FDTD MICROWAVES SIMULATOR USING ADVANCE COMPUTER GRAPHIC TECHNOLOGIES

Nicolae CRISAN
Technical University of Cluj-Napoca, 26-28 Baritiu Street, 400027, Cluj-Napoca, Romania
E-mail:Nicolae.Crisan@com.utcluj.ro
Phone:40-64-401241

Abstract: The paper presents several technical issues for finding and rendering solutions for Maxwell's differential equations. The finite difference time domain (FDTD) method used in the paper is the most wanted numerical analysis for microwaves rendering in the time domain. Nowadays computational electrodynamics (electromagnetics) comes with more advanced and agile techniques used in computer games. Frameworks as DirectX or OpenGL accelerate hardware using graphical processor units (GPUs) and enabling dynamic rendering in real time. These frameworks could be used to speed up the 3D rendering according to the FDTD demands in terms of the framerate, and, bring some new capabilities as rotation, translation, and scaling from computer graphic to computational electrodynamics. The presented software is intended to replace some obsolete graphic technologies or software (like Mefisto-2D or the graphic simulations based on Matlab) which are in use today in universities, for understanding the main properties of the microwaves with new and agiler rendering techniques that are coming from computer graphics world.

Keywords: *Computer-aided, FDTD, DirectX, OpenGL, Maxwell's equations, electromagnetic modeling*

I. INTRODUCTION

A modern simulator is using the most advanced technologies for rendering dynamic images (frame by frame) with the help of GPUs (Graphic Processor Unit). A dynamic process is calling, in fact, in a continuous loop, the solver to perform calculations and brings the representation of the one frame after another.

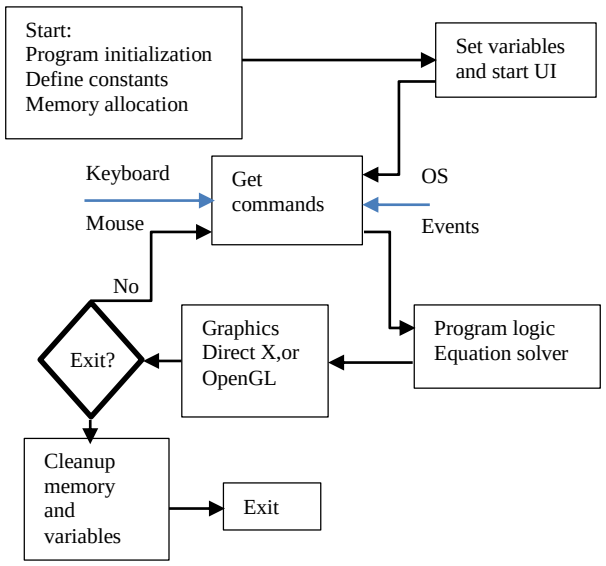


Figure 1. Schematic of the program framework

A user interface (UI) stands between the user actions (commands) and input data in the rendering process. So the main phases of the program framework are shown in figure 1. Behind the scene stands the MFC (Microsoft Foundation Classes) for passing the events to the operating system (OS) and back to the user interface (UI). This allows the user to interact with the program and exchange commands and parameters. The rendering pipeline manipulates the graphic hardware by throwing data in the rapid exchange between GPUs and CPU (central processing unit). These details are invisible for the programmer who can pay more attention to data processing directly in the system memory (RAM). This concept enables the possibility to work more efficient by focusing more on the main problems, not over every detail behind the scene. In the paper, the DirectX version 11 has been used. In this case, the DXGI (Direct X Graphics Infrastructure) is dealing with the exchange between hardware and Direct3D.

This approach offers the possibility to render dynamic processes at a frame rate up to 100 frames/second. The data exchange between CPU and GPU doesn't overload the CPU and allowing it to deal with the most intensive process, the electromagnetic solver. This technique enables rendering during the simulation in the time domain.

The proposed simulator brings the graphics technologies from computer games to the Maxwell solver to replace old simulators used in microwaves laboratory for understanding the main properties of the electromagnetic waves. In this way, the students could come closer to the

Manuscript received March 2, 2017; revised March 10, 2017

electromagnetic phenomena by contributing directly to the solver engine and to the three-dimensional rendering.

The paper is structured as follows. Section II shortly describes the editor interface. Section III is about Maxwell's equations in numerical form and their C++ implementations and the boundary conditions. Section IV describes the use of the DirectX to represent the results on the PC screen. Section V comes with 3D rendering and with the special effects of the boundary conditions over the field's envelope and propagation modes. Finally, section VI concludes the paper.

