Module Code
COM00174M

MSc and MEng Degree Examinations 2021–2

DEPARTMENT OF COMPUTER SCIENCE

High-Performance Parallel and Distributed Systems

Open Individual Assessment

Issued: 9th March 2022, 12:00 noon
Submission due: 20th April 2022, 12:00 noon
Feedback and marks due: 25th May 2022, 12:00 noon

All students should submit their answers through the electronic submission system: http://www.cs.york.ac.uk/student/assessment/submit/ by 20th April 2022, 12:00 noon. An assessment that has been submitted after this deadline will be marked initially as if it had been handed in on time, but the Board of Examiners will normally apply a lateness penalty.

Your attention is drawn to the section about Academic Misconduct in your Departmental Handbook: https://www.cs.york.ac.uk/student/handbook/.

Any queries on this assessment should be addressed by email to Steven Wright at steven.wright@york.ac.uk. Answers that apply to all students will be posted on the VLE.

Rubric:

Carry out the whole task as described in the following pages. Your report must not exceed 8 sides of A4, with a minimum 11pt font, minimum 120% line spacing (what Word calls “Multiple 1.08”), and minimum 2cm margins on all sides. This does not include any covering page, table of contents, reference list or appendices. Excess pages will not be marked.

References must be listed at the end of the document and do not count towards page limits. Appendices may be used for supplementary data only.

Your exam number should be on the front cover of your assessment. You should not be otherwise identified anywhere on your submission.

The Problem

Maxwell's equations are a set of coupled partial differential equations (PDEs), that together with the Lorentz force law, form the foundation of classical electromagnetism. The equations describe how electric and magnetic fields are generated by charges, currents and changes in the fields. The four equations are:

$$\nabla \cdot \vec{E} = \frac{\rho}{\epsilon} \qquad \text{(Gauss's law)}$$

$$\nabla \cdot \vec{B} = 0 \qquad \text{(Gauss's law for magnetism)}$$

$$\frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E} \qquad \text{(Faraday's law of induction)}$$

$$\frac{\partial \vec{E}}{\partial t} = \frac{1}{\mu\epsilon}\nabla \times \vec{B} - \frac{1}{\epsilon}\vec{J} \qquad \text{(Ampère's law)}$$

In these equations, $\vec{E}$ is the electric field, $\vec{B}$ is the magnetic field, ϵ and μ are the permittivity and permeability of free space, ρ is the charge density and $\vec{J}$ is the current density. **Luckily, it is not necessary that you understand these equations for this assessment.** This assessment will focus on a method for solving these equations using finite differencing that was proposed by Kane Yee in 1966 [1].

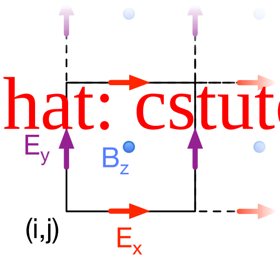

Figure 1: A two-dimensional Yee grid

Yee's approach to solving Maxwell's equations relies first on using central-difference approximations to the space and time partial derivatives, and then on using leapfrog integration to solve these equations. First, the space is discretised (in our case, in two-dimensions) into a grid of cells (a so-called "Yee grid"), as seen in Figure 1, where the electric fields are calculated on the edges of each grid square, and the magnetic fields are calculated in the middle of each grid square[1]. Then, Faraday's law and Ampère's law are rewritten into the following system of scalar

[1]You may notice that the electric field (E) has an x and a y component, while the magnetic field (B) has only a z component. This is because an electric field flowing in the x and y direction will create a magnetic field in the z direction (as per the "right-hand rule").

equations (with current density ($\vec{J}$) omitted for simplicity):

$$\frac{\partial \vec{B}_z}{\partial t} = \frac{\partial \vec{E}_x}{\partial y} - \frac{\partial \vec{E}_y}{\partial x}$$
$$\frac{\partial \vec{E}_x}{\partial t} = \frac{\partial B_z}{\partial y} \times \frac{1}{\mu \times \epsilon}$$
$$\frac{\partial \vec{E}_y}{\partial t} = -\frac{\partial \vec{B}_z}{\partial x} \times \frac{1}{\mu \times \epsilon}$$

These equations can then be written into a series of finite difference equations that can be solved computationally using a leapfrog method. In the leapfrog method, we update the magnetic field on a half time step, and then update the electric field on each time step (i.e. each half time step, the magnetic field and the electric field updates “leap frog” each other).

