Project 2

Alessio Canclini, Filip von der Lippe
(Dated: October 8, 2022)

Github repository: *https://github.com/Fslippe/FYS4150/tree/main/project2*

INTRODUCTION

The main purpose of this project is to familiarize ourselves with the topics of equation scaling, code testing and numerical solutions to eigenvalue problems. The eigenvalue problem we will be working with is a special case of a one-dimensional buckling beam.
Problem 1 demonstrates how the initial second order differential equation is scaled. Problems 2 through 4 focus on writing and testing the Jacobi rotation algorithm which will be used to solve the eigenvalue problem. These do not appear in the report, but can be found in the Github repository linked above. Problem 5 looks at the algorithm's scaling behavior, specifically how the the number of similarity transformations is related to the size N of the input matrix. Finally in problem 6 the constructed code is used to solve the eigenvalue problem and make a plot of the three eigenvectors corresponding to the three lowest eigenvalues, visualizing the solutions to the differential equation for the buckling beam.

We are working with:

- A horizontal beam of length L.
- Let $u(x)$ be the vertical displacement of the beam at horizontal position x, with $x \in [0, L]$.
- A force F is applied at the endpoint ($x = L$), directed into the beam, i.e. towards $x = 0$.
- The beam is fastened with pin endpoints, meaning that $u(0) = 0$ and $u(L) = 0$, but the endpoints are allowed to rotate ($u'(x) \neq 0$).

Second order differential equation describing the buckling beam situation:

$$\gamma \frac{d^2u(x)}{dx^2} = -Fu(x) \tag{1}$$

Throughout this project we will be working with the scaled equation:

$$\frac{d^2u(\hat{x})}{d\hat{x}^2} = -\lambda u(\hat{x}) \tag{2}$$

Where $\hat{x} \equiv x/L$ is a dimensionless variable, $\hat{x} \in [0, 1]$ and $\lambda = \frac{FL^2}{\gamma}$.

PROBLEM 1

Using the definition $\hat{x} \equiv x/L$ to show that Eq. 1 can be written as Eq. 2.
First applying the chain rule on the differential operator $\frac{d}{dx}$. This gives us:

$$\frac{d^2}{dx^2} = \frac{d^2}{d\hat{x}^2}\frac{d\hat{x}^2}{dx^2}$$

Then using that $\left(\frac{\hat{x}}{x}\right)^2 = \frac{1}{L^2}$.

$$\frac{d^2}{dx^2} = \frac{d^2}{d\hat{x}^2}\frac{1}{L^2}$$

Putting this into Eq. 1 we have:

$$\gamma\frac{d^2u(x)}{d\hat{x}^2}\frac{1}{L^2} = -Fu(x)$$

Then dividing both sides by γ and multiplying by L^2.

$$\frac{d^2u(x)}{dx^2} = -\frac{FL^2}{\gamma}u(x)$$

Inserting $x = \hat{x}L$ we can rename $u(\hat{x}L)$ to $\hat{u}(\hat{x})$ and again rename this $u(\hat{x})$.

$$\frac{d^2u(\hat{x}L)}{dx^2} = -\frac{FL^2}{\gamma}u(\hat{x}L)$$
$$\frac{d^2u(\hat{x})}{dx^2} = -\frac{FL^2}{\gamma}u(\hat{x})$$

This finally gives us Eq. 2:

$$\frac{d^2u(\hat{x})}{d\hat{x}^2} = -\lambda u(\hat{x})$$

Where $\lambda = \frac{FL^2}{\gamma}$.

PROBLEM 2

PROBLEM 3

PROBLEM 4

PROBLEM 5

We will now look at how many similarity transformations are needed before a result where all non-diagonal matrix elements are close to zero is reached.

Running the program for different choices of N gives us the following scaling data in figure 1. We obeserve that the number of required transformations T are square proportional to the matrix size N. So $T = N^2$.

Figure 1 demonstrates that the program shows close to the same scaling behavior for both tridiagonal and dense matrixes. This is as expected, since the tridiagonal matrix is is no longer tridiagonal after the first Jacobi rotation of the algorithm. Therfore the tridiagonal matrix is treated just like a dense matrix after a few rotations in this case.

