

Project 2 - Computational Physics FYS3150/FYS4150

Émilie Valle, Josef Ayman
(Dated: September 23, 2024)

GitHub repo: [github.uio.no/josefam/FYS3150](https://github.com/josefam/FYS3150)

I. PROBLEM 1

We have to show that eq(1) can be written as eq(2).

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

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

with $\hat{x} = \frac{x}{L}$ and $\lambda = \frac{FL^2}{\gamma}$

We can begin from the end to back up in the beginning.

$$\frac{d^2 u(\hat{x})}{d\hat{x}^2} = -\lambda u(\hat{x}) \quad (3)$$

$$\frac{d^2 u(\frac{x}{L})}{d(\frac{x}{L})^2} = -\frac{FL^2}{\gamma} u(\frac{x}{L}) \quad (4)$$

$$\gamma \frac{d^2 u(x)}{dx^2} \frac{L^2}{L} = -F \frac{L^2}{L} u(x) \quad (5)$$

$$\gamma \frac{d^2 u(x)}{dx^2} = -Fu(x) \quad (6)$$

II. PROBLEM 2

As asked, we have set up the tridiagonal 6x6 matrix and solves $A \vec{v} = \lambda \vec{v}$ using Armadillo. Then, we have checked that the eigenvalues and eigenvectors agrees with the analytical result for $N = 6$. The thing with scaling of vectors has been done with the normalization as discussed in the introduction of the project.

And then, we have found that the checking is good and all the values are quite the same.

III. PROBLEM 3

a) The function coded here has permitted to find the largest off-diagonal element and to give its absolute value using Armadillo.

b) With the following matrix :

$$\begin{bmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & -0.7 & 0 \\ 0 & -0.7 & 1 & 0 \\ 0.5 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

we can see that, the function has found -0.7 as largest off-diagonal element and 0.7 as its absolute value, which is the right answer.

IV. PROBLEM 4

a) The function coded here, using the code snippets can solve the following equation with Jacobi's rotation algorithm. Note that, the suggested code structure has been used but not the helper functions.

$$A\vec{v} = \lambda\vec{v} \quad (8)$$

b) With this code, we have check if the eigenvalues and eigenvectors agree with the analytical results for $N = 6$.

We can see in both case that, the difference between the analytical and the Jacobi's method is very small. For the eigenvectors, we have a difference like 1×10^{-12} and for the eigenvalues, 1×10^{-14} .

V. PROBLEM5

a) With the Jacobi's method, we can run our program with different choices of N and see the number of required transformation we need to solve eq(8).

Here the table of results : (with i as iterations)

$$\begin{cases} N = 10, i = 137 \\ N = 20, i = 655 \\ N = 40, i = 2762 \\ N = 80, i = 11214 \\ N = 160, i = 45319 \\ N = 320, i = 182906 \end{cases}$$

We can then, do the plot with a linear scale and with a log scale.

