FYS3150 oppgavesett 1

Jon Aleksander Prøitz and Marius Torsvoll

Relevant code can be found at: https://github.com/Jonaproitz/Project_1

PROBLEM 1.

Given

$$u(x) = 1 - (1 - e^{-10})x - e^{-10x} \tag{1}$$

Inserting $x = 0$

$$u(0) = 1 - (1 - e^{-10}) \cdot 0 - e^{-10 \cdot 0} = 1 - 0 - 1 = 0$$

and $x = 1$

$$u(1) = 1 - (1 - e^{-10}) \cdot 1 - e^{-10 \cdot 1} = 1 - 1 + e^{-10} - e^{-10} = 0$$

furthermore the one-dimensional possion equation can be written

$$-\frac{d^2u}{dx^2} = -\frac{d^2}{dx^2}\left(1 - (1 - e^{-10})x - e^{-10x}\right) = -\frac{d}{dx}\left((1 - e^{-10}) + 10e^{-10x}\right) = 100e^{-10x} = f(x)$$

Hence equation 1 is an exact solution to our problem. ■

PROBLEM 2.

Figure 1 shows a plot of the exact solution to the one-dimensional poisson equation given by equation 1.

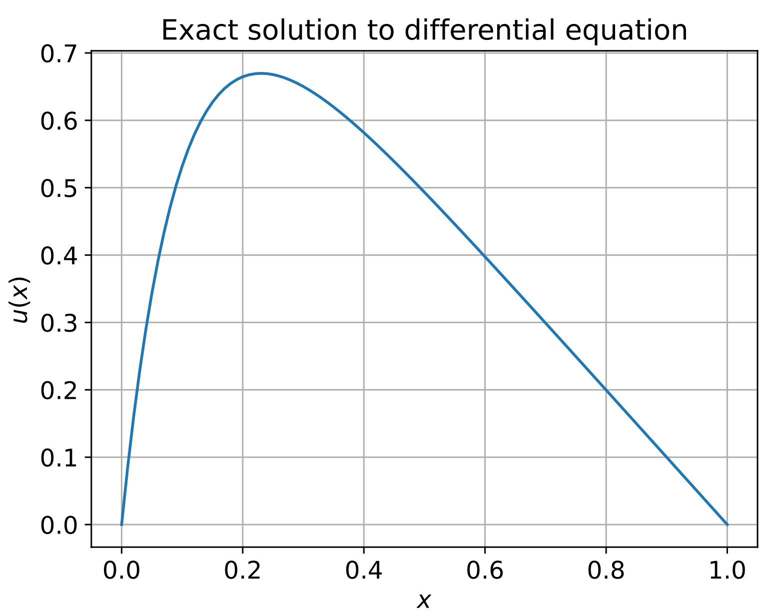

FIG. 1. Plot of equation 1 in the given area $x \in [0, 1]$.

PROBLEM 3.

The one-dimensional poisson equation can be written

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

Discretizing x with m values and a given distance h between each distinct value then gives

$$\begin{aligned} x &\rightarrow x_0, x_1, x_2, ..., x_{m-1} \\ u(x) &\rightarrow u_0, u_1, u_2, ..., u_{m-1} \\ f(x) &\rightarrow f_0, f_1, f_2, ..., f_{m-1} \end{aligned}$$

with $u_i = v_i$, such that

$$-\frac{d^2v_i}{dx^2} = -v_{i-1} + 2v_i - v_{i+1} = f_i h^2$$

PROBLEM 4.

The set of equations from problem 3 can be written as

$$\begin{aligned} -v_0 + 2v_1 - v_2 &= h^2 f_1 \\ -v_1 + 2v_2 - v_3 &= h^2 f_2 \\ -v_2 + 2v_3 - v_4 &= h^2 f_3 \\ &\vdots \\ -v_{m-3} + 2v_{m-2} - v_{m-1} &= h^2 f_{m-2} \end{aligned}$$

Wich for

$$\begin{aligned} g_1 =& h^2 f_1 + v_0 \\ g_2 =& h^2 f_2 \\ g_3 =& h^2 f_3 \\ &\vdots \\ g_{m-3} =& h^2 f_{m-3} \\ g_{m-1} =& h^2 f_{m-2} + v_{m-1} \end{aligned}$$

can be written as the matrix equation

$$\begin{pmatrix} 2 & -1 & 0 & 0 & 0 & \dots \\ -1 & 2 & -1 & 0 & 0 & \dots \\ 0 & -1 & 2 & -1 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{m-3} \\ v_{m-2} \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_{m-3} \\ g_{m-2} \end{pmatrix}$$

on the form $A\vec{v} = \vec{g}$.

