STUDY ON TRANSMISSION PROPERTIES OF FIBER BRAGG GRATING WITH GPU SUPPORT

MARCH 2017

ANGGER ABDUL RAZAK

Abstract

This research aims to analyze one of the important components in the optical telecommunication which called as Fiber Bragg Grating (FBG). FBG is a fiber core with many gratings on its core. With these gratings, it reflects some of the specific wavelengths that suit to Bragg condition. In optical telecommunication, FBG acts as a band reject filter to flatten the output gain of erbium-doped fiber amplifier (EDFA). It also could be used as dispersion compensator to reduce the noise in the data after traveling a long trip in the fiber optic systems.

Another function of FBG is that it can be used in the sensor region called as optical fiber sensor (OFS). OFS could be used is to measure temperature, strain, humidity, vibration and seismic activity, hazardous gas concentration, index refractive, and so on.

With the important function of FBG, we try to analyze it numerically both in two-dimensional and three-dimensional approach and evaluate its output characteristic by using FDTD method. This program aimed to be able to imitate characteristic of FBG based on the design parameters. Furthermore, this FDTD program is also targeted to be able to analyze the future structure of FBG that are not proposed or exist yet.

As result, FBG structures with different parameters successfully simulated by using FDTD Method. Results show that higher grating refractive index value affects in a smaller value of transmission in a specific wavelength, and the center of wavelength drop is shifted to a higher value. Increasing grating period will affect to the center of wavelength drop shifted to higher wavelength value, with a linear correlation between them. Longer grating length will reduce the transmission value with a linear correlation between them. Chirped FBG (CFBG) show a weaker reflectance, but with wider broadband wavelength reflected due to more grating period variations contained in FBG. Higher temperature will result in wavelength shift to the higher wavelength value, with a linear correlation between them.

For three-dimensional analysis, general CPU calculation is too slow to simulate FDTD program. Thus, GPU calculation is proposed to get faster simulation time. In terms of memory usage, CPU calculation only uses PC RAM with the similar amount used both single and multi-thread calculation, while GPU calculation uses PC RAM and GPU memory with higher usage on both of them. Comparison of CPU and GPU results show a very similar characteristic between them, with median error only around 0.007%. Furthermore, GPU calculation could reach more than 100 times compared to single thread calculation in 1st PC system. This means that GPU calculation could substitute CPU calculation based on result's similarity with much faster execution time. The performance gain of GPU calculation is affected by several factors, mainly by the specification of GPU and others PC systems components.

Results from FBG parameter changes in three-dimensional calculation have a similar property compared to two-dimensional case. Even the center of reflected wavelength is a bit shifted and the reflection power is different, both systems have a similar property when the parameter of FBG changes, and reflect real FBG characteristics.

If FBG structure could be constructed by using two-dimensional approach, there is no need to use three-dimensional calculations to get its output characteristic. When two-dimensional approach can't imitate the proposed structures, the three-dimensional calculation could be promoted to simulate these structures.

Furthermore, even the results reported here are only based on current FBG structures, FDTD program already prepared to simulate FBG structure that not proposed or exist yet. As long as the structures could be modeled by mathematics equation, even when every grating structures are different each other, FDTD program could be used to simulate and evaluate its output characteristics.

Table of Content

List of Figures

List of Tables

I. Introduction

1.1 BACKGROUND

Optical fiber has made a big revolution in the landlines telecommunication systems. With the existence of this technology, we are able to communicate between two long distances in a high-speed connection. The high-speed internet also one of the services that arise from the use of optical telecommunication systems. In the optical systems, like any other long distance data transmission systems, information will need to be amplified in the process of transmission.

One of the widely used amplifiers for optical fiber is erbium doped fiber amplifier (EDFA). EDFA have a very high gain for amplifying the optical signals, but it has non-flat gain in its characteristic. With a non-flat gain power, data that transmitted through the system will be unbalanced in term of the power, and it will be difficult to read the data for the whole bandwidth used. To flatten this gain, Fiber Bragg Grating (FBG) is needed [1]. It is working by lowering some wavelength that is amplified too high after data are passing the EDFA and make the final amplified output are similar and flat in term of power. In another word, FBG acts as a band reject filter here. In optical communication, the function of FBG are not limited to a band reject filter, it could be used as dispersion compensator also to reduce the noise in the data after traveling a long trip in the fiber optic systems [2] [3].

Other than in optical telecommunication system, FBG could be used also in a sensor region, called as optical fiber sensor (OFS) [4]. The most widely used functions of the OFS are to measure temperature and strain. Temperature and strain sensor could be implemented in a single-mode optical fiber [5] as well as in single-mode-multimode-single-mode structures [6]. Implementation of hybrid FBG also proposed to differentiate between strain and temperature sensing by using long period FBG [7]. For even further application, processing and using FBG for harsh environment also proposed [8] [9], as well as very low temperature down to -80°C by using organic polymer package [10] and very high temperature up to 800°C by using type-II FBG [11] and 1500 °C by using sapphire FBG [12] also possible. For remote sensing application, a long-distance strain and temperature sensor for more than 50km could be obtained by using a fiber Raman laser with etched FBG [13]. To measure strain sensor only, compensating temperature shift could be done by using FBG-based Fabry-Perot (FBG-FP) as a reference [14], and semi-distributed strain measurement could be obtained by incorporating both wavelength division multiplexing (WDM) and time division multiplexing (TDM) [15]. Other than strain sensor that usually coupled with temperature sensor, other sensor function also possible to be combined with a temperature sensor such as humidity sensor [16]. While for the application for humidity sensor without coupled with temperature sensor could be applied by using tapered optical fiber [17] or tilted FBG [18]. Durability and performance of FBG based humidity sensor already conducted to measure the accuracy in an aggressive gaseous environment with very good results [19].

In addition to these, there are many other functions of measurement of the OFS other than sensing temperature, strain and humidity; such as measuring vibration and seismic activity [20], hazardous gas concentration [21], surrounding refractive index (SRI) [22] [23] [24], hydraulic pressure [25], composite structure damage [26], torsion [27], and so on. Characteristic of FBG as a band reject filter and as a sensor are depends on its design parameters, such as fiber core type, grating period, grating refractive index value, grating type, and other design parameters [28]. These parameters will be needed to fabricate and create FBG based on the purpose of the design.

Numerical analysis of FBG could be done by using finite difference time domain (FDTD) method [29] [30] [31]. The numerical analysis could be conducted both in two-dimensional and three-dimensional case. Latter case will require a much higher demand for the computational resource since parameter involved in the equation are multiplied and the simulation area is tens to hundreds time bigger compared to two-dimensional approach. These lead to the long time period needed for three-dimensional analysis executed by using general central processing unit (CPU), namely hours to days depend on the size of the systems calculated.

To get faster execution time, several methods could be applied to the systems. Since FDTD code could be highly parallelized, parallel computing is one of the answers to getting faster simulation time. This parallel computing could be applied by using multiple CPU in one personal computer (PC) system or by using multiple PC systems at once. By using multiple CPU in one system PC, the limitation will come from the number of CPU and number of memory controller found in the system itself and usually can't get performance gain of more than 6 times. While for multiple PC systems, it can achieve higher performance gain, but with the much higher cost since the number of PC will be similar to the number of performance increase that desired.

Another approach application of parallel computing for FDTD is by using graphics processing unit (GPU) calculation [32] [33] [34]. This method uses GPU to calculate the numerical equation rather than to be used to process graphics output like what it usually does in gaming, or 2D and 3D rendering. GPU calculation has a merit that comes from up to thousands cores in a single system PC. With this potential, GPU computation comes with difficulties. Since the architecture of GPU is different from CPU, it will need special programming that different compared to CPU programming also.

This study could produce programs that could be used to evaluate FBG structures based on the design parameter. Both two and three-dimensional cases are evaluated and compared. Results from both cases are evaluated to see the advantages and disadvantages for each case. Further feedback could be used to optimize and increase the performance of both program cases. With the help from these programs, the design of FBG could be predicted before entering manufacturing process. New FBG structures also could be predicted if the designer found new interesting parameter changes that could be simulated.

With the aim to increase higher FDTD performance, result from this study could be used also in other research or study that use FDTD or similar computational methods in their simulation process, especially in the three-dimensional case.

1.2 OBJECTIVE OF THE STUDY

With the important function of FBG, we try to analyze it numerically and evaluate the output characteristic of FBG by using FDTD method. Both FBG function as a band reject filter and as a sensor are numerically analyzed. This program aimed to be able to imitate characteristic of FBG based on the design parameters. Furthermore, this FDTD program is also targeted to be able to analyze the future structure of FBG that are not proposed yet.

Both two-dimensional and three-dimensional cases of FDTD method will be conducted. Various different parameter changes will be applied to see the effect of its output characteristics. Trend changes in the output from FDTD program will be compared to real FBG characteristic also. Simulation of FBG as a thermometer sensor also included in the worklist.

With the highly demand of computational resource for three-dimensional simulation, this study also tried to increase the performance of the FDTD program and reduce the time needed for the simulation. Some of the approaches will be conducted in this study, such as using higher performance memory, using newer CPU and memory technology, as well as by using GPU to get faster simulation time. Each approach will be observed and the increasing performance will be recorded. Performance benchmark here will be measured by comparing the time needed for simulating same structure in the three-dimensional simulation case. The faster simulation time needed to finish the job means a better performance and vice versa.

This research also intends to find shortcoming from each dimensional simulation that has been conducted. Finding the cause, and the possible solution also become one of the final targets. Furthermore, describing future work that arises from the shortcomings of this study will be listed in the last section.

1.3 THESIS OUTLINE

This thesis consists of five chapters.

Chapter 1 summarizes the general introduction, including the background, objectives of this study, and the thesis outline. In this chapter, the significance function of FBG in the optical telecommunication is mentioned. Usefulness of FBG in a sensor region also listed which cover basic sensor function as a thermometer, strain sensor, humidity sensor, and so on.

Chapter 2 consists of some literature review that used in this study. A first literature review is discussing Fiber Bragg Grating (FBG) itself. How FBG founded, where the name of FBG come from, what is the law that lies inside it, as well as what is the equation that applied in FBG is mentioned in FBG portion.

After discussing FBG, literature review discusses Finite Difference Time Domain (FDTD). History and the origin of FDTD are mentioned first. After that, fundamental equations that become the roots of FBG follow. The root equation for one-dimension, two-dimension, and three-dimension are shown in this section. In each dimension, the equation is further processed and described. Illustration of the electric and magnetic field in each dimension also displayed. An example of a flowchart of FDTD program is shown in the last section of FDTD.

Another section of this chapter discusses Graphics Processing Unit (GPU) computing. Explanation of what is GPU computing is listed in the first portion. Discussing the history of GPU computing does not leave CPU calculation. How the early CPU program is constructed, the method that used to increase its performance in the 30 years of CPU computing also mentioned. The reason why early construction and 30 years of CPU optimization need to be changed is discussed. After that, comparison of CPU and GPU structure will take place. Comparison of different architecture that applied in the CPU and GPU also mentioned. Advantages and obstacles of using GPU are described with its reason. Program diagram of GPU programming is shown here with the execution diagram follows and discussed. A sample of programming code for GPU is shown here in the last section.

Chapter 3 includes numerical analysis in the two-dimensional case of FBG. Simulation setup and methodology fill the first three pages of this chapter. List of the software that used and initial parameter setting for FBG is shown also here. Workflow for this chapter is described in detail as well as a flowchart for this flow also shown. Following the methodology, several parameter changes are conducted in this chapter. Changes in grating refractive index value, grating period, total grating length, as well as chirped FBG are simulated in the two-dimensional case. The description of the parameters that used, graphics of output characteristic, and discussion of the results is presented in every simulation. Some simulation also includes a table for more detail overview of the data. After parameter changes simulation, FBG as a temperature sensor is shown. The description on how FBG could be used as a temperature sensor is shown also. Functions that involved for the temperature sensor is mentioned and described. Since the computational resource is relatively small, only CPU calculation is applied in all simulation in the two-dimensional chapter. Conclusion for two-dimensional simulation is shown in the last section of the chapter.

Chapter 4 contains numerical calculation in three-dimensional approach. Similar to the two-dimensional case, simulation setup and methodology fill first pages of this chapter. The quantity of the software that used in this chapter is greater than the software that used in the two-dimensional analysis. Parameters value is mentioned in this section with most of them is similar to the two-dimensional case with addition for supporting three-dimensional environment. Workflow also contains and describes here with its flowchart shown. With the high requirement for the three-dimensional case, optimization is needed to make the simulation time faster. Several tests conducted to see the increase of the performance, including changing memory, comparing the performance of two different PC systems generation, and try to obtain much faster simulation with the help of GPU computation. Involving GPU computing show much higher performance gain, thus GPU calculation is applied to the rest of the chapter. Similar to the two-dimensional case, several parameter changes will be applied here with similar changes also. Result in the graph will be shown and described in every parameter changes. Simulation for FBG as a temperature sensor also conducted. Conclusion for three-dimensional simulation is shown in the last section if this chapter.

Chapter 5 is a general conclusion. It mentions major objective in this study, and follow by conclusion from two and three-dimensional case. Comparison results between two and three-dimensional case are discussed here. Shortcoming from each dimensional case also mentioned including the possibility of the cause. The final summary based on all findings is described in detail in this chapter. Future work and its reason are listed in the last section.

II. Literature review

2.1 FIBER BRAGG GRATING

Fiber Bragg Grating (FBG) is an optical fiber that has the grating on its core, which formed by a periodic modulation of the refraction index in the core. This grating is lies along the longitudinal direction and could be produced by various techniques. The name of FBG was taken from the Bragg law that applied in the periodical structure inside the fiber core.

The Bragg law was discovered in 1912 by Sir William Lawrence Bragg, a British physicist and X-ray crystallographer who was born in 1890. This law has been used for the study and determination of crystal structure. When electromagnetic radiation entering to a crystalline material, Bragg diffraction will occur if it has the same order of magnitude to the atomic spacing. If there is a crystalline structure whose lattice plane layer has a distance d, X-ray that entering the crystal scatter and this scatter interferes constructively when different path length for each wave is same with the integer multiplication wavelength of the X-ray. Bragg's law equation is shown in Eq. (2.1), while illustration of Bragg's law is shown in Fig. II.1.

$$2d \sin\theta = n\lambda \tag{2.1}$$

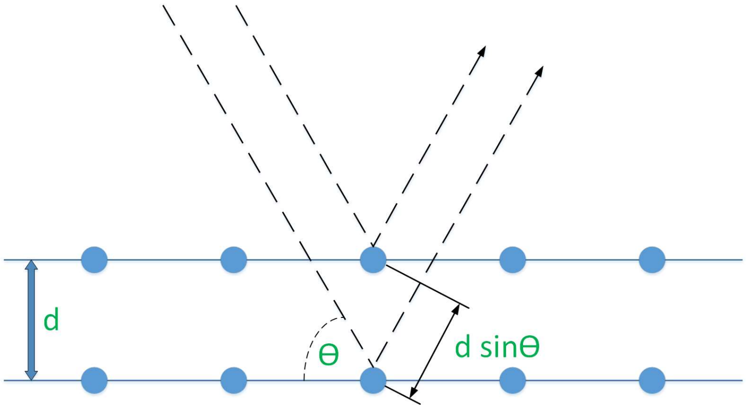

Fig. II.1. Bragg diffraction.

where θ is the incident angle, n is an integer number, while λ is the wavelength. When Bragg condition is not satisfied, the scattered radiation is weak. In contrary, when the angle θ is tuned to match Bragg condition, strong intensity of diffraction will occurs. This phenomenon call as Bragg peak in graph of scattered radiation as a function of angle θ.

The discovery of FBG initially was founded by chance. Report of index refraction changes that observed in germane silicate fiber is submitted by Kennet Hill and co-worker in 1978 [35]. At that time, they put a high power blue light into the fiber and the transmitted light unexpected decayed after a few minutes. In the report, they said that permanent grating could be written in the fiber core by using argon ion laser with a 488nm wavelength that launched into the fiber. This grating still has a very weak index modulation, thus reflection occurs on a narrow band only. When Hill used a beam splitter, he found that there is a missing light as a result of the reflection [36]. At the end of the fiber, he found 4% of the light that reflected by Fresnel effect from the fiber and interfered with ongoing light that producing the pattern. At first, the reflected light intensity is low, but since the light is continuously changing in the refractive index, the reflected light grows until almost all of the light is reflected back. The growth of the reflection back of the light is called as "photosensitivity".

With applying Eq (2.1) to this grating, we have $\theta = 90°$ and d is distance between peaks of the grating. Thus, $\lambda = 2d$ for $n = 1$ which is the approximation peak of the reflected wavelength. With this condition, the fiber now act as a mirror for specific band of wavelength. When adapted for silica, Bragg's condition that previously developed for vacuum now could be applied to these fiber gratings. Bragg's condition for FBG is shown in Eq. (2.2), [28]

$$\lambda_B = 2n_{eff}\,\Lambda \tag{2.2}$$

where λ_B is the wavelength reflected by FBG that satisfies Bragg condition or called as Bragg wavelength, n_{eff} is an effective refractive index of the fiber core, and Λ is a grating period of FBG. Furthermore, an overview of FBG structure is shown in Fig. II.2 while FBG characteristic in wavelength domain is shown in Fig. II.3.

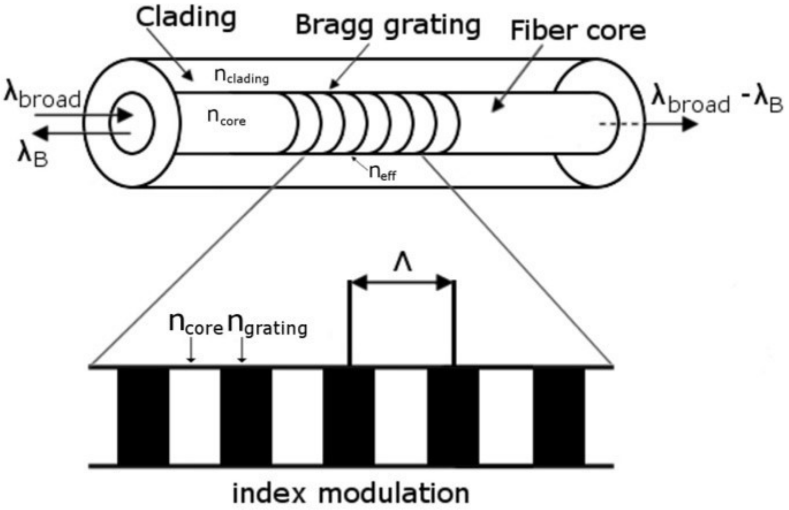

Fig. II.2. Overview of FBG.

