A Simple GPU Implementation of FDTD/PBC Algorithm for Analysis of Periodic Structures

Veysel Demir

Department of Electrical Engineering
Northern Illinois University, DeKalb, IL 60115, USA
vdemir@niu.edu

Abstract ─ Constant horizontal wavenumber approach is a simple method to model Periodic Boundary Conditions (PBC) in the Finite-Difference Time-Domain (FDTD) method proposed for efficient analysis of periodic structures; however, it requires execution of the FDTD simulations many times, each time for a different value of horizontal wavenumber to achieve useful results. Therefore, although each simulation may take a short time to complete, a sweep of simulations still takes a long time and there is a need to employ methods to speed-up the simulations. In this contribution we present an implementation of the FDTD/PBC algorithm using the Compute Unified Device Architecture (CUDA) to run the simulations on Graphics Processor Unit (GPU) devices to speed-up the the FDTD/PBC simulations. We also present a method in which a problem space is extended by one padded cell on each of the four periodic sides. As a consequence, programming is simplified, especially for the GPU code for the field update process at the boundaries, the problem space and efficiency of calculations as well is improved.

Index Terms ─ Finite-Difference Time-Domain (FDTD) method, Graphics Processor Unit (GPU), Periodic Boundary Conditions (PBC).

I. INTRODUCTION

Many electromagnetic applications require the use of periodic structures such as Frequency Selective Surfaces (FSS), Electromagnetic Band Gap (EBG) structures, corrugated surfaces, phased antenna arrays, periodic absorbers or negative index materials. These structures extend to several wavelengths in size; therefore, their analyses are time-consuming and memory-extensive using the conventional Finite-Difference Time-Domain (FDTD) [1]-[2] method. To overcome the limitation of the conventional FDTD method, a class of techniques, referred to as Periodic Boundary Conditions (PBC), have been developed. These techniques consider a periodic structure as infinitely periodic and then utilizes the infinite periodicity to analyze only one unit cell of the periodicity instead of the entire structure and obtains the results for the entire infinite size structure.

The PBC algorithms are generally divided into two main categories: field transformation methods and direct field methods [3]. The field transformation methods introduce auxiliary fields to eliminate the need for time-advanced data. The transformed field equations are then discretized and solved using FDTD techniques. The split-field method [4] and multi-spatial grid method [5] are two approaches in this category. The direct field category methods work directly with Maxwell's equations, and hence, there is no need for any field transformation. The sine-cosine method [6] is an example of this category. It should be noted that this method is a single frequency method and does not maintain the wide-band capability of FDTD.

A direct field method, referred to as constant horizontal wavenumber approach, is introduced in [7]-[10]. In this approach, the FDTD simulation is performed by setting a Constant Horizontal Wavenumber (CHW) instead of a specific angle of incidence. With this approach, one can achieve wideband results; however, the results will be valid for a different angle for each frequency. Therefore, the results from a single simulation usually are not meaningful. In order to obtain

Submitted On: May 14, 2014
Accepted On: August 16, 2014

useful results, one can sweep the horizontal wavenumber; i.e., run the simulation for a number of different horizontal wavenumbers, and construct a two-dimensional image of reflection or transmission coefficient distribution on the horizontal wavenumber-frequency plane. Although, running the simulation in a single unit cell makes the problem space much smaller than what it would be, a sweep of simulations takes a long time to achieve useful results. Therefore, still there is a need to employ methods to speed up the simulations. In this context, we consider to speed up the simulations using Graphic Processor Units (GPUs).

In this contribution, we present a GPU implementation of the CHW approach using Compute Unified Device Architecture (CUDA) to improve the computation speed. We also present a method in which a problem space is extended by one padded cell on each of the four periodic sides. Though this treatment increases the size of a problem space, it allows a simpler field update process at the boundaries. As a consequence, programming is simplified, especially for the GPU code, and efficiency of calculations as well is improved.

We present a summary of the CHW algorithm in Section II, and then we discuss the GPU implementation of the algorithm using CUDA in Section III. In Section IV, we illustrate speed up factors achieved using the presented implementation.

II. HORIZONTAL WAVENUMBER METHOD

Consider an infinitely periodic structure with periodicity in the *xy* plane. The periodic structure is illuminated by an obliquely incident plane wave. Figure 1 shows the computational domain of a unit cell of a periodic structure. The unit cell is terminated by PBC boundaries on four sides and Convolutional Perfectly Matched Layer (CPML) [11] absorbing boundaries on top and bottom sides. The incident plane wave is injected into this domain on a source plane.