PROBLEM 5

a

When finding the matrix, A, in problem 4 it is assumed that v_0 and v_{m-1} are known. Hence theese values are not calculated and

$$n = m - 2$$

b

As discussed above the the vector $\vec{v}$ is equal to $\vec{v}^*$ excluding the first and last element. Hence

$$\vec{v}^* = [v_0, \vec{v}, v_{m-1}] \tag{2}$$

PROBLEM 6

a

A general tridiagonal matrix on the form below, with with each row marked as (R_i)

$$\begin{array}{c|ccccccc|c} (R_1) & b_1 & c_1 & 0 & 0 & 0 & \dots & | & g_1 \\ (R_2) & a_2 & b_2 & c_2 & 0 & 0 & \dots & | & g_2 \\ (R_3) & 0 & a_3 & b_3 & c_3 & 0 & \dots & | & g_3 \\ & & & & \ddots & & & | & \vdots \\ (R_n) & 0 & 0 & 0 & 0 & a_n & b_n & | & g_n \end{array} \tag{3}$$

can be solved by first forward substituting, giving

$$R_2 \to R_2 - \frac{a_2}{b_1} R_1 = 0 \;\; b_2 - \tfrac{a_2}{b_1} c_1 \;\; 0 \;\; 0$$

Continuing this substitution gives the following relation by introducing the variables $\tilde{b}_i$ and $\tilde{g}_i$:

$$\tilde{b}_1 = b_1 \quad \text{and} \quad \tilde{b}_i = b_i - \frac{a_i}{\tilde{b}_{i-1}} c_{i-1}, \qquad \text{For i = 2,3,4....,n}$$
$$\tilde{g}_1 = g_1 \quad \text{and} \quad \tilde{g}_i = g_i - \frac{a_i}{\tilde{g}_{i-1}} \tilde{g}_{i-1}, \qquad \text{For i = 2,3,4....,n}$$

Wich then gives the rowequivalent matrix

$$\begin{array}{cccccc|c} \tilde{b} & c_1 & 0 & 0 & 0 & \dots & \tilde{g}_1 \\ 0 & \tilde{b}_2 & c_2 & 0 & 0 & \dots & \tilde{g}_2 \\ 0 & 0 & \tilde{b}_3 & c_3 & 0 & \dots & \tilde{g}_3 \\ & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \tilde{b}_n & \tilde{g}_n \end{array}$$

Thus

$$\frac{\tilde{g}_n}{\tilde{b}_n} = v_n$$

And bacwards substitution gives the relation

$$v_i = \frac{\tilde{g}_i - c_i v_{i+1}}{\tilde{b}_i}, \text{For i} = n-1, n-2, n-3, \dots 2, 1$$

This can be interpreted as the following algorithm.

Algorithm 1 General alorithm

$\tilde{b}_1 = b_1$
$\tilde{g}_1 = g_1$
for $i = 2, 3, ..., n$ **do**
  $\gamma = \frac{a_i}{\tilde{b}_{i-1}}$
  $\tilde{b}_i = b_i - \gamma c_{i-1}$
  $\tilde{g}_i = g_i - \gamma \tilde{g}_{i-1}$
$v_n = \frac{\tilde{g}_n}{\tilde{b}_n}$
for $i = n-1, n-2, ..., 1$ **do**
  $v_i = \frac{\tilde{g}_i - c_i v_{i+1}}{\tilde{b}_i}$

b

Notice that the first for loop contains $5(n-1)$ FLOPs while the second for loop contains $3(n-1)$ FLOPs. Hence the total number of FLOPs is roughly $8(n-1)$.

PROBLEM 7

A comparison of the analytical and numerical solutions to the one-dimensional poisson equation is shown in fifure 2. Relevant code can be found at the linked github.

