Project 1
FYS3150

Ann-Silje Kirkvik

github.com/annsilje/fys3150

September 14, 2016

1 Introduction

This project will explore different numerical methods for solving linear second order differential equations of the form

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

This problem can be rewritten into a set of linear equation $\boldsymbol{Ax} = \boldsymbol{b}$ with the nice property that the design matrix $\boldsymbol{A}$ is tridiagnoal. This particular matrix even has an analytical solution, which enables the creation of tailored alogrithms that are fast and memory efficient compared to a brute force linear equation solver.

2 Description

Starting with a Taylor expansion of $u(x_i + h) = u_{i+1}$ and $u(x_i - h) = u_{i-1}$ and adding these togheter yields the three point formula for the second order derivative:

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

The numerical approximation $v_i''(x_i)$ to $u''(x)$ is then

$$v_i''(x_i) = \frac{v_{i+1} + v_{i-1} - 2v_i}{h^2}.$$

Inserting this approximation into the original differential equation in 1 yields the following formula:

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

for a given set of grid points $(x_i, f_i(x_i))$ for $i = 1, ..., n$, where $h = 1/(n+1)$ is the step length between each x_i. With the boundary conditions $v_0 = v_{n+1} = 0$ this becomes n linear equations:

$$\begin{aligned}
2v_1 - v_2 &= h^2 f_1 \\
-v_1 + 2v_2 - v_3 &= h^2 f_2 \\
-v_2 + 2v_3 - v_4 &= h^2 f_3 \\
&\vdots \\
-v_{n-1} + 2v_n - v_{n+1} &= h^2 f_n
\end{aligned}$$

In matrix form this becomes $\boldsymbol{Av} = h^2 \boldsymbol{f}$ or

$$
\begin{bmatrix}
2 & -1 & 0 & \cdots & \cdots & \cdots & 0 \\
-1 & 2 & -1 & 0 & \cdots & \cdots & 0 \\
0 & -1 & 2 & -1 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & 0 & -1 & 2 & -1 \\
0 & \cdots & \cdots & \cdots & 0 & -1 & 2
\end{bmatrix}
\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix}
= h^2 \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix}
$$

This linear equation system can be solved numerically in at least three different ways:

- By LU-decomposition and backward substitution.
- With a tridiagonal solver and backward subtitution.
- With a analytical custom made solver and backward substitution.

2.1 LU-decomposition

LU-decomposition consists of finding a factorization $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ such that $\boldsymbol{L}$ is a lower triangular matrix and $\boldsymbol{U}$ is an upper triangular matrix. For instance if $\boldsymbol{A}$ is a 4×4 matrix one such factorization is:

$$
\boldsymbol{L} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
l_{21} & 1 & 0 & 0 \\
l_{31} & l_{32} & 1 & 0 \\
l_{41} & l_{42} & l_{43} & 1
\end{bmatrix}
\text{ and } \boldsymbol{U} = \begin{bmatrix}
u_{11} & u_{12} & u_{13} & u_{14} \\
0 & u_{22} & u_{23} & u_{24} \\
0 & 0 & u_{33} & u_{34} \\
0 & 0 & 0 & u_{44}
\end{bmatrix}
$$

Once this factorization is found the equation system $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{L}\boldsymbol{U}\boldsymbol{v} = h^2\boldsymbol{f}$ can be solved by setting $\boldsymbol{L}\boldsymbol{y} = h^2\boldsymbol{f}$ and $\boldsymbol{U}\boldsymbol{v} = \boldsymbol{y}$ and applying backward substitution twice.

2.2 Tridiagonal solver

A tridiagonal solver exploits the fact that only the diagonal elements and the elements directly above and below the diagonal is different from zero. The matrix $\boldsymbol{A}$ then has the form:

$$
\boldsymbol{A} = \begin{bmatrix}
b_1 & c_1 & 0 & \cdots & \cdots & \cdots & 0 \\
a_2 & b_2 & c_2 & 0 & \cdots & \cdots & 0 \\
0 & a_3 & b_3 & c_3 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\
0 & \cdots & \cdots & \cdots & 0 & a_n & b_n
\end{bmatrix}
$$

By performing Gauss elimination on the equation system it can be shown that the elements directly above the diagonal remains unchanged and the diagonal elements d_i become

$$d_i = b_i - a_i c_{i-1}/d_{i-1} \text{ with } d_1 = b_1$$

The elements directly below the diagonal becomes zero, as is the goal of the Gauss elimination. The column vector elements w_i becomes

$$w_i = f_i - a_i w_{i-1}/d_{i-1} \text{ with} w_1 = f_1$$

2.3 Custom solver

3 Solution

If $f(x) = 100e^{-10x}$ the analytical solution to the differential equation in 1 is $u(x) = 1 - (1 - e^{-10})x - e^{-10x}$.