II. THE EDITOR

The purpose of the program's editor is to make the link between the input data and the solver entries. All boundary conditions are translated from the graphics lines, dots or rectangles into the mathematical parameters (coordinates and geometric properties - vertices). All lines represent here walls and come with colors: electric wall (red line) magnetic (blue line) or reflection walls (green line). Dots represented by circles stand for punctiform sources and rectangles for the propagation medium or for the animation region (figure 3). The color selects which boundary conditions are applied, and the coordinates of the lines establish where the field bounds to the structure. These conditions are required to confine to the size of the propagation medium and to find the solutions. All lines, dots or rectangles are seen in the program as elements in a sequence container known as a simple linked list. The list allows the insertion, the deletion or the modification of any element that has been selected or drawn previously in the editor. The editor itself prepares the list for further use keeping elements and eventually sorting the elements after some specific criteria before rendering (figure 2).

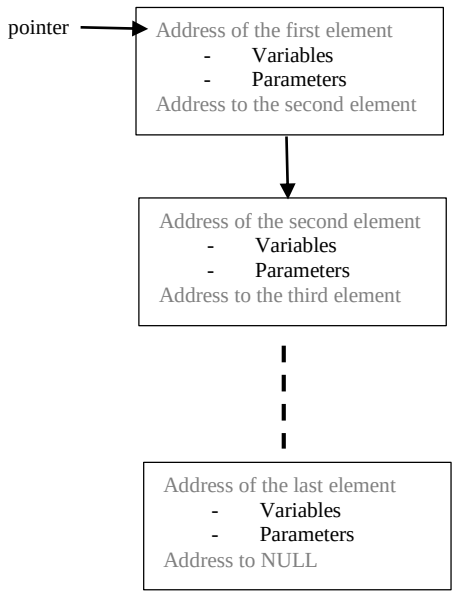


Figure 2. Elements' sequence in the list

Each element comes with its data structure and a pointer to the next element in the list. The work with pointers and lists is an efficient way to manipulates and prepare the elements during the editing of the microwave structure. This could be rectangular or square only. In order to write Maxwell's equations in the numeric form, a wire-frame grid approximates the entire structure. The size of the grid is set by the user at the beginning of the design, and the cell size is adjusted automatically to the size of the active window. The application is an MDI type (Multi-Document Graphic Interface) that allows the visualization of many types of windows in a common window framework. This is the interface that comes with the most advanced editors like the MSOffice. Two types of windows are allowed here by the application: the first type is for editing (known as the child window or the editor) and the second is for 3D rendering with DirectX (the render). All programs that work with MFC (Microsoft Foundation Classes) framework are following the same concept. The editor is written in the class EMSView derived from CScrollView class that allows scrolling up and down or from the left to the right of the view. The MFC executable is faster than any interpreted code or compiled code into any intermediate language. The code speed is here the key in making the solver faster, to keep up the solver speed with the demanded frame rate of the screen. For a realistic motion, the frame rate must be at least 30 frames/second to avoid screen freezing or frames dropping. This condition makes the solver execution time critical [1][2].

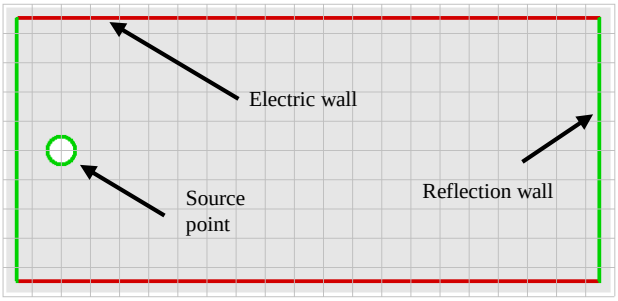


Figure 3 Editing of the boundary conditions behind of each element with lines and colors [1]

III. THE SOLVER

The solver starts from the time-dependent Maxwell's curl equations (eq. 1 & eq. 2) and uses the list of elements from figure 3 to set the boundary conditions and parameters. The simplest case is the one-dimension representation of the wave propagation.

$$\frac{\delta \bar{E}}{\delta t} = \frac{1}{\varepsilon_0} \nabla x \bar{H} \quad (1)$$

$$\frac{\delta \bar{H}}{\delta t} = -\frac{1}{\mu_0} \nabla x \bar{E} \quad (2)$$

In this form, equation 1 and equation 2 stand for a plane wave in free space. In order to simplify the understanding of the problem, one can consider only the one dimension case. The wave is then propagating only along the z-axes and equations 1&2 become more simple.

