of the  $P$  and  $Q$  components must be considered in order to compute the specific outputs desired such as diffraction efficiencies or Stokes parameters.

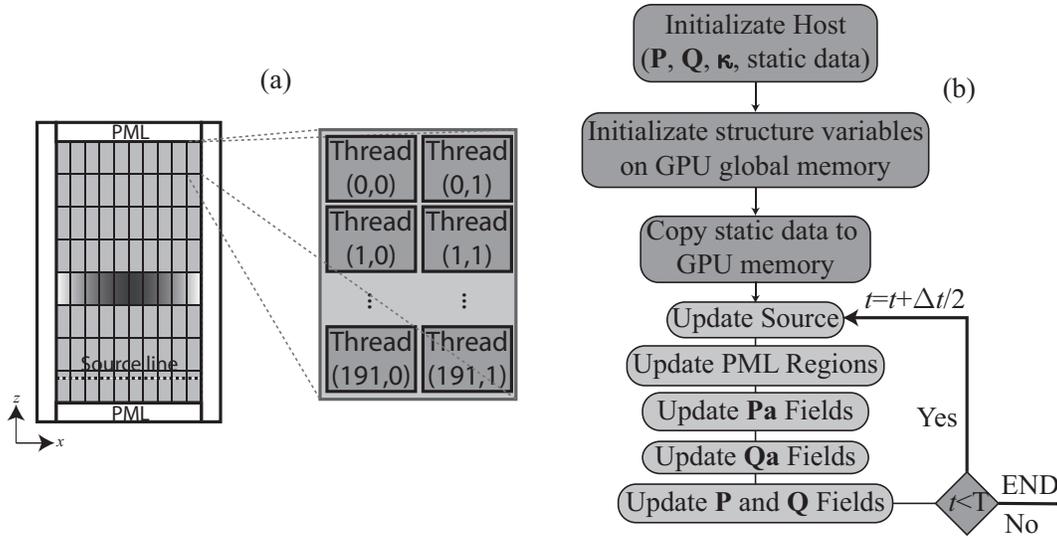


Figure 2. (a) Arrangement scheme of the grid of blocks invoked by CUDA Kernels with a detail of the threads that conform each block. (b) Flowchart of GPU parallel programming for SF-FDTD.

Table 1. SF-FDTD parameters.

$\lambda$ (nm)	$\Lambda$	$\Delta$ (nm)	$\Delta t$ (ps)	$R$	Rows	Columns	Time Steps
633	$3\lambda$	31.65	59.81	20	60	500	1200

The segmentation of the kernels is based on updating each electromagnetic component by means of a grid of blocks. This arrangement improves the efficient use of the shared memory in the device and also the correct usage of the cache. This effect is maximized in the new Fermi architecture, where each SM has 64 KB of on-chip memory that can be configured as 48 KB of Shared memory with 16 KB of L1 cache or as 16 KB of shared memory with 48 KB of L1 cache. Nonetheless, applications that do not use Shared memory automatically benefit from the L1 cache, allowing high performance CUDA programs to be built with minimum time and effort.

## 4. RESULTS

In this section the validation results of the implementation here described are summarized. This validation is done for both the isotropic and the anisotropic version. The systems simulated are based on the analysis performed in<sup>3,19</sup>. In all cases the software runs under a Unix based platform with an Intel Core i7-950 Processor with 8MB of cache, a clock speed of 3.06 GHz and 6 GB of global DDRAM3. Regarding the GPU, all the computations are processed by a NVIDIA GTX-470 card.

### 4.1 Validation of the isotropic version

In order to validate the isotropic version of the method a binary grating based on dielectric media is simulated. The binary grating has a period of  $\Lambda = 3\lambda$  and refractive index of  $n_g = 1.5$ . The output substrate is also a dielectric media of  $n_g$ . For illustrating this scheme the  $E_y$  field has been represented in Fig. 3. In this example, the binary grating can be easily identified inside the grid simulation. The angle of incidence for this case is  $30^\circ$  and the values of the FDTD parameters are listed in the Table 1. The spatial and time resolutions are denoted by  $\Delta$  and  $\Delta t$  respectively, whereas  $R$  is the cells density per wavelength.

