Project 1

Erlend Lima[1], Frederik J. Mellbye[2], and Aram Salihi[3]

[1,2,3]Oslo University

August 31, 2017

Abstract

In this project we investigate the speed and numerical precision of several methods of solving the one-dimensional Poisson equation with Dirchelet boundary conditions. The equation is discretized and written as a system of linear equations. This equates to a tridiagonal matrix equation, which is solved using the general Thomas algorithm, a specialized version of the Thomas algorithm and by LU-decomposing the matrix.

The specialization of the algorithm for our specific problem resulted in a x percent reduction in computation time.

Table 1: Summary of the errors

n	log error
10	-0.647811
100	-1.69284
1000	-2.69923
10000	-3.69988
100000	-4.69994
1000000	-5.9374

Figure 1: Caption

1 Theory

1.1 Discretizing the Poisson equation

The one-dimensional Poisson equation with Dirchelet boundary conditions is

$$\frac{d^2u}{dx^2} = -f$$

We can approximate the second derivative of $u(x)$ using a Taylor expansion and solving for $\frac{d^2u}{dx^2}$:

$$\frac{d^2u}{dx^2} \approx \frac{u(x+h) + u(x-h) - 2u(x)}{h^2} + O(h^2)$$

We now discretize the equation with grid points $x_1, x_2, \ldots, x_n$. Imposing Dirchelet boundary conditions forces $x_1 = x_n = 0$, using the handy notation $u(x_i + h) = u_{i+1}$ and inserting the approximated second derivative into the Poisson equation yields

$$\frac{u_{i+1} + u_{i-1} - 2u_i}{h^2} = -f_i$$

where $h = \frac{1}{n+1}$ is the step size, i.e. the distance between grid points, and $f_i = f(x_i)$. The above equation is a linear system of equations, and can therefore be written as the matrix equation

$$A\mathbf{u} = -h^2\mathbf{f}$$

where

$$A = \begin{bmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & & \vdots \\ 0 & -1 & 2 & -1 & 0 & \\ \vdots & 0 & -1 & 2 & -1 & 0 \\ & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & -1 & 2 \end{bmatrix}$$

is a $n \times n$ tridiagonal matrix, the only non-zero elements are on, directly above or directly below the diagonal. Row reducing the matrix can therefore be done in an efficient way using the Thomas algorithm, as we shall see in the next paragraph.

1.2 Solving a general tridiagonal matrix problem

A general tridiagonal matrix equation is on the form

$$\begin{bmatrix} b_1 & c_1 & 0 & \dots & \dots & 0 \\ a_2 & b_2 & c_2 & 0 & & \vdots \\ 0 & a_3 & b_3 & c_3 & 0 & \\ \vdots & 0 & a_4 & b_4 & c_4 & 0 \\ & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ \\ u_n \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ \\ v_n \end{bmatrix}$$

where $v_i = h^2 f_i$ in the case examined in this project. To get the matrix in row reduced echelon form and solve the problem, we first eliminate the a_i's. This is done by subtracting $\frac{a_2}{b_1}$ times the first row from the second. We now have

$$\begin{bmatrix} b_1 & c_1 & 0 & \dots & \dots & 0 \\ 0 & b_2' & c_2 & 0 & & \vdots \\ 0 & a_3 & b_3 & c_3 & 0 & \\ \vdots & 0 & a_4 & b_4 & c_4 & 0 \\ & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ \\ u_n \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2' \\ v_3 \\ \vdots \\ \\ v_n \end{bmatrix}$$

where $b_2' = b_2 - \frac{a_2}{b_1} c_1$ and $v_2' = v_2 - \frac{a_2}{b_1} v_1$. We now repeat this operation for every row (i.e. subtract $\frac{a_i}{b_{i-1}'}$ times row $i-1$ from row i), which eliminates the a_i's. A general expression for the diagonal elements is therefore

$$b_i' = b_i - \frac{a_i}{b_{i-1}'} c_{i-1}$$

where $b_1' = b_1$ and $i = 2, 3, ..., n$. Similarly, elements in the rhs vector $\mathbf{v}'$ are

$$v_i' = v_i - \frac{a_i}{b_{i-1}'} v_{i-1}$$

Now, dividing each row by its b_i', the matrix is in row reduced form.