3.1 Algorithm

3.2 Results

Steps (n)	Generic	Specific	G - S	LUD	G - LUD
10^1	1.20×10^{-5}s	4.00×10^{-6}s	8.00×10^{-6}s	1.90×10^{-5}s	-7.00×10^{-6}s
10^2	8.50×10^{-5}s	9.00×10^{-6}s	7.60×10^{-5}s	3.54×10^{-3}s	-3.45×10^{-3}s
10^3	2.40×10^{-4}s	8.00×10^{-5}s	1.60×10^{-4}s	1.20s	-1.20s
10^4	3.20×10^{-4}s	2.72×10^{-4}s	4.80×10^{-5}s		
10^5	3.19×10^{-3}s	2.49×10^{-3}s	6.93×10^{-4}s		
10^6	2.81×10^{-2}s	2.59×10^{-2}s	2.23×10^{-3}s		
10^7	2.85×10^{-1}s	2.54×10^{-1}s	3.06×10^{-2}s		

Table 1: Run times for the different algorithms.

Step size (h)	Max relative error(ϵ)	$log(\epsilon)$
9×10^{-2}	2.00329251	$6.94792080 \times 10^{-1}$
1×10^{-2}	1.09015546	$8.63203083 \times 10^{-2}$
1×10^{-3}	1.00890166	$8.86227046 \times 10^{-3}$
1×10^{-4}	1.00088902	$8.88626716 \times 10^{-4}$
1×10^{-5}	1.00008889	$8.88864595 \times 10^{-5}$
1×10^{-6}	1.00000923	$9.22752276 \times 10^{-6}$
1×10^{-7}	1.00000307	$3.07023540 \times 10^{-6}$

Table 2: Maximum relative error for each step length using the generic solver.

4 Conclusions

A Gauss elimintaion

Gauss elimination of the equation system $\boldsymbol{A}\boldsymbol{v} = h^2\boldsymbol{f}$, where a_{ij} is the matrix element of matrix $\boldsymbol{A}$ at row i and column j and R_i is row number i.

$$\begin{bmatrix} b_1 & c_1 & 0 & \cdots & \cdots & \cdots & 0 \\ a_2 & b_2 & c_2 & 0 & \cdots & \cdots & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & \cdots & \cdots & \cdots & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} = h^2 \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix}$$

Set $b_1 = d_1$, $f_1 = w_1$ and apply $R_2 = R_2 - R_1 * a_{j1}/a_{11}$.

$$\begin{bmatrix} d_1 & c_1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & b_2 - \frac{a_2 c_1}{d_1} & c_2 - \frac{a_2 * 0}{b_1} & 0 & \cdots & \cdots & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & \cdots & \cdots & \cdots & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} = h^2 \begin{bmatrix} w_1 \\ f_2 - \frac{a_2 w_1}{d_1} \\ f_3 \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix}$$

Set $d_2 = b_2 - a_2 c_1/d_1$, $w_2 = f_2 - a_2 w_1/d_1$ and apply $R_3 = R_3 - R_2 * a_{j2}/a_{22}$.

$$\begin{bmatrix} d_1 & c_1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & d_2 & c_2 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & b_3 - \frac{a_3 c_2}{d_2} & c_3 - \frac{a_3 * 0}{d_2} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & \cdots & \cdots & \cdots & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} = h^2 \begin{bmatrix} w_1 \\ w_2 \\ f_3 - \frac{a_3 w_2}{d_1} \\ \vdots \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix}$$

Continuing this process for every row finally yields

$$\begin{bmatrix} d_1 & c_1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & d_2 & c_2 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & d_3 & c_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & 0 & d_{n-1} & c_{n-1} \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & d_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} = h^2 \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ \vdots \\ w_{n-1} \\ w_n \end{bmatrix}$$

With $d_i = b_i - a_i c_{i-1}/d_{i-1}$ and $w_i = f_i - a_i w_{i-1}/d_{i-1}$.