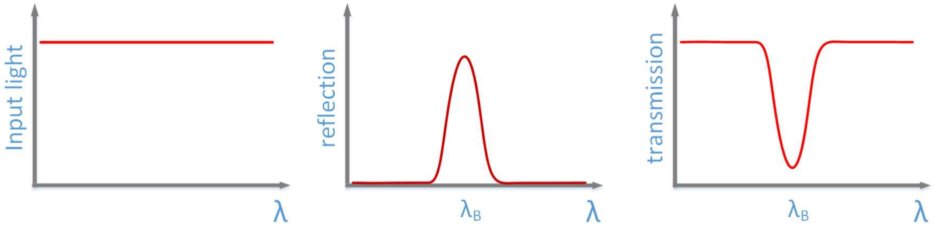

Fig. II.3. FBG characteristic in the wavelength domain.

2.2 FINITE DIFFERENCE TIME DOMAIN (FDTD)

Finite difference time domain (FDTD) is a numerical analysis technique that could be used to simulate electromagnetic with its interaction with physical object and the environment. FDTD is solving time-dependent Maxwell's equation by using several partial differential equations. This method originally proposed by Kane S. Yee in 1966 [37] that make this method also called Yee's method. The time-dependent Maxwell's equation in the free space is shown in Eq. (2.3), [38].

$$\frac{\partial \boldsymbol{E}}{\partial t} = \frac{1}{\varepsilon} \nabla \times \boldsymbol{H} \tag{2.3a}$$

$$\frac{\partial \boldsymbol{H}}{\partial t} = -\frac{1}{\mu} \nabla \times \boldsymbol{E} \tag{2.3b}$$

where ***E*** and ***H*** are electric fields and magnetic fields vector in three dimensions. From these equations, the vector components of the rotation operators in Cartesian coordinates produce six coupled scalar equations. Eq. (2.4) are derived from Eq. (2.3a), while Eq. (2.5) are derived from Eq. (2.3b).

x-components $$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z}\right) \tag{2.4a}$$

y-components $$\frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x}\right) \tag{2.4b}$$

z-components $$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}\right) \tag{2.4c}$$

x-components $$\frac{\partial H_x}{\partial t} = -\frac{1}{\mu}\left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z}\right) \tag{2.5a}$$

y-components $$\frac{\partial H_y}{\partial t} = -\frac{1}{\mu}\left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x}\right) \tag{2.5b}$$

z-components $$\frac{\partial H_z}{\partial t} = -\frac{1}{\mu}\left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y}\right) \tag{2.5c}$$

There is three kinds of finite difference approximation: forward finite-difference, backward finite-difference, and central finite-difference approximation [39] [40].

forward $$\frac{df}{dx}|_{x=x_0} = \frac{f(x_0+\Delta x)-f(x_0)}{\Delta x} \tag{2.6a}$$

backward $$\frac{df}{dx}|_{x=x_0} = \frac{f(x_0)-f(x_0-\Delta x)}{\Delta x} \tag{2.6a}$$

central $$\frac{df}{dx}|_{x=x_0} = \frac{f\left(x_0+\frac{1}{2}\Delta x\right)-f\left(x_0-\frac{1}{2}\Delta x\right)}{\Delta x} \tag{2.6a}$$

Forward and backward finite-difference approximation have first-order accurate, while central finite-difference approximation has second-order accurate [41]. Thus, smaller error achieved when using a central finite-difference approximation. To get a smaller error, the central finite-difference approximation is selected in this research. This approximation applied in first partial space and time derivative of Maxwell's equations.

$$\frac{\partial F}{\partial x} = \frac{F\left(x+\frac{\Delta x}{2},y,z,t\right)-F\left(x-\frac{\Delta x}{2},y,z,t\right)}{\Delta x} \tag{2.7a}$$

$$\frac{\partial F}{\partial t} = \frac{F\left(x,y,z,t+\frac{\Delta t}{2}\right)-F\left(x,y,z,t-\frac{\Delta t}{2}\right)}{\Delta t} \tag{2.7b}$$

where $\Delta x, \Delta y, \Delta z$ is cell size, and Δt is time step. With analytical region is divided into the small cell $(x, y, z, t) = (i\Delta x, j\Delta y, k\Delta z, n\Delta t)$. Notation for the function is determined as:

$$F(x, y, z, t) = F^n(i, j, k) \tag{2.8}$$

With central-difference applied, the leapfrog in time derivatives is used:

Electric field: $t = (n-1)\Delta t, n\Delta t, (n+1)\Delta t, \ldots$

Magnetic field: $t = \left(n-\frac{1}{2}\right)\Delta t, \left(n+\frac{1}{2}\right)\Delta t, \ldots$

Thus, finite-difference for the electric and magnetic field in time derivative are:

$$\frac{\partial \boldsymbol{E}}{\partial t}\Big|_{t=\left(n-\frac{1}{2}\right)\Delta t} = \frac{\boldsymbol{E}^{n}-\boldsymbol{E}^{(n-1)}}{\Delta t} \tag{2.9a}$$

$$\frac{\partial \boldsymbol{H}}{\partial t}\Big|_{t=n\Delta t} = \frac{\boldsymbol{H}^{\left(n+\frac{1}{2}\right)}-\boldsymbol{H}^{\left(n-\frac{1}{2}\right)}}{\Delta t} \tag{2.9a}$$

Substituting Eq. (2.3) with Eq. (2.9) produce the following results:

$$\frac{\boldsymbol{E}^{n}-\boldsymbol{E}^{(n-1)}}{\Delta t}=\frac{1}{\varepsilon}\nabla\times\boldsymbol{H}^{\left(n-\frac{1}{2}\right)} \tag{2.10a}$$

$$\frac{\boldsymbol{H}^{\left(n+\frac{1}{2}\right)}-\boldsymbol{H}^{\left(n-\frac{1}{2}\right)}}{\Delta t}=-\frac{1}{\mu}\nabla\times\boldsymbol{E}^{n} \tag{2.10b}$$

Arranging $\boldsymbol{E}^{n}$ and $\boldsymbol{H}^{\left(n+\frac{1}{2}\right)}$ we can obtain:

$$\boldsymbol{E}^{n}=\boldsymbol{E}^{(n-1)}+\frac{\Delta t}{\varepsilon}\nabla\times\boldsymbol{H}^{\left(n-\frac{1}{2}\right)} \tag{2.11a}$$

$$\boldsymbol{H}^{\left(n+\frac{1}{2}\right)}=\boldsymbol{H}^{\left(n-\frac{1}{2}\right)}-\frac{\Delta t}{\mu}\nabla\times\boldsymbol{E}^{n} \tag{2.11b}$$

From Eq. (2.11) we can see data started from the electric field $\boldsymbol{E}^{(n-1)}$ and magnetic field $\boldsymbol{H}^{\left(n-\frac{1}{2}\right)}$. The next step is calculate $\boldsymbol{E}^{n}$ from them. After that, magnetic field $\boldsymbol{H}^{\left(n+\frac{1}{2}\right)}$ could be calculated from $\boldsymbol{H}^{\left(n-\frac{1}{2}\right)}$ and $\boldsymbol{E}^{n}$. The calculation will continue for $\boldsymbol{E}^{(n+1)}$, $\boldsymbol{H}^{\left(n+1\frac{1}{2}\right)}$, and so on. Illustration of this process is shown in Fig. II.4.

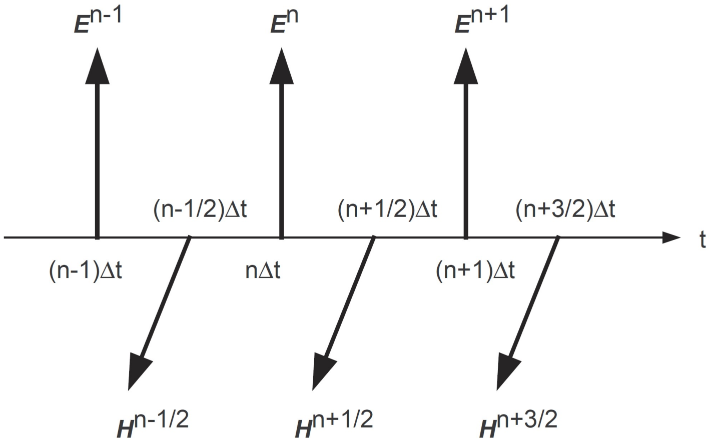

Fig. II.4. Illustration of **E** and **H** in FDTD formulation in time series.

2.2.1 ONE-DIMENSIONAL CASE

Maxwell's equation could be implemented in one-dimensional, two-dimensional, and three-dimensional case. For one-dimensional case, with an example of an electric field oriented in the x direction, magnetic field oriented in the y direction, and traveling in the z direction, we can get differential equation electric field ***E*** and magnetic field ***H*** in Eq. (2.4), [38].

$$\frac{\partial E_x}{\partial t} = -\frac{1}{\varepsilon}\frac{\partial H_y}{\partial z} \tag{2.12a}$$

$$\frac{\partial H_y}{\partial t} = -\frac{1}{\mu}\frac{\partial E_x}{\partial z} \tag{2.12b}$$

Eq. (2.12) are derived from Eq. (2.4) and Eq. (2.5) with ${}^{\partial}/_{\partial x} = 0$ and ${}^{\partial}/_{\partial y} = 0$. Then we substitute Eq. (2.9a) to Eq. (2.12a) resulting as follow:

$$\frac{E_x^n - E_x^{(n-1)}}{\Delta t} = -\frac{1}{\varepsilon}\frac{\partial H_y^{\left(n-\frac{1}{2}\right)}}{\partial z} \tag{2.13a}$$

$$E_x^n = E_x^{(n-1)} - \frac{\Delta t}{\varepsilon}\frac{\partial H_y^{\left(n-\frac{1}{2}\right)}}{\partial z} \tag{2.13b}$$

Similar to the process for center-difference in the time domain, center-difference in the space domain also applied.

$$\left.\frac{\partial H_y^{\left(n-\frac{1}{2}\right)}}{\partial z}\right|_{z=k\Delta z} = \frac{H_y^{\left(n-\frac{1}{2}\right)}\left(k+\frac{1}{2}\right) - H_y^{\left(n-\frac{1}{2}\right)}\left(k-\frac{1}{2}\right)}{\Delta z} \tag{2.14}$$

Substituting Eq. (2.13) and Eq. (2.14) we obtain:

$$\frac{E_x^n(k) - E_x^{(n-1)}(k)}{\Delta t} = -\frac{1}{\varepsilon}\frac{H_y^{\left(n-\frac{1}{2}\right)}\left(k+\frac{1}{2}\right) - H_y^{\left(n-\frac{1}{2}\right)}\left(k-\frac{1}{2}\right)}{\Delta z} \tag{2.15a}$$

$$E_x^n(k) = E_x^{(n-1)}(k) - \frac{\Delta t}{\varepsilon \Delta z}\left[H_y^{\left(n-\frac{1}{2}\right)}\left(k+\tfrac{1}{2}\right) - H_y^{\left(n-\frac{1}{2}\right)}\left(k-\tfrac{1}{2}\right)\right] \quad (2.15b)$$

Similar center-difference process both time and space for the magnetic field in Eq. (2.12b) resulting as follow:

$$\frac{H_y^{\left(n+\frac{1}{2}\right)} - H_y^{\left(n-\frac{1}{2}\right)}}{\Delta t} = -\frac{1}{\mu}\frac{\partial E_x^n}{\partial z}$$

$$\left.\frac{\partial E_x^n}{\partial t}\right|_{z=\left(k+\frac{1}{2}\right)\Delta z} = \frac{E_x^n(k+1) - E_x^n(k)}{\Delta z}$$

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

$$H_y^{\left(n+\frac{1}{2}\right)}\left(k+\tfrac{1}{2}\right) = H_y^{\left(n-\frac{1}{2}\right)}\left(k+\tfrac{1}{2}\right) - \frac{\Delta t}{\mu \Delta z}\left[E_x^n(k+1) - E_x^n(k)\right] \quad (2.16)$$

In the equations above, n is the parameter indicating time, while k is parameter indicating the position. Illustration of FDTD formulation in one-dimensional case is shown in FIG. II.5.

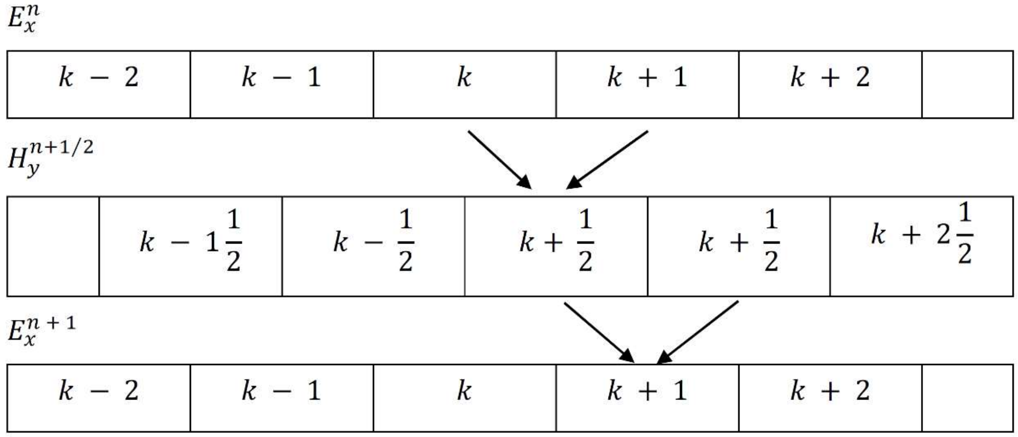

Fig. II.5. Interleaving of the E_x and H_y field in FDTD formulation both in space and time.

This picture illustrate that to calculate $H_y(k+\frac{1}{2})$, surrounding value of E_x at k and $k+1$ at time n are needed. This also apply for electric field, to calculate $E_x(k+1)$, data value from H_y at $k+\frac{1}{2}$ and $k+1\frac{1}{2}$ at time $n+\frac{1}{2}$ are necessary. All electric field $\boldsymbol{E}$ for time n will be calculated in the same time step. After that, calculating all magnetic field $\boldsymbol{H}$ in time $n+\frac{1}{2}$. Calculation continues to $\boldsymbol{E}^{(n+1)}$, $\boldsymbol{H}^{(n+1\frac{1}{2})}$, $\boldsymbol{E}^{(n+2)}$, $\boldsymbol{H}^{(n+1\frac{1}{2})}$, continuously until the time step reaching the limit.

2.2.2 TWO-DIMENSIONAL CASE

In two-dimensional case, the value of $^{\partial}/_{\partial z}$ is equal to 0. Applying it to Eq. (2.4) and Eq. (2.5) we got two different vector groups that separated each other. These groups are the transverse magnetic (TM) mode which consist of Ez, Hx, Hy, and the transverse electric (TE) mode which composed of Ex, Ey, Hz. In this research, we use TM mode for two-dimensional approach.

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}\right) \tag{2.17a}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\frac{\partial E_z}{\partial x} \tag{2.17b}$$

$$\frac{\partial H_x}{\partial t} = -\frac{1}{\mu}\frac{\partial E_z}{\partial y} \tag{2.17c}$$

When the center difference procedure for time domain applied to Eq. (2.17a), we got the equation as follow:

$$E_z^n = E_z^{(n-1)} + \frac{\Delta t}{\varepsilon}\left[\frac{\partial H_y^{\left(n-\frac{1}{2}\right)}}{\partial x} - \frac{\partial H_x^{\left(n-\frac{1}{2}\right)}}{\partial y}\right] \tag{2.18}$$

Furthermore, center-difference in space domain for Eq. (2.17) are:

$$\left.\frac{\partial H_y^{\left(n-\frac{1}{2}\right)}}{\partial x}\right|_{(x,y)=(i\Delta x,j\Delta \mathrm{y})} = \frac{H_y^{\left(n-\frac{1}{2}\right)}\left(i+\frac{1}{2},j\right)-H_y^{\left(n-\frac{1}{2}\right)}\left(i-\frac{1}{2},j\right)}{\Delta z} \tag{2.19a}$$

$$\left.\frac{\partial H_y^{\left(n-\frac{1}{2}\right)}}{\partial y}\right|_{(x,y)=(i\Delta x,j\Delta \mathrm{y})} = \frac{H_y^{\left(n-\frac{1}{2}\right)}\left(i,j+\frac{1}{2}\right)-H_y^{\left(n-\frac{1}{2}\right)}\left(i,j-\frac{1}{2}\right)}{\Delta z} \tag{2.19b}$$

Substitution from Eq. (2.18) and Eq. (2.19) above result in FDTD expression for electromagnetic field E as follows:

$$E_z^n(i,j) = E_z^{n-1}(i,j) + \frac{\Delta t}{\varepsilon(i,j)\Delta x}\left[H_y^{n-\frac{1}{2}}\left(i+\frac{1}{2},j\right) - H_y^{n-\frac{1}{2}}\left(i-\frac{1}{2},j\right)\right]$$

$$-\frac{\Delta t}{\varepsilon(i,j)\Delta y}\left[H_x^{n-\frac{1}{2}}\left(i,j+\frac{1}{2}\right) - H_x^{n-\frac{1}{2}}\left(i,j-\frac{1}{2}\right)\right] \tag{2.20}$$

where n is the parameter indicating time, while i and j is parameter indicating the position in x and y axis.