$$\frac{\delta E_x}{\delta t} = -\frac{1}{\varepsilon_0} \frac{\delta H_y}{\delta z} \quad (3)$$

$$\frac{\delta H_y}{\delta t} = -\frac{1}{\mu_0}\frac{\delta E_x}{\delta z} \quad (4)$$

The electric field is oriented in the x-direction and magnetic field in y-direction according to the equations 3 and 4. Both derivatives can be expressed in the numeric form using finite differences in the time domain [3]. These are approximations that consider finite intervals $(\Delta t, \Delta x)$ for time and space representation. Both differentials equalities stand if and only if the interval Δx is at least ten times smaller than a wavelength.

$$\frac{E_x^{n+\frac{1}{2}}(k) - E_x^{n-\frac{1}{2}}(k)}{\Delta t} = -\frac{1}{\varepsilon_0}\frac{H_x^n\left(k+\frac{1}{2}\right) - H_x^n\left(k-\frac{1}{2}\right)}{\Delta z} \quad (5)$$

$$\frac{H_y^{n+1}\left(k+\frac{1}{2}\right) - H_y^n\left(k+\frac{1}{2}\right)}{\Delta t} = -\frac{1}{\mu_0}\frac{E_x^{n+\frac{1}{2}}(k+1) - E_x^{n+\frac{1}{2}}(k)}{\Delta z} \quad (6)$$

Index k stands for the distance and index n for the time according to $z = k\Delta z$ and $t = n\Delta t$. The equations 5 and 6 are taking the so called the central difference approximation form which is taking into account the interleaving between the E and H fields. This interleaving are in space and time simultaneously. The final form of the fields is:

$$E_x^{n+\frac{1}{2}}(k) = E_x^{n-\frac{1}{2}}(k) - \frac{\Delta t}{\varepsilon_0 \Delta z}\left[H_x^n\left(k+\frac{1}{2}\right) - H_x^n\left(k-\frac{1}{2}\right)\right] \quad (7).$$

In equation 7 the current E field can be calculated from the previous E field and from the difference between the current H field along z-axes. From the *Courant* condition [3], the time step interval is given by:

$\Delta t = \frac{\Delta z}{2c_0}$, where $c_0 = 3 * 10^8 m/s$ is the speed of light in vacuum. Making the substitution $\tilde{E} = \sqrt{\frac{\varepsilon_0}{\mu_0}}E$ we can simplify the equation 7, because $\frac{1}{\sqrt{\mu_0 \varepsilon_0}}\frac{\Delta t}{\Delta z} = 0.5$.

Table 1. Runtime in microseconds for 19x111 nodes

Solver	**Runtime [μsec]**	**Iterations**
Simple *For Loop*	21	2109
Two *For Loops*	43	4218
Two Loops + Boundary conditions	49	4478
Four Loops + Boundary conditions	91	8696
Four Loops + Boundary conditions + Color Palette + Video ram buffer	932	-

This change of the variable allows us to write the equation 7 in C code:

```
for (k = 1; k < nrofsegments; k++)
    ex[k] = ex[k] + 0.5 * (hy[k - 1] - hy[k]);
```

Here the number of segments must be higher or at least equal to the number of wavelengths times ten. If we intend to visualize at least three wavelengths, then we need at least 30 segments. Even more, an over-segmentation for a fine resolution must be taken into consideration. Here twenty segments for each wavelength brings the finest resolution [3]. Having in mind the critical framerate, we should have at least 30 frames/second for a good animation. So we get the number of the iterations of roughly 60*30 per second. This isn't too much, but when the problem goes two-dimensional than, for a wire-grid, we could have 10*60*30 iterations per second. Therefore the solver must be faster when it goes to a higher spatial dimension.

In table 1 are shown the average values of the runtime for the main routines of the solver. The measurements are made progressively from the simplest solver to the more complex one. Anyway, the overall runtime for the solver drops under one millisecond. Normally this is too less for an ordinary time counter (written in C) to count. If we program a simple timer, the result will always be under the resolution of the clock counter. The code execution is much faster, so to solve the problem it has been used a special procedure of measuring based on the source code in C++ imagine by Shade Gavin. The idea is to redesign the timer and to increment the counter every microsecond instead of one millisecond. This means that the counter is incremented one million times for each second. A normal unsigned integer is overloaded after one hour of measurements. A special variable is defined to cope with the overloading problem by initiating the counter with a new type of variable that resembles 64 bits instead of 32. This allows hours of measurements to search for the average runtime value which fluctuates after every measurement.