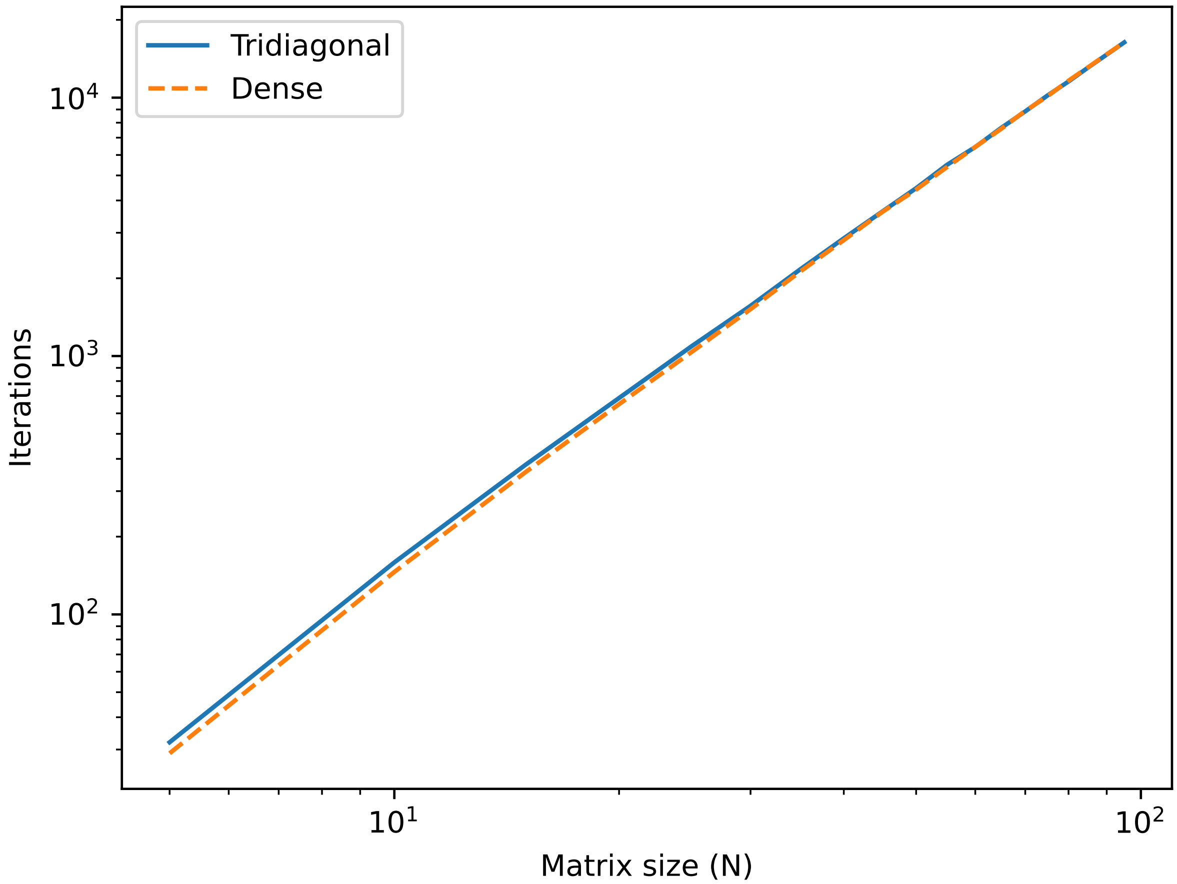
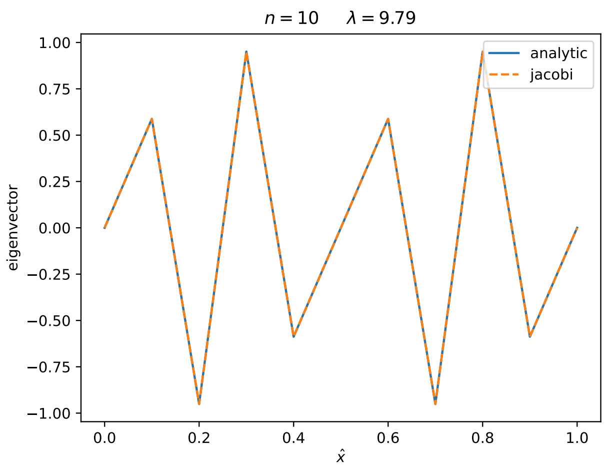


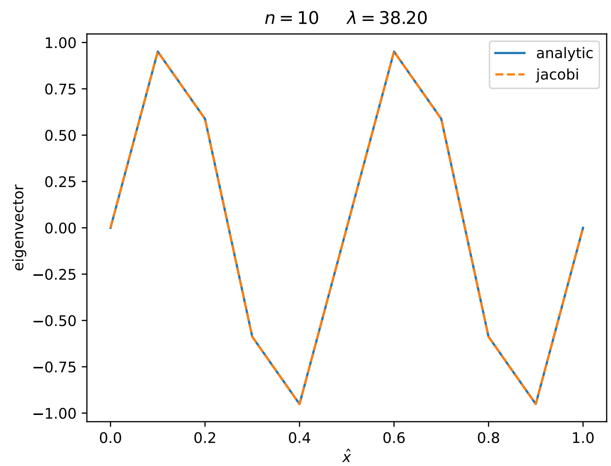
Figure 1: Transformation scaling data for a matrixes on a Log scale.