Fig. 4 shows the comparison of the diffraction efficiency of different orders with RCWA and SF-FDTD. The diffraction efficiency of the 0th,  $\pm 1$  and  $\pm 2$  orders is computed for a grating with period  $\Lambda = 3\lambda$  as a function of the grating thickness. This analysis is also performed for oblique angle of incidence in Fig. 5 only for the zeroth

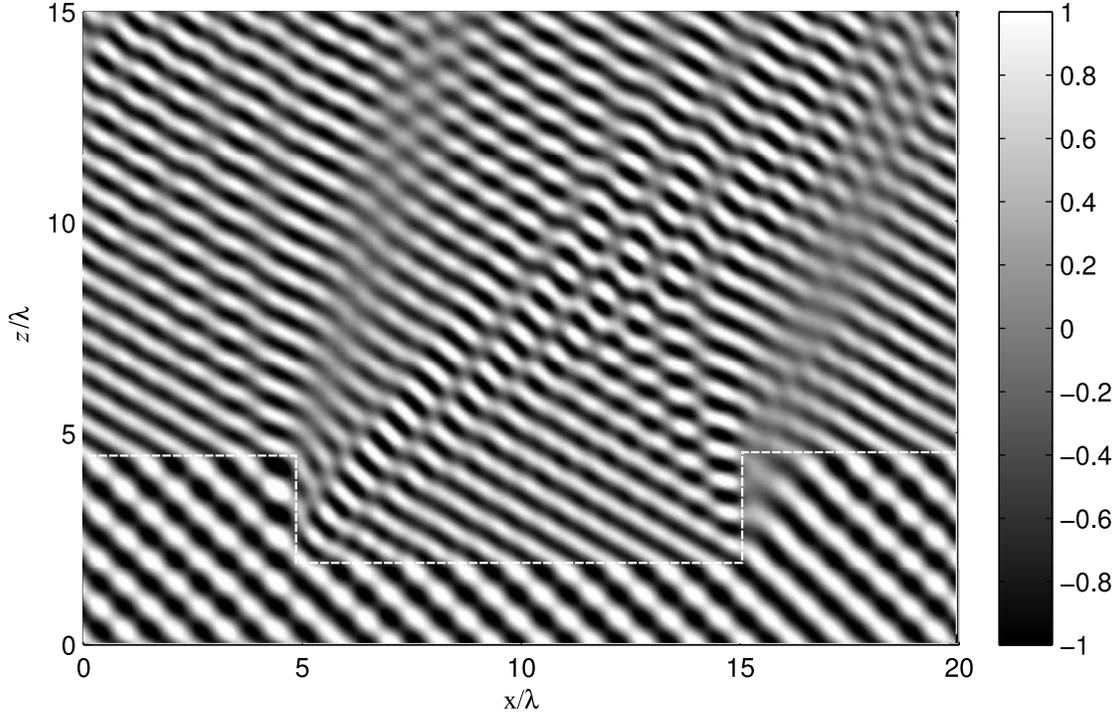


Figure 3. Near-field image of the  $E_y$  field at  $30^\circ$  incidence.

and first orders. In both cases there is a good agreement between RCWA and SF-FDTD results, thus validating our implementation.

## 4.2 Validation of the anisotropic version

For validating the anisotropic version a polarization grating with the same characteristics detailed in the work of Oh *et al.*<sup>3</sup> has been considered. The validation of this system gives us the opportunity of extrapolating this formalism to other optical devices such as twisted-nematic liquid crystal cells or Parallel Aligned LCoS. In this case, the periodic anisotropic structure varies along the  $x$ -direction with the following structure:

$$\epsilon(x) = \begin{bmatrix} n_{\perp}^2 \cos^2 \alpha + n_{\parallel}^2 \sin^2 \alpha & (n_{\perp}^2 - n_{\parallel}^2) \sin \alpha \cos \alpha & 0 \\ (n_{\perp}^2 - n_{\parallel}^2) \sin \alpha \cos \alpha & n_{\perp}^2 \sin^2 \alpha + n_{\parallel}^2 \cos^2 \alpha & 0 \\ 0 & 0 & n_{\perp}^2 \end{bmatrix}, \quad (6)$$

where  $\epsilon_{1,3} = n_{\perp}^2$ ,  $\epsilon_2 = n_{\parallel}^2$ ,  $\alpha(x) = \pi x/\Lambda$ , and  $\beta = \gamma = 0$ . The material parameters are  $n_{\perp} = 1.5$ ,  $n_{\parallel} = 1.7$  and  $\Delta n_l = n_{\parallel} - n_{\perp} = 0.2$ . In addition, gradient-index anti-reflection (AR) coatings are applied in air-PG boundaries<sup>3,20</sup>.

