FYS4150 - Project 1: Differential Equations and Linear Algebra

Tore Klungland (torekl)

August 27, 2019

1 Introduction

Poisson's equation, given in general as

$$\nabla^2 u(\mathbf{r}) = f(\mathbf{r}) \tag{1}$$

where $f(\mathbf{r})$ is some known function, is an often-used equation in physics as it describes how a field (in this case u) is affected by some source (in this case f). If f is a complicated function this equation may not be analytically solvable, so one must use numerical methods methods instead, to find an approximate solution. In one dimension this can be done by using an approximate expression for the second derivative of u at discrete set of points; then equation 1 becomes a set of linear equations for the values of u at these points, which can be expressed as a matrix involving a tridiagonal matrix. Thus the problem of solving Poisson's equation numerically reduces to inverting such an equation numerically.

In this project I have examined three different methods for solving this matrix equation: One "brute-force" method which inverts the entire matrix numerically, and two which exploit the fact that the matrix is tridiagonal by storing data in vectors instead of a matrix, which reduces memory usage. One of these uses a general algorithm for row-reducing tridiagonal matrices, while the other takes advantage of the specific form of the matrix in question. In this text I will first outline these methods, and then compare their results, in particular the difference in CPU time used. I will also discuss the effects of round-off errors as the step size between the different data points is changed.

2 Methods

2.1 Turning the Poisson equation into a matrix equation

The target of this project was solving Poisson's equation,

$$u''(x) = -f(x), \tag{2}$$

for $f(x) = 100e^{-10x}$ for $x \in [0, 1]$ with the boundary conditions $u(0) = u(1) = 0$. This equation has an exact solution given by [1]

$$u(x) = \left(e^{-10} - 1\right)x + 1 - e^{-10x} \tag{3}$$

The first step is dividing this interval into a discrete set of points x_i for $i = 0, 1, 2, \ldots, n+1$, where $x_0 = 0$ and $x_{n+1} = 1$. Thus the step size is given by $h = x_{i+1} - x_i = 1/(n+1)$. The second derivative can be approximated by [2]

$$u_i'' \approx \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} \tag{4}$$

(obtained by manipulating the Taylor expansion of $u(x)$) where the error is of order h^2. Here $u_i \equiv u(x_i)$, etc. Inserting this into equation 2 gives

$$-u_{i-1} + 2u_i - u_{i+1} = g_i$$

where I have defined $g_i \equiv h^2 f(x_i)$. This is a set of linear equations for u_i; it can thus be rewritten on matrix form as (using the fact that $u_0 = u_{n+1} = 0$, by the boundary conditions):

$$A\mathbf{u} = \mathbf{g} \tag{5}$$

Here A is a matrix whose non-zero elements on the i'th row are $A_{i,i-1} = -1$, $A_{i,i} = 2$, $A_{i,i+1} = -1$ (except the first row which only has $A_{0,0} = 2$ and $A_{0,1} = -1$, and the n'th row which only has $A_{n,n-1} = -1$ and $A_{n,n} = 2$). Thus it is a tridiagonal matrix with 2 on the diagonal and -1 on the elements directly adjacent to the diagonal. Furthermore $\mathbf{u}^T \equiv [u_1, u_2, \ldots, u_n]$, and $\mathbf{g}^T \equiv [g_1, g_2, \ldots, g_n]$.

References

[1] Assignment text for project 1 in FYS3150/FYS4150, Dept. of Physics, University of Oslo, Norway

[2] Hjorth-Jensen, Morten, 2015, Computational Physics - lecture notes