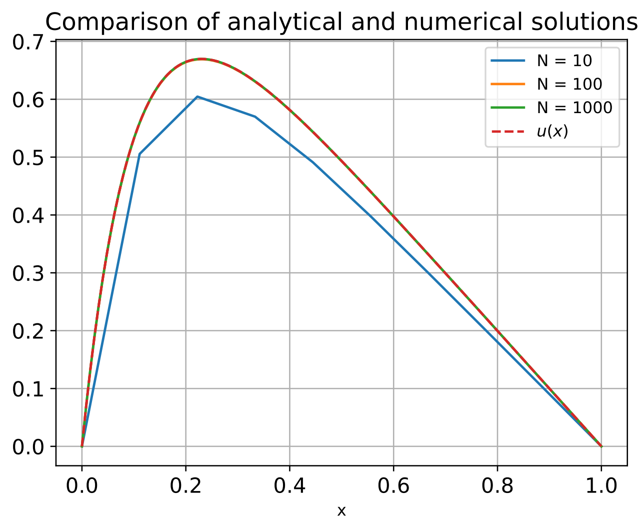

FIG. 2. Comparison of the analytical and numerical solutions of the one-dimensional poisson equation.

PROBLEM 8

a

Figure 3 shows the difference between the numerical and exact solution.

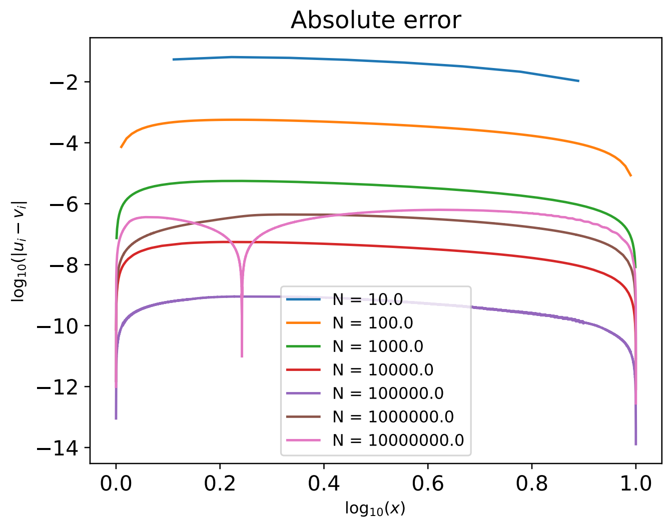

FIG. 3. Absolute error of the numerical solution for different number of steps.

b

Figure 4 shows the relative error in the numerical solution.

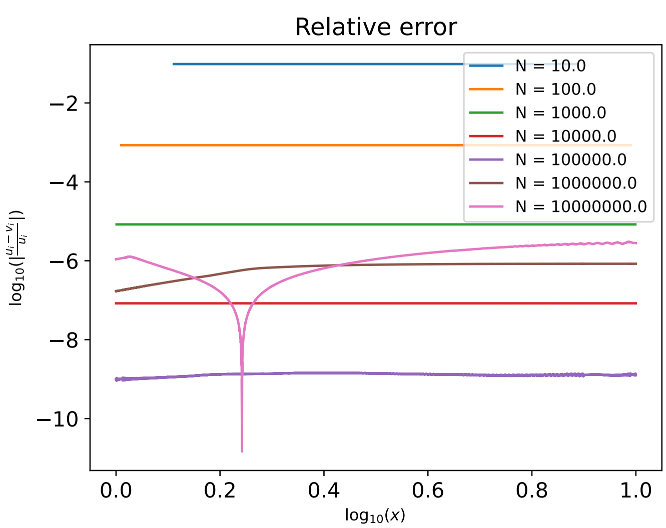

FIG. 4. Relative error of the numerical solution for different number of steps.

c

The table below shows the maximum relative error for a given number of points. This data is also represented in figure 5.

Number of points	Output
10	9.682820e-02
100	8.498198e-04
1000.0	8.350052e-06
10000.0	8.342473e-08
100000.0	1.452374e-09
1000000.0	8.405051e-07
10000000.0	2.983866e-06

TABLE I. Write a descriptive caption here, explaining the content of your table.