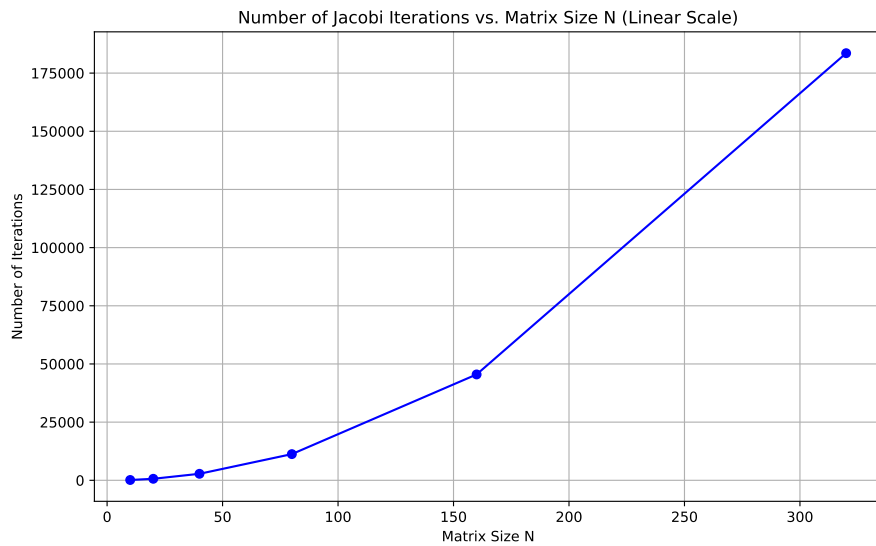


FIG. 1. Plot of Jacobi's iterations in linear scale

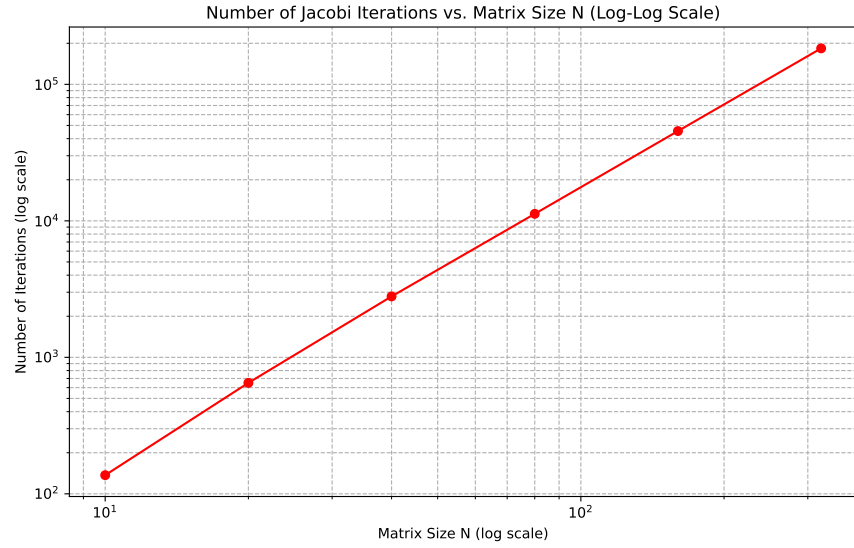


FIG. 2. Plot of Jacobi's iterations in log scale

And then, try to find a function to fit with the plot. With this, we have the scale of the number of required transformations with the matrix N size when we use our code to solve eq(8).

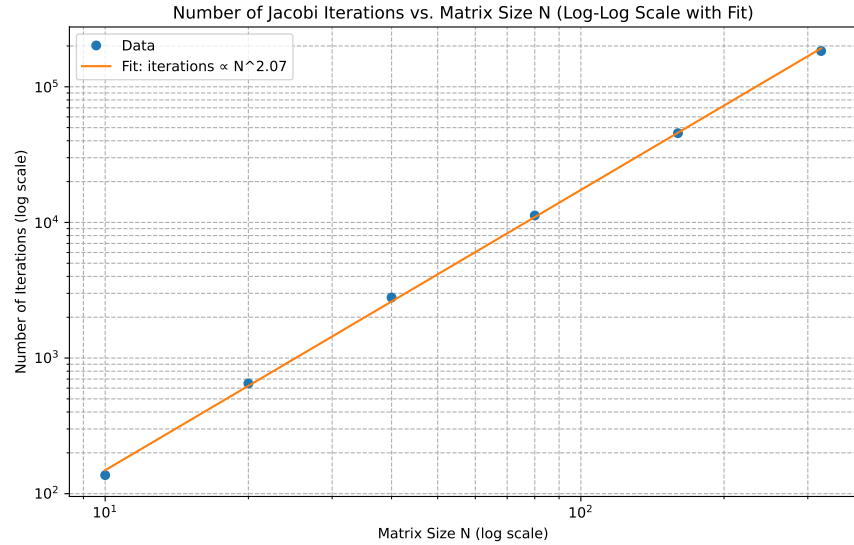


FIG. 3. Fit function with log scale

So, the scale is $N * 2.07$ in log scale.

b) For a dense symmetric matrix, we would expect the scaling behavior to be similar or even worse than $O(N^2)$, due to the larger number of off-diagonal elements that need to be eliminated.