For magnetic wave H in Eq. (2.17b) and Eq. (2.17c), FDTD expression results of central-difference processing both in time and space domain are:

$$H_y^{\left(n+\frac{1}{2}\right)}\left(i+\tfrac{1}{2},j\right) = H_y^{\left(n-\frac{1}{2}\right)}\left(i+\tfrac{1}{2},j\right)$$

$$+\frac{\Delta t}{\mu\Delta y}\left[E_z^n(i+1,j) - E_z^n(i,j)\right] \tag{2.21a}$$

$$H_x^{\left(n+\frac{1}{2}\right)}\left(i,j+\tfrac{1}{2}\right) = H_x^{\left(n-\frac{1}{2}\right)}\left(i,j+\tfrac{1}{2}\right)$$

$$-\frac{\Delta t}{\mu\Delta y}\left[E_z^n(i,j+1) - E_z^n(i,j)\right] \tag{2.21b}$$

Interleaving between E and H for two-dimensional TM FDTD case is shown in Fig. II.6

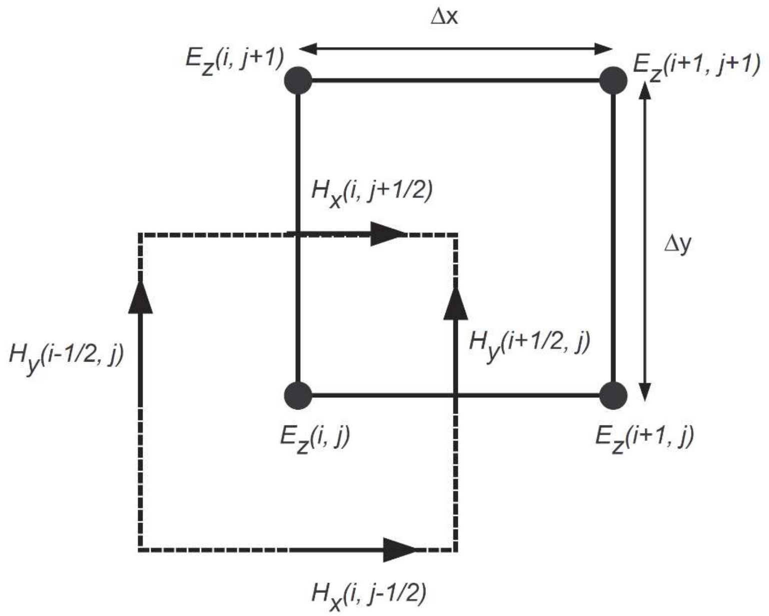

Fig. II.6. Interleaving of the ***E*** and ***H*** field in two-dimensional TM formulation

2.2.3 THREE-DIMENSIONAL CASE

For three-dimensional simulation case, all electric and magnetic field in Eq. (2.4) and Eq. (2.5) are calculated. Original FDTD concept in the three-dimensional case is described by *Yee cell* shown in Fig. II.7. The name was taken from *Kane Yee* that first time proposed this approach.

This picture is only containing one box of the cell. In the simulation, there will be many boxes that construct the three-dimensional environment. In the simulation, these boxes will be arranged side by side in all three axes with facing the same direction and creating much bigger boxes where the simulated structure lies in it. The size of this box will depend on the frequency that will be simulated since if the size of the box too big or the frequency is too high, FDTD calculation could not work well.

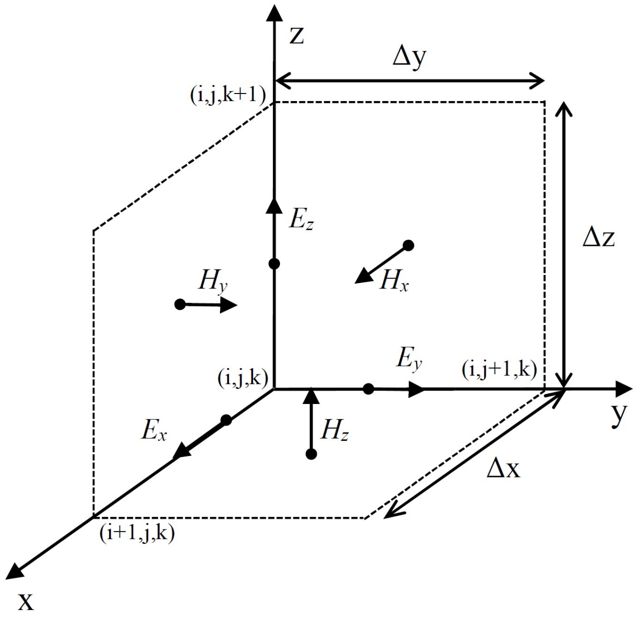

Fig. II.7. Yee cell

For finite difference approximation, we use (2.4c) and (2.5c) as examples:

$$E_z^n\left(i,j,k+\tfrac{1}{2}\right) = E_z^{n-1}\left(i,j,k+\tfrac{1}{2}\right) + \frac{\Delta t}{\varepsilon\Delta x}\left[H_y^{n-\frac{1}{2}}\left(i+\tfrac{1}{2},j,k+\tfrac{1}{2}\right) - H_y^{n-\frac{1}{2}}\left(i-\tfrac{1}{2},j,k+\tfrac{1}{2}\right)\right] - \frac{\Delta t}{\varepsilon\Delta y}\left[H_x^{n-\frac{1}{2}}\left(i,j+\tfrac{1}{2},k+\tfrac{1}{2}\right) - H_x^{n-\frac{1}{2}}\left(i,j-\tfrac{1}{2},k+\tfrac{1}{2}\right)\right] \quad (2.22a)$$

$$H_z^{n+\frac{1}{2}}\left(i+\tfrac{1}{2},j+\tfrac{1}{2},k\right) = H_z^{n-\frac{1}{2}}\left(i+\tfrac{1}{2},j+\tfrac{1}{2},k\right) - \frac{\Delta t}{\mu\Delta x}\left[E_y^n\left(i+1,j+\tfrac{1}{2},k\right) - E_y^n\left(i,j+\tfrac{1}{2},k\right)\right] + \frac{\Delta t}{\mu\Delta y}\left[E_x^n\left(i+\tfrac{1}{2},j+1,k\right) - E_x^n\left(i+\tfrac{1}{2},j,k\right)\right] \quad (2.23b)$$

2.2.4 FDTD FLOWCHART

An example flowchart of FDTD is shown in Fig. II.8.

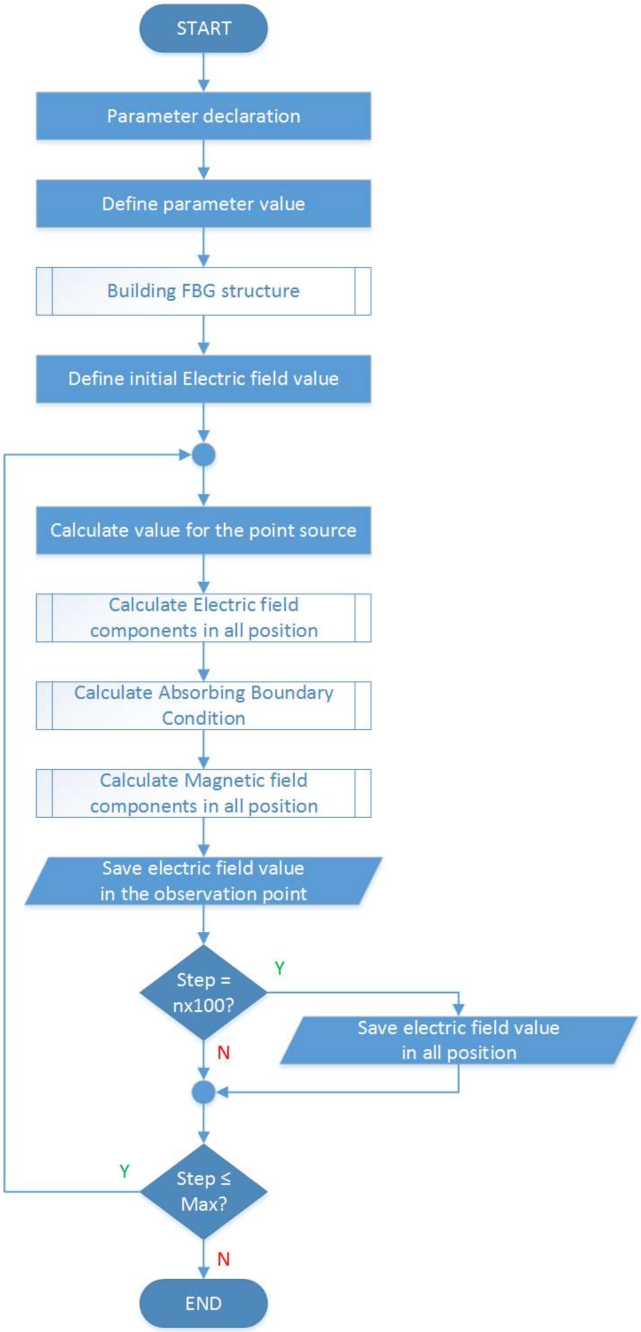
START
Parameter declaration
Define parameter value
Building FBG structure
Define initial Electric field value
Calculate value for the point source
Calculate Electric field components in all position
Calculate Absorbing Boundary Condition
Calculate Magnetic field components in all position
Save electric field value in the observation point
Step = nx100?
Y
Save electric field value in all position
N
Y
Step ≤ Max?
N
END

Fig. II.8. Flowchart of an FDTD program

In FDTD simulation, some of the computer codes provided in the previous subsection are used in the calculation of the electric and magnetic field in all position. By seeing these codes, we can see that electric and magnetic variable are placed in the array and calculated as a matrix with the dimension that equal to the dimension of the simulated structure. Since these arrays are saved in the memory, these arrays will be frequently accessed by the processor (CPU or GPU) during the simulation. Thus, memory bandwidth is an important factor in FDTD simulation. Other than these codes, we need to declare the parameter, build the environment and FBG structure that is simulated, defining the initial condition, saving electrical field value, and so on.

2.3 GRAPHICS PROCESSING UNIT (GPU) COMPUTING

Graphics processing unit (GPU) computing is the use of GPU that usually used for taking computation for graphic to perform a calculation that traditionally handled by Central Processing Unit (CPU) in a computer system. This calculation uses parallelization in GPU structure to get a higher overall bandwidth of the running program.

Traditionally, a program that runs on the computer in the early era of programming is running in a sequential process. The process is easily understood by human being since the process is stepped sequentially through the program. With this process, the program will run faster by using faster CPU that usually came in the newer generation. Since the sequential program only uses a single core, the performance gain in the newer CPU generation is usually implemented by increasing CPU frequency. For 30 years, this one only important approach is implemented in the increasing performance for consumer computing [42].

In the recent generation, higher frequency is not efficient to gain higher performance since there is a limitation for working frequency of the CPU. Since 2003, semiconductor industry tried to maintain execution speed of the sequential program while moving into multiple cores [43]. Multi-cores for the CPU begins with two cores which could run two simultaneous processes in each core. The sequential program couldn’t get the benefit if running alone in this type of CPU, while parallelized program could boost its performance by using both CPU cores at the same time. For GPU itself, it has much higher number of cores that contain in one unit. Illustration of CPU and GPU is shown in Fig. II.9.

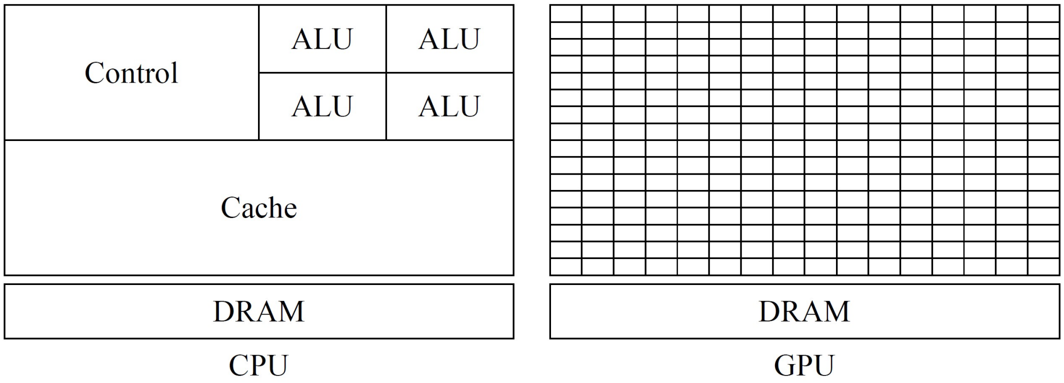

Fig. II.9. Illustration construction of CPU and GPU

CPU and GPU have a different design approach. CPU usually have a small number of cores, up to 8 cores for a consumer product, and each one has a good power for sequential performance and capable of doing many instructions variation supported by its complex control hardware. While for GPU, it consists of many cores with much smaller and simpler architecture for doing simpler instruction for each core. With more cores used in the parallelized program, performance of the program could be multiplied also. Even with slower and simpler instruction for GPU cores, the parallel program could benefit much faster performance from these hundreds to thousands of GPU cores.

These higher potential doesn't come at a cost. With GPU that have a simpler control unit, instructions that its cores could process is different compared to the CPU cores. Furthermore, language program style that used in GPU also different to the programming code for the CPU. Thus, program code that runs on the CPU can't be run on GPU directly and must be rewritten with GPU style programming if the programmer wants to use GPU computing. While using GPU computing, GPU will work as a coprocessor beside the CPU. CPU will manage the whole program, while GPU is doing the large number multi-thread processing.

In this research, GPU programming that used is (Compute Unified Device Architecture) CUDA. CUDA is parallel computing platform that created by NVidia. By using this platform, we can use a CUDA-enabled graphics processing unit for general purpose processing (GPGPU). The list of CUDA-enabled GPU is consisting of GPU that produces by NVidia only such as NVidia Tesla, NVidia Quadro, NVidia NVS, NVidia GeForce, and TEGRA/Jetson. Not all of NVidia cards have similar CUDA compute compatibility. Older cards tend to support more basic instructions or features, while newer cards support higher capability. CUDA platform support general programming languages such as C, C++, and Fortran. But, as mentioned in the previous paragraph, since the structure is different with CPU, program code that already written in C, C++, and Fortran for the CPU will need to be rewritten in GPU programming style. Other programs also support the use of CUDA such as F#, Java, Mathematica, MATLAB, .NET, Phyton, etc. Illustration of CUDA program Diagram is shown in Fig. II.10.

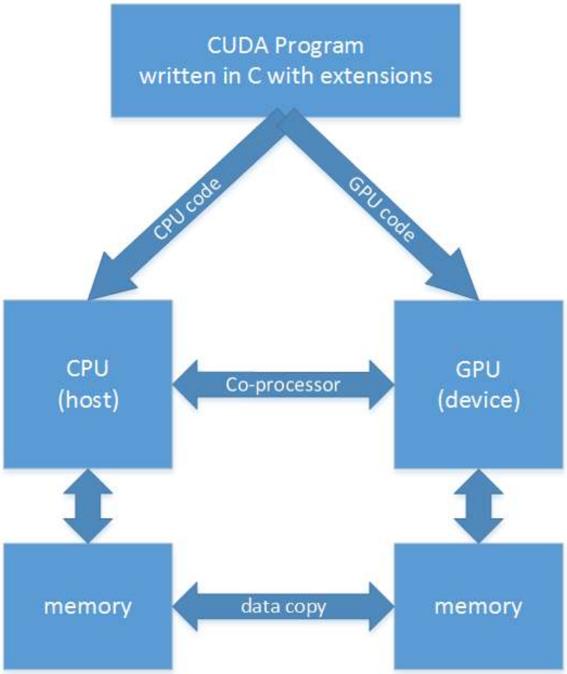

Fig. II.10. CUDA program diagram

With CUDA program, program code for CPU and GPU will be divided separately. When the program running on the CPU, it will process the data in CPU memory (RAM).

When GPU is processing the program inside it, GPU will access to its own memory. GPU can't access CPU RAM, and CPU can't access GPU memory directly. Data copy between CPU RAM and GPU RAM is possible, and this is one of the important features. This process needed to prepare the data in GPU before kernel in GPU launched, and take back the result from GPU memory to CPU memory when the kernel finished. When result data needed to be displayed, stored or modified further by other CPU application, the data must reside in the CPU memory since GPU can't access to the storage device directly and other CPU application didn't access to GPU memory. With the program diagram that differs from usual CPU program, execution process of a CUDA code also differ to the usual execution process of the CPU program. The illustration of the execution process of a CUDA program is shown in Fig. II.11.

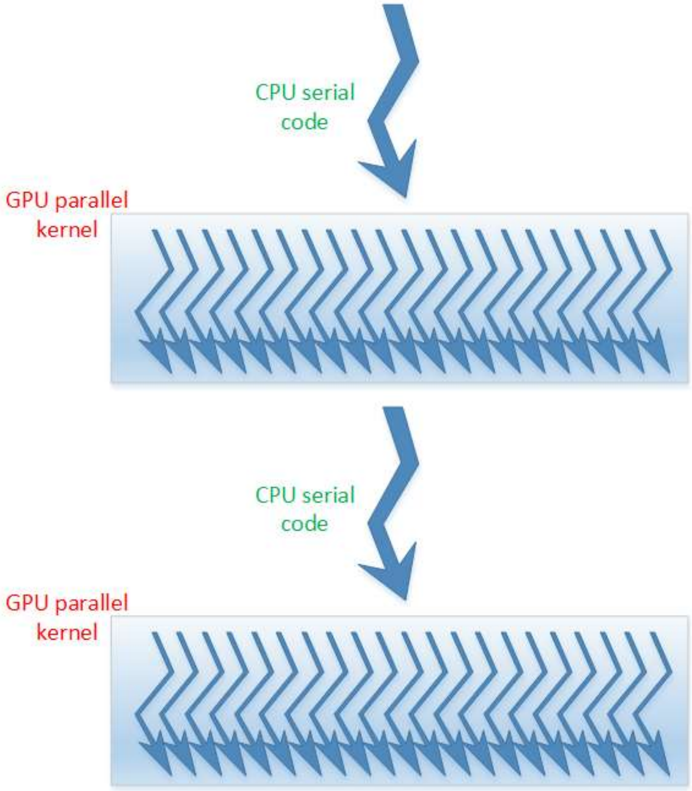

Fig. II.11. Execution diagram of CUDA program

The execution of the program is started with CPU (host) execution with preparation for GPU kernel launch. When a kernel function is launched, execution is moved to GPU (device) where a numerous number of threads are generated. Usually, there is more than one process of the kernel that involved in the program. In that case, the CPU will prepare the kernel again and the other kernel will be launched again for another task in GPU.