Fig. 6a shows the near field image of a PG captured from SF-FDTD simulations when input polarization is linear. It can be identified how the polarization grating provides circular polarization with both senses right and left hand at the output plane. This effect can be easily identified by means of the representation of the  $S_3$  Stokes parameter in Fig. 6b. These results are consistent with those detailed in previous works<sup>3</sup>.

In order to validate the library here implemented also in non-normal incidence a similar analysis has been performed varying the angle of incidence of the input plane wave. The new angle of incidence is  $\theta_0 = 10^\circ$  and the  $E_x$  and the  $S_3$  Stokes parameter are also represented in Fig. 7a and Fig. 7b respectively. This results are consistent with those obtained in Fig. 6, thus validating our implementation for anisotropic media.

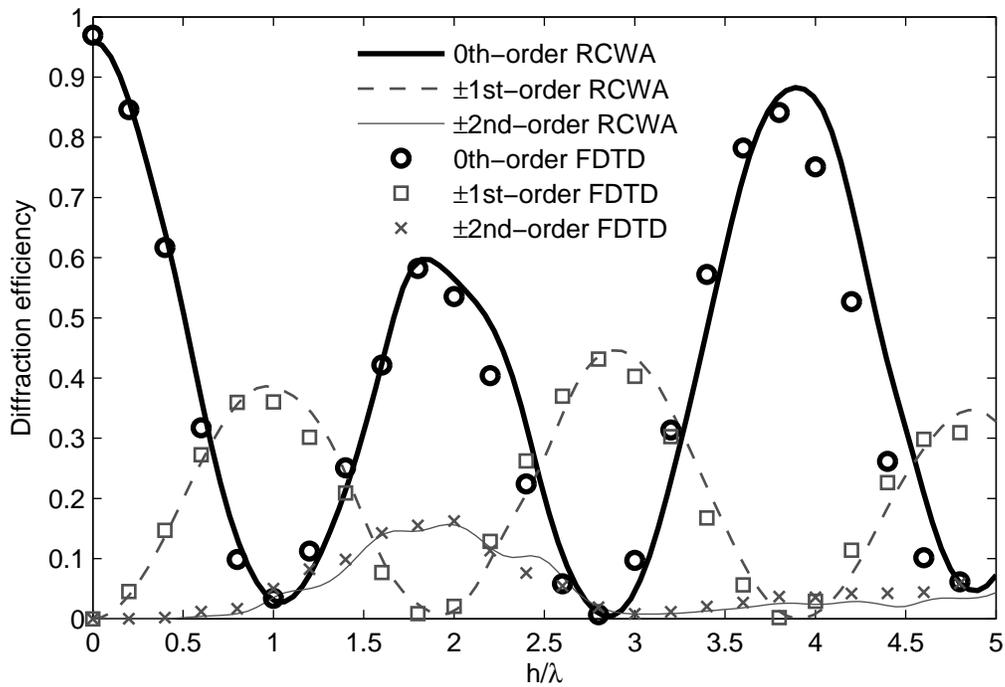


Figure 4. Comparison between RCWA and SF-FDTD of the diffraction efficiency. Parameters:  $\Lambda = 3\lambda$ ,  $\theta_0 = 0^\circ$ ,  $n_g = 1.5$ .

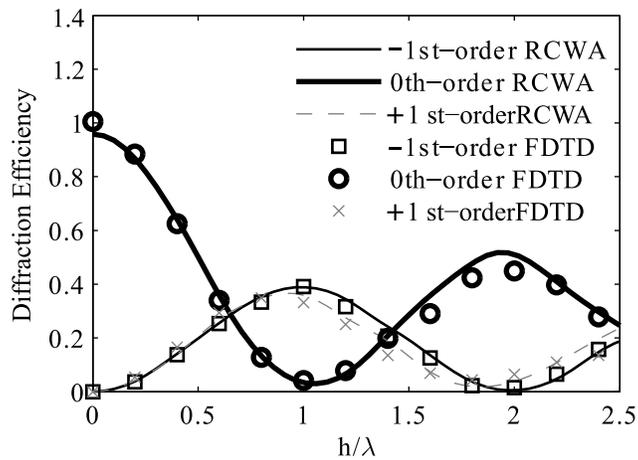


Figure 5. Comparison between RCWA and SF-FDTD of the diffraction efficiency. Parameters:  $\Lambda = 3\lambda$ ,  $\theta_0 = 15^\circ$ ,  $n_g = 1.5$ .

