FYP I Progress Report

|  |
| --- |
| Modeling of Electromagnetic Wave Propagation in NIM using FDTD Technique |

|  |
| --- |
| Attique Dawood |
|  |

Project Supervisor:

|  |
| --- |
| Hina Ashraf |
|  |

Project Co-supervisor:

Project Group Members:

|  |  |  |
| --- | --- | --- |
| Junaid Nasir |  | I10-0361 |

Semester:

|  |
| --- |
| Fall 2013 |
|  |

Project Objective:

|  |
| --- |
| Project Objective is to simulate and observe the behavior of Electromagnetic waves in Negative Index Material (NIM) at specific frequency of 3GHz. For this Finite Difference Time Domain (FDTD) algorithm will be used which is a numerical analysis technique, it is fairly accurate and works with wide range of frequency band.  FDTD algorithm only works for positive index material so it has to be modified to work with negative index materials accurately and should produce correct results.  Project will also cover the performance comparison across MATlab, C++ and GPU. |

Introduction and Background:

|  |
| --- |
| A numerical analysis technique, Finite Difference Time Domain (FDTD) used for modeling conventional electromagnetic problems can be implemented on Graphical processing Unit (GPU) to accelerate computations to manifolds as compared to central processing unit (CPU).In this project, our aim is to incorporate the Drude dispersive Model into Maxwell’s equations of FDTD algorithm to characterise and model an NIM problem and implement it on GPU for accelerated results.  Finite-difference time-domain (FDTD) is one of the most efficient techniques of differential time domain numerical modeling methods. Since it is a time-domain method, FDTD solutions can cover a wide frequency range with a single simulation run, and treat nonlinear material properties in a natural way. Furthermore, it is readily parallelizable on hardware like GPU. It solves the Maxwell’s differentia equations in leap-frog manner. At a given time instant, first electric field and then magnetic field is solved. These two steps are repeated until the solution is obtained.  Negative Index Material (NIM) or Left Handed Material (LHM) is a material which has negative permeability and permittivity. It involves the natural phenomenon of optics that is refraction. In refraction, when a light enters from one medium to another with different refractive indices, it alters its path depending upon the difference between the refractive indices. For all the naturally existed materials the value of refractive index is always positive thus NIM is artificially designed and fabricated engineered structure for having a negative refractive index that leads to surprisingly new results for different laws of physics.  To study the science and unusual behavior of one dimensional electromagnetic wave in Negative index Materials, FDTD technique will be used. |

Approved FYP I Plan:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Task** | **Duration** | **Status** |
| 1 | MATlab simulation of 1D EM wave | 6 weeks | Complete |
| 2 | Frequency Domain Analysis | 4 weeks | Complete |
| 3 | Implementation on C++ | 4 weeks | Complete |
| 4 | File Handling | 3 weeks | Complete |