- The Jacobi algorithm does not take advantage of the sparsity of the initial matrix.

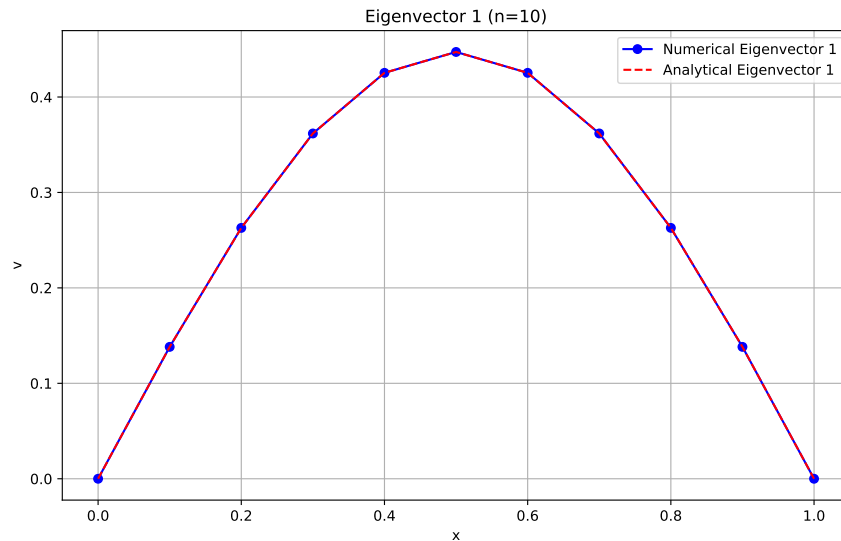
- Even starting with a sparse matrix, the rotations introduce non-zero elements (fill-in effect), effectively making the matrix dense as the algorithm progresses.
- Therefore, the scaling behavior for a dense matrix would also be quadratic or worse.

Scaling behavior:

- In a dense matrix, the number of off-diagonal elements is $N(N - 1)/2$, which scales as $O(N^2)$.
- Each Jacobi rotation zeroes out one off-diagonal element.
- In the worst case, we may need to apply a number of rotations proportional to the number of off-diagonal elements.
- Each rotation involves updating $O(N)$ elements in the matrix and the eigenvector matrix.
- The total cost is roughly $O(N^3)$ for dense matrices.

VI. PROBLEM 6

a) With our Jacobi code, we have made a plot of the three eigenvectors corresponding to the three lowest eigenvalues comparing with the analytical answers.



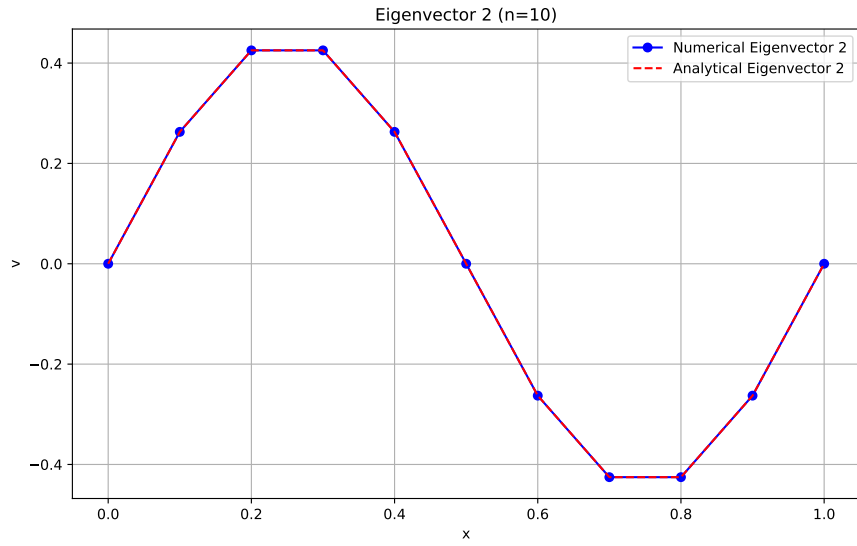


FIG. 4. Second eigenvector with $n = