Avances 17-04-2017

Carlos Pérez - 103753

Manuel Ríos - 159284

Background Teórico

Las ecuaciones de Maxwell están constituidas por un conjunto de ecuaciones diferenciales parciales, las cuales junto con una ley de fuerza de Lorentz, conforman el fundamento del electromagnetismo y óptica clásicos así como de los circuitos eléctricos.

Las ecuaciones son nombradas en honor al físico y matemático James Clerk Maxwell, quien entre 1861 y 1862 publicó las ecuaciones así como una proposición de que la luz es un fenómenos electromagnético.

$$\nabla \cdot \mathbf{E} = 0 \qquad \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t},$$
$$\nabla \cdot \mathbf{B} = 0 \qquad \nabla \times \mathbf{B} = \frac{1}{c^2}\frac{\partial \mathbf{E}}{\partial t}$$

o bien, utilizando ciertas identidades, en su forma de ecuación de onda se tiene que

$$\frac{1}{c^2}\frac{\partial^2 \mathbf{E}}{\partial t^2} - \nabla^2 \mathbf{E} = 0$$
$$\frac{1}{c^2}\frac{\partial^2 \mathbf{B}}{\partial t^2} - \nabla^2 \mathbf{B} = 0$$

La formulación considerada en CUDA está fundamentada en actualizar las ecuaciones para propiedades anisotropicas de materiales en las se que se incluyen permisividad, permeabilidad y conductividades eléctrica y magnética. El dominio para el problema FDTD es una celda, referida en la literatura como la celda de Yee, como se muestra a continuación.

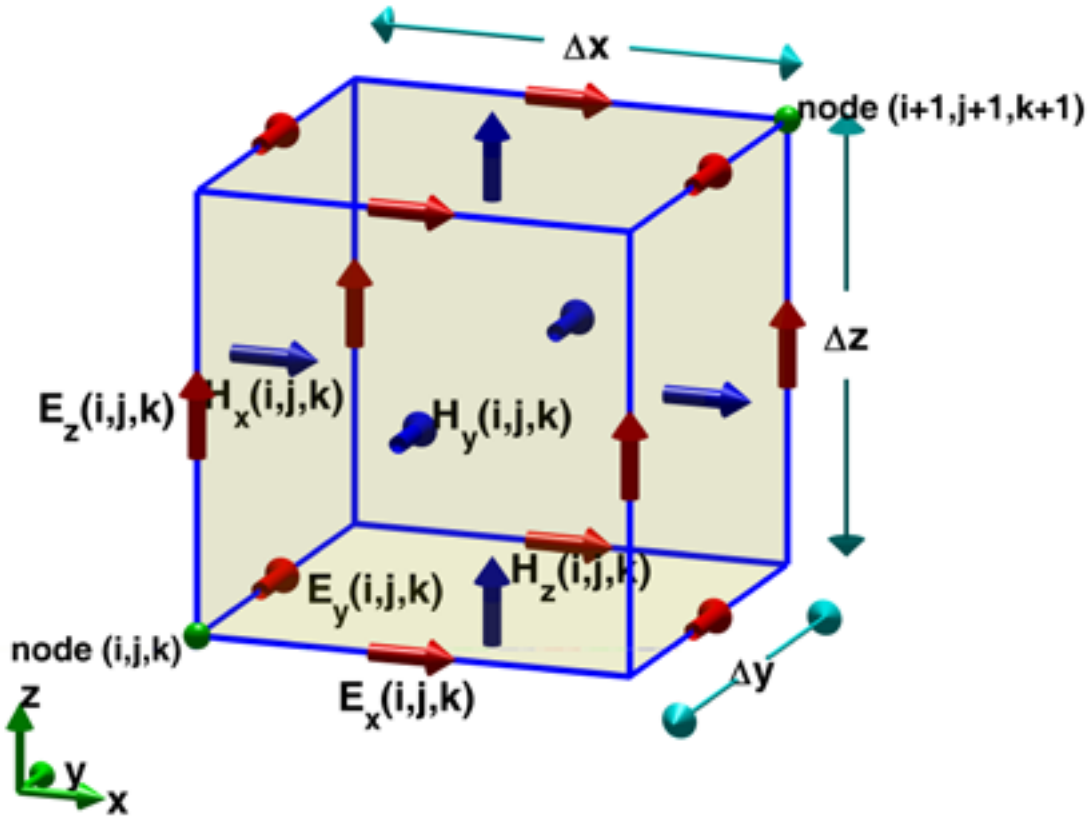

Figure 1: Celda de Yee

Código BASE

A continuación llevamos a cabo la descripción del código de los archivos integrados en la carpeta de entrega el Lunes 17 de Abril. Parte de este cdigo viene de Nvidia.

Descripción de la carpeta:

- inc
  - FDTD3d.h
  - FDTD3dGPU.h
  - FDTD3dGPUKernel.cuh
  - FDTD3dReference.h
- src
  - FDTD3d.cpp
  - FDTD3dGPU.cu
  - FDTD3dReference.cpp
- FDTD3d.txt
- Makefile
- NsightEclipse.xml
- readme.txt

INC

En la carpeta **inc** incluimos todos los archivos a incluir en el codigo de C. Principalmente *header files* tanto para el caso paralelo como para el no-paralelo. Estos archivos despues seran "incluidos" `#include` en las partes "centrales" del codigo de C.