Figure 2 shows the field components on the *xy* plane-cut of the grid of a unit cell problem space. The size of the problem space is $P_x = N_x \Delta x$ in the *x* direction and $P_y = N_y \Delta y$ in the *y* direction, where N_x and N_y are number of cells, and Δx and Δy are the cell sizes in respective directions. At steady state, a field component on the right boundary of the grid is a time delayed equivalent of a field on the left boundary. For instance, one can write:

$$E_y(x = P_x, y, z, t) = E_y(x = 0, y, z, t - \frac{P_x}{c} \sin\theta_{inc}), \quad (1)$$

where c is the speed of the wave propagating in free space and θ_{inc} is the incident angle. Similarly, a field component on the back boundary of the grid is a time delayed equivalent of a field on the front boundary. When transformed from time-domain to frequency-domain, (1) can be written as:

$$E_y(x = P_x, y, z) = E_y(x = 0, y, z) e^{-jk_x P_x}, \quad (2)$$

which implies that the field component on the right boundary of the grid is a phase delayed equivalent of a field on the left boundary. Here, k_x is *x* component of the wavenumber.

The phase relation between the fields that have a periodic distance of P_x or P_y is utilized to develop an FDTD algorithm that simulates the infinite periodic structure. For instance, in order to calculate an electric field component on the front boundary in Fig. 2, that is indexed as $E_x^{n+1}(i,1,k)$, one needs the value of the magnetic field component below it that could be indexed as $H_z^{n+0.5}(i,0,k)$. While $H_z^{n+0.5}(i,0,k)$ is not in the computational space of the unit cell in consideration, thus, its value is not known; a phase shifted equivalent of it, $H_z^{n+0.5}(i,N_y,k)$, can be used instead following the Floquet theory as $H_z^{n+0.5}(i,0,k) = H_z^{n+0.5}(i,N_y,k) e^{jk_y P_y}$. Then, the electric field updating equation can be written for $E_x^{n+1}(i,1,k)$ as:

$$\begin{aligned} E_x^{n+1}(i,1,k) &= C_{exe}(i,1,k) \times E_x^n(i,1,k) \\ &+ C_{exhz}(i,1,k) \times [H_z^{n+0.5}(i,1,k) - H_z^{n+0.5}(i,N_y,k) e^{jk_y P_y}] \\ &+ C_{exhy}(i,1,k) \times [H_y^{n+0.5}(i,1,k) - H_y^{n+0.5}(i,1,k-1)], \end{aligned} \quad (3)$$

where C_{exe}, C_{exhz}, and C_{exhy} are the updating coefficients. All other fields on the four side boundaries are treated in the same manner and their updates are completed using the phase shifted equivalents of their periodic components.

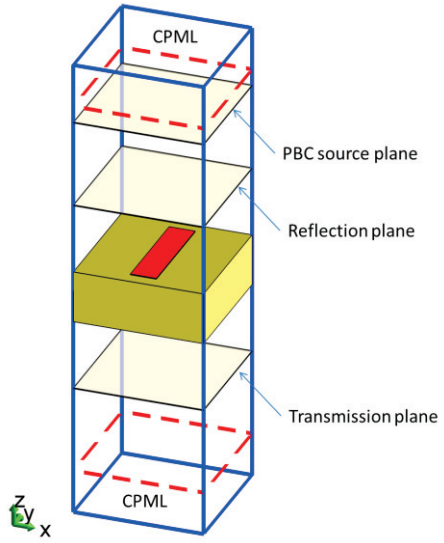


Fig. 1. Problem space of a unit cell of a periodic structure.

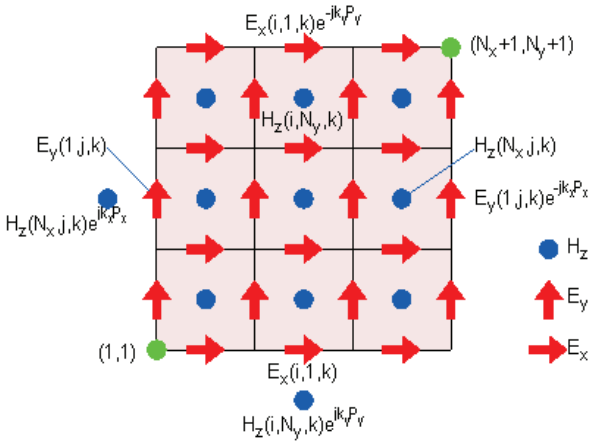


Fig. 2. Field components on the *xy* plane-cut of a unit cell in a three-dimensional domain.

