FYS3150-Project 1

Marcus Berget, Sebastian Amundsen, Andreas Wetzel

August 2020

Abstract

1 Introduction

In this project we will use numerical methods to solve the one dimensional Poisson equation. We will be rewriting the Dirichlet boundary conditions as a set of linear equations.

2 Method

By multiplying **A** and **v** we get

$$\begin{bmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \dots \\ 0 & -1 & 2 & -1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots & & \\ 0 & \vdots & \vdots & -1 & 2 & -1 \\ 0 & \vdots & \vdots & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_n \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} 2v_0 - v_1 \\ -v_0 + 2v_1 - v_2 \\ -v_1 + 2v_2 - v_3 \\ \vdots \\ -v_{n-1} + 2v_n - v_{n+1} \\ -v_n + 2v_{n+1} \end{bmatrix}$$

$$= \begin{bmatrix} h^2 f_0 \\ h^2 f_1 \\ h^2 f_2 \\ \vdots \\ h^2 f_n \\ h^2 f_{n+1} \end{bmatrix} = \widetilde{\mathbf{b}}$$

2.1 General algorithm

2.2 Algorithm for specific tri-diagonal matrix

In our special case we can implement a solver that is even simpler than what is described previously. We will exploit the fact that the matrix has identical matrix elements along the diagonal and identical values for the non diagonal elements $\vec{e}_i$. In this case we can precalculate the new values for the updated matrix elements d_i without taking into account the values for $\vec{e}_i$:

$$d_i = 2 - \frac{1}{\tilde{d}_{i-1}} = \frac{i+1}{i} \tag{1}$$

Here the initial value is $\tilde{d}_1 = 2$. The new righthand side solution $\tilde{f}_i$ is given by:

$$\tilde{f}_i = f_i + \frac{(i-1)\tilde{f}_{i-1}}{i} \tag{2}$$

Here the initial value is $\tilde{f}_1 = f_1$. The last step is to make a backward substitution which gives the final solution u_i:

$$u_{i-1} = \frac{i-1}{i}(\tilde{f}_{i-1} + \tilde{u}) \tag{3}$$

This method requires that we know the last value u_n in the u_i array. This value is given by $u_n = \tilde{f}_n / \tilde{b}_n$.

2.3 Relative error

Algorithms have a varying degree of uncertainty. We will test how precise our algorithm is for the specific tri-diagonal matrix case. The numerical solution will be compared to the analytical solution given the relative error ϵ_i:

$$\epsilon_i = \log_{10}\left(\left|\frac{v_i - u_i}{u_i}\right|\right) \tag{4}$$

For the numerical v_i and analytical u_i function values.

2.4 LU decomposition

3 Implementation

All programs used is available at:
`https://github.com/Sebamun/FYS3150_Projekter`

We implement the algorithms numerically with varying values of grid points n.

4 Results

4.1 General algorithm

4.2 Algorithm for specific tri-diagonal matrix

4.3 Relative error

4.4 LU decomposition

5 Concluding remarks