In the CUDA program, code script for both CPU and GPU could reside in the same file. Usually, the script for the CPU is used for parameter setting, variable declaration, kernel setting, kernel data preparation, including another task to prepare kernel launching in GPU and data processing after kernel finished. The script for launching kernel is containing the kernel's name, kernel's parameter, and kernel's variable. An example of CUDA program is shown below.

```
#include <stdio.h>

//kernel for GPU
__global__ void cube(float * d_out, float * d_in){
    int idx = threadIdx.x;
    float f= d_in[idx];
    d_out[idx] = f * f * f;
}

int main(int argc, char ** argv) {
    const int ARRAY_SIZE = 64;
    const int ARRAY_BYTES = ARRAY_SIZE * sizeof(float);

    // generate the input array on the host
    float h_in[ARRAY_SIZE];
    for (int i = 0; i < ARRAY_SIZE; i++) {
        h_in[i] = float(i);
    }
    float h_out[ARRAY_SIZE];

21      // declare GPU memory pointers
22      float * d_in;
23      float * d_out;
24
25      // allocate GPU memory
26      cudaMalloc((void**) &d_in, ARRAY_BYTES);
27      cudaMalloc((void**) &d_out, ARRAY_BYTES);
28
29      // transfer the array to GPU
30      cudaMemcpy(d_in, h_in, ARRAY_BYTES, cudaMemcpyHostToDevice);
31
32      // launch the kernel
33      cube<<<1, ARRAY_SIZE>>>(d_out, d_in);
34
35      // copy back the result array to the CPU
36      cudaMemcpy(h_out, d_out, ARRAY_BYTES, cudaMemcpyDeviceToHost);
37
38      // print out the resulting array
39      for (int i =0; i < ARRAY_SIZE; i++) {
40          printf("%f", h_out[i]);
41          printf(((i % 4) != 3) ? "\t" : "\n");
42      }
43
44      cudaFree(d_in);
45      cudaFree(d_out);
46
47      return 0;
48  }
```

Line 4 to 8 is initialization for GPU kernel, as well as what job that needs to be processed in GPU in line 7. Line 10 to 48 is the main program, consist of array declaration, GPU memory declaration, GPU memory allocation, copy the array to GPU, kernel launching, retrieving result data, and displaying result data. When the kernel is not launched yet, GPU is not running the kernel that declared in the first place. Before launching the kernel, it is important to make sure that every kernel needs is already prepared and declared, otherwise, the error could arise when the kernel is launched.

This program is trying to make cube value of the number from 1 to 64, which done by using 64 threads in GPU at once in the same time. Since the process is done at once in 64 different threads, time needed for the execution is similar to 1 thread that cubing a single data number. By this example, we can see that 64 thread involved in this program could multiply the performance compared to a sequential program that needs to do the same cube calculation 64 times.

III. Two-dimensional analysis

3.1 Simulation Setup and Methodology

Software that used for this simulation are:

1. Visual Studio Community 2013

   This software is a fundamental application that is needed for other following programming software to work well. This software is a free version of software provided by Microsoft.

2. Intel Visual Fortran XE 2013

   This software is Fortran language programming provided by Intel. This language program is used for creating two-dimensional FDTD calculation, as well as to create FBG structure, and evaluate its output by using fast Fourier transform (FFT). All numerical examinations in two-dimensional simulations are using this programming language.

For simulation parameter, the analytical region is z=430μm times x=7μm. The length of the grating structure is up to 200μm, with the core width is 3μm. The refractive index of the cladding is n_3=1.44, while core refractive index is n_2=1.46. The cell sizes are set to be Δx=Δz=100nm, and the time step size Δt=$2.06475\text{x}10^{-16}$s is used.

The Gaussian pulse is applied as the incident wave in the simulations. The value of grating period is set as Λ_0=529nm. Illustration of FBG simulation area for two-dimensional examination is shown in Fig. III.1.

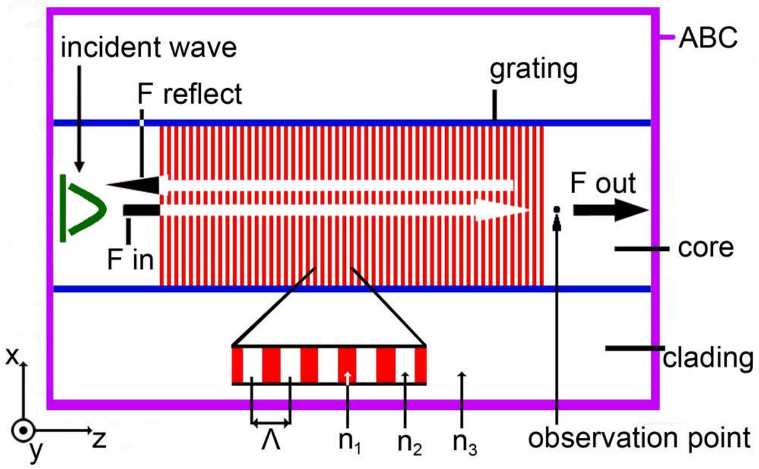

Fig. III.1. Illustration of FBG simulation area.

Since the data from FDTD is collected in the time domain, fast Fourier transform (FFT) technique is applied to the output data to get output characteristic in the frequency or wavelength domain. Absorbing boundary condition (ABC) is using 2nd order MUR ABC [44].

In this chapter, many processes are needed to get the final data that ready to be displayed or compared. This process including studying FBG parameter, creating mathematical FBG structure, preparing FDTD code, creating FBG sub-program, parameter tuning for FBG and FDTD, building and running FDTD program, gathering data for FFT, modify FFT program if necessary, building and running FFT program, and processing final data. These processes are needed since FDTD program needs to be built from the program code. FDTD program also couldn’t run by itself and the final result can’t be obtained only by using FDTD code. With this condition, many of these steps are need to be done. The workflow that applied in this chapter is shown in Fig. III.2. Evaluation and feedback are done both in every step or in the wider chain if something went wrong.

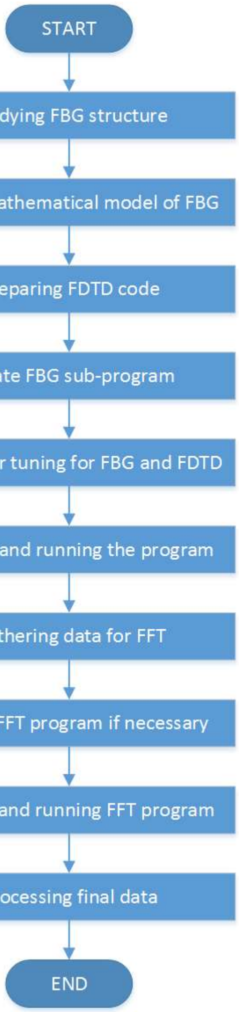
START
Studying FBG structure
Create mathematical model of FBG
Preparing FDTD code
Create FBG sub-program
Parameter tuning for FBG and FDTD
Building and running the program
Gathering data for FFT
Modify FFT program if necessary
Building and running FFT program
Processing final data
END

Fig. III.2. Workflow for two-dimensional simulation.

In studying FBG structure, how the real FBG structure constructed with its parameter value are gathered. After that, creating a mathematical model from real FBG structure is conducted. If more than one structure is needed, each structure will need its own mathematical model. Preparing FDTD code is the next thing to do, including preparing point source, the equation for the electric and magnetic field, the equation for absorbing boundary condition, and preparing data recorder. Usually, modification to this area is necessary if something goes wrong. After FDTD program ready, a mathematical model of FBG is placed in the subprogram of FDTD. This process needs to convert the equation to computer code correctly to make the simulation work well. After FBG sub-program ready, parameter tuning for both FBG and FDTD program are necessary. This process aims to create correct FBG parameter value that will be simulated and construct FDTD environment that suits to FBG that lies within the program. If everything already prepared, compiling and running FDTD program is the next process to do. Compiling FDTD program involving Intel Fortran compiler for Windows, and running FDTD program is done by using windows command prompt. After FDTD program finished without error, output data is prepared for FFT simulation, including adding zero data to fill the necessary rows using Maruo editor. By default, FFT program will process 2^{19} data, produce the result in a db format for y axis and show frequency in the x axis. If this number or parameter need to be changed, then the modification to FFT program is needed. Similar like FDTD program, FFT program will be compiled and running in the Windows environment. Result from FFT program will be further processed by using wgnuplot program for display or comparison purpose.

3.2 FBG PARAMETERS CHANGES ANALYSIS

3.2.1 GRATING REFRACTIVE INDEX VALUE CHANGES

The transmission property of FBG for several different values of grating refractive index n_1 is examined. Fig. III.3 shows the transmittance characteristic results for various grating refractive indexes n_1=1.46, n_1=1.462, n_1=1.464, n_1=1.466, n_1=1.468 and n_1=1.470. When n_1=n_2=1.46, this condition corresponds to no grating structure. The sinusoidal grating refractive index is applied, with the grating period Λ=529nm.

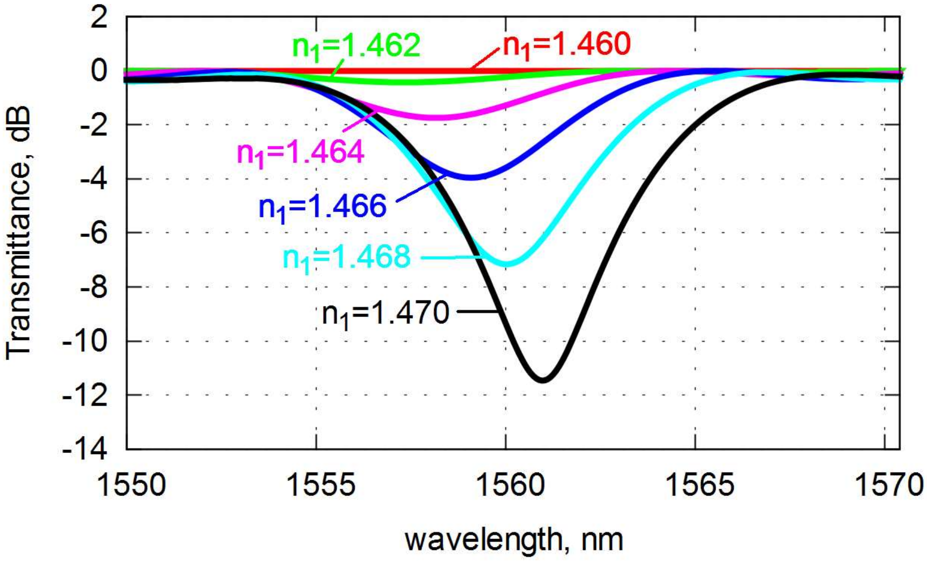

Fig. III.3. Transmittance for FBG with grating refractive index n_1 changes.

By observing Fig. III.3, it is shown that if there is no grating (n_1=1.46), FBG transmits all range of wavelength in this area. For n_1=1.462, the transmission for some wavelength is dropped. While for n_1=1.464 and larger value, we can see different transmission amplitude occur and center of transmission wavelength drop also shifted. Comparison between five difference grating refractive indexes shown that higher grating refractive index value affects to a smaller value for transmission amplitude and the center of transmission wavelength drops slightly shifted to a higher value.

3.2.2 GRATING PERIODS CHANGES

Several FBG structures with a different grating period are simulated to see the effect in their transmission characteristic properties. For this simulation, the refractive index of the cladding is n_3=1.44, core refractive index n_2=1.46, grating refractive index n_1=1.47, and the sinusoidal type grating refractive index shape applied. Grating period structure ranged from Λ=525nm to Λ=533nm with gap 1nm for each simulation.

Fig. III.4 show 5 different grating period results from 527nm to 531nm, while Fig. III.5 shows the relation between the center of transmission wavelength drop and all grating period results from 515nm to 533nm. The mark indicates numerical results for every center of transmission wavelength drop and the solid line means the linear polynomial regression.

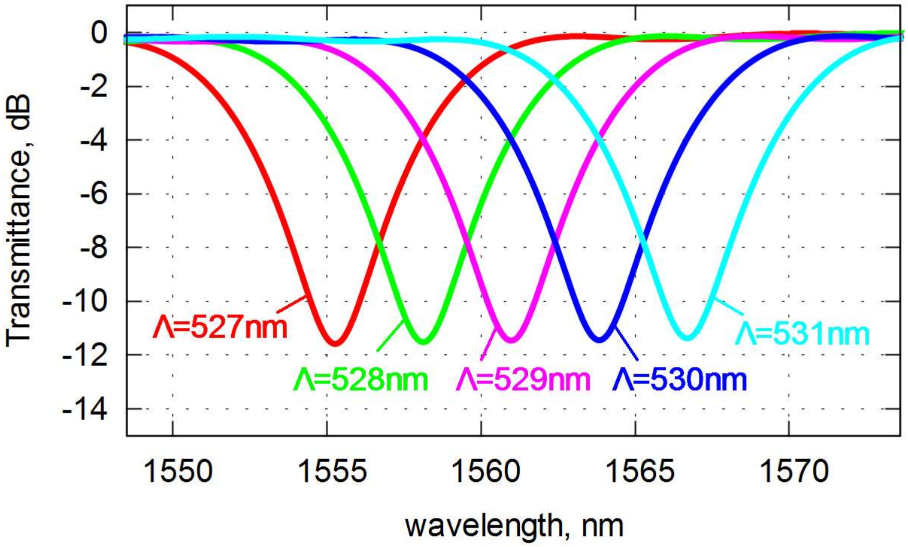

Fig. III.4. Transmission of FBG with several grating periods Λ.

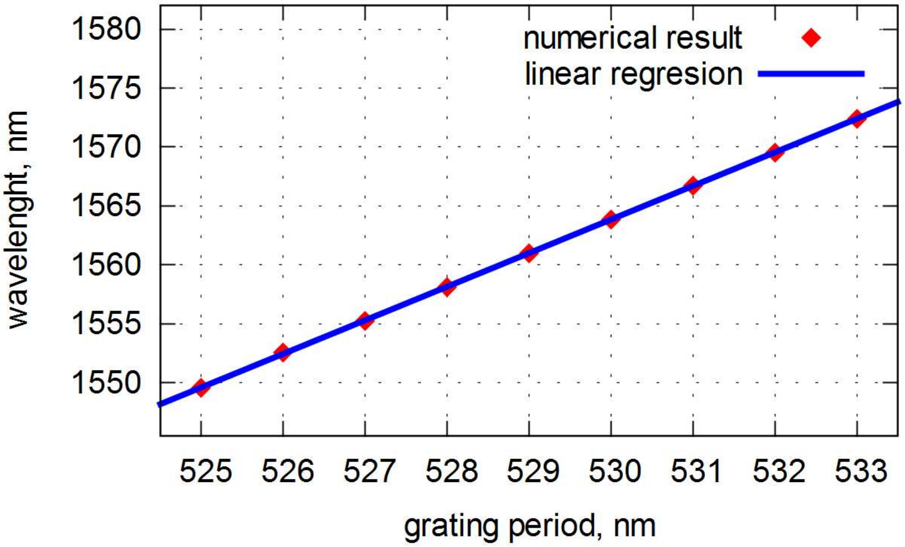

Fig. III.5. Center of wavelength drop for FBG with several grating periods Λ.

Results show that the grating period and the reflected wavelength have a positive correlation. The Higher grating period will correspond to the higher wavelength. From this picture and the table, we can observe that correlation between them is a linear correlation. This is suits to the Bragg condition. Data list for Fig. III.5 are shown in Table III.1.

Table III.1. Center wavelength drop for several grating periods.

Grating period (in nm)	Wavelength (in nm)
525	1549.55
526	1552.56
527	1555.27
528	1558.12
529	1560.97
530	1563.83
531	1566.69
532	1569.54
533	1572.40

3.2.3 GRATING LENGTH CHANGES

Several FBG with different grating length are simulated. The grating length that applied to FBG is ranged from 0mm (correspond to no grating structure) to 200µm, with the differences of 50µm for each FBG structures simulated. Grating period Λ=529nm and grating refractive index value n_l=1.470 are used.

Fig. III.6 show comparison from some result from the simulation in wavelength domain, while Fig. III.7 show comparison between the center of wavelength drop with the transmittance value from 50µm to 200µm with 25µm different total length between the data.

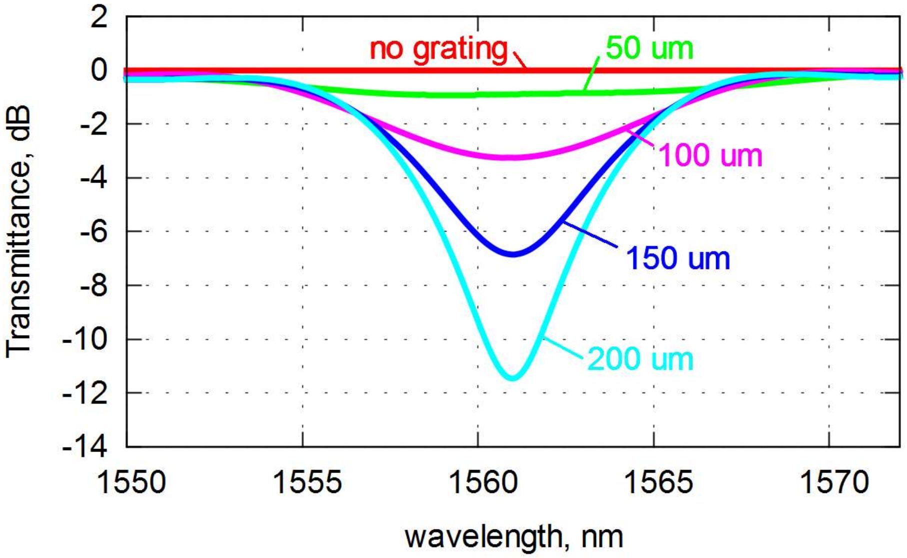

Fig. III.6. Transmittance for FBG with different total grating length.

