ENEL464 Embedded Software and Advanced Computing 2019
Group Assignment 2

Michael Hayes

1 Introduction

The purpose of this assignment is to implement a numerical algorithm on a desktop computer to make the most efficient use of its caches and multiple cores.

The algorithm is Jacobi relaxation. This is an iterative algorithm used to approximate differential equations, for example, Poisson's and Laplace's equations. Poisson's equation can be used to find the electric potential given a specified charge distribution or temperature given a specified heat source. There are more efficient ways to solve this problem using Green's functions and Fourier transforms but that is not the purpose of this assignment.

2 Jacobi relaxation

The discrete form of Poisson's equation,

$$\nabla^2 V_{i,j,k,n} = f_{i,j,k,n}, \tag{1}$$

can be solved iteratively using Jacobi relaxation, where

$$V_{i,j,k,n+1} = \frac{1}{6}\left(V_{i+1,j,k,n} + V_{i-1,j,k,n} + V_{i,j+1,k,n} + V_{i,j-1,k,n} + V_{i,j,k+1,n} + V_{i,j,k-1,n} - \Delta^2 f_{i,j,k}\right). \tag{2}$$

Here $\Delta = \Delta x = \Delta y = \Delta z$ is the spacing between voxels in metres, $0 \leq i < N$, $0 \leq j < N$, and $0 \leq k < N$. Voxels on the boundary ($i = -1$, $i = N$, $j = -1$, $j = N$, $k = -1$, $k = N$) have a value V_{bound}. This is a Dirichlet boundary condition, equivalent to enclosing the problem in a metal box at potential V_{bound}.

3 Implementation

Implement an algorithm for Jacobi relaxation in either C or C++. Your goal is to find a fast implementation that will run on a CAE lab computer, making best use of the caches and multiple cores.

Your implementation needs to work with an arbitrary 3-D source distribution f. The potential V needs to be stored as a double data type.

4 Testing

Test your algorithm with a single point charge in the centre of the volume, i.e.,

$$f_{i,j,k} = \begin{cases} 1 & i = N/2, j = N/2, k = N/2, \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where the volume is comprised of $N \times N \times N$ voxels. Note, you algorithm must work with arbitrary source distributions.

5 Support

Only questions submitted via the ENCE464 assignment forum will be answered. Emails will be quietly ignored.

6 Reports

The reports are to be submitted as PDF documents through the ENCE464 Learn page. They will be submitted to TurnItIn for plagiarism checking.

Guidelines for writing a report are available at
`https://eng-git.canterbury.ac.nz/mph/report-guidelines/blob/master/report-guidelines.pdf`.

Each report is to use a 12 point font and be no longer than four pages, including appendices.

Your report will be marked in terms of:

- Written style
- Presentation
- Cache analysis
- Multithreading analysis
- Profiling analysis
- Optimisation analysis
- Overall excellence

Your report should present the average time and standard deviation of the time to run 500 iterations of your implementation for $N = 101, 201, 301, 401, 501, 601, 701, 801, 901$.

7 Code

You must submit your code as a `.zip` file so it can be checked for numerical accuracy and plagiarism. Your fastest implementation must be able to built by running `make`.