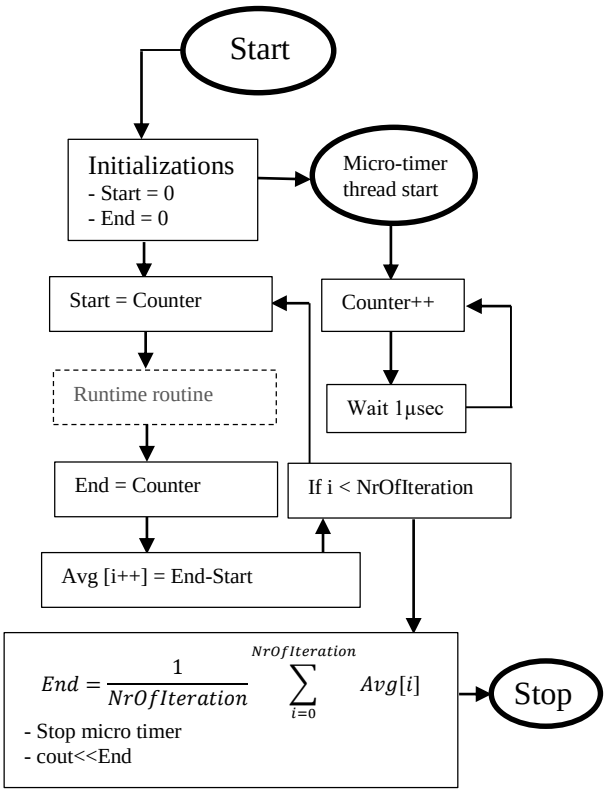


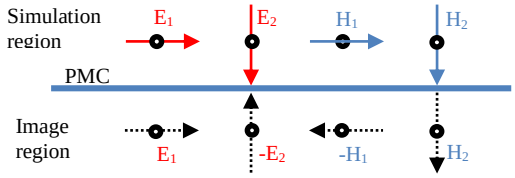
Figure 4. Logic diagram of the runtime measurement approach using the micro-timer method

In table 1 the solver runtime is lower than one millisecond. An active timer that normally increments the counter every other millisecond it isn't appropriate for measuring the

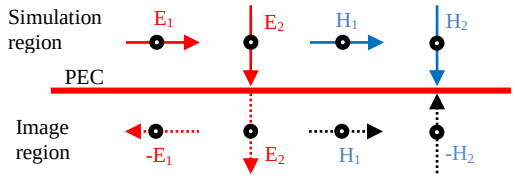
efficiency of the solver routines. Usually, with this solver the runtime is under one millisecond even for the most complex approach, and one can maintain up to 100 frames per second easily.

IV. THE BOUNDARY CONDITIONS

One of the most important operations inside the solver is related to the boundary conditions (walls). Maxwell's equations are in fact differential equations with an infinite number of solutions. These are limited by the number of nodes. In finding these values for each node, near the boundary of the grid, it is necessary to set some special conditions at the limit of it, along with a closed curve line, that is bounding the structure. One could have three types of the boundary conditions: electric wall (PEC), magnetic wall (PMC) or Perfectly Match Layer (PML) [4]. The PML boundary condition is more complex in its schematic detail for two or three dimensions, and its presentation here is beyond the scope of the paper. But the first two conditions: PEC and PMC that have been used in figure 6, are depicted in figure 5.a and b in every aspect.



a. Fields near Perfect Magnetic Conductor wall (PMC / Magnetic Wall)



b. Fields near Perfect Electric Conductor wall (PEC / Electric Wall)

Figure 5. PMC&PEC Boundary Conditions

As one can see in figure 5.a the perpendicular E vector to the PMC plane is canceled by the image of the vector field as long as the magnetic H vector will be in phase with its image. The parallel E vector with the PMC wall is in phase with its parallel image vector. Finally, the H parallel vector is canceled by its image along the PMC wall. In contrast to the magnetic wall an electric wall changes the behavior of the fields as one can remark from figure 5.b. As a general rule, the E vector will be attenuated when goes parallel with an electric wall. This is normal as long the electric field it breaks down near a perfect conductor which exhibits a large conductivity due to the induced currents.

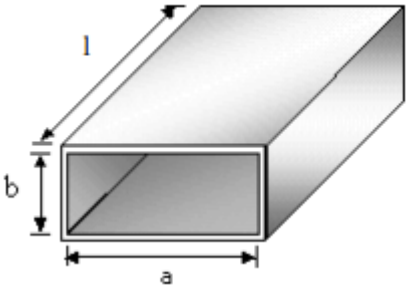
V. PLOTTING THE RESULTS

The experiment here envisages the effect of the boundary conditions over the transmission line. If the left and right lines represent magnetic walls (blue), then the wave will have a plane front (TEM – Transversal Electromagnetic front). The front of a wave is, in fact, the locus of points having equal phases. Generally, the TEM transmission mode enables a plane front during propagation. When both left and right walls are electric (PEC), and sidewalls are magnetic then the transmission line enables the TE or the TM propagation mode, and the wavefront is then curved. As long the TEM mode characterized the propagation in free space or coaxial lines, then, the second (TE or TM mode), is characterizing the propagation along the waveguides. In figure 6 b is the electric fields along y-axes inside the WR-90 waveguide. This could be used in X-band between 8.2-12.4 GHz.