III. IMPLEMENTATION OF PBC

A. General considerations

An existing FDTD code can be modified to accommodate PBC calculations as well. Some PBC specific considerations are discussed in this section.

One main consideration in PBC programming is that, as equation (3) includes the complex exponential phase term, the fields are evaluated in complex time domain rather than real time domain. Therefore, it should be noted that the time domain simulation results are not meaningful, while the frequency domain results such as reflection and transmission coefficients are still valid. Since all fields are complex valued, their respective three dimensional arrays need to be defined as complex data types instead of real data types in the code. To minimize the modification to an existing code, additional three dimensional field arrays with real data type can be used to store the imaginary parts of the complex field values instead of changing the data types of existing arrays. Then field updates can be performed first for the real parts and then for the imaginary parts of the fields in separate subroutines since the updating coefficients are the same as the regular FDTD coefficients.

Other additional steps to the time-marching loop of an existing FDTD program are calculation and addition of incident fields to the fields on the source plane, and capturing the averaged fields after phase corrections.

B. Programming for GPU using CUDA

Recent developments in the design of graphics processing units, introduced a new generation of graphical computation cards which can be programmed to run scientific codes orders of magnitudes faster than Central Processor Units (CPUs). Especially, the introduction of the Compute Unified Device Architecture (CUDA) development environment from NVIDIA made GPU computing much easier and widespread. CUDA has been reported as the programming environment for implementation of FDTD in several articles, which include [12]-[13] as some of the earlier implementations. We also have presented an implementation of an FDTD code in [14]. In this contribution, we discuss the implementation of PBC as an extension to that of [14], therefore we refer the reader to [14] for further details.

As discussed in the previous subsection, one needs to use additional three dimensional arrays for the imaginary part of field values to have the existing code accommodate the PBC calculations. Hence, three dimensional arrays for the imaginary parts of the fields are defined. For a minimum modification to the existing code, the field update functions are executed for a second time, after they are executed to update the real parts of the fields to update the imaginary parts of the fields. For instance, the code section in Listing 3 of [14] is called a second time as shown in Listing 1

below.

```
update_magnetic_fields_on_kernel
<<<grid, threads, shared_memory_size>>>
(nxx, nyy, nx, ny, nz,
Ex_im, Ey_im, Ez_im, Hx_im, Hy_im, Hz_im,
Chxh,Chyh,Chzh,Chxey,
Chxez,Chyez,Chyex,Chzex,Chzey);
```

Listing 1. CUDA code to call kernel function for imaginary part of magnetic field updates.

In the CHW method, calculation of incident field on the source plane requires multiplication of each field at each field point with a phase shift term. Since the phase shift term is a constant value for each point on the source plane, a two dimensional array that carries the phase shift value for each respective field point is constructed before the time-marching loop starts. Then, these arrays are used to adjust the phases of the incident field components during each time step and these incident field components are added to the respective field components on the source plane to excite the problem space. Similar phase shift terms are needed while capturing the fields on reflection and transmission planes. These phase shift terms as well are stored in two dimensional arrays and they are used during the time marching loop to adjust the phases of the fields on the reflection and transmission planes before they are averaged to obtain a single value for reflection and transmission at each time step.

PBC formulation requires the special update of electric field components on the side boundaries, such as shown in (3). For instance, as presented in [10], first, all electric fields in the problem space except for the ones on the side boundaries are updated following the usual updating equations. Then, the electric fields on the left, right, top, and bottom boundaries, except for the corner components, are updated using the phase shifted periodic magnetic field components. Then, as the last step, each of the four corner field components are updated, however, this time using two of the phase shifted periodic magnetic field components for each corner. The steps of this procedure are illustrated in Fig. 3. One can notice that there are many steps as well as exceptions in this flow chart. This kind of granular treatment of fields makes it difficult to write the code for both the CPU and GPU. It may also detriment the efficiency on a GPU since efficiency of computations on GPU mainly relies on the data parallelism of the algorithm.