$$\frac{B_z^{n+\frac{1}{2}}(i+\frac{1}{2},j+\frac{1}{2}) - B_z^{n-\frac{1}{2}}(i+\frac{1}{2},j+\frac{1}{2})}{\Delta t} = \frac{E_x^n(i+\frac{1}{2},j+1) - E_x^n(i+\frac{1}{2},j)}{\Delta y} - \frac{E_y^n(i+1,j+\frac{1}{2}) - E_y^n(i,j+\frac{1}{2})}{\Delta x}$$

$$\frac{E_x^{n+1}(i+\frac{1}{2},j) - E_x^n(i+\frac{1}{2},j)}{\Delta t} = \frac{B_z^{n+\frac{1}{2}}(i+\frac{1}{2},j+\frac{1}{2}) - B_z^{n+\frac{1}{2}}(i+\frac{1}{2},j-\frac{1}{2})}{\Delta y \times \mu \times \epsilon}$$

$$\frac{E_y^{n+1}(i,j+\frac{1}{2}) - E_y^n(i,j+\frac{1}{2})}{\Delta t} = -\frac{B_z^{n+\frac{1}{2}}(i+\frac{1}{2},j+\frac{1}{2}) - B_z^{n+\frac{1}{2}}(i-\frac{1}{2},j+\frac{1}{2})}{\Delta x \times \mu \times \epsilon}$$

This approach to solving Maxwell’s equations is formalised in the `maxwell` C application that can be downloaded from the VLE. The application implements the above equations on a rectilinear grid, and provides functionality to start the application for various simple problems.

Download the source code and build the application using:

```
$ make
```

You can then run the application with:

```
$ ./maxwell
```

At the end of execution, a VTK file will be written that can be loaded into a visualisation application such as VisIt (https://visit-dav.github.io/visit-website/). The output file will contain information about the geometry of the problem, and the values of the electric and magnetic fields at the grid points (i.e. the intersections of the grid). The final simulation state will be based upon the default configuration options, but you can customise the parameters by changing the runtime arguments passed to the application.

```
$ ./maxwell --help
```

The `maxwell` application consists of 5 C files and 4 associated header files.

`args.c` contains functionality to handle the command line options, and can override some simulation parameters based on these options.

`data.c` contains the definitions for the shared storage and simulation parameters that are used in the application, and also functions to allocate addressable, contiguous 2D and 3D arrays.

`setup.c` has functionality to set up a simulation based on the input problem that is chosen, and assigns and allocates the required variables and storage.

`vtk.c` handles the file input and output for the application.

`maxwell.c` contains the `main` method, and the methods required to progress the simulation.

References

[1] Kane Yee, "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media" in: IEEE Transactions on Antennas and Propagation; 1966

Assignment

The `maxwell` application is currently a simple single-threaded implementation of the Yee method described above. Your task is to produce **three** parallelisations of the `maxwell` application and write a report detailing the parallelisation process, the scaling performance of your three parallelisations, and a comparative study of these parallelisations.

You are expected to produce parallelisations using the following three programming models:

- OpenMP
- The Message Passing Interface (MPI)
- CUDA

Your parallelisations should produce the same result (within some small tolerance due to floating point arithmetic) as the original single-threaded application for equivalent starting parameters. You are expected to fully evaluate your applications using the HPC hardware that is available within the University (e.g. CUDA-capable workstations within the Department, the Viking cluster, etc.), and that is appropriate for each particular programming model.

Submission

Your submission should be a **.zip** file that will contain your three application "ports" and a report that documents:

- **Parallelisation Approach** [30 marks]
  The approach taken to parallelise the application for each programming model.
- **Validation** [15 marks]
  The validation process used to verify the correctness of your applications.
- **Experimental Setup** [15 marks]
  A summary of the systems used for performance evaluation and an account of the process of collecting results.
- **Performance Evaluation** [20 marks]
  Appropriate data demonstrating the performance and scaling behaviour of your applications.
- **Comparative Analysis** [20 marks]
  A comparative analysis of your three applications.

Your report must not exceed 8 sides of A4, with a minimum 11pt font, minimum 120% line spacing (what Word calls "Multiple 1.08"), and minimum 2cm margins on all sides. This does not include any covering page, table of contents, reference list or appendices. Excess pages will not be marked.

References must be listed at the end of the document and do not count towards page limits. Appendices may be used for supplementary data only.

End of examination paper