Computational Physics: Project 2

Nigar Abbasova, Leah Hansen, Adam Jakobsen and Daniel Haas
https://github.com/Adamjakobsen/FYS4150/tree/main/Project2

Problem 1

Given $\hat{x} \equiv x/L$, we have that $d\hat{x} = (1/L)dx$. Therefore, the second-order differential operator becomes:

$$\begin{aligned}\frac{d^2}{dx^2} &= \frac{d}{d\hat{x}}\frac{d\hat{x}}{dx}\frac{d}{d\hat{x}}\frac{d\hat{x}}{dx} \\ &= \frac{1}{L^2}\frac{d^2}{d\hat{x}^2}\end{aligned}$$

And it follows

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

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

Problem 2

Our intentions were to write a code that sets up a $N = 6$ tridiagonal matrix, solves $\mathbf{A}\vec{v} = \lambda\vec{v}$ and checks that the analytical eigenvalues and eigenvectors agree with the numerical ones. The results were as expected.

The code can be found on `https://github.com/Adamjakobsen/FYS4150/tree/main/Project2` **under the main.cpp file, as will be the case for all of the written programs.**

Problem 3 & 4

We wrote a program that finds the largest absolute value off-diagonal element of a $N \times N$ matrix. We assumed the matrix to be symmetric. We then compared our numerical results to the analytical

matrix A, given by

$$A = \begin{bmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & -0.7 & 0 \\ 0 & -0.7 & 1 & 0 \\ 0.5 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The results were as expected.

We then expanded the code by implementing the Jacobi's rotation algorithm to solve the eigenvalue problem:

$$\mathbf{A}\vec{v} = \lambda\vec{v}$$

This time we tested our code for a $N = 6$ matrix, with successful results.

Problem 5

We plot the relationship between number of iterations and size of the matrix in Figure 1.

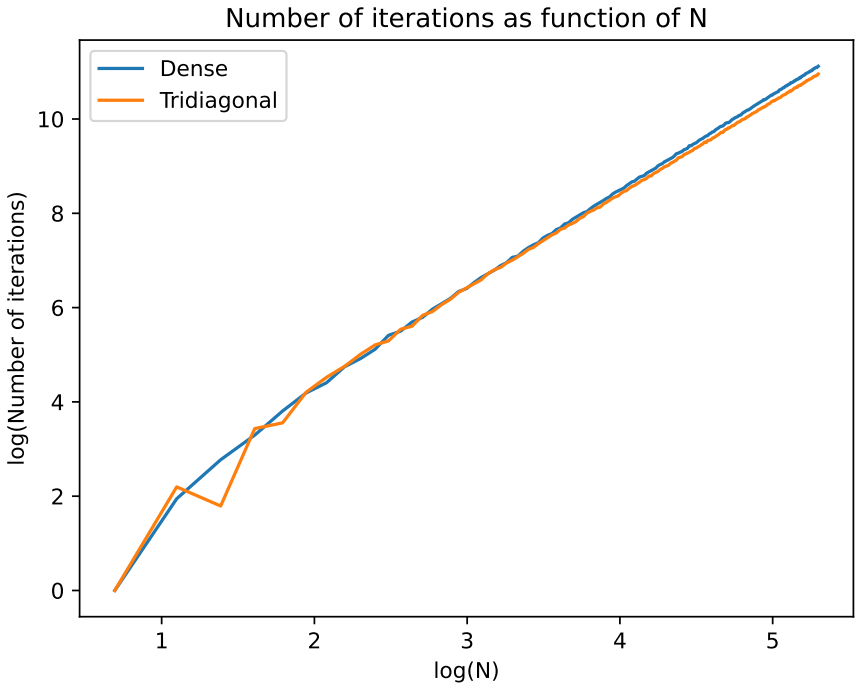

Figure 1: Number of iterations as a function of matrix size, N, (log-scale) until $N = 200$. "Dense" is the number of iterations for a dense matrix of size N, and "Tridiagonal" is the number of iterations for a tridiagonal matrix of size N.

We see that the number of iterations increase following an apparent quadratic profile with an increase in the size N of the matrix.

From Figure 1 we see that the slope of the log-log graph for the given relation is about 2. The log of our function, y, has the following relation with $\log(N)$

$$\log(y) = A\log(N) \Rightarrow y = N^A \tag{4}$$

The apparent slope of 2 in 1 corroborates with the initial supposition that the algorithm should have, in general, a quadratic relationship with the number of iterations. Indeed, since we have estimated our slope to be 2, we get from 4 that $A = 2$, meaning

$$y = N^2$$

Contrary to our expectations the relationship remains about the same for both a tridiagonal and a dense matrix. The Jacobi's rotation algorithm goes through the matrix about the same amount of times before reaching a diagonalized matrix regardless of whether the matrix is dense or not.

Intuitively, one would think that a matrix with many zero elements, like the tridiagonal matrix, would be much faster to run through than a dense matrix. However, the Jacobi's rotation algorithm still goes through the zero elements of the matrix and updates them like the rest of the off-diagonal elements. This means that the size of the matrix is the main determining factor deciding the number of iterations.