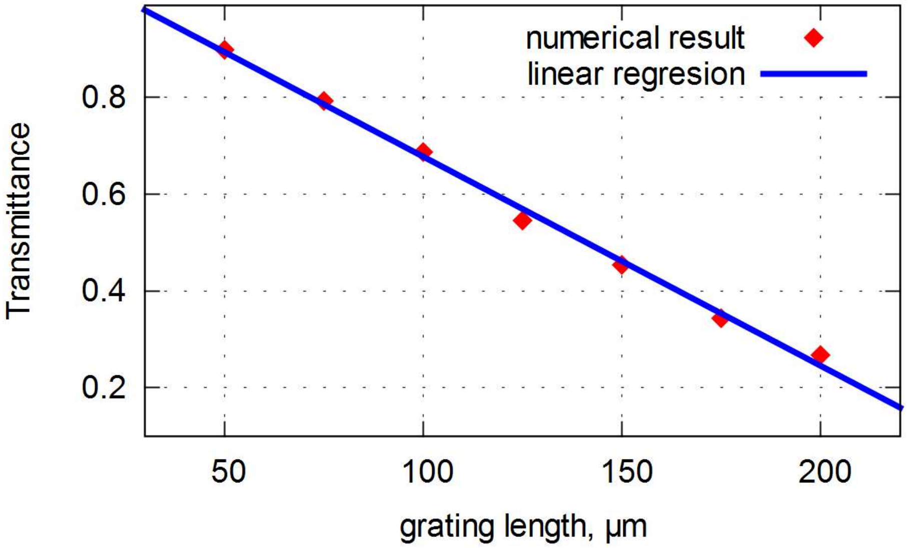

Fig. III.7. Transmission power drop of FBG with different total grating periods.

Results show that all FBG with gratings structure give similarity in the center of the wavelength that dropped. We can see clearly also that increasing grating total length will affect the lower transmission value. **Error! Not a valid bookmark self-reference.** show all data that presented in Fig. III.7.

Table III.2. Transmittance for different total grating periods.

Grating length (in µm)	Transmittance (in unit)
50	0.8989
75	0.7922
100	0.6871
125	0.5451
150	0.4544
175	0.3428
200	0.2670

3.2.4 CHIRPED FIBER BRAGG GRATING (CFBG)

CFBG is an FBG with a different grating period in the core. The grating period is gradually changing from the first grating toward the end of FBG structure. Illustration of CFBG is shown in Fig. III.8. CFBG will be simulated with same first grating periods while having different end grating period between each structure. The first grating period is fixed at 529nm, while end grating period simulated here is 529nm (uniform FBG), 531nm, 533nm, and 535nm. The result from the simulation is shown in Fig. III.9.

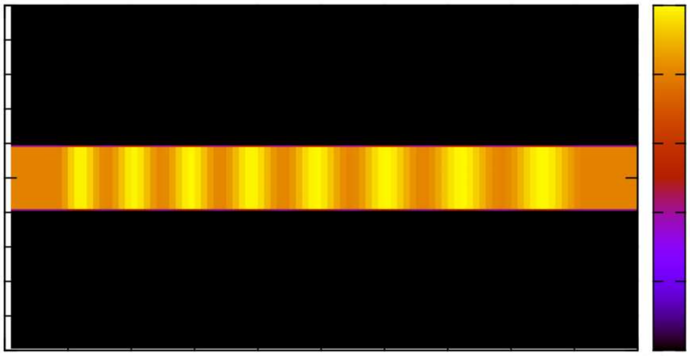

Fig. III.8. Illustration of simulation model area for Chirped FBG structure.

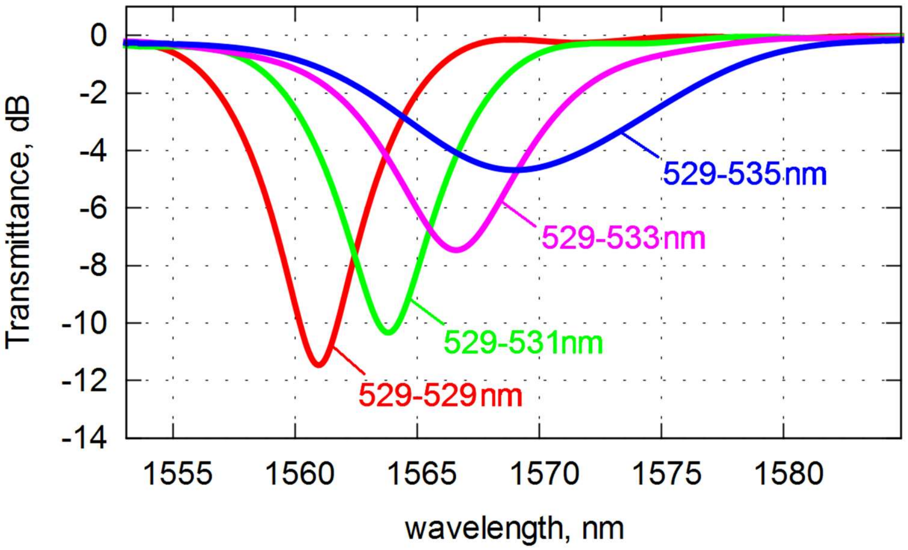

Fig. III.9. The transmittance of CFBG with a different end grating period.

The result shows that CFBG with the different end grating period will generate different output characteristic. If we increase end grating, wider wavelength broadband will be reflected by CFBG, but the reflection power is decreased. This result is reasonable, since increasing end grating period while maintaining first grating period will create more variation of the grating periods in the structure. Thus, will reflect the wider number of wavelengths. With same total grating structures length, increasing end grating period will reduce the number of grating that reflects a certain wavelength, which affects to lower reflectance power.

Another simulation for CFBG is simulating structure that has its first and end grating period swapped. Results from CFBG that have first grating period 529nm and last grating period 533nm is compared to the similar structure with a 533nm grating period as the first and 529nm grating period as the last structure. The result is shown in Fig. III.10.

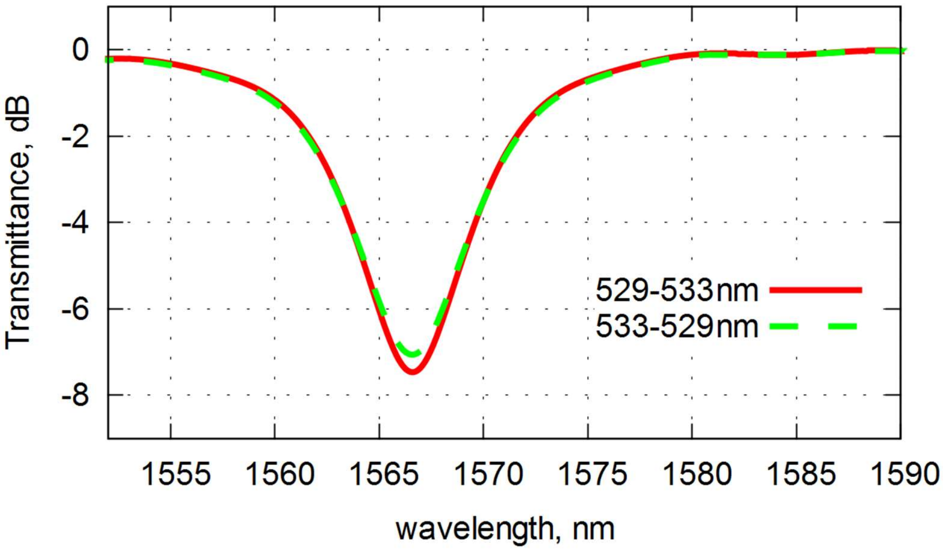

Fig. III.10. Transmittance for FBG with exchanged the first and the end grating period.

3.3 TEMPERATURE SENSING

By seeing equation (2.2) $\lambda_B = 2n_{eff}\,\Lambda$, we can observe that Bragg wavelength only depends on the grating period and refractive index. Any external factor that could change these two values could shift the center of peak reflected wavelength. This is the fundamental property to sense changes in the surrounding of FBG. Thus, other than act as wavelength or frequency filter, FBG could be used as an optical fiber sensor (OFS). One of the applications of the OFS is a temperature sensor. With the increasing of the temperature, FBG will be expanded which will expand the grating period with it. Even this expansion is very small, we still need to calculate it. Another parameter that will be changed with the increasing of the temperature is the refractive index of the fiber core, cladding, and the grating.

To calculate temperature sensitivity for FBG, we start from Bragg equation with paying attention that the sensitivity is partial derivative of the temperature [4]:

$$\frac{\Delta\lambda_B}{\Delta T} = 2n_{eff}\frac{\partial\Lambda}{\partial T} + 2\Lambda\frac{\partial n_{eff}}{\partial T} \tag{3.1}$$

Substitute the original Bragg equation twice will result:

$$\frac{\Delta\lambda_B}{\Delta T} = \frac{1}{\Lambda}\frac{\partial\Lambda}{\partial T}\lambda_B + \frac{1}{n_{eff}}\frac{\partial n_{eff}}{\partial T}\lambda_B$$

With rearrangement we get

$$\frac{\Delta\lambda_B}{\lambda_B} = \frac{1}{\Lambda}\frac{\partial\Lambda}{\partial T}\Delta T + \frac{1}{n_{eff}}\frac{\partial n_{eff}}{\partial T}\Delta T$$

Where the first part is thermal expansion of the silica (α) that affect the period of FBG, while the second part is thermo-optic coefficient (η) which represent temperature dependence for the refractive index $\left({dn}/{dT}\right)$. With both term substituted, total wavelength shift that caused by temperature changes is shown in Eq. (3.2) [4]

$$\Delta\lambda_B/\lambda_B = (\alpha + \eta)\Delta T \qquad (3.2)$$

If we separate and process further for each α and η, we get:

$$\alpha = \frac{1}{\Lambda}\frac{\partial\Lambda}{\partial T}$$

$$\alpha\,\Lambda = \frac{\partial\Lambda}{\partial T}$$

Thus:

$$\Delta\Lambda = \frac{\partial\Lambda}{\partial T}\Delta T$$

$$\Delta\Lambda = \alpha\,\Lambda\,\Delta T$$

New grating period with effect from temperature included will become:

$$\Lambda' = \Lambda + \Delta\Lambda$$

$$\Lambda' = \Lambda + \alpha\,\Lambda\,\Delta T \qquad (3.3)$$

For η we also get:

$$\eta = \frac{1}{\mathrm{n}}\frac{\partial \mathrm{n}}{\partial T}$$

$$\eta\ \mathrm{n} = \frac{\partial \mathrm{n}}{\partial T}$$

Thus:

$$\Delta \mathrm{n} = \frac{\partial \mathrm{n}}{\partial T}\Delta T$$

$$\Delta \mathrm{n} = \eta\ \mathrm{n}\ \Delta T$$

New refractive index with effect from temperature included will become:

$$\mathrm{n}' = \mathrm{n} + \Delta \mathrm{n}$$

$$\mathrm{n}' = \mathrm{n} + \eta\ \mathrm{n}\ \Delta T \tag{3.4}$$

The parameters for silica with a germanium doped core are having values α =0.55 x 10^{-6}/°C and η=8.6 x 10^{-6}/°C [4]. For the refractive index changes, it affects all values in the simulation including cladding refractive index, fiber core refractive index, as well as grating refractive index values. With these parameters included in the programming code, we can simulate the effect of temperature changes to FBG output characteristic.

Since the equation is mentioning ΔT rather than exact temperature value, we need to give reference point to make the simulation easier to understand. Recalling the determining values for cladding and fiber core refractive index, these values are taken for silica that have surrounding temperature of 20°C [45]. Thus, $\Delta T = 0$ is equal to 20°C, $\Delta T = 5$ is equal to 25°C, and so on.

For the simulation, several different Δ T is given to FBG. This ΔT value is ranged from 0°C to 80°C (T=20°C to 100°C) with 10°C difference in every simulation. Fig. III.11

show output characteristic for 6 different temperatures. Fig. III.12 and Table III.3 show comparison between Δ T and center of wavelength drop from all data.

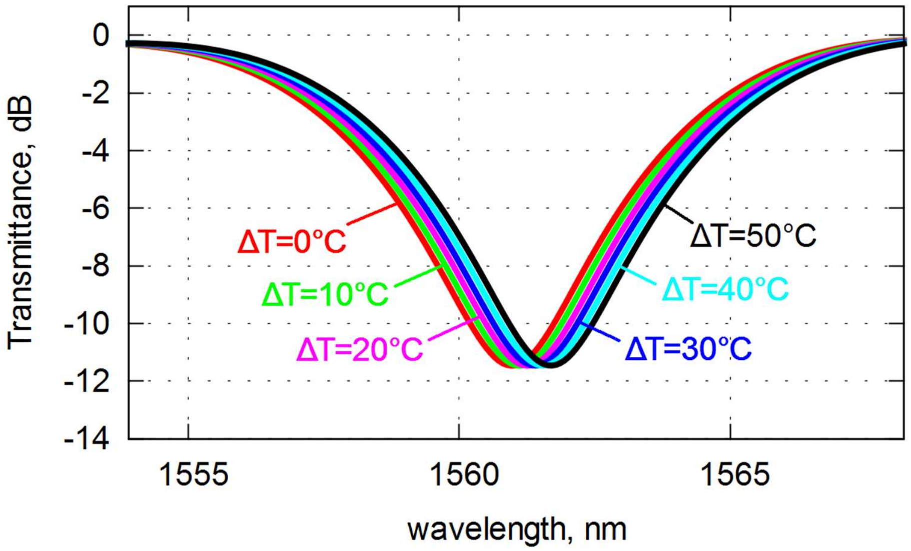

Fig. III.11. Transmission of FBG with different ΔT.

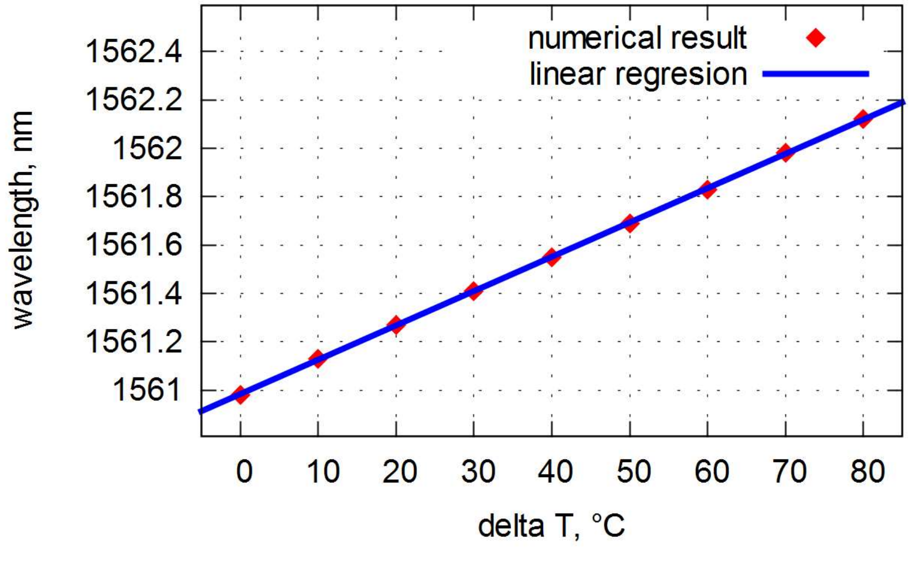

Fig. III.12. Center of wavelength drop for FBG with different ΔT.

Table III.3. Center of wavelength drop for different ΔT.

Delta T (in °C)	Wavelength (in nm)
0	1560.98
10	1561.13
20	1561.27
30	1561.41
40	1561.55
50	1561.69
60	1561.83
70	1561.98
80	1562.12

3.4 CONCLUSION

FBG structures with different parameters have been simulated by using FDTD Method. Higher grating refractive index value affects in a smaller value of transmission in a specific wavelength, and the center wavelength drop is shifted to a higher value. Increasing grating period will affect to reflected wavelength that shifting to a higher value with a linear correlation between them. Longer grating length will reduce the transmission value with a linear correlation between them. Chirped FBG (CFBG) show a weaker reflectance, but with wider broadband wavelength reflected due to more grating period variations contained in FBG. Higher temperature will result in wavelength shift to the higher wavelength value, with a linear correlation between them.

All of the results from the two-dimensional numerical analysis show a similar property to the real FBG characteristics. FBG that act as a band reject filter in the optical area and as an optical fiber sensor in the sensor region are simulated well. This could be concluded that two-dimensional program created could imitate real FBG structures.

IV. Three-dimensional analysis with GPU support

4.1 Simulation Setup and Methodology

Software that used for this simulation are:

1. Visual Studio Community 2013

   This software is a fundamental application that is needed for other following programming software to work well. This software is a free version software provided by Microsoft.

2. Intel Visual Fortran XE 2013

   This software is Fortran language programming provided by Intel. This language program is used for creating FBG structure, as well as to evaluate its output by using fast Fourier transform (FFT).

3. CUDA Toolkit 8.0 2016

   This software is baseline toolkit for using GPU calculation on NVidia cards. With the existence of this toolkit, another application that uses CUDA language for calculation could access and use GPU resources. This toolkit is provided freely by NVidia.

4. Open source FDTD CUDA

   Open source FDTD program code that used in this chapter is provided by P. Klapetek, found at http://gsvit.net/ and http://gwyddion.net/. This program is general FDTD solver that supporting the use of NVidia GPU for faster simulation time. This software could be used for scientific research in nanotechnology and nanoscale optics, like scanning near-field optical microscopy, tip-enhanced Raman scattering, rough surface scattering, etc.

For three-dimensional case, similar simulation parameters are applied with several different changes compared to two-dimensional analysis. As the analytical region is set as x = 210 µm, y = z = 7 µm. The length of the grating structure is still as up to 200µm, with the core width is 3µm. All refractive indexes are similar to the two-dimensional case, the refractive index of the cladding is n_3=1.44, while core refractive index is n_2=1.46. The cell sizes are set to be Δx = Δy = Δz =100nm, and the time step size Δt=1.92583x10^{-16}s is used. The Gaussian pulse is also applied as the incident wave in the simulations. The value of grating period is also same as Λ_0=529nm. Illustration of FBG for three-dimensional examination is shown in Fig. IV.1.

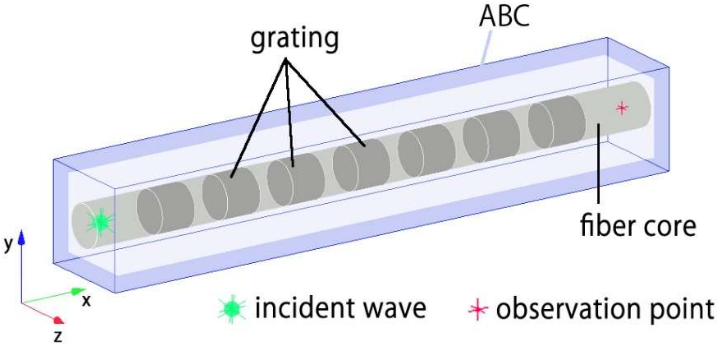

Fig. IV.1. Illustration of 3D FBG structure simulation area.