FDTD3d.h

Header file. Definimos las variables a usar para el caso no paralelo.

```
#ifndef _FDTD3D_H_
#define _FDTD3D_H_
```

Definimos las dimensiones minimas y maximas de las matrices. Estos se pueden ajustar pero cuando son operaciones de grandes dimensiones puede tomar muchisimo tiempo en correr.

```
#define k_dim_min           96
#define k_dim_max           376
#define k_dim_qa            248
```

Definimos el radio que usara el kernel, lo definimos como 4 ya que se necesita una constante. Si se ajusta este variable se debe de hacer su respectivo ajuste en el kernel.

```
#define k_radius_min        4
#define k_radius_max        4
#define k_radius_default    4
```

```
#define k_timesteps_min     1
#define k_timesteps_max     10
#define k_timesteps_default 5
```

FDTD3dGPU.h

Header file. En esta parte definimos el codigo para el caso paralelo.

```
#ifndef _FDTD3DGPU_H_
#define _FDTD3DGPU_H_
```

```
#include <cstddef>
#if defined(WIN32) || defined(_WIN32) || defined(WIN64) || defined(_WIN64) && defined(_MSC_VER)
typedef unsigned __int64 memsize_t;
#else
#include <stdint.h>
typedef uint64_t memsize_t;
#endif
```

```
#define k_blockDimX    32
#define k_blockDimMaxY 16
#define k_blockSizeMin 128
#define k_blockSizeMax (k_blockDimX * k_blockDimMaxY)
```

Definimos todas las variables usadas para el caso paralelo. Como el radio, las 3 dimensiones a usar, etc.

```
bool getTargetDeviceGlobalMemSize(memsize_t *result, const int argc, const char **argv);
bool fdtdGPU(float *output, const float *input, const float *coeff, const int dimx, const int dimy, cons
```

FDTD3dGPUKernel.cuh

Header file de cuda. Definimos las variables a usar en el kernel de CUDA:

FDTD3dGPUReference.h

Header file. Declaramos todas las variables a usar en partes posteriores del codigo.

```
void generateRandomData(float *data, const int dimx, const int dimy, const int dimz, const float lowerB
void generatePatternData(float *data, const int dimx, const int dimy, const int dimz, const float lower
bool fdtdReference(float *output, const float *input, const float *coeff, const int dimx, const int dim
bool compareData(const float *output, const float *reference, const int dimx, const int dimy, const int
```

SRC

En esta parte incluimos el *source code.* Esta es la parte "central" del programa.

FDTD3d.cpp

*Codigo para implemententacion de FDTD. No-paralelo.

FDTD3dGPU.cpp

*Codigo para implemententacion de FDTD usando GPU. Modo paralelo.

FDTD3dGPUReference.cpp

Definicion de variables a usar en el codigo de paralelo.

Makefile

Makefile para la compilacion del program. Incluye las partes del codigo de CUDA.

NsightEclipse.xml

Project file. Contiene la informacion acerca del proyecto.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE entry SYSTEM "SamplesInfo.dtd">
<entry>
  <name>FDTD3d</name>
  <description><![CDATA[This sample applies a finite differences time domain progression stencil on a 3
```

```
<devicecompilation>whole</devicecompilation>
<includepaths>
  <path>inc</path>
  <path>./</path>
  <path>../</path>
  <path>../../common/inc</path>
</includepaths>
<keyconcepts>
  <concept level="advanced">Performance Strategies</concept>
</keyconcepts>
<keywords>
  <keyword>GPGPU</keyword>
  <keyword>CUDA</keyword>
  <keyword>finite difference</keyword>
  <keyword>fdtd</keyword>
  <keyword>differential equation</keyword>
  <keyword>pde</keyword>
  <keyword>ode</keyword>
</keywords>
<libraries>
</libraries>
<librarypaths>
</librarypaths>
<nsight_eclipse>true</nsight_eclipse>
<primary_file>FDTD3d.cpp</primary_file>
<scopes>
  <scope>1:CUDA Advanced Topics</scope>
  <scope>1:Performance Strategies</scope>
</scopes>
<sm-arch>sm20</sm-arch>
<sm-arch>sm30</sm-arch>
<sm-arch>sm35</sm-arch>
<sm-arch>sm37</sm-arch>
<sm-arch>sm50</sm-arch>
<sm-arch>sm52</sm-arch>
<sm-arch>sm60</sm-arch>
<supported_envs>
  <env>
    <arch>x86_64</arch>
    <platform>linux</platform>
  </env>
  <env>
    <platform>windows7</platform>
  </env>
  <env>
    <arch>x86_64</arch>
    <platform>macosx</platform>
  </env>
  <env>
    <arch>arm</arch>
  </env>
  <env>
    <arch>ppc64le</arch>
```

```
      <platform>linux</platform>
    </env>
  </supported_envs>
  <supported_sm_architectures>
    <include>all</include>
  </supported_sm_architectures>
  <title>CUDA C 3D FDTD</title>
  <type>exe</type>
</entry>
```