B Numerical vs Analytical solutions

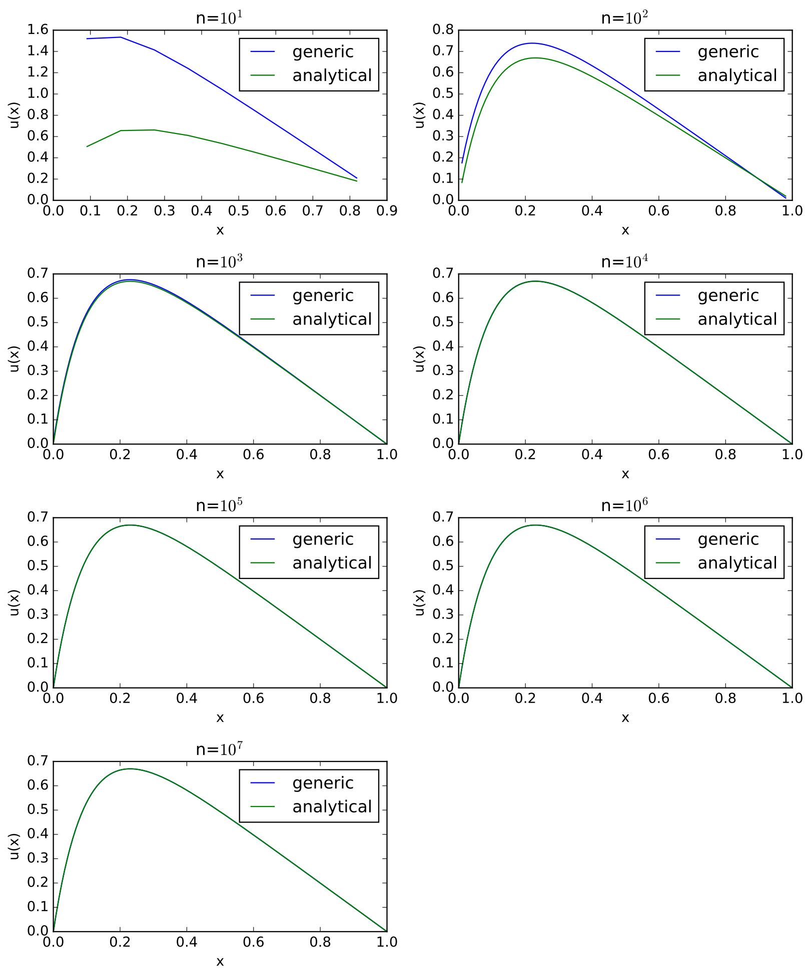

Figure 1: Numerical solution using the generic tridiagonal solver compared to the analytical solution.