Absorbing boundary condition (ABC) is using convolutional perfectly matched layer (CPML) [46] with 20 layers thick in the boundary that is touched by fiber core, and 15 layers thick for the boundaries that parallel to the fiber core. Similar to two-dimensional analysis, since the data from FDTD method is collected in the time domain, fast Fourier transform (FFT) technique is applied to the output data to get output characteristic in the frequency or wavelength domain.

Similar to two-dimensional case, there are many processes that involved in this chapter. Since the use of the software that involved is different, this causes a slightly different process in this chapter. The processes that involved in this chapter are studying FBG parameter, creating mathematical FBG structure, studying open source FDTD program, creating FBG program, Parameter tuning for FBG and FDTD, attaching FBG structure and running FDTD program, gathering data for FFT, modify FFT program if necessary, building and running FFT program, and processing final data. The workflow for this chapter is shown in below.

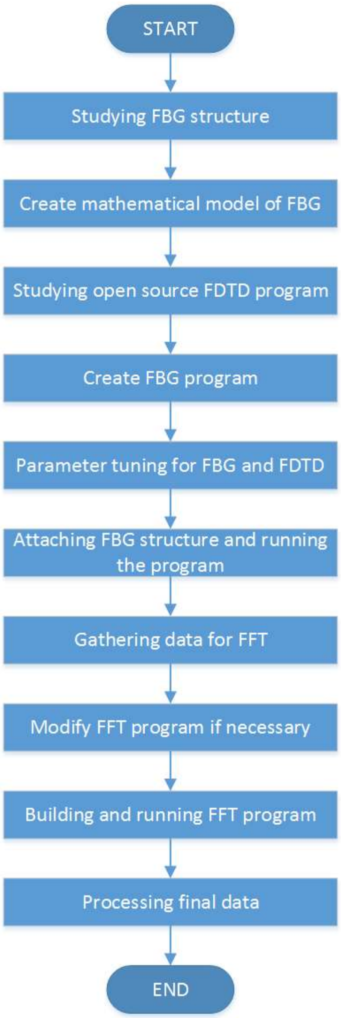
START
Studying FBG structure
Create mathematical model of FBG
Studying open source FDTD program
Create FBG program
Parameter tuning for FBG and FDTD
Attaching FBG structure and running the program
Gathering data for FFT
Modify FFT program if necessary
Building and running FFT program
Processing final data
END

Fig. IV.2. Workflow for three-dimensional simulation.

The slight different process compared to the two-dimensional case are started from studying open source FDTD program. While in two-dimensional case all of FDTD program are created in the lab, the three-dimensional FDTD program is obtained from the open source software. Studying requirement, behavior, characteristic, and limitation of this software is necessary. This also includes studying on how the program could simulate an FBG structure inside its environment. The next difference is creating FBG program rather than a sub-program. This is needed since it is found that it is difficult to build FBG structure inside the open source program. Separated program is created solely for creating FBG and its parameter that suit to open source FDTD program. The output from this program will create a file containing scripts that could be read by the open source software used here. After that, attaching FBG structures file to the program, preparing appropriate parameter setting in the program, and running FDTD program is the next thing to do. Since the program is made by other persons, it is found that output file from the program is different compared to the two-dimensional case. To make the output data can be read properly by the FFT program, output data need a further process by using Notepad, Excel, and Maruo editor. After that, the rest of the process is similar to two-dimensional case.

4.2 ENHANCING PERFORMANCE

Three-dimensional FDTD analysis is involving a high quantity of calculation, with long execution time is needed when using general CPU calculation. With some approaches could be chosen to increase the performance, we choose to use GPU calculation to get faster simulation time. The reasons behind this decision are GPU calculation is much cheaper that multi-PC systems, while only use few times wattage usage compared one PC system, but the performance gain from it could reach up to hundreds of times compared to single thread CPU calculation.

To see the performance of GPU calculation across different PC systems, two PC systems are used in this chapter. Specification of the 1st PC system is:

- Dell Precision T5500 (Desktop PC)
- Intel® Xeon® E5520 4 cores 8 threads @2.25-2.53GHz
- 24GB DDR3-1066 (PC3-8500 8-7-7-20) triple channel
- Windows 10 64 bit
- GeForce GTX Titan X (Maxwell version)
  3072 CUDA cores @1-1.2GHz, 12GB GDDR5 384-bit 7Gbps

While the specification of the 2nd PC system is:

- MSI GS43VR (Notebook PC)
- Intel® Core® i7-6700HQ 4 cores 8 threads @2.6-3.5GHz
- 16GB DDR4-2400 (PC4-19200 17-17-17-39) dual channel
- Windows 10 64 bit
- GeForce GTX 1060
  1280 CUDA cores @1.4-1.8GHz, 6GB GDDR5 192-bit 8Gbps

But, before using GPU calculation for the analysis, we need to evaluate its performance and result's similarity compared to general CPU performance. This step is necessary to confirm whether GPU calculation has good output results.

4.2.1 CPU-GPU MEMORY USAGE

The first comparison between CPU and GPU calculation is comparing memory usage in both scenarios. This step is confirming if the memory usage is still in the PC system's memory capacity, and to see memory usage trend for both calculation methods. For this purpose, both of the systems will simulate uniform 3D FBG with the length of 200µm. The refractive index of the cladding is n_3=1.44, while core refractive index is n_2=1.46, and the grating refractive index is n_1=1.44. The grating period is Λ_0=529nm. PC random access memory (RAM) and GPU memory usage will be recorded. Table IV.1 show comparison of memory usage for CPU and GPU calculation in both systems.

Table IV.1. Memory usage for different systems and simulation scheme.

Simulation scheme	1st PC system		2nd PC system	
	RAM	GPU	RAM	GPU
CPU 1 thread	970 MB	0 MB	970 MB	0 MB
CPU 8 threads	970 MB	0 MB	970 MB	0 MB
GPU computing	1,556 MB	639 MB	1,556 MB	669 MB

From Table IV.1 we could see that both system show similarity in the memory usage in all cases. CPU computation in both system as well as single and multi-threads calculation show 0 MB usage for GPU, which mean they did not involve GPU memory for the CPU computation only. Single and multi-thread calculation use a similar amount of PC RAM. This may be caused by the program that already designed to run multi-thread calculation that leads to a similar memory of usage even only used as a single thread calculation. GPU computation involving both PC RAM and GPU memory for the calculation. PC RAM usage is also getting bigger for GPU computation because it also contains variable that calculated in GPU memory. Furthermore, all memory usage is still in both PC RAM and GPU memory capacity, memory size shouldn't be a problem.

4.2.2 CPU-GPU RESULT'S COMPARISON

In this subsection, the comparison between CPU and GPU computing results will be conducted. Both results in the time domain and frequency domain will be observed to see the similarity. In CPU calculation, single thread and multi-threads simulation also conducted. Output's results in the time domain and frequency domain is shown in Fig. IV.3 and Fig. IV.4 respectively.

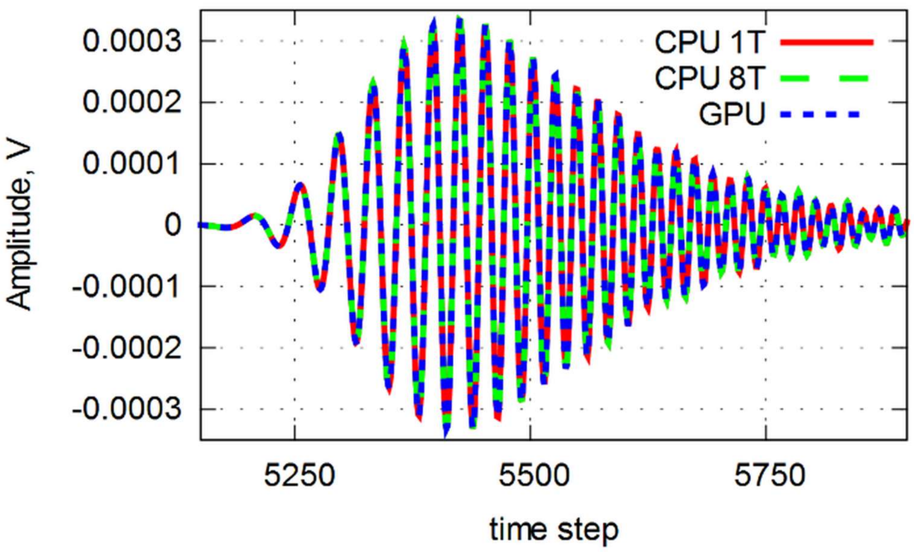

Fig. IV.3. CPU-GPU comparison results in the time domain.

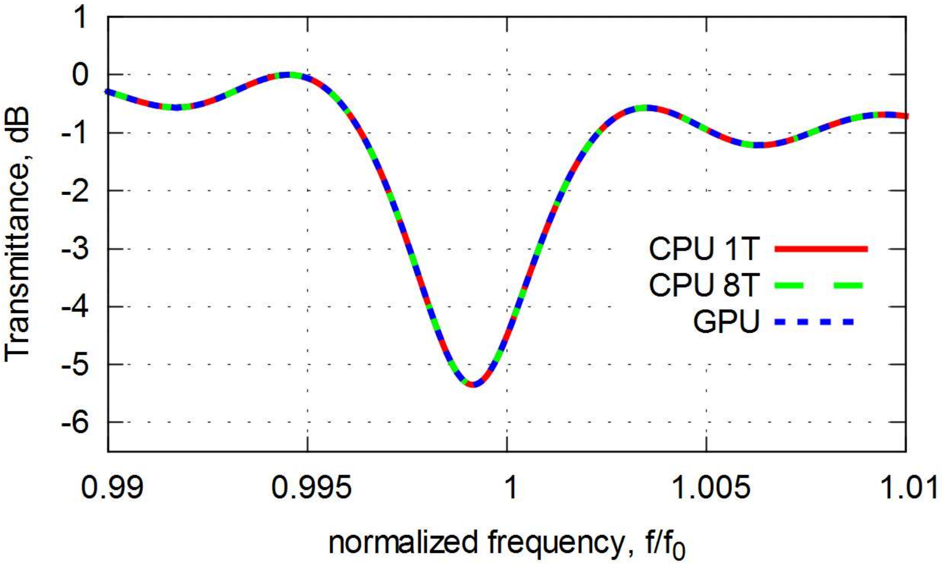

Fig. IV.4. CPU-GPU comparison results in the frequency domain.

We can observe from Fig. IV.3 and Fig. IV.4 that results from CPU and GPU computing is indistinguishable both in time and frequency domain. Further examination from time domain results in the Fig. IV.3 shows that median error between CPU and GPU result is 0.007%, with average error is 0.03%. This means that GPU results are similar to CPU results, and GPU computation could substitute CPU computation based on the result's similarity.

4.2.3 CPU-GPU EXECUTION TIME

With the result's similarity between CPU and GPU calculation, we move forward to see the performance gain of GPU calculation in both PC systems. Similar to the previous subsection, both CPU and GPU calculation is conducted, with single and multi-thread calculation are evaluated in CPU calculation. Table IV.2 show execution time for the 1st PC system, while Table IV.3 show data for 2nd PC system.

Table IV.2. Time performance comparison for 1st PC system.

Simulation scheme	Simulation time		Performance Gain
	hh:mm:ss	second	
CPU 1 thread	10:55:15	39,315	1 x
CPU 8 threads	03:35:42	12,942	3.038 x
GPU computing	00:06:14	374	105.120 x

Table IV.3. Time performance comparison for 2nd PC system.

Simulation scheme	Simulation time		Performance Gain
	hh:mm:ss	second	
CPU 1 thread	5:47:56	20,874	1 x
CPU 8 threads	02:09:42	7,782	2.683 x
GPU computing	00:06:28	388	53.804 x

From Table IV.2 we can see that GPU computation speed could reach more than 100 times faster than a single thread CPU computing for 1st PC system, and around 34.6 times faster than by using all 8 CPU threads. While Table IV.3 show performance gain of 53.804 times and 20.054 times from GPU computing compared to single CPU and all 8 CPU thread simulation respectively in the 2nd PC system.

Even by using all 8 threads that are available from both PC system, performance gains are much less than 8 times compared to single thread calculation in CPU only calculation. This phenomenon is caused by FDTD method that bound to memory speed rather than CPU speed. Thus, even with 8 thread CPU are available to the execution, the performance gain is not far from the number of the channel that memory controller architecture applied, which is around 2-3 times only. On the other hand, even with similar GPU execution time, the performance gain from GPU computing in the 2nd system is not as high as the 1st system, due to the 1st system PC having a weaker CPU.

Additionally, since the specification of 1st GPU is higher than the 2nd system, similar GPU execution time for both PC system is showing that GPU isn't working optimally in the 1st PC system. There is a bottleneck that probably occurred in the 1st PC since the speed of 1st GPU is around 50% faster than 2nd GPU. This bottleneck could come from the motherboard and CPU that can't deliver data as fast as GPU could process. 1st GPU card is using peripheral component interconnect express (PCI-E) generation 3 in its interface, while the motherboard is only having version 2 in the board architecture which only have half of the maximum theoretical speed of generation 3. Another factor that could contribute to the bottleneck is the architecture of the 1st board chipset that has a relatively slower speed compared to recent PC systems.

This subsection shows that performance gain of GPU calculation is affected by several factors especially specification of GPU and others PC systems components follow. Even with similar GPU execution time and lower performance gain in the 2nd PC system, GPU performance results are still showing a much faster simulation time compared to CPU calculation both single and multi-threads analysis.

4.2.4 RAM MEMORY TIMING

Since FDTD is bound to memory speed, this subsection wants to evaluate different memory timing and see the effect of execution time. 2nd PC system is used, with two different pairs of memory is tested, and other components of the system remain the same.

Both pairs of memory tested have a similar working frequency, but with different memory timing. Memory timing is the amount of the clock needed for the memory module to respond the instruction from the memory controller. Thus, lower memory timing is equal to faster respond of the memory in the same frequency and lead to lower latency and higher maximum throughput of the memory. General applications usually designed not to depends on the memory timing, and faster memory timing fairly has an effect on them. But, since FDTD method depends on memory speed, the performance increase could be gained.

Two pairs of memory that used are:

- Kingston DDR4-2400 17-17-17-39 (1st RAM)
- Kingston DDR4-2400 14-14-14-35 (Hyper X Series, 2nd RAM)

DDR4-2400 corresponds to DDR4 technology, with 2400MHz data frequency applied in the system. While 4 pair of numbers sequentially behind the frequency are:

- CAS latency (CL)
- RAS to CAS delay (tRCD)
- RAS Precharge (tRP), and
- RAS active time (tRAS) respectively.

These numbers are corresponding to each task when accessing the memory.

To get a relatively good comparison, a structure that used to examine is still the same to the structure that is applied in the three previous subsections above. Data obtained for both pairs of memory in GPU and CPU calculations are shown in Table IV.4.

Table IV.4. Time performance comparison for different memory timing.

Simulation scheme	1st RAM duration		2nd RAM duration		Performance Gain
	hh:mm:ss	second	hh:mm:ss	second	
CPU 1 thread	05:47:56	20,874	05:24:41	19,482	7.16 %
CPU 8 threads	02:09:42	7,782	01:59:27	7,167	8.58 %
GPU computing	00:06:28	388	00:06:28	388	0.00 %

Results show that different memory timing has an effect to FDTD simulation, but are only applied to CPU calculation. This phenomenon makes sense since GPU calculation takes most of the calculation in GPU with its own memory, rather than in the CPU with PC RAM that compared here. These results also confirm that FDTD that highly use memory speed have its performance increased when using memory with lower latency, even with same working frequency.

4.3 FBG PARAMETERS CHANGES ANALYSIS

4.3.1 GRATING REFRACTIVE INDEX VALUE CHANGES

The transmission property of FBG for several different values of grating refractive index n_1 is examined in this subsection to confirm its effect in output characteristics. Fig. IV.5 shows the transmittance characteristic results for various grating refractive indexes n_1=1.46, n_1=1.462, n_1=1.464, n_1=1.466, n_1=1.468 and n_1=1.47. When the refractive index n_1=n_2=1.46, this condition corresponds to no grating structure. The grating period is set as Λ=529nm. Transmittance is indicated in decibel (dB) as the function of wavelength (nm).

By observing Fig. IV.5, it is shown that if there is no grating (n_1=1.46), FBG transmits all range of wavelength in this area. For n_1=1.462, the transmission for some wavelength is slightly dropped. While for n_1=1.464 and larger value, we can see different transmission amplitude occur and center of transmission wavelength drop also shifted. Comparison between five difference grating refractive indexes shown that higher grating refractive index value affects to a smaller value for transmission amplitude and the center of transmission wavelength drops slightly shifted to higher wavelength value.

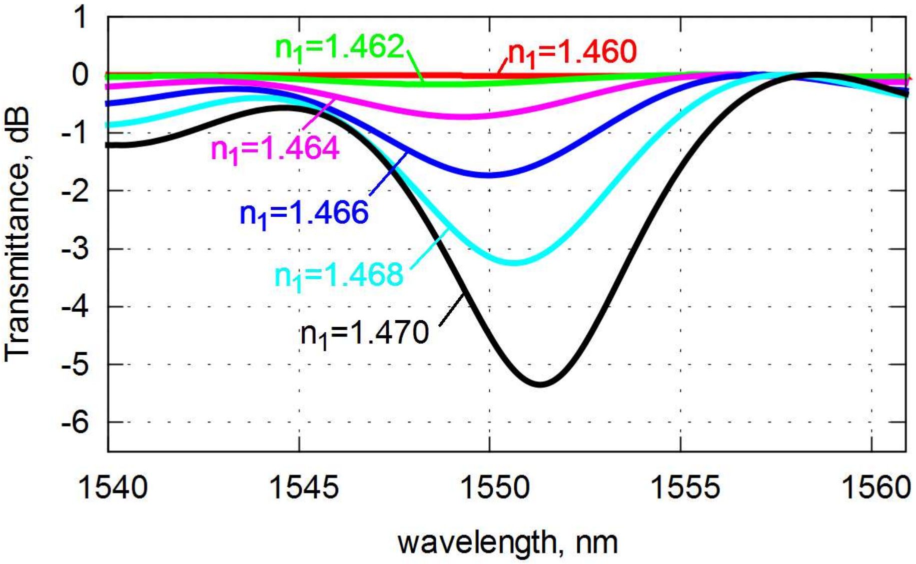

Fig. IV.5. Transmittance for 3D FBG with different grating refractive index n_1.

