SOLVING THE POISSON-EQUATION IN ONE DIMENSION: TRIDIAGONAL MATRIX ALGORITHM AND LU-DECOMPOSITION

FYS3150: COMPUTATIONAL PHYSICS

SIGURD SANDVOLL SUNDBERG
GITHUB.COM/SIGURDSUNDBERG

ABSTRACT. Abstract write last.

CONTENTS

1. Introduction

The use of differential equation has a sentral role in every branch of science. Linear second-order differential equations cover many of the important differential equations(DEs). Many of which can be manipulated to be solved as a linear algebra problem, for which we can develop algorithms to solve to problems numerically. There are limitations when it comes to numerically solutions. If the number of floating point operations(FLOPs) becomes too big the calculations can get exceedingly slow. One can also risk numerical inaccuracies simply due to a high amount of FLOPs. This puts the emphasis on developing algorithms which are stable, accurate and quick. In this project we will take a look at solving the one-dimensional Poisson equation with Dirichlet boundary conditions by rewriting it as a set of linear equations. Excplicitly we want to solve the following equation

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

We will device an algorithm for a tridiagonal matrix, both the general case and a specialized case. Including, we will take a look at the more general method of solving matrix equations using LU-decomposition, comparing run times and space between the different algorithms. In addition to looking at relative error and finding the mesh-grid which gives the best approximation. Lastly, the findings will be discussed.

2. Theory

2.1. **Poisson equation.**

2.2. **Approximating the second derivative.**

To solve one-dimensional Poisson equiation numerically we will need to discretize the equation. Instead of having continous functions, we get

$$\begin{aligned} x &\to x_i \in [x_0, x_1, \dots, x_i, \dots, x_{n+1}] \\ u(x) &\to u(x_i) = u_i \in [u_0, u_1, \dots, u_i, \dots, u_{n+1}] \\ f(x) &\to f(x_i) = f_i \in [f_0, f_1, \dots, f_i, \dots, f_{n+1}], \end{aligned} \tag{2}$$

using grid-points $x_i = ih$ in the interval from $x_0 = 0$ to $x_{n+1} = 1$ since we have an open interval. The step length h is defined as $h = 1/(n+1)$. We then have the boundary conditions $x_0 = v_{n+1} = 0$. The approximation for the second derivative we need from 1, can be found through Taylor expansion of $u(x \pm h)$ around x.

$$u(x \pm h) = u(x) \pm hu'(x) + \frac{h^2 u''(x)}{2!} \pm \frac{h^3 u'''(x)}{3!} + \mathcal{O}(h^4) \tag{3}$$

If we look at the two equation we get from $u(x+h)$ and $u(x-h)$, we can see that the first and third derivatives cancel eachother out when we add the two equations together. If we solve the resulting equation for $u''(x)$ we get

$$u''(x) = \frac{u(x+h) + u(x-h) - 2u(x)}{h^2} + \mathcal{O}(h^2). \tag{4}$$

Using our discretization of the continous functions given by equation 2, we have the following equation

$$-u''(x) \simeq f_i = \frac{u_{i+1} + u_{i-1} - 2u_i}{h^2} + \mathcal{O}(h^2). \tag{5}$$

for $i = 1, \cdots, n$. If we define $f_i^* = h^2 f_i$, equation 5 becomes $-u_{i+1} - u_{i-1} + 2u_i = f_i^*$. If we try inserting values of $i = 1, 2, 3, 4, \ldots, n-1n$ we get the following

$$\begin{aligned} -u_2 - u_0 + 2u_1 &= f_1^* \quad i = 1 \\ -u_3 - u_1 + 2u_2 &= f_2^* \quad i = 2 \\ -u_4 - u_2 + 2u_3 &= f_3^* \quad i = 3 \\ -u_5 - u_3 + 2u_4 &= f_4^* \quad i = 4 \\ &\vdots \\ -u_n - u_{n-2} + 2u_{n-1} &= f_{n-1}^* \quad i = n-1 \\ -u_{n+1} - u_{n-1} + 2u_n &= f_n^* \quad i = n \end{aligned} \tag{6}$$

With the Dirichlet boundary conditions $u(0) = u(n+1) = 0$ we see resembles a matrix with 2 along the diagonal and -1 directly above and below the leading diagonal. We see that we can solve as a linear algebra problem, where we want to find $\vec{x}$, on the following form.

$$\mathbf{A}\vec{x} = f^*,$$

where $\mathbf{A}$ is an $n \times n$ tridiagonal matrix which we found in 6.

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & \ldots & \ldots & 0 \\ -1 & 2 & -1 & 0 & \ldots & \ldots \\ 0 & -1 & 2 & -1 & 0 & \ldots \\ \ldots & \ldots & \ddots & \ddots & \ddots & \ldots \\ 0 & \ldots & 0 & -1 & 2 & -1 \\ 0 & \ldots & \ldots & 0 & -1 & 2 \end{bmatrix} \tag{7}$$

3. Algorithms

3.1. **Tridiagonal Matrix Algorithm.** Section on the TDMA » Problem b » Implementation » FLOPS

Looking at a general case of solving a tridiagonal matrix on the form $\mathbf{A}\vec{x} = f^*$ we have

$$\begin{bmatrix} b_1 & c_1 & 0 & \ldots & \ldots & \ldots \\ a_1 & b_2 & c_2 & 0 & \ldots & \ldots \\ 0 & a_2 & b_3 & c_3 & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & a_{n-2} & b_{n-1} & c_{n-1} \\ \ldots & \ldots & \ldots & \ldots & a_{n-1} & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ \ldots \\ \ldots \\ x_n \end{bmatrix} = \begin{bmatrix} f_1^* \\ f_2^* \\ \ldots \\ \ldots \\ \ldots \\ f_n^* \end{bmatrix} \tag{8}$$

Let's take general case for a 4×4 tridiagonal matrix and use Gaussian elimination to see if we can spot a pattern. Our system can then be written

$$\mathbf{A}\vec{x} = f^*$$

$$\begin{bmatrix} b_1 & c_1 & 0 & 0 \\ a_1 & b_2 & c_2 & 0 \\ 0 & a_2 & b_3 & c_3 \\ 0 & 0 & a_3 & b_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} f_1^* \\ f_2^* \\ f_3^* \\ f_4^* \end{bmatrix} \tag{9}$$

The set of equations we will use Gaussian elimination on is therefore

$$\left[\begin{array}{cccc|c} b_1 & c_1 & 0 & 0 & f_1^* \\ a_1 & b_2 & c_2 & 0 & f_2^* \\ 0 & a_2 & b_3 & c_3 & f_3^* \\ 0 & 0 & a_3 & b_4 & f_4^* \end{array}\right] \tag{10}$$

First we will use forward substitution to remove all the elements along the diagonal below the leading diagonal. We start with $\text{II} - \text{I}(a_1/b_1)$ to remove a_1 from II, where I refers to row number one and so on.

3.2. **Specialized algorithm.** Section on the optimization » Specialized algorithm problem c » FLOPS » CPU time

3.3. **LU Decomposition.** Section on LU-decomposition » Alogrithm for LU-decomposition » FLOPS » CPU time

4. Results

Results from the report. » CPU time difference » Plots » Difference in relative error

5. Discussion

Discussion of the report.

6. Conclusion

Conclusion of the report.