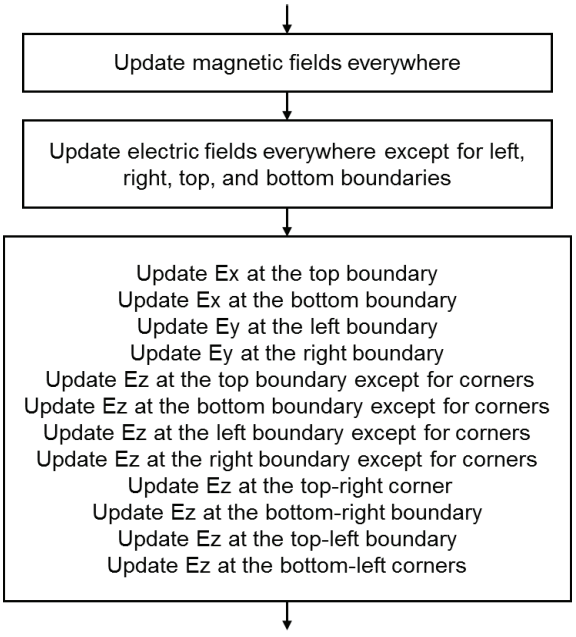


Fig. 3. Conventional field update process for treatment of periodic boundary conditions.

We employed a different approach to avoid the granular treatment of electric field components on the boundaries; we extended the problem space by one padded cell on each of the four sides as illustrated in Fig. 4. Here, the shaded region is the original problem space which is the same as the one in Fig. 2. The extended problem space is terminated by PEC boundaries on four sides; thus, it does not require any special boundary treatment. In each time step, all of the magnetic field components in the extended problem space are updated as usual. Then, the magnetic field components in row 1 are multiplied by the phase shift term $e^{-jk_yP_y}$ and copied to row N_y+1. The magnetic field components in row N_y are multiplied by the phase shift term $e^{jk_yP_y}$ and copied to row 0. Similarly, the magnetic field components in column 1 are multiplied by the phase shift term $e^{-jk_xP_x}$ and copied to column N_x+1. The magnetic field components in column N_x are multiplied by the phase shift term $e^{jk_xP_x}$ and copied to column 0. This operation is a simple product and copy and it

is more efficient to perform on GPU compared with the procedure described in [10]. Then, there is no need for treatment of electric field components on the boundary; all electric field components are updated in the entire problem space using the usual updating equations, where the particular field components on the periodic boundary will find the required phase shifted magnetic fields available in the extended cells. The steps of this procedure are illustrated in Fig. 5. Comparing the flow charts in Figs. 3 and 5, reveals the simplicity of the presented approach.

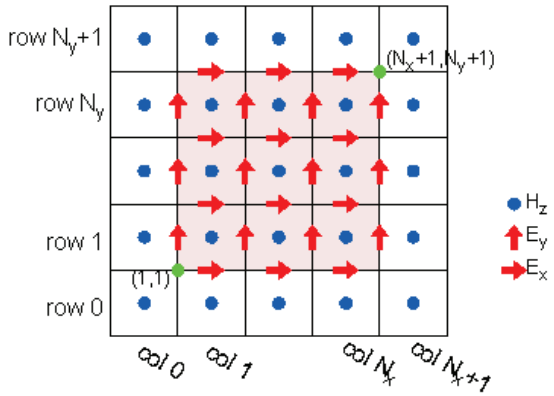


Fig. 4. Field components on the *xy* plane-cut of an extended unit cell.

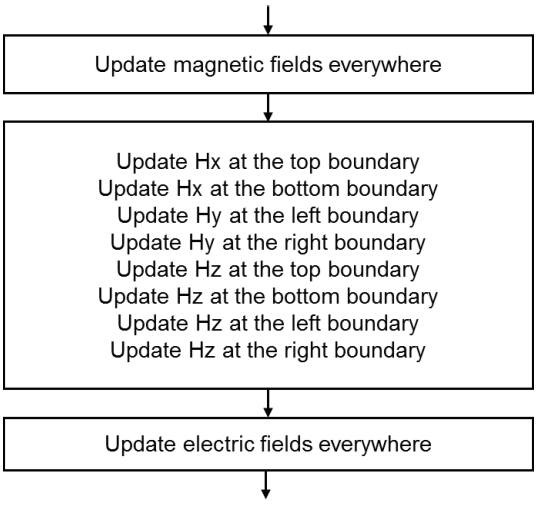


Fig. 5. Proposed field update process for treatment of periodic boundary conditions.