|  |
| --- |
| **TASK 1: MATlab implementation of FDTD for 1D EM wave**  The finite differences time domain method approximates both the spatial and temporal Derivatives that appear in Maxwell’s equations, The FDTD algorithm as first proposed by Kane Yee in 1966 employs second-order central differences.  Summary of Algorithm is as follow:  1. Derivatives in Ampere’s and Faraday’s law is replaced by finite differences. Both space and time is discretized so that electric and magnetic fields are staggered in space and time.  2. Compute the new equations to obtain “update equations”, which gives the future values of fields using past values of fields  3. Magnetic field is evaluated one time step at a time to obtain the future value for magnetic field.  4. Electrical field is evaluated one time step at a time to obtain the future value of electric field.  5. step 3 and 4 is repeated until the fields have been obtained over the desired duration  Update equations obtained are  Work completed: FDTD method is successfully implemented for both free space and medium with double permittivity with respect to free space  Problems:  -Gaussian and Sinusoidal waves implementation  -Boundary Conditions  -unstable behavior at medium boundary  -Absorbing boundary conditions  -Additive sources  Conclusion:  All the above mentioned problems were solved and following results were obtained  C:\Users\Hina\Desktop\New folder\1.PNG  Figure Free space simulation  C:\Users\Hina\Desktop\New folder\2.PNG  Figure Simulation with dense medium slab |
| **TASK 2:** **Frequency Domain Analysis**  Description:  Frequency domain analysis compares the theoretical values with that of simulated values. In Frequency domain analysis spectrum of transmitted and Incident waves are plotted. Refractive index is also calculated which should be equal to calculated refractive index given by  Work completed:  Frequency domain analysis was implemented in MATlab, and was compared with actual theoretical values.  Problems:  -Implementation of Fast Fourier Transform  -Extraction of Incident and reflected waves values from simulation  Conclusion:  Following results were obtained from Frequency domain analysis of simulation results which are accurate when compared with calculated values for the given system.  C:\Users\Hina\Desktop\New folder\3.PNGC:\Users\Hina\Desktop\New folder\4.PNG  Refractive Index value is equal to 1.414 at 3 GHz which is equal to that of theoretical value.  C:\Users\Hina\Desktop\New folder\5.PNGC:\Users\Hina\Desktop\New folder\6.PNG |
| **TASK 3: Implementation in C++**  Description:  One of the objectives of project is to compare the computational power and performance factor between MATlab, C++ and GPU. GPU is also programmed using C++ hence the entire code is first converted into C++. However it will be only used to compute the main FDTD algorithm all the plotting of graphs and additional computation will be done in MATlab, because MATlab handle these things more efficiently. Implementing Graphs or Fourier functions in c++ is out of scope for this project.  Work Completed:  FDTD algorithm is implemented into C++ using Visual Studio 2010. Results are accurate when compared to MATlab results of algorithm proving the correct implementation of algorithm.  Problems:  -Data type precision  -Dynamic Memory Allocation  -Array index difference between MATlab and C++  -conversion of MATlab functions  Conclusion:  MATlab handles the precision of variables by itself but in C++ we have to choose and declare every variable individually .Arrays in C++ is allocated using dynamic memory allocation because of different time simulation for Gaussian and Sinusoidal waves, dynamic memory should also be de-allocated at end of program to free up the memory used. Every equation implemented in MATlab had to be changed because of different array index in C++ and functions. |
| **TASK 4: File Handling**  Description:  Visual Studio 2010 natively doesn’t support graphics processing, hence it is required to export all the data of simulation to MATlab for plotting and comparing. The best solution is to write the data into files. For each variable (array) a file should be created individually so that it can be kept separately. There are 1000 instances of two main arrays (Electrical field and magnetic field component) in case of Gaussian pulse and 5000 instances in case of sinusoidal, each array contains 1000 entries at each point of space present in simulated field. There are ten more arrays carrying data required as mentioned above  Work Completed:  File handling is efficiently implemented in C++ for writing the data and in MATlab for reading the data. Total size of 2009 data files is approx. 15.6 MB.  Problems:  -Saving 2000 different files with same array (updated)  -Saving files in a loop  -Reading files in MATlab  -Organizing files into folder  Conclusion:  Data is written into binary format because of fixed size of data, fstream is used to write into files. Stringstream is handling the names of files and organize them with respect to simulation loop number. All the files are stored into results folder, results folder is first created by windows.h library’s function if it doesn’t already exists.  File handling is also dynamic hence we can change system parameters as well as simulation time and all the changes into variables will also be exported to matlab for analysis automatically  In MATlab fullfile() function gets the current path using pwd variable present in MATlab. Fopen along with strcat is used to open the files with different names into loop. At the end results are plotted using plot functions |

**FYP I Progress Report Assessment:**

|  |  |  |
| --- | --- | --- |
| Marks should be awarded on a scale of 0 to 20  *A+ A B C D F*  20 19 18 17 16 15 14 13 12 11 10 9 8 – 0 | | |
|  | | |
| Progress Report Assessment | Weight | Marks |
| a. Project Objective | 10% |  |
| b. Introduction and Background | 10% |  |
| c. Description of work completed/in progress | 60% |  |
| d. Quality of Progress Report | 10% |  |
| e. Fonts, Type, Margins, Spacing, Figures etc. | 10% |  |
| Total (0.1 x a + 0.1 x b + 0.6 x c + 0.1 x d + 0.1 x d)/20 |  |  |
| Total Marks (Total x 10) 10 |  |  |

**Supervisor’s Comments:**

|  |
| --- |
|  |

Supervisor’s Name: \_Attique Dawood\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ Supervisor’s Signature Date