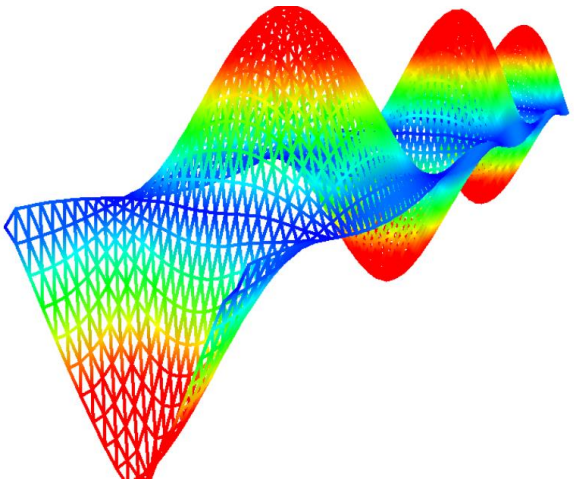
Table 2. WR-90 dimensions

Frq. [GHz]	a [mm]	b [mm]	l [mm]
8.2-12.4	22.86	10.16	228.6

The plot shows the variation of the field inside the waveguide subject to the displacement along x (horizontal), y (vertical) and z-axes. Electric field approaches zero V/m near the side walls where it goes parallel with the PEC. In the middle of the waveguide, the magnitude of the electric field has a maximum value as long the field goes perpendicular to the up and down PEC walls. The waveguide is open terminated, and the wave sees magnetic walls at both ends of the line. This explains why the field ends with a maximum amplitude here (figure 6.b).



a. WR-90 rectangular waveguide (see Table 2 for a, b and l parameters value) for X-band (8.2-12.4 GHz)



b. Electric field along y-axes inside the WR-90 waveguide in TE_{10} mode

Figure 6. The PEC boundary conditions for left&right walls and the PMC conditions for both ends of the line – TE_{10} – propagation mode inside the WR90 waveguide.

VI. CONCLUSIONS

The paper presents how the computer graphics technologies that are coming from computer games and multimedia applications can accelerate the simulations used in electrodynamics. The main idea is to decrease the processor payload when it deals with the electromagnetic solver, using DirectX technologies. This approach drops the solver runtime below one millisecond and paves the way for the frame rate increasing up to 100 frames/sec. Although, Maxwell's equations are not solved using CUDA (Compute Unified Device Architecture) technologies on the GPU, but on the CPU in a classical way. Only the graphic approach is based on the DirectX technique, not the solver itself. The runtime results are shown in table 1 (for the solver only) and a frame for a given moment in the time domain is depicted in figure 6 b. The methods based on CUDA architecture developed by NVIDIA can accelerate more the simulation if the solver is parallelized on GPUs. Nevertheless, a non-CUDA based approach is running on every graphical card. Therefore, the CUDA approach demands compatible cards only made by NVIDIA.

REFERENCES

[1] M. F. Vaida, P. G. Pop, C. Strilețchi, L. D. Chiorean, L. Alboaie, "*Programarea în limbajul C/C++ Algoritmi de bază în C/C++*", Ed. Casa cărții de știință, ISBN 978-606-17-0004-2, 2011

[2] E. Lengyel, "*Mathematics for 3D Programming and Computer Graphics*", Charles River Media Inc. Hingham, Massachusetts Second Edition, ISBN 1-58450-277-0, 2004

[3] X. Wang, S. Liu, X. Li, "*GPU – Accelerated Finite-Difference Time-Domain Method for Dielectric Media Based on CUDA*", International Journal of RF and Microwave Computer-Aided Engineering, doi: 10.1002/mmce.20997, Wiley Periodicals Inc., 2016

[4] G. Mur, "*Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations*", IEEE Trans Electromag Compatibility, 1981

[5] A. Pekmezci, E. Topuz, L. Sevgi, "*Finite Difference Time Domain Formulation for Epsilon-Negative Medium Using Wave Equation*", International Journal of RF and Microwave Computer-Aided Engineering, doi: 10/1002/mmce.20965, Wiley Periodicals Inc., 2016