

## Project 2

Alessio Canclini, Filip von der Lippe  
(Dated: September 27, 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.

In problem 1 we will show how our initial second order differential equation is scaled. Problems 2 through 4 focus on writing and testing the Jacobi rotation algorithm we will be using to solve our eigenvalue problem. These do not appear in the report, but can be found in the Github repository linked above. In problem 5 we look at the algorithm's scaling behavior, specifically how the number of similarity transformations is related to the size  $N$  of the input matrix. Finally in problem 6 we use the constructed code to solve our eigenvalue problem and make a plot of the three eigenvectors corresponding to the three lowest eigenvalues, visualizing the solutions to the differential equation for our buckling beam.

We are working with:

- A horizontal beam of length  $L$ .
- We 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 our buckling beam situation:

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

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

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

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.

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

Dividing both sides by  $\gamma$ .

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

Multiplying both sides by  $L^2$ .

$$\begin{aligned}\frac{d^2 u(x) L^2}{dx^2} &= -\frac{FL^2}{\gamma} u(x) \\ \frac{d^2 u(x)}{d\frac{x^2}{L^2}} &= -\frac{FL^2}{\gamma} u(x)\end{aligned}$$

Using that  $\hat{x} \equiv x/L$  so that  $\hat{x}^2 \equiv x^2/L^2$ .

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

We also know that  $x = \hat{x}L$ , so  $u(x) = u(\hat{x}L) = u(\hat{x}) + u(L) = u(\hat{x})$ . Since  $u(L) = 0$ . Finally, using  $\lambda = \frac{FL^2}{\gamma}$  gives us the dimensionless Eq. 2:

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

## PROBLEM 5

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

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

In figure 1 we see that our program shows the close to the same scaling behavior for both tridiagonal and dense matrixes. This is as expected, since our tridiagonal matrix is no longer tridiagonal after the first jacobi rotation of our algorithm. Therefore the tridiagonal matrix is treated just like a dens matrix after a few rotations in this case.

