Solving the 1D-Poisson Equation Numerically

Kristian Gjestad Vangsnes

Department of physics, University of Oslo

Date: 9. September 2019

1 Introduction

2 Theory

In this project we study the one-dimentional Poisson equation with Dirichlet boundary conditions which reads

$$-u''(x) = f(x), \quad x \in (0,1), \quad u(0) = u(1) = 0.$$

We will use the source term $f(x) = 100e^{-10x}$. This gives the particular solution $u(x) = 1 - (1 - e^{-10})x - e^{-10x}$. We will compare our numerical solution to this exact solution.

3 Numerical methods

In order to model this equation in a computer we need to define the discretized approximated solution to $u(x)$ as $v_i = v(x_i)$ where $x_i = x_0 + ih = ih$ since $x_0 = 0$. We let $x_{n+1} = 1$, this means $h = \frac{1}{n+1}$. From Taylor expanding $u(x \pm h)$ we get

$$u(x \pm h) = u(x) \pm hu'(x) + \frac{h^2}{2!}u''(x) \pm O(h^3)$$

We see that $u''(x) = \frac{u(x+h)+u(x-h)-2u(x)}{h^2} - O(h^4)$. Hence for the approximated solution we get that

$$-\frac{v_{i+1} + v_{i-1} - 2v_i}{h^2} = f_i \quad \text{for i} = 1, ..., n$$

where $f_i = f(x_i)$. If we set $g_i = h^2 f_i$ we get that $2v_i - v_{i+1} - v_{i-1} = g_i$. This is just n equations, given by $i = i, ..., n$. We can write this in matrix form

$$\mathbf{A}\mathbf{v} = \mathbf{g},$$

where $\mathbf{A}$ is a $n \times n$ tridiagonal matrix on the form

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

$\mathbf{v}$ and $\mathbf{g}$ are vectors on the form

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \qquad \mathbf{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}$$

3.1 General algorithm

The general algorithm to solve a set of equation on a tridiagonal form is done by Gaussian elimination, which is a forward- and a backward substitution. On matrix form the problem is $\mathbf{Av} = \mathbf{g}$. Where $\mathbf{v}$ contains the unknowns v_i. In our case the unknowns are the solution to the Poisson equation at $x_i = ih$, i.e. $v(x_i) = v_i$.

$$\mathbf{A} = \begin{bmatrix} d_1 & b_1 & 0 & \dots & ... & 0 \\ a_1 & d_2 & b_2 & 0 & ... & 0 \\ 0 & a_2 & d_3 & b_3 & 0 & ... \\ \vdots & 0 & \ddots & \ddots & \ddots & ... \\ 0 & ... & ... & a_{n-2} & d_{n-1} & b_{n-1} \\ 0 & ... & ... & 0 & a_{n-1} & d_n \end{bmatrix}, \qquad \mathbf{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}.$$

Component wise the forward substitution is given by

$$\tilde{d}_{i+1} = d_{i+1} - \frac{a_i b_i}{\tilde{d}_i}$$

$$\tilde{g}_{i+1} = g_{i+1} - \frac{a_i \tilde{g}_i}{\tilde{d}_i}$$

Where $\tilde{d_1} = d_1$ and $\tilde{g}_1 = g_1$. Now our problem in on the form $\tilde{\mathbf{A}}\mathbf{v} = \tilde{\mathbf{g}}$ where

$$\tilde{\mathbf{A}} = \begin{bmatrix} \tilde{d_1} & b_1 & 0 & \dots & ... & 0 \\ 0 & \tilde{d_2} & b_2 & 0 & ... & 0 \\ 0 & 0 & \tilde{d_3} & b_3 & 0 & ... \\ \vdots & 0 & \ddots & \ddots & \ddots & ... \\ 0 & ... & ... & 0 & \tilde{d}_{n-1} & b_{n-1} \\ 0 & ... & ... & 0 & 0 & \tilde{d_n} \end{bmatrix}, \qquad \mathbf{g} = \begin{bmatrix} \tilde{g}_1 \\ \tilde{g}_2 \\ \vdots \\ \tilde{g}_n \end{bmatrix}.$$

To get the problem to a diagonal form we have to perform a backward substitution, this is done component wise by

$$v_{n-i} = \frac{\tilde{g}_{n-i} - b_{n-i}v_{n+1-i}}{\tilde{d}_{n-i}}$$

where $v_n = \frac{\tilde{g}_n}{\tilde{d}_n}$. The number of floating points operations (FLOPS) performed in this general algorithm is $9n$, $6n$ FLOPS in the forward substitution, two subtractions, two multiplications and two divisions. $3n$ FLOPS in the backward substitution, one subtraction, one multiplication and one division.

3.2 Specialized algorithm

4 results

This algorithm was programed using c++ and the solution for matrix size n=10, 100 and 1000 was compared to the exact solution and plotted using python as shown below.

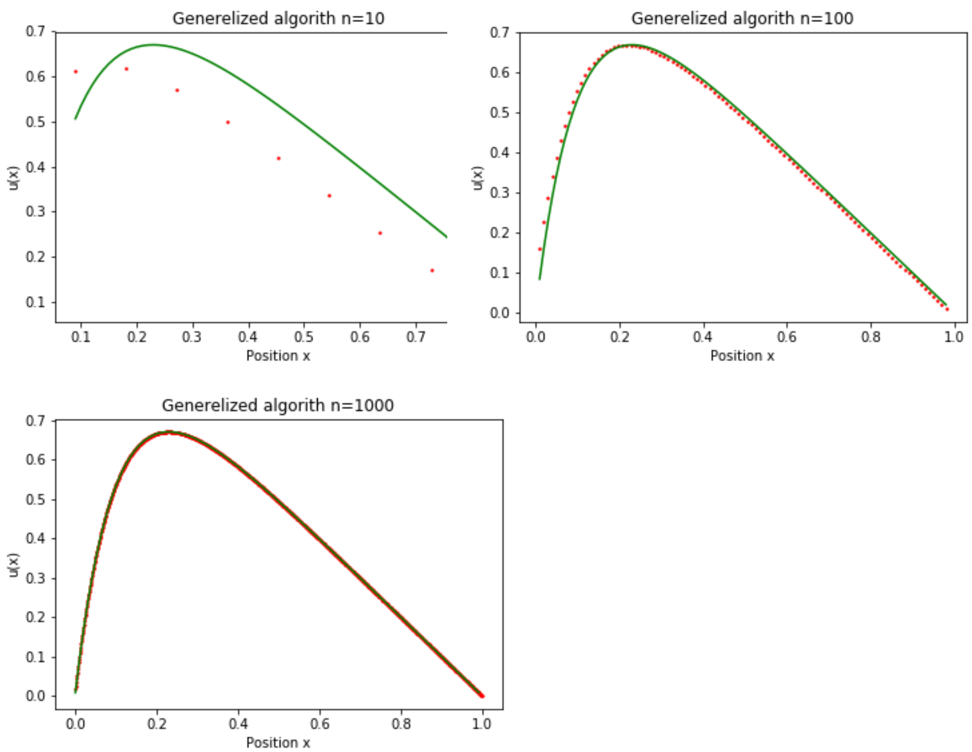

Figure 1: Numerical solution of the Poisson equation for matrix size n=10, 100 and 1000 is plotted as red dots. The exact solution is plotted in green.