IV. RESULTS

The periodic structure in Fig. 1 is used as a test case to demonstrate the speed up obtained by the GPU implementation. This structure is referred to as a dipole Frequency Selective Surface (FSS) [15]. The metal patch of rectangular shape is placed on a dielectric substrate. The PEC patch has a length of 12 mm and a width of 3 mm. The substrate has a thickness of 6 mm and relative permittivity of 2.2. The periodicity is 15 mm in both x and y directions. The reflection plane is placed 16 mm above the substrate, while the transmission plane is at 3 mm below the substrate. The source plane is 18 mm above the substrate. The simulations are performed using cubic Yee cells of 0.5 mm on a side. The problem space is composed of $32\times32\times82=83{,}968$ cells. The structure is illuminated by incident plane wave of TE mode. Simulations are repeated for a sweep of horizontal wavenumber k_x by varying it from 21 to 100 for 80 distinct values. In all these simulations the horizontal wavenumber k_y is kept as a constant value of 7.8.

The code that runs on CPU is developed using FORTRAN, while the code for GPU is developed using CUDA for C. Simulations are performed on a computer with a CPU of Intel® Core™2 Quad Processor Q9550 at 2.83 GHz, and an NVIDIA GTX480 graphics card. Results are obtained for reflection and transmission coefficients to construct distribution of these coefficients on the horizontal wavenumber-frequency plane. Figure 6 shows the result for magnitude of reflection coefficient; whereas, Fig. 7 shows the result for magnitude of transmission coefficient for a frequency range between 3 GHz to 13 GHz. It should be noted that the results are the same for both the GPU and CPU computations. Simulations are repeated 80 times, each time for a different k_x value. Each simulation is run for 10,000 time steps. Total simulation time is recorded as 59 minutes using the CPU, while it is recorded as 7.8 minutes using the GPU. The GPU/CPU speed-up factor is obtained as 7.5. It should be noted that problem size is rather small in terms of number of cells in this example. As discussed in [16], it is more efficient to solve larger FDTD domains than smaller domains on GPU. In order to demonstrate the performance improvement for a larger problem, the dipole FSS simulations are repeated using cubic Yee cell of 0.25 mm on a side. The problem space is composed of $62\times62\times144=553{,}536$ cells. Total simulation

time is recorded as 490 minutes using the CPU, while it is recorded as 18 minutes using the GPU. The GPU/CPU speed-up factor is obtained as 27.

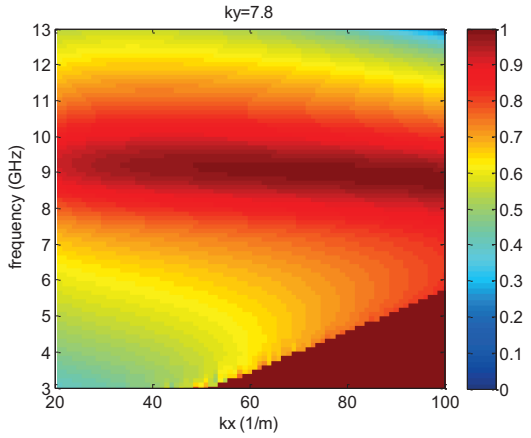


Fig. 6. Magnitude of reflection coefficient in the horizontal wavenumber-frequency plane.

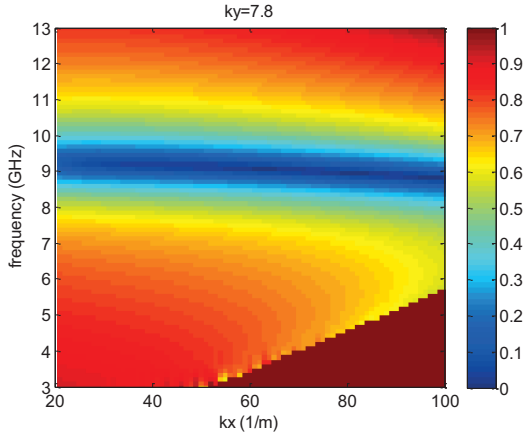


Fig. 7. Magnitude of transmission coefficient in the horizontal wavenumber-frequency plane.

V. CONCLUSION

In this contribution we presented implementation of constant horizontal wavenumber periodic boundary condition as an extension to an existing FDTD implementation using CUDA to speed up the periodic boundary analyses utilizing the computational power of GPU devices. Presented implementation is programmed with the goal of minimum modification to the existing code. It has been shown that results can be achieved in a much shorter time using a GPU card for computations. It should be noted that efficiency can be further improved by an implementation which uses three dimensional arrays of complex data type to store the fields and optimize the kernel functions that run in GPU for these arrays. Moreover, as the simulations run more efficiently on GPU devices when the problem spaces are larger, as illustrated in [16], an algorithm can be developed to stack multiple PBC problem spaces and run them in a single simulation to further speed up the PBC analyses.