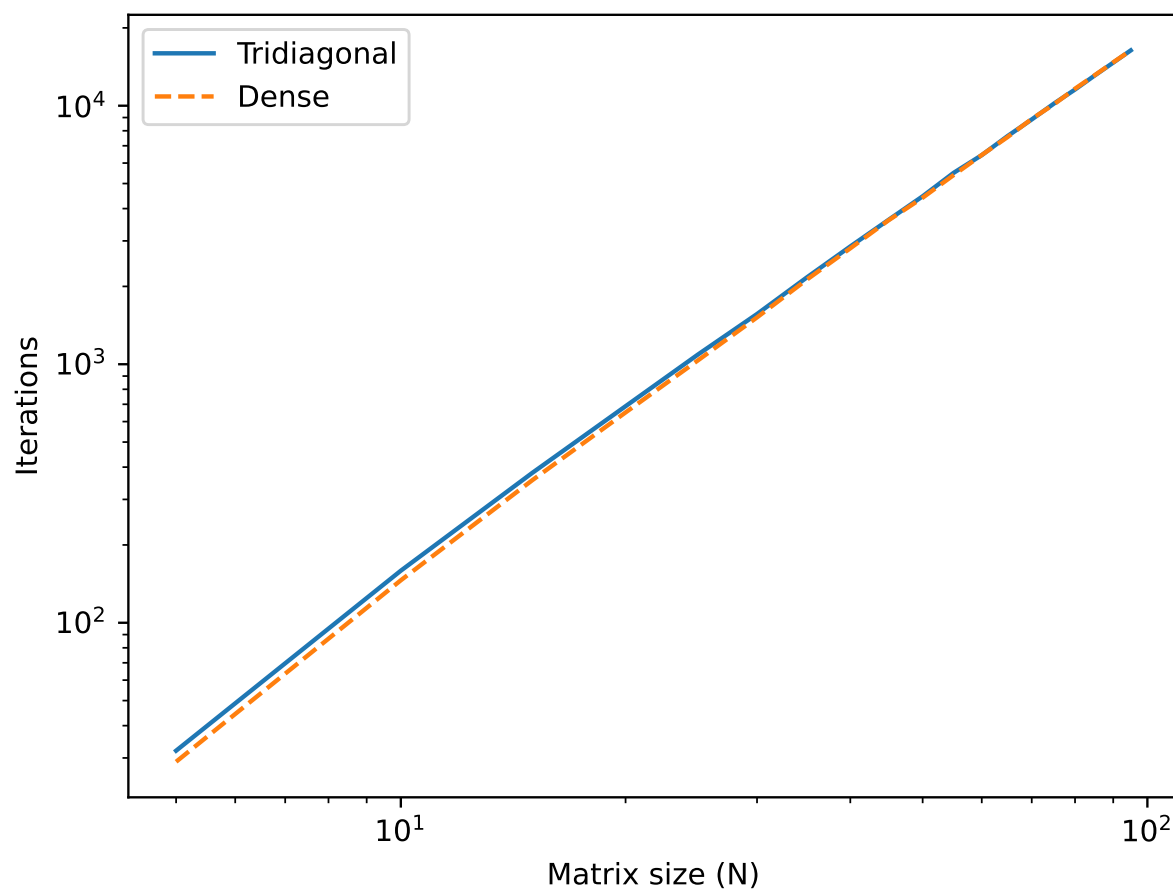


Figure 1: Transformation scaling data for a matrixes on a Log scale.

### 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

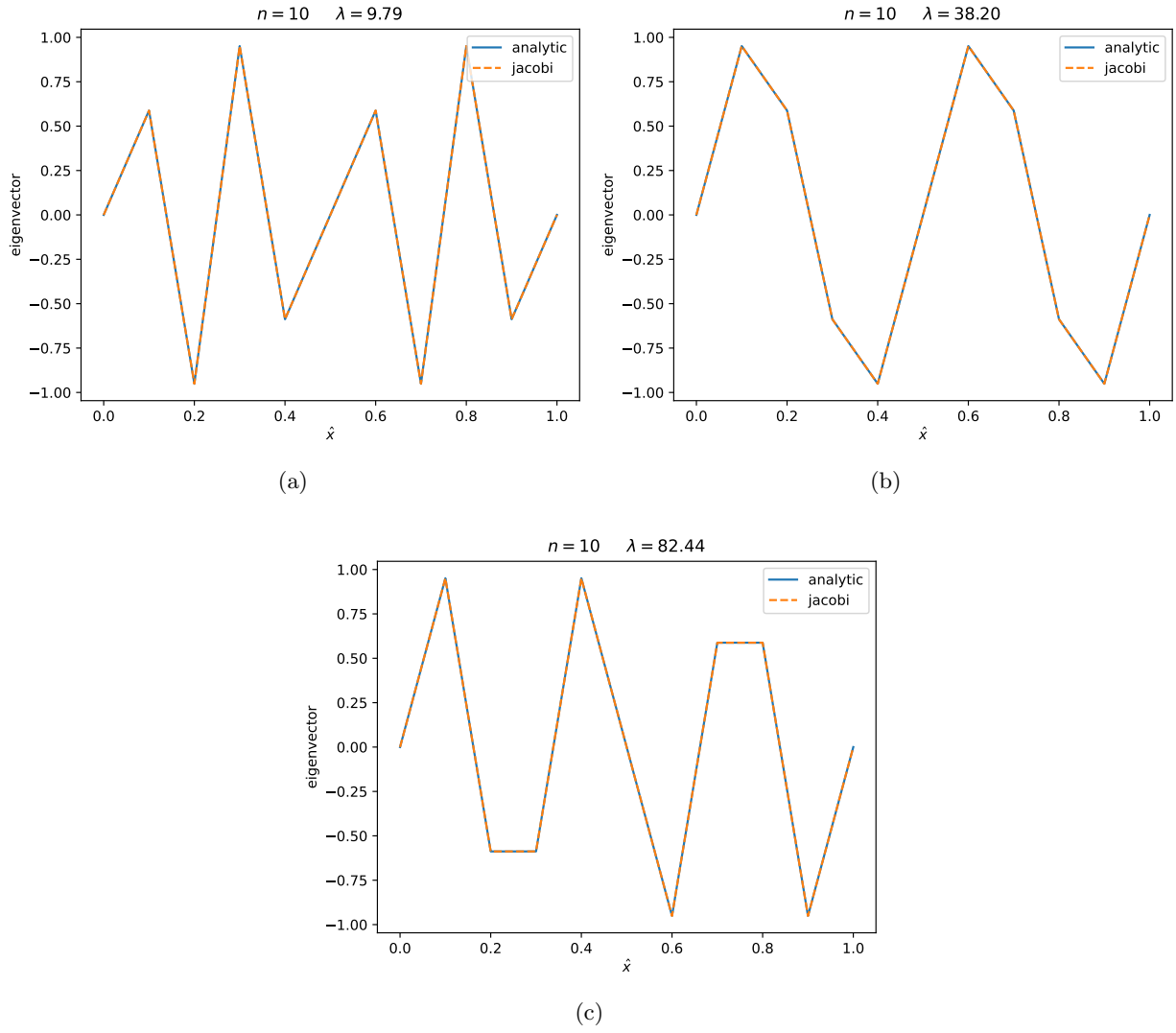


Figure 2: Eigenvectors corresponding to the three lowest eigenvalues for  $n = 10$ . Shows the vector  $v_i$  elements against the corresponding positions