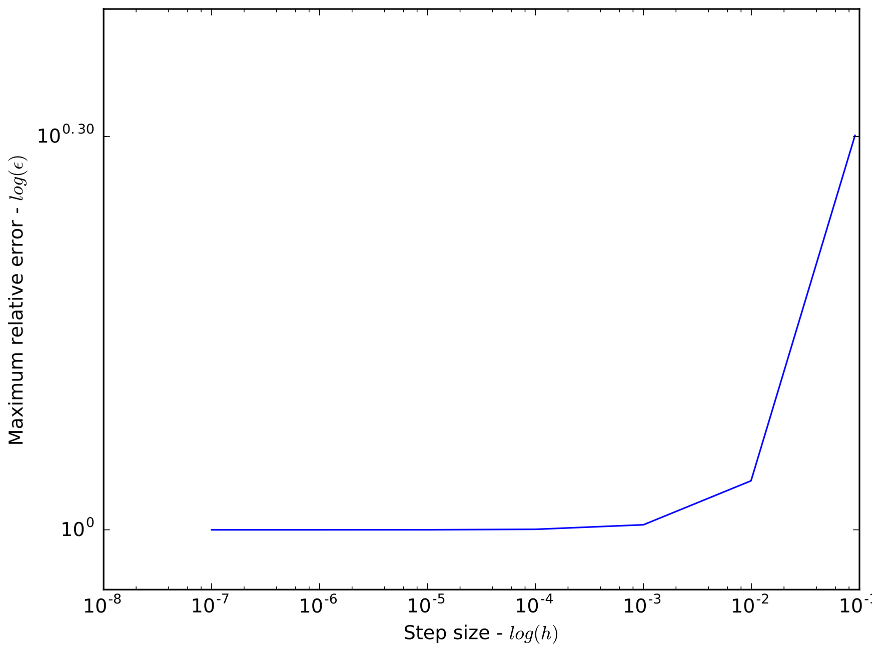

Figure 2: Error

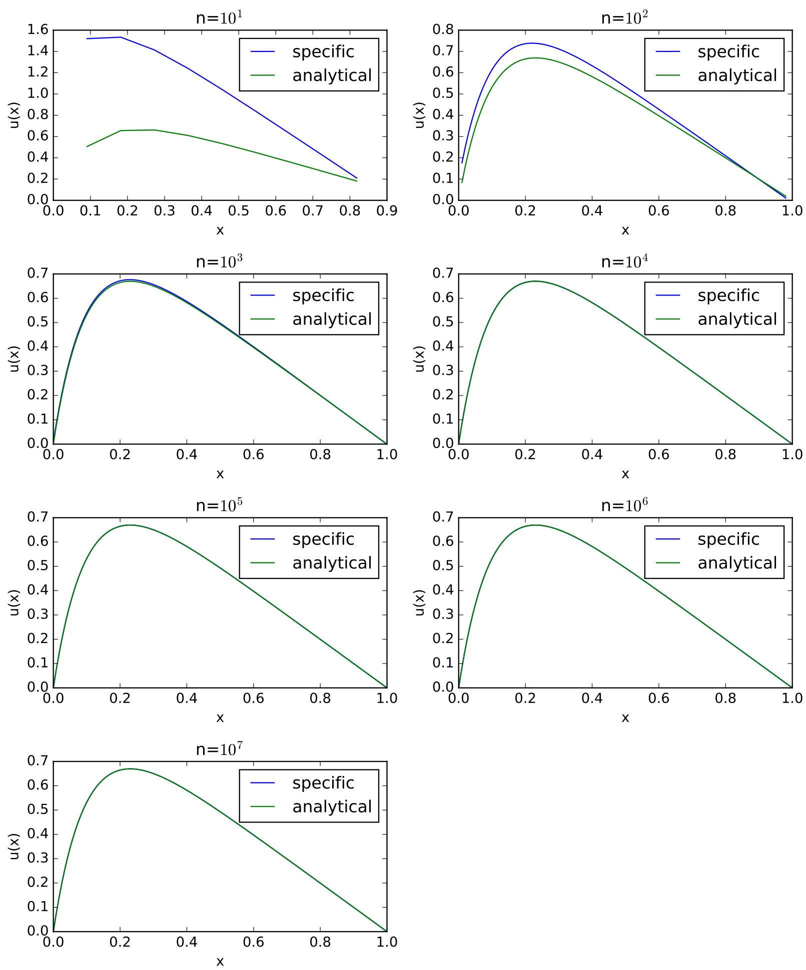

Figure 3: Numerical solution using the tridiagonal solver tailored to this specific matrix compared to the analytical solution.

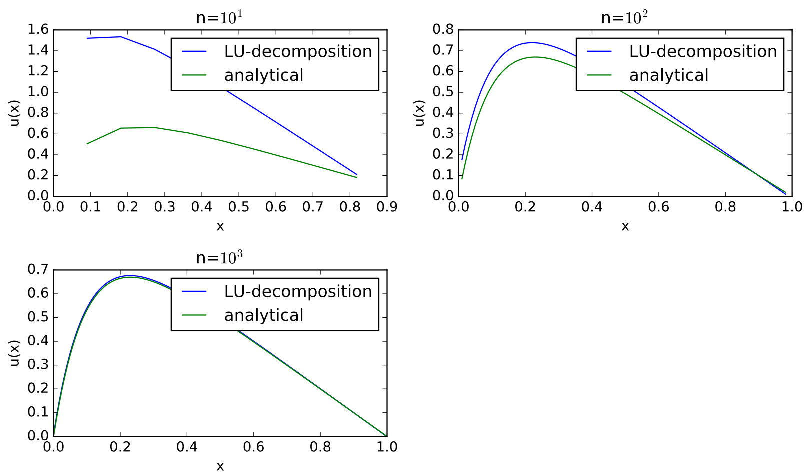

Figure 4: Numerical solution using the LU-decomposition and backward substiution compared to the analytical solution.