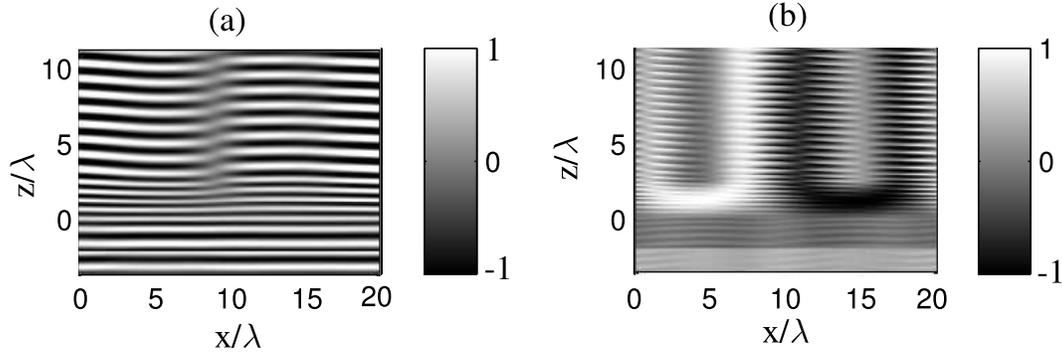


Figure 6. Polarization grating analysis for normal incidence: (a) Near-field image of the  $E_x$  field as a function of the grid size. (b)  $S_3$  Stokes parameter.

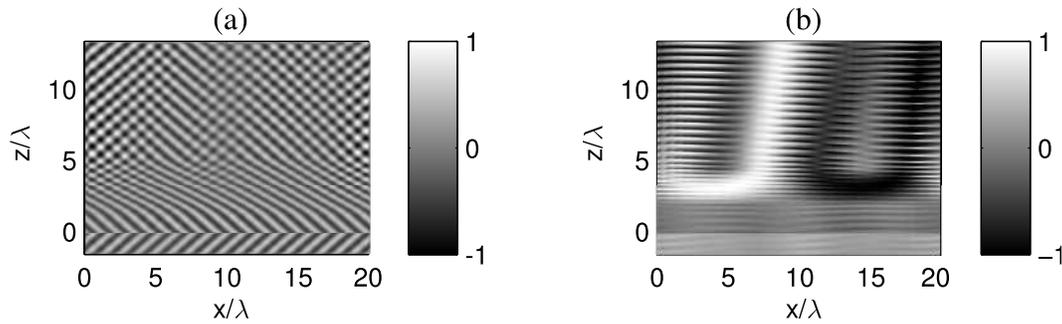


Figure 7. Polarization grating analysis for oblique incidence ( $\theta_0 = 10^\circ$ ): (a) Near-field image of the  $E_x$  field as a function of the grid size. (b)  $S_3$  Stokes parameter.

### 4.3 Computational performance

A set of simulations with both the isotropic and the anisotropic versions have been performed. In order to compare both scenarios, the time simulation costs have been compared for the analysis of a binary dielectric grating for the isotropic version and a phase grating for the anisotropic version. In both cases the time costs of the CPU and GPU versions are compared as a function of the grid size. The variation of the grid size has been established by means of the density parameter  $R$  that has been modified from  $R = 10$  to  $R = 400$ . Fig. 8 shows the SpeedUp, defined as  $T_{CPU}/T_{GPU}$ , for both versions implemented as a function of the grid size in MCells. The number of steps of each simulation has been also modified in order to ensure the steady-state in all cases. For the specific case of GPU computing, the time measurements here shown also include the time that GPU invest in downloading the field components into the host memory. The authors consider that this process is inherent to GPU computing and it also must be considered for accurately measure the degree of improvement of this platform.