4.3.2 Grating periods changes

After simulating different grating refractive index value, several FBG structures with a different grating period are simulated to examine their transmission characteristic properties. For this simulation, the refractive index of the cladding is n_3=1.44, core refractive index n_2=1.46, grating refractive index n_1=1.47. Grating period structure ranged from Λ=525nm to Λ=533nm with a gap of 1nm for each simulation.

Fig. IV.6 shows several transmission properties results from 5 different grating period from Λ=527nm to Λ=531nm. While Fig. IV.7 and Table IV.5 show data for all simulation results.

We could see clearly that increasing grating period will result in higher wavelength reflected. This is suits to the Bragg condition.

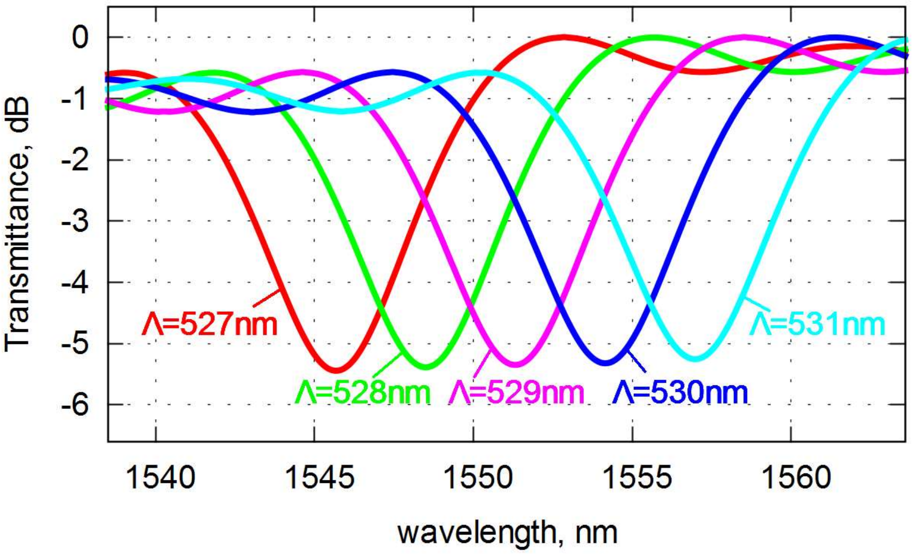

Fig. IV.6. Transmission of 3D FBG with several grating periods Λ.

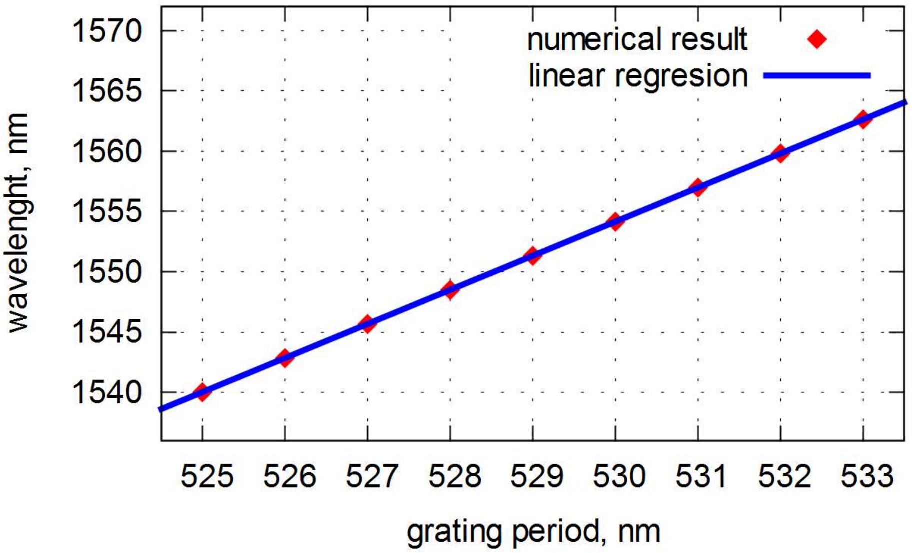

Fig. IV.7. Center of wavelength drop of 3D FBG with several grating periods Λ.

Table IV.5. Center wavelength drop for several grating periods of 3D FBG.

Grating period (in nm)	Wavelength (in nm)
525	1540.02
526	1542.85
527	1545.68
528	1548.51
529	1551.33
530	1554.14
531	1556.99
532	1559.80
533	1562.62

4.3.3 GRATING LENGTH CHANGES

Afterward, several FBG with different grating length are simulated. The grating length that applied to FBG is ranged from 0µm (no grating structure) to 200µm, with the differences of 50µm for each FBG structures simulated. Grating period Λ=529nm and grating refractive index value n_1=1.47 are used.

Fig. IV.8 shows a comparison from some result from the simulation. While Fig. IV.9 and Table IV.6 shows whole data simulation results. It shows that FBG with gratings structure gives similarity in the center of wavelength drop. We can see clearly also that increasing grating total length will affect the lower transmission value with a linear correlation between them.

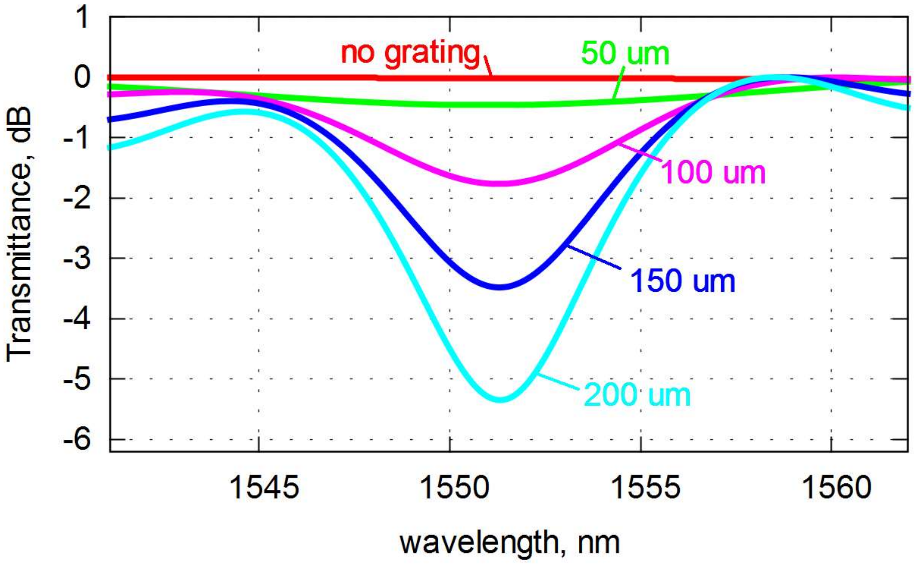

Fig. IV.8. Transmittance for 3D FBG with different grating length.

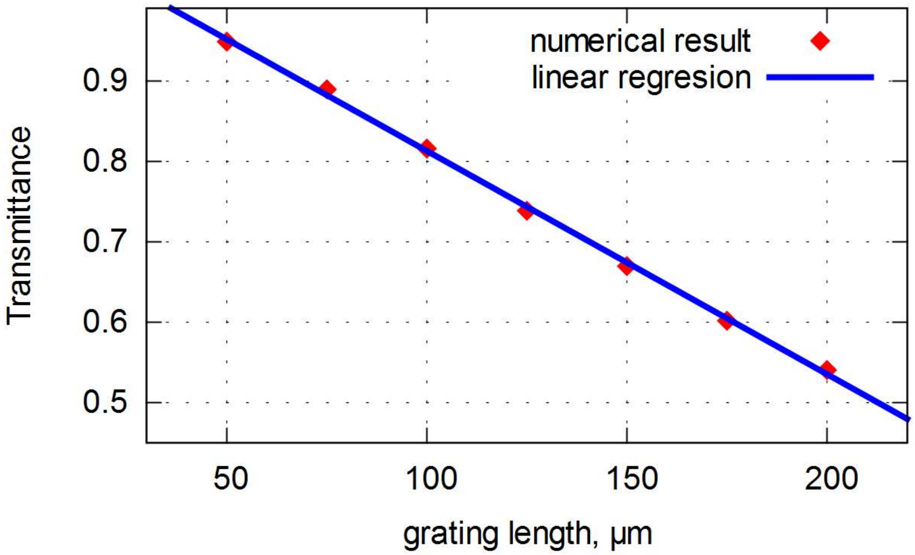

Fig. IV.9. Transmission power drop of 3D FBG with different total grating periods.

Table IV.6. Transmission power drop of 3D FBG with different total grating periods.

Grating length (in μm)	Transmission (in unit)
50	0.9481
75	0.8892
100	0.8158
125	0.7383
150	0.6697
175	0.6020
200	0.5401

4.3.4 CHIRPED FIBER BRAGG GRATING (CFBG)

Chirped Fiber Bragg Grating (CFBG) is an FBG which contain a different grating period. The grating period in the CFBG is gradually changing from the first grating toward the end of the grating structures. We compare CFBG output characteristic with the first grating period of 529nm and the end grating period ranged from 529nm to 535nm with a 2nm difference between simulations. End grating period of 529nm corresponds to uniform FBG structure. The result from the simulation is shown in Fig. IV.10.

In this figure, we can see that different end grating period will generate different output characteristic. If we increase end grating period while still maintaining first grating period, the wider broadband wavelength will be reflected by CFBG, but the reflection power is decreased. This result is reasonable because when the different period of first and last grating become larger, more variation of the grating period will reflect wider wavelength regions. But if total grating structures length remains the same, increasing difference between first and last grating period will reduce the number of the grating period that reflects a specific wavelength, which affects to lower reflectance power.

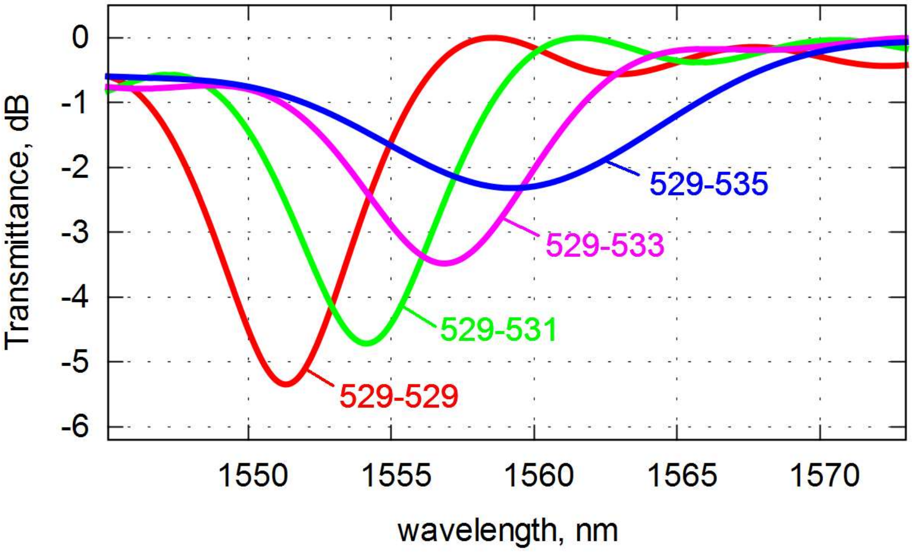

Fig. IV.10. Transmission drop of 3D CFBG with different end grating periods.

4.4 Temperature sensing

Similar to two-dimensional analysis, temperature sensing also calculated in the three-dimensional analysis. Refractive index of the cladding is n_3=1.44, core refractive index n_2=1.46, grating refractive index n_1=1.47, and the sinusoidal type grating refractive index shape applied. The grating period of the structure is set at Λ=530nm. Δ T values are ranged from 0°C to 80°C (T=20°C to 100°C) with 10°C difference in every simulation.

Fig. IV.11 shows 6 different ΔT as the function of wavelength (nm), while all simulation data shown in Fig. IV.12 and Table IV.7.

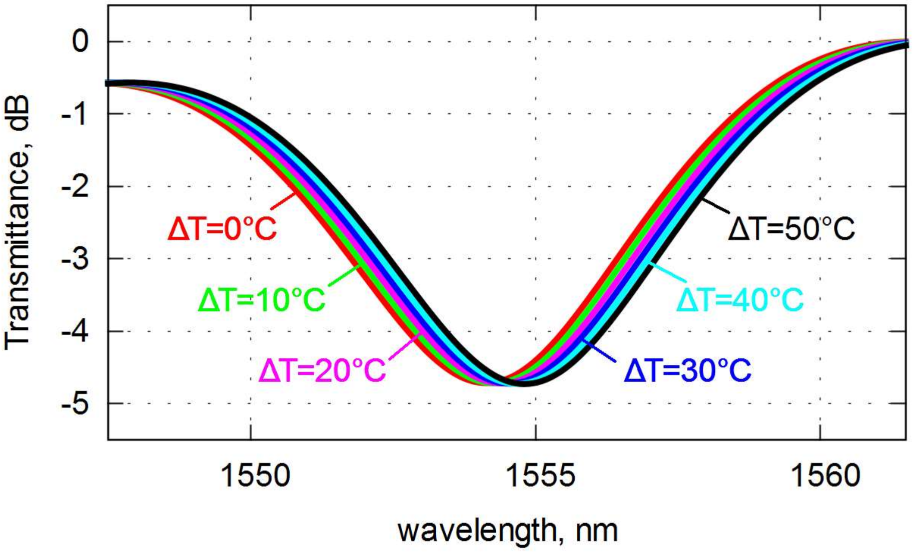

Fig. IV.11. Transmission of 3D FBG with different ΔT.

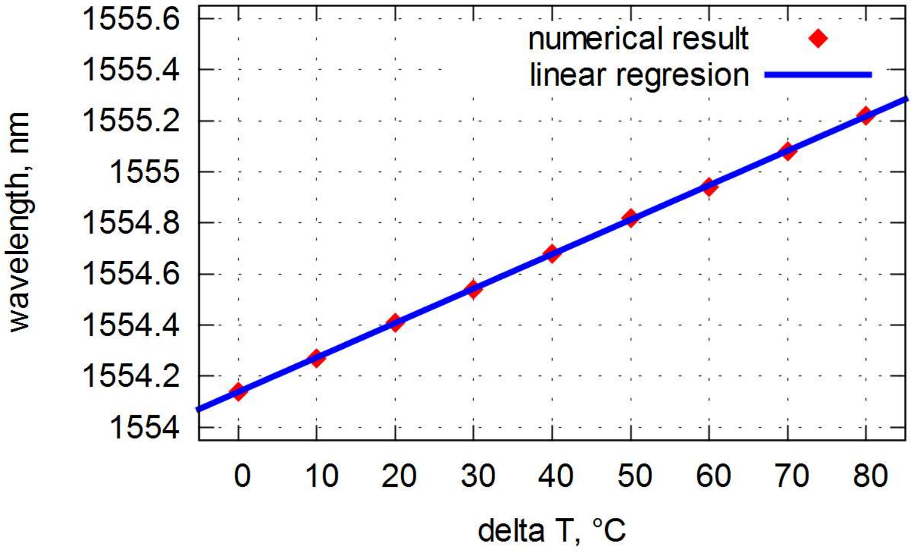

Fig. IV.12. Center of wavelength drop for 3D FBG with different ΔT.

Table IV.7. Center of wavelength drop for different ΔT for 3D FBG.

Delta T (in °C)	Wavelength (in nm)
0	1554.14
10	1554.27
20	1554.41
30	1554.54
40	1554.68
50	1554.82
60	1554.94
70	1555.08
80	1555.22

These results confirm that increasing of the temperature leads to higher value of wavelength reflected with linear correlation occurred.

4.5 CONCLUSION

FBG structures with several parameters change successfully simulated by using FDTD method in the three-dimensional analysis. CPU calculation is to slow to simulate FDTD in a three-dimensional area. Thus, GPU calculation is proposed to get faster simulation time. In terms of memory usage, CPU calculation only uses PC RAM with the similar amount used both single and multi-thread calculation, while GPU calculation uses PC RAM and GPU memory with higher usage on both of them. Comparison of CPU and GPU results show a very similar characteristic between them, with median error only around 0.007%. Furthermore, GPU calculation could reach more than 100 times compared to single thread calculation in 1st PC system. This means that GPU calculation could substitute CPU calculation based on result's similarity with much faster execution time. The performance gain of GPU calculation is affected by several factors, mainly by the specification of GPU and others PC systems components.

Results from FBG parameter changes in three-dimensional calculation have a similar property compared to two-dimensional case. Even the center of reflected wavelength is a bit shifted and the reflection power is different, both systems have a similar property when the parameter of FBG changes, and reflect real FBG characteristics.

If FBG structure that will be simulated could be constructed by using two-dimensional approach, there is no need to use three-dimensional calculations to get its output characteristic. But, when two-dimensional approach can't imitate the proposed structures, the three-dimensional calculation could be proposed to simulate these structures.

Furthermore, even the results examined here are only based on current FBG structures, FDTD program already prepared to simulate FBG structure that not proposed yet. As long as the structures could be modeled by mathematics equation, even when every grating structures are different each other, FDTD program could be used to simulate and evaluate its output characteristics.

V. General conclusion

The major objective of this study is trying to create programs that could simulate Fiber Bragg Grating (FBG) structure both in two-dimensional and three-dimensional case, evaluate output characteristic of FBG from FDTD program, as well as try to find factors that could increase the performance of the simulation. This study employs numerical analysis that used for modeling electromagnetic field. The method that applied to simulate FBG structure here is Finite Difference Time Domain (FDTD) Method.

Results from two-dimensional simulation show that higher grating refractive index value affects in a smaller value of transmission in a specific wavelength, and the center wavelength drop is shifted to a higher value. Increasing grating period will affect reflected wavelength shifting to a higher value with a linear correlation between them. Longer grating length will reduce the transmission value with a linear correlation between them. Chirped FBG (CFBG) shows a weaker reflectance, but with wider broadband wavelength reflected due to more grating period variations contained in FBG. Higher temperature will result in wavelength shift to the higher wavelength value, with a linear correlation between them.