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

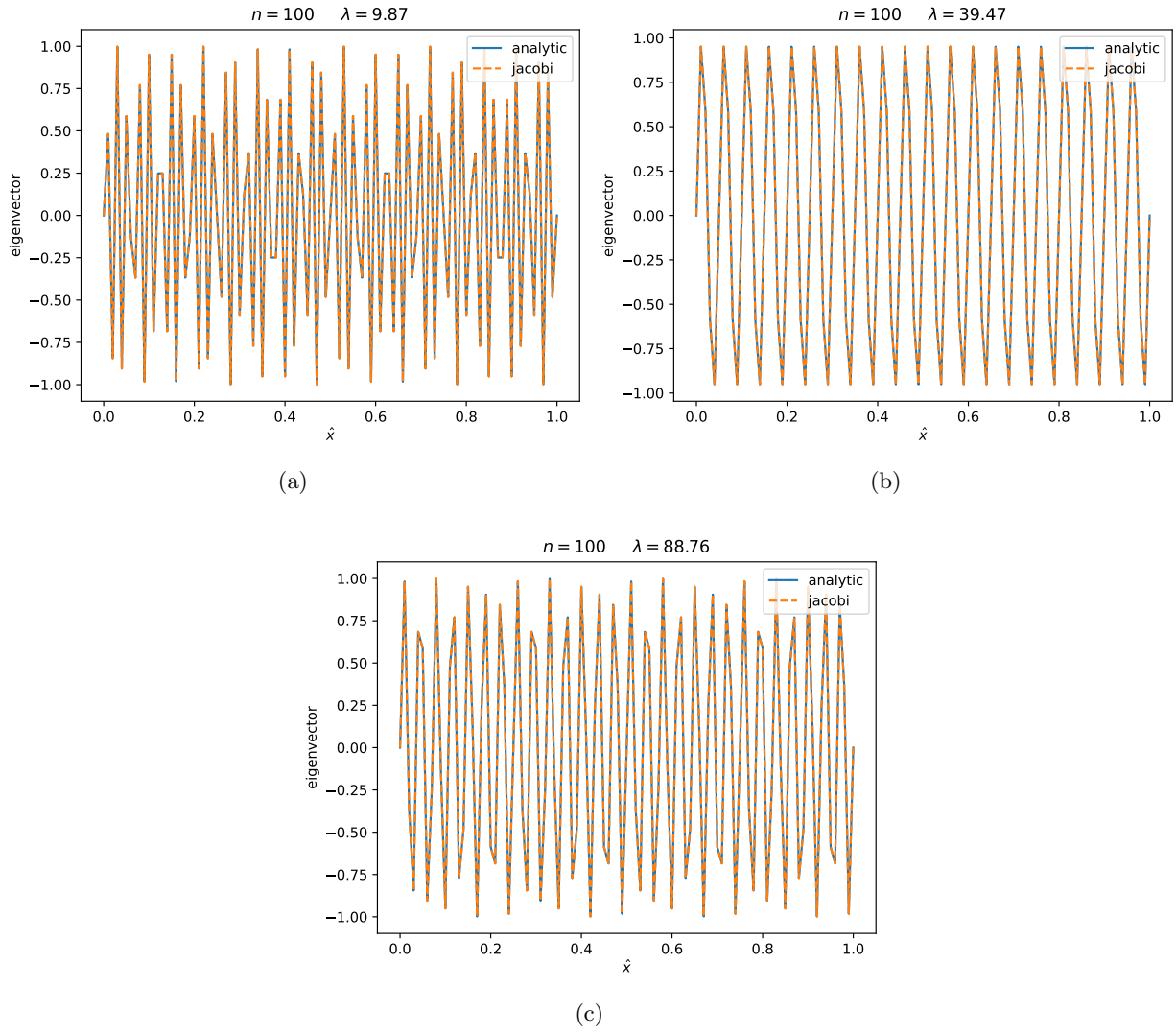


Figure 3: Eigenvectors corresponding to the three lowest eigenvalues for  $n = 100$ .