As can be seen in Fig. 8a the SpeedUp has a rapidly upward trend for both versions as the simulation size becomes greater. This trend is modified for bigger simulation sizes and remains close to a upper limit. For the isotropic GPU implementation a factor near of 15 is achieved in the best cases, whereas the anisotropic version with GPU computing is more that 22 times faster than the CPU version. Fig. 8b represents the time simulation costs for the isotropic and non isotropic media for both platforms. A dramatic growth of the time costs requirements for the non isotropic version can be easily identified in this graph reaching values of almost an hour for the biggest simulation size here considered. Nevertheless, the GPU time costs for both kind of problems remain under four minutes in the worst case.

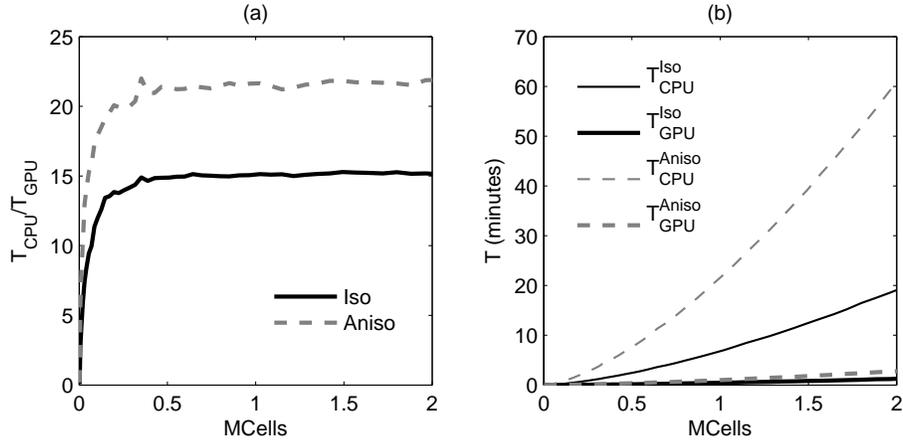


Figure 8. Computational results: (a) SpeedUp of the isotropic and anisotropic versions of SF-FDTD as a function of the grid size. (b) Time simulation costs as a function of the grid size.

## 5. CONCLUSIONS

In this work, the SF-FDTD for the analysis of anisotropic periodic media has been accelerated by means of GPU computing. Firstly, the isotropic version has been implemented in CPU and also in GPU and both have been validated by means of the analysis of the diffraction efficiency of binary phase gratings. This analysis has been carried out in normal incidence and also in the non-paraxial domain achieving a good agreement with those results obtained via RCWA. Secondly, the anisotropic version has been also validated by means of the study of the near-field pattern and the Stokes parameters of a polarization grating achieving also good results and thus validating the implementation here described. The computational results are based on the analysis of the SpeedUp obtained as the difference of the sequential and parallel simulation times and also of the time cost values. This analysis has been performed as a function of the grid simulation size for both cases isotropic and anisotropic media. The degree of improvement of the isotropic GPU version is near of 15 compared with the CPU version, whereas the anisotropic GPU version of the library provides better values of the SpeedUp, being near of 22 times faster than the CPU version. These results joint the values of the time run simulation for anisotropic media demonstrates that considering acceleration techniques for the anisotropic formulation of the SF-FDTD is mandatory for massively computations. The SF-FDTD formalism in its anisotropic form requires considering all the field components, a full tensor formulation of the dielectric permittivity and complex notation. This last factor increases dramatically the number of operations and the complexity of the calculus of the method, degrading the performance in CPU computing. The authors are currently working in optimization techniques based on improving the performance in modern multi-core CPU by means of shared memory approaches. Moreover, the authors are considering to improve the GPU implementation considering tiling techniques and more complex memory strategies, since the memory requirements of the anisotropic SF-FDTD can degrade the performance due to the limiting factor of memory in GPUs.

## ACKNOWLEDGMENTS

This work was supported by the “Ministerio de Economía y Competitividad” of Spain under projects FIS2011-29803-C02-01, FIS2011-29803-C02-02 and by the “Generalitat Valenciana” of Spain under projects PROMETEO/2011/021, ISIC/2012/013 and GV/2012/099.

## REFERENCES

- [1] Taflove, A., H. S. C., [*Computational Electrodynamics: The Finite-Difference Time-Domain Method*], Artech House, Boston (2000 (2nd Edition)).