Hence, the algorithm is only slightly slower for a dense matrix where you would need more iterations to turn the non-zero elements zero.

For our choice of epsilon, we had $\varepsilon = 10^{-3}$. Any larger values would make the step sizes too small, leading us to eventually divide by zero in our algorithm. This is still sufficient as our smallest eigenvalue is ≈ 10, having off-diagonal values of order 10^{-3} which will still give us a diagonally dominant matrix.

Problem 6

We choose to plot the dimensionless solution to our eigenvalue problem, including the boundary solutions. We plot the lowest energy solutions, corresponding to the three smallest eigenvalues. See Figure 2 for the result with $n = 10$ steps, and Figure 3 for the result with $n = 100$ steps.

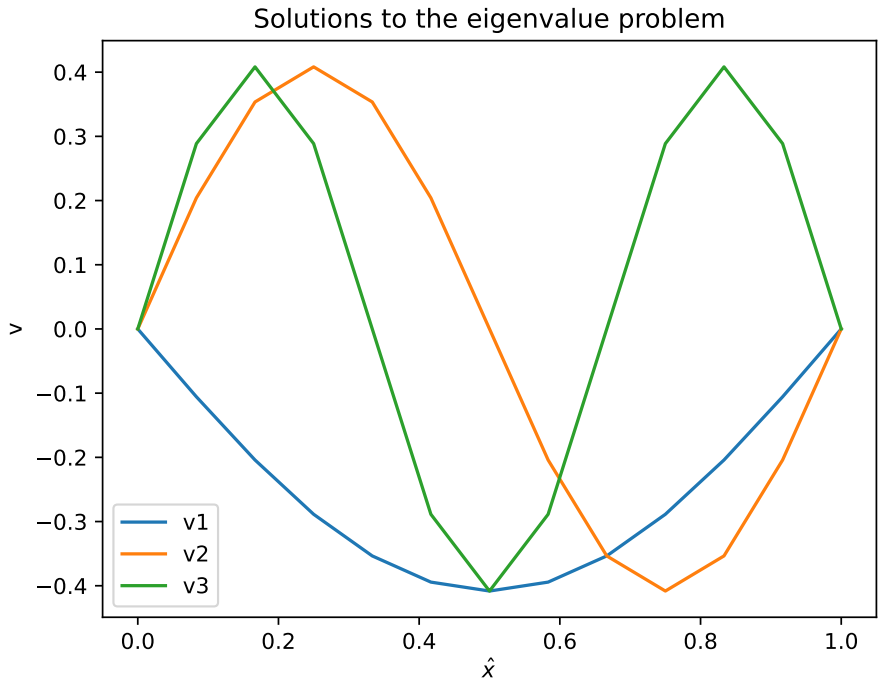

Figure 2: Solutions with lowest eigenvalues (energies) plotted with $n = 10$ steps. Dimensionless $\hat{x}$ is defined as $\hat{x} \equiv x/L$.

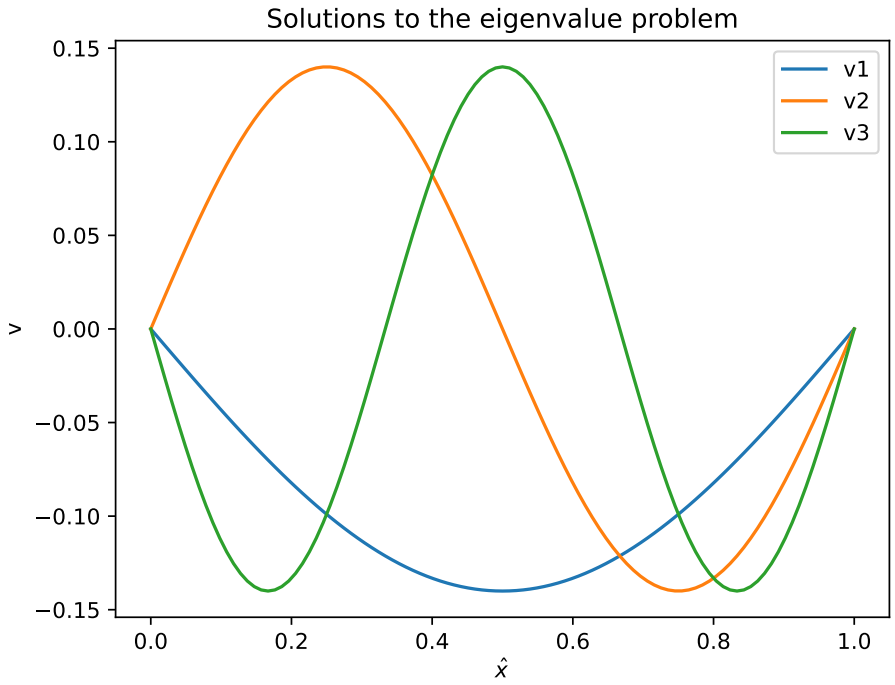

Figure 3: Solutions with lowest eigenvalues (energies) plotted with $n = 100$ steps. Dimensionless $\hat{x}$ is defined as $\hat{x} \equiv x/L$.