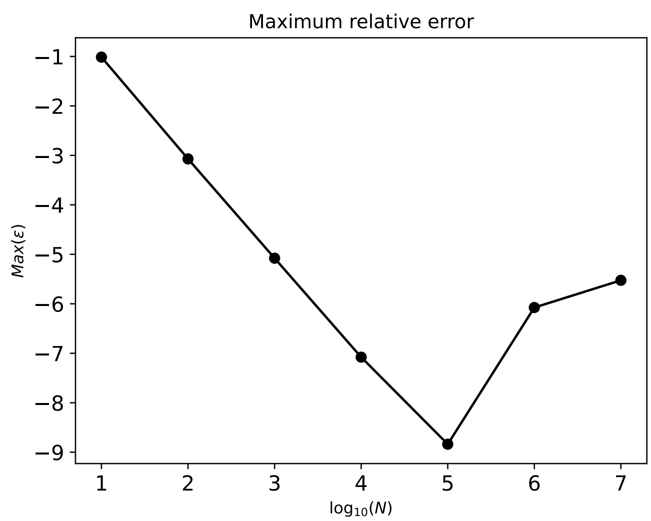

FIG. 5. Maximum error of the numerical solution for different number of steps.

All code can be found at the linked github.

PROBLEM 9

a

By making the special algorithm we then get the following results.

Algorithm 2 General alorithm

$\tilde{b}_1 = 2$
$\tilde{g}_1 = g_1$
for $i = 2, 3, ..., n$ **do**
  $\gamma = \frac{1}{\tilde{b}_{i-1}}$
  $\tilde{b}_i = 2 - \gamma$
  $\tilde{g}_i = g_i + \gamma \tilde{g}_{i-1}$
$v_n = \frac{\tilde{g}_n}{\tilde{b}_n}$
for $i = n-1, n-2, ..., 1$ **do**
  $v_i = \frac{\tilde{g}_i + v_{i+1}}{\tilde{b}_i}$

b

Notice that the first for loop contains $4(n-1)$ FLOPs while the second for loop contains $2(n-1)$ FLOPs. Hence the total number of FLOPs is roughly $6(n-1)$.

c

See the GitHub link

PROBLEM 10

Figure 6 shows a plot of the difference in time between the special and general algorithm. Notice both axis are logarithmic. It is clear that the general algorithm needs more time to run and that the difference grows by a constant factor, as expected. It is not clear what units the time is given in, note that the snippet used to get the time was clear that it would be in seconds.

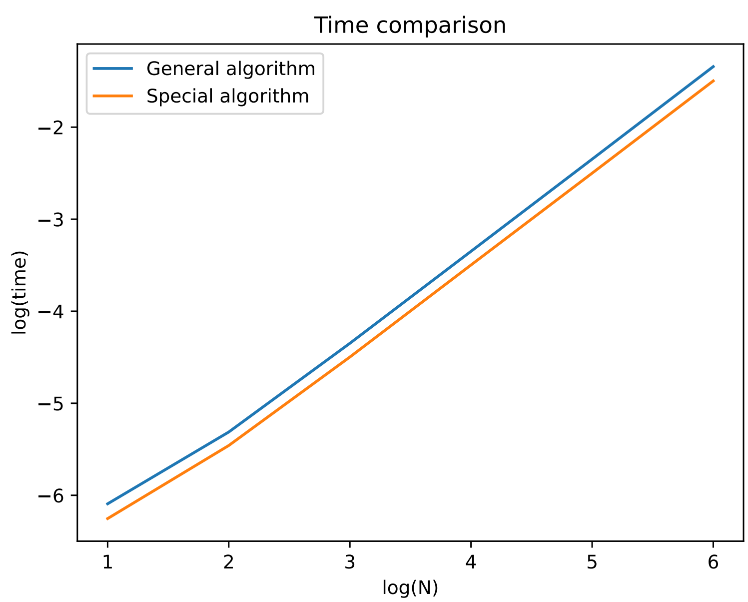

FIG. 6. Comparison of the times the two algorithm needed to run.

PROBLEM 11

Unfortunately we did not have time to answer this quation.