- [2] Roden, J. A., Gedney, S. D., Kesler, M. P., Maloney, J. G., and Harms, P. H., "Time-domain analysis of periodic structures at oblique incidence: Orthogonal and nonorthogonal fdtd implementations," *IEEE Trans. Microwave Theory Tech.* **46**(4), 420–427 (1998).
- [3] Oh, C. and Scuti, M. J., "Time-domain analysis of periodic anisotropic media at oblique incidence: an efficient fdtd implementation," *Opt. Express* **14**(24), 11870–11884 (2006).
- [4] Miskiewicz, M. N., Bowen, P. T., and Escuti, M. J., "Efficient 3d fdtd analysis of arbitrary birefringent and dichroic media with obliquely incident sources," *Proc. of SPIE* **8255**, 82550W–1–10 (2012).
- [5] Kirk, D. B. and Hwu, W. W., [*Programming Massively Parallel Processors: A Hands-on Approach*], Morgan Kaufmann, Burlington (2010).
- [6] Sanders, J. and Kandrot, E., [*CUDA by example: an introduction to general-purpose GPU programming*], Addison-Wesley, Upper Saddle River (2010).
- [7] Bernaschi, M., Parisi, G., and Parisi, L., "Benchmarking GPU and CPU codes for heisenberg spin glass over-relaxation," *J. Comput. Phys. Commun.* **182**(6), 1265–1271 (2011).
- [8] Ferrando, N., Gosálvez, M., Cerdá, J., Gadea, R., and Sato, K., "Octree-based, GPU implementation of a continuous cellular automaton for the simulation of complex, evolving surfaces," *Computer Physics Communications* **182**(3), 628 – 640 (2011).
- [9] Ford, E. B., "Parallel algorithm for solving Keplers equation on graphics processing units: Application to analysis of Doppler exoplanet searches," *New Astronomy* **4**, 406–412 (2009).
- [10] Harris, C., Haines, K., and Staveley-Smith, L., "GPU accelerated radio astronomy signal convolution," *Experimental Astronomy* **22**, 129–141 (2008).
- [11] Bianchi, S. and Di Leonardo, R., "Real-time optical micro-manipulation using optimized holograms generated on the GPU," *J. Comput. Phys. Commun.* **181**(8), 1444–1448 (2010).
- [12] Gu, X., Pan, H., Liang, Y., Castillo, R., Yang, D., and Choi, D., "Implementation and evaluation of various demons deformable image registration algorithms on a GPU," *Physics in Medicine and Biology* **54**, 207–219 (2009).
- [13] Gross, D., Heil, U., Schulze, R., Schmer, U., and Schwanecke, U., "GPU-based volume reconstruction from very few arbitrarily aligned X-ray images," *SIAM Journal on Scientific Computing* **31**, 4204–4221 (2009).
- [14] Taylor, Z. A., Comas, A., Cheng, M., Passenger, J., Hawkes, D. J., Atkinson, D., and Ourselin, S., "On modelling of anisotropic viscoelasticity for soft tissue simulation: Numerical solution and GPU execution," *Medical Image Analysis* **13**, 234–244 (2009).
- [15] Trapnell, C. and Chatz, M. C., "Optimizing data intensive GPGPU computations for DNA sequence alignment," *Parallel Computing* **35**, 429–440 (2009).
- [16] Yang, C.-T., Huang, C.-L., and Lin, C. F., "Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU clusters," *J. Comput. Phys. Commun.* **182**(1), 266–269 (2011).
- [17] Shahmansouri, A. and Rashidian, B., "Gpu implementation of split-field finite difference time-domain method for drude-lorentz dispersive media," *Progress In Electromagnetics Research* **125**, 55–77 (2012).
- [18] Arfken, G. B. and Weber, H. J., [*Mathematical methods for physicists, 4th ed*], Academic Press, San Diego (1995).
- [19] Francés, J., Neipp, C., Gallego, S., Bleda, S., Márquez, A., Pascual, I., and Beléndez, A., "Comparison of simplified theories in the analysis of the diffraction efficiency in surface-relief gratings," *Proceedings of the SPIE* **8429**, 84291U–84291U–10 (2012).
- [20] Southwell, W. H., "Gradient-index antireflection coatings," *Opt. Lett.* **8**, 584–586 (1983).