REFERENCES

[1] A. Taflove and S. C. Hagness, "Computational electrodynamics: the finite-difference time-domain method," 3rd edition, *Norwood, MA: Artech House*, 2005.

[2] A. Elsherbeni and V. Demir, "The finite difference time domain method for electromagnetics: with MATLAB simulations," *SciTech Publishing*, 2009.

[3] J. Maloney and M. P. Kesler, "Analysis of periodic structures, computational electrodynamics: the finite-difference time-domain method," 3rd ed., *Norwood, MA: Artech House*, 2005.

[4] J. A. Roden, S. D. Gedney, P. Kesler, J. G. Maloney, and P. H. Harms, "Time-domain analysis of periodic structures at oblique incidence: orthogonal and non-orthogonal FDTD implementations," *IEEE Transaction on Antennas and Propagation*, vol. 46, no. 4, pp. 420-427, 1998.

[5] Y. C. A. Kao and R. G. Atkins, "A finite-difference time-domain approach for frequency selective surfaces at oblique incidence," *IEEE Antennas and Propagation Society International Symposium*, vol. 2, pp. 21-26, 1996.

[6] P. Harms, R. Mittra, and W. Ko, "Implementation of the periodic boundary condition in the finite-difference time-domain algorithm for FSS structures," *IEEE Transaction on Antennas and Propagation*, vol. 42, no. 9, pp. 1317-1324, 1994.

[7] F. Yang, J. Chen, R. Qiang, and A. Z. Elsherbeni, "A simple and efficient FDTD/PBC algorithm for scattering analysis of periodic structures," *Radio Science*, vol. 42, RS4004, 2007.

[8] F. Yang, J. Chen, R. Qiang, and A. Z. Elsherbeni, "FDTD analysis of periodic structures at arbitrary incidence angles: a simple and efficient implementation of the periodic boundary conditions," *IEEE Antennas and Propagation Society International Symposium*, pp. 2715-2718, 2006.

[9] F. Yang, A. Z. Elsherbeni, and J. Chen, "A hybrid spectral-FDTD/ARMA method for periodic structure analysis," *IEEE Antennas and*

Propagation Society International Symposium, pp. 3720-3723, 2007.

[10] K. ElMahgoub, F. Yang, and A. Z. Elsherbeni, "Scattering analysis of periodic structures using finite-difference time-domain method," *Morgan and Claypool*, 2012.

[11] Roden and S. Gedney, "Convolution PML (CPML): an efficient FDTD implementation of the CFS-PML for arbitrary media," *Microwave and Optical Technology Letters*, vol. 27, no. 5, pp. 334-339, 2000.

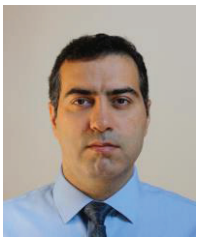
[12] P. Sypek, A. Dziekonski, and M. Mrozowski, "How to render FDTD computations more effective using a graphics accelerator," *IEEE Transactions on Magnetics*, vol. 45, no. 3, pp. 1324-1327, 2009.

[13] N. Takada, T. Shimobaba, N. Masuda, and T. Ito, "High-speed FDTD simulation algorithm for GPU with compute unified device architecture," *IEEE International Symposium on Antennas & Propagation & USNC/URSI National Radio Science Meeting, 2009*, North Charleston, SC, United States, p. 4, 2009.

[14] V. Demir and A. Z. Elsherbeni, "Compute unified device architecture (CUDA) based finite-difference time-domain (FDTD) implementation," *Journal of the Applied Computational Electromagnetics Society (ACES)*, vol. 25, no. 4, pp. 303-314, April 2010.

[15] B. A. Munk, "Frequency selective surface," *New York: Wiley*, 2000.

[16] V. Demir, "A stacking scheme to improve the efficiency of finite-difference time-domain solutions on graphics processing units," *Journal of the Applied Computational Electromagnetics Society (ACES)*, vol. 25, no. 4, pp. 323-330, April 2010.



Veysel Demir is an Assistant Professor with the Department of Electrical Engineering at Northern Illinois University. His research interests include numerical analysis techniques as well as microwave and Radio Frequency (RF) circuit analysis and design. He is a member of IEEE, Sigma Xi, and Applied Computational Electromagnetics Society (ACES).