PROBLEM 6

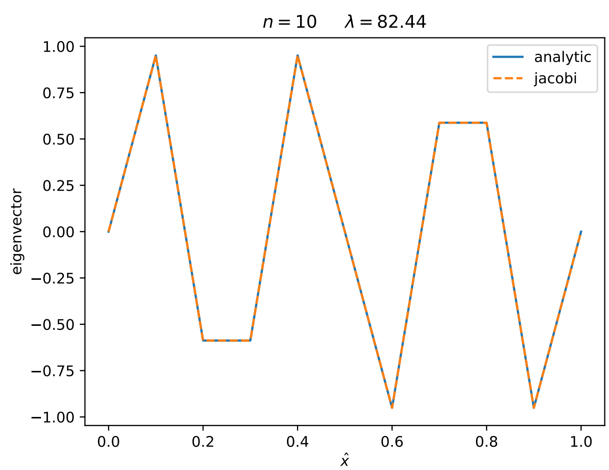
a) Solving the equation $\mathbf{A}\tilde{\mathbf{v}} = \lambda\tilde{\mathbf{v}}$ for a discretization of $\hat{x}$ with steps $n = 10$, steps gives us the three eigenvectors in figure 2



(a)



(b)



(c)

Figure 2: Eigenvectors corresponding to the three lowest eigenvalues for $n = 10$. Plots show the vector v_i elements against the corresponding positions

b) Figure 3 shows the same plots for discretization of with $n = 100$ steps.

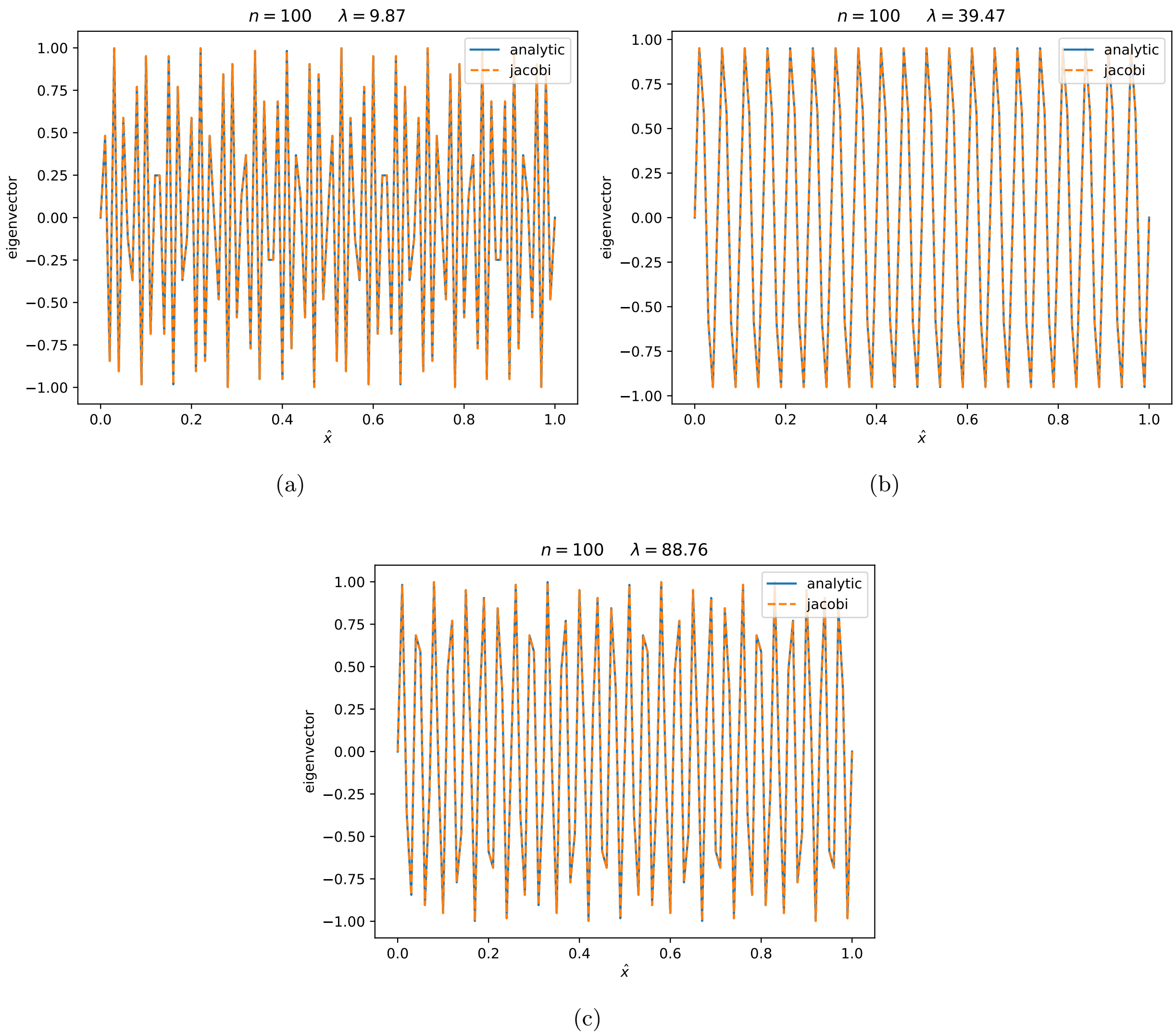


Figure 3: Eigenvectors corresponding to the three lowest eigenvalues for $n = 100$.