While in three-dimensional case, it is found that CPU calculation is to slow to simulate FDTD in a three-dimensional area. Thus, development is necessary. In this research, GPU calculation is proposed to get faster simulation time. Comparison between CPU and GPU calculation in term of memory usage, CPU calculation only use PC RAM with the similar amount used both single and multi-thread calculation, while GPU calculation uses PC RAM and GPU memory with higher usage on both of them. Comparison of CPU and GPU results show a very similar characteristic between them, with median error only around 0.007%. Furthermore, GPU calculation could reach more than 100 times compared to single thread calculation in 1st PC system. This means that GPU calculation could substitute CPU calculation based on result's similarity with much faster execution time. The performance gain of GPU calculation is affected by several factors, mainly by the specification of GPU and others PC systems components.

Results from FBG parameter changes in three-dimensional calculation have a similar property compared to two-dimensional case. Even the center of reflected wavelength is a bit shifted and the reflection power is different, both systems have a similar property when the parameter of FBG changes, and reflect real FBG characteristics.

Further recalling the design if FBG, with a refractive index of the cladding n_3=1.44, core refractive index is n_2=1.46, grating refractive index n_1=1.47, the effective refractive index is n_{eff}=1.465. With grating period is 529nm, based on Bragg condition, the center of the reflected frequency should around 1.550nm. In the two-dimensional simulations, the simulated FBG have the center reflection of 1560.98nm, while for the three-dimensional case it has center reflection frequency 1551.33nm. This show that three-dimensional simulation has a better accuracy for the center of reflected frequency. The shifting of center reflected frequency in the two-dimensional case could be caused by 2nd order MUR that still reflect around 2% of the incoming electromagnetic field, while the three-dimensional case with its CPML only reflects around 0.5%.

Even the two-dimensional simulation has a slight shift for the center of reflected frequency from the design value, it has a deeper power of the reflection. In the two-dimensional case, the transmitted value is around -11.5db in the center of reflected frequency, while in three-dimensional simulation it has -5.3db. The lower reflection power in the three-dimensional case could be caused by the absorbing boundary condition that lies too close to the core, which only 5 cells. This distance can’t be increased because the area for the simulation is limited by the software.

Based on all of those findings, we summarize the results of this study as follows:

1. Two-dimensional and three-dimensional simulation periodic waveguide has been successfully simulating the model of FBG by using FDTD Method. Both simulations could imitate FBG characteristic based on changes in the design parameter. Basic Optical Fiber Sensor as a temperature sensor also could be imitated well.

2. The two-dimensional case has a slight shifting in the center of reflected frequency from 1550nm of FBG design to 1560.98nm that appear in the simulation result. The shifting could be caused by absorbing boundary condition 2nd order MUR that still have around 2% of reflection.

3. The three-dimensional case has lower reflectance result compared two-dimensional simulation. The transmittance of around -5.3db in the three-dimensional case clearly shows that the reflectance is not as high as the result in the two-dimensional case which has around -11.5db transmittance in the center of reflected frequency. This phenomenon may be caused by the absorbing boundary condition Convolutional Perfectly Matched Layer that lies too close to the fiber core which only has 5 cells distance.

4. Other than slight shifting in the center of reflected frequency, two-dimensional simulation already shows good performance for imitating the output characteristic of FBG that related to design parameter changes. It also has a much less computational requirement for the simulation that results in much faster simulation time. As long as two-dimensional approach could fulfill the need of the design, evaluating FBG design by the two-dimensional case is enough. But, sometimes FBG design could not be fulfilled by two-dimensional simulation, especially complex FBG structures that still not proposed yet. When this problem arises, the three-dimensional simulation could be used to predict the output characteristic.

5. When using three-dimensional case, computation by using CPU will take considerably long time. If the simulation time is considered too long, faster simulation time could be achieved by several methods. The first method that the cheapest one is by using faster memory module, especially the one that have tighter memory timing since FDTD method depends highly on memory bandwidth. This method could increase 7-8% of the performance. The second one is by upgrading PC system to the newer technology since newer technology generally has a faster speed for CPU and memory. Depend on the upgraded system, this method could

increase around 66-88% in this research. The third one is the most expensive and most difficult one, that is a move to GPU computing for much faster simulation time. In this research, performance gain that observed for using GPU computing is ranged between 53-105 times or around 5,300%-10,500% increased.

Future works

Some improvement needs to be developed from the current results, such as:

1. Applying Perfectly Matched Layer type for absorbing boundary condition in the two-dimensional case, since 2nd order MUR ABC still reflect some electromagnetic fields.

2. Re-write 3D FDTD from beginning to overcome limitation from the open source program. With this approach, this research could simulate longer fiber structure, with absorbing boundary condition lies in good distance from the fiber core.

3. Analyze complex or combined FBG structure that not proposed or exist yet. This approach tries to discover new FBG structure that could be found by changing the various parameter of FBG and observe the output characteristic.

4. Include more physical parameter changes in the simulation of FBG as a sensor. Thus, more FBG sensor could be simulated and open the new possibility of a new combination of the sensor functions of FBG.

5. Optimize FDTD CUDA based on latest GPU architecture. GPU architectures are continuously developed. New architecture usually offers more performance gain or have the features increased. This new improvement won’t have utilized maximally if the code did not employ the new features in the new architecture.

Acknowledgment

I am heartily grateful to The Almighty ALLAH SWT, for giving me the opportunity, health, strength, guidance, easiness, and *barakah* in all progress of my study. Without His will, I couldn’t finish this study. I also want to grateful to my parents Hariyadi and Poppy Shofiyah, their infinite love, assistances, and supports lead to ALLAH’s pleasure to me. *Alhamdulillaahi Rabbil ’aalamiin.*

I would like to express my deepest and sincere gratitude to my supervisor, Dr. Mitsuhiro Yokota, Professor of Electrical and Systems Engineering, University of Miyazaki, for providing excellent supervision, valuable insight, encouragement, and generous support all throughout my study period. His patience in guiding this study is very sincere.

I would like to extend my gratitude to the members of dissertation committee: Dr. Koichi Tanno, Professor of Electrical and Systems Engineering, University of Miyazaki; Dr. Ichijo Hodaka, Professor of Environmental Robotics, University of Miyazaki; Dr. Amane Takei, Associate Professor of Electrical and Systems Engineering, University of Miyazaki; and Dr. Osamu Kinoshita, Associate Professor of Agricultural and Environmental Sciences, Faculty of Agriculture, University of Miyazaki, for their valuable discussion, suggestions, review, and insightful comments for my dissertation.

I would like to express my gratitude to Dr. Thi Thi Zin, Professor of Electrical and Systems Engineering, University of Miyazaki, for helping in the submission to one of the journal. I would like to express my gratitude to Dr. Ir. Sholeh Hadi Pramono, M.S., M. Aziz Muslim, S.T., M.T., Ph.D., Panca MudjiRahardjo, S.T., M.T., Ph.D., and all Electrical Engineering lecturers and staffs for their support and assistance to this study.

I am truly grateful to my beloved wife Irawati Putri Cahyani, my children Aisyah Nuur Abidah and Abdullah Abdul Razak, and all my family members for their unconditional understanding, consistent encouragement, and continued moral support.

I also express my gratitude to the Japanese Government especially to the Ministry of Education, Culture, Sports, Science, and Technology (MEXT) for funding my study.

I appreciate the supporting, conducive environment and research facilities at Information and Communication Engineering Laboratory, University of Miyazaki, which is chaired by Dr. Mitsuhiro Yokota. Special thanks are given to lab mates: Reya Tanaka, Tatsuaki Mitsui, Shota Sakakibara, Shota Marushita, Barry Ibrahim, and all of my lab colleagues for helpful and kind support, the great friendship, collaboration, and encouragement during my study.

Last but not least, I also want to thanks to all of the members of Muslim Community Miyazaki for the great brotherhood, friendship, and kinship that warmth the heart and strengthen the faith. Not forgetting also to thanks to all the members of Indonesian Students Community in Miyazaki, their existence become a new family for me when I am away from my families in Indonesia.

References

[1] A. Orthonos and K. Kalli, Fiber Bragg grating fundamental and application in telecommunication and sensing, Norwood: Artech House, 1999.

[2] I. Riant and e. al, "Chirped Fiber Bragg Gratings for WDM Chromatic Dispersion Compensation in Multispan 10-Gb/s Transmission," *IEEE Journal of Selected Topics In Quantum Electronics,* vol. 5, no. 5, pp. 1312-1324, 1999.

[3] P. Singh, "Dispersion compensation in an optical fiber by using chirp grating," *International Journal of Research in Engineering and Technology,* vol. 3, no. 7, pp. 506-508, 2014.

[4] M. M. Werneck, R. C. S. B. Allil, B. A. Ribeiro and F. V. B. d. Nazaré, "A Guide to Fiber Bragg Grating Sensors," in *Current Trends in Short- and Long-period Fiber Gratings*, Dr. Christian Cuadrado-Laborde (Ed.), InTech, DOI: 10.5772/54682, 2013, pp. 1-24.

[5] M. Silva-López, A. Fender, W. N. MacPherson, J. S. Barton and J. D. C. Jones, "Strain and temperature sensitivity of a single-mode polymer optical fiber," *Optics Letters,* vol. 30, no. 23, pp. 3129-3131, 2005.

[6] S. M. Tripathi, A. Kumar, R. K. Varshney, Y. B. P. Kumar, E. Marin and J.-P. Meunier, "Strain and Temperature Sensing Characteristics of Single-Mode–Multimode–Single-Mode Structures," *Journal of Lightwave Technology,* vol. 27, no. 13, pp. 2348-2355, 2009.

[7] H. J. Patrick, G. M. Williams, A. D. Kersey, J. R. Pedrazzani and A. M. Vengsarkar, "Hybrid Fiber Bragg Grating Long Period Fiber Grating Sensor for

Strain/Temperature Discrimination," *IEEE Photonics Technology Letters,* vol. 8, no. 9, pp. 1223-1225, 1996.

[8] V. P. Wnuk, A. Méndez, S. Ferguson and T. Graver, "Process for Mounting and Packaging of Fiber Bragg Grating Strain Sensors for use in Harsh Environment Applications," in *Smart Structures Conference, SPIE*, 2005.

[9] A. Pal, A. Dhar, A. Ghosh, R. Sen, B. Hooda, V. Rastogi, M. Ams, M. Fabian, T. Sun and K. T. V. Grattan, "Sensors for Harsh Environment: Radiation Resistant FBG Sensor System," *Journal of Lightwave Technology,* no. DOI 10.1109/JLT.2016.2598666, 2016.

[10] W. WU, Z. QIN, X. LIU and T. CHEN, "Investigation on low-temperature characteristics of FBG sensors and the technology to enhance sensitivity," *IEEE,* no. 978-1-4244-7113-3/10, 2010.

[11] O. Prakash, J. Kumar, R. Mahakud, S. K. Agrawal, S. K. Dixit and S. V. Nakhe, "Enhanced Temperature (~800 °C) Stability of Type-IIa FBG Written by 255 nm Beam," *IEEE Photonics Technology Letters,* vol. 26, no. 1, pp. 93-95, 2014.

[12] D. Grobnic, S. J. Mihailov, C. W. Smelser and H. Ding, "Sapphire Fiber Bragg Grating Sensor Made Using Femtosecond Laser Radiation for Ultrahigh Temperature Applications," *IEEE Photonics Technology Letters,* vol. 16, no. 11, pp. 2505-2507, 2004.

[13] T. V. A. Tran, Y.-G. Han, Y. J. Lee, S. H. Kim and S. B. Lee, "Performance Enhancement of Long-Distance Simultaneous Measurement of Strain and Temperature Based on a Fiber Raman Laser With an Etched FBG," *IEEE Photonics Technology Letters,* vol. 17, no. 9, pp. 1920-1922, 2005.

[14] W. Huang, W. Zhang, T. Zhen, F. Zhang and F. Li, "π-Phase-Shifted FBG for High-Resolution Static-Strain Measurement Based on Wavelet Threshold Denoising

Algorithm," *Journal of Lightwave Technology,* vol. 32, no. 22, pp. 4294-4300, 2014.

[15] L. Yan, Z. Wu, Z. Zhang, W. Pan, B. Luo and P. Wang, "High-Speed FBG-Based Fiber Sensor Networks for Semidistributed Strain Measurements," *IEEE Photonics Journal,* vol. 5, no. 5, 2013.

[16] Q. Chen and P. Lu, "Fiber Bragg Gratings and Their Applications as Temperature and Humidity Sensors," in *Atomic, Molecular and Optical Physics*, Nova Science Publishers, Inc., 2008, pp. 235-260.

[17] J. M. Corres, F. J. Arregui and I. R. Matias, "Design of Humidity Sensors Based on Tapered Optical Fibers," *Journal of Lightwave Technology,* vol. 24, no. 11, pp. 4329-4336, 2006.

[18] Y. Miao, B. Liu, H. Zhang, Y. Li, H. Zhou, H. Sun, W. Zhang and Q. Zhao, "Relative Humidity Sensor Based on Tilted Fiber Bragg Grating With Polyvinyl Alcohol Coating," *IEEE Photonics Technology Letters,* vol. 21, no. 7, pp. 441-443, 2009.

[19] L. S. M. Alwis, H. Bustamante, B. Roth, K. Bremer, T. Sun and K. T. V. Grattan, "Evaluation of the durability and performance of FBG-based sensors for monitoring moisture in an aggressive gaseous waste sewer environment," *Journal of Lightwave Technology,* no. DOI 10.1109/JLT.2016.2593260, 2016.

[20] Q. Haifeng, L. Tongyu, W. Chang, W. Jinyu and L. Xiaohui, "High sensitive multiplexed FBG micro-seismic monitoring system," *IEEE,* no. 978-1-4244-6554-5/11, 2011.

[21] A. Khare and J. Singh, "Design and Study of Chirped Fiber Bragg Grating for Sensing of Hazardous Gases," *International Journal of Computer Application,* vol. 23, no. 9, pp. 40-43, 2011.

[22] T. Guo, H.-Y. Tam, P. A. Krug and J. Albert, "Reflective Tilted Fiber Bragg Grating Refractometer Based on Dtrong Cladding to Core Recoupling," *Optics Express,* vol. 17, no. 7, 2009.

[23] A. Sun and Z. Wu, "A hybrid LPG/CFBG for highly sensitive refractive index measurement," *Sensor-Open Access Journal,* pp. 7318-7325, 2012.

[24] L.-Y. Shao, J. Liang, X. Zhang, W. Pan and L. Yan, "High-Resolution Refractive Index Sensing With Dual-Wavelength Fiber Laser," *IEEE Sensors Journal,* vol. 16, no. 23, pp. 8463-8467, 2016.

[25] Y.-F. Gu, Y. Zhao, R.-Q. Lv and Y. Yang, "A Practical FBG Sensor Based on a Thin-Walled Cylinder for Hydraulic Pressure Measurement," *IEEE Photonics Technology Letters,* vol. 28, no. 22, pp. 2569-2572, 2016.

[26] S. Man, Z. Yong, L. Yong and L. Haowu, "Detection of Interface damage of composite structure using FBG sensing," *IEEE,* no. 978-1-4244-6554-5/11, 2011.

[27] W. Zhang, G. Kai, X. Dong, S. Yuan and Q. Zhao, "Temperature-Independent FBG-Type Torsion Sensor Based on Combinatorial Torsion Beam," *IEEE Photonics Technology Letters,* vol. 14, no. 8, pp. 1154-1156, 2002.

[28] R. Kasyap, Fiber Bragg Grating 2nd Ed, Burlington: Elsevier, 2010.

[29] A. Tavlove, Computational electrodynamics, Norwood, USA: Artech House, 1995.

[30] T. Kudou, K. Shimizu, Y. Takimoto and T. Ozeki, "Bragg grating filter synthesis using Fourier transform with iteration," *IEEE Trans. Electron,* Vols. E83-C, no. 6, pp. 898-902, 2000.

[31] T. Miyamoto, M. Momoda and K. Yasumoto, "Numerical analysis for 3-dimentional optical waveguide with periodic structure using Fourier series

expansion method," *IEICE Trans. Japan (section J),* Vols. J86-C, no. 6, pp. 591-600, 2003.

[32] S. Adams, J. Payne and R. Boppana, "Finite Difference Time Domain (FDTD) Simulations Using Graphics Processors," in *HPCMP Users Group Conference*, 2007.

[33] P. Klapetek and M. Valtr, "Near-field optical microscopy simulations using graphics processing units," *Surf. Interface Analysis,* vol. 42, pp. 1109-1113, 2010.

[34] P. Klapetek and e. al., "Rough surface scattering simulations using graphics cards," *Applied Surface Science,* vol. 256, pp. 5640-5643, 2010.

[35] K. O. Hill, Y. Fujii, D. C. Johnson and B. S. Kawazaki, "Photosensitivity in optical fiber waveguides: application to reflection fiber fabrication," *Applied Physics Letters,* vol. 32, no. 10, pp. 647-649, 1978.

[36] K. O. Hill, "Photosensitivity in Optical Fiber Waveguides: From Discovery to Commercialization," *IEEE Journal on Selected Topics in Quantum Electronics,* vol. 6, no. 6, pp. 1186-1189, 2000.

[37] K. S. Yee, "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media," *IEEE Transactions on Antennas and Propagation,* pp. 302-307, 1966.

[38] D. M. Sullivan, Electromagnetic Simulation Using the FDTD Method, New York: IEEE, 2000.

[39] P. Wilmott, S. Howison and J. Dewynne, The Mathematics of Financial Derivatives: A Student Introduction, New York: Cambridge University Press, 1995.

[40] P. J. Oliver, Introduction to Partial Differential Equations, Springer Science + Business Media, 2014.

[41] M. H. Chaundhry, Open-Channel Flow Second Edition, New York: Springer Science+Business Media, 2008.

[42] J. Sanders and E. Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Boston: Addison-Wesley, 2011.

[43] D. B. Kirk and W.-m. W. Hwu, Programming Massively Parallel Processors, Elsevier, 2010.

[44] G. Mur, "Absorbing Boundary Conditions for the Finite-Difference Approximation of the Time-Domain Electromagnetic-Field Equations," *IEEE Transactions on Electromagnetic Compability,* Vols. EMC-23, no. 4, pp. 377-382, 1981.

[45] I. H. Malitson, "Interspecimen Comparison of the Refractive Index of Fused Silica," *Journal of the Optical Society of America,* vol. 55, no. 10, pp. 1205-1209, 1965.

[46] J. A. Roden and S. D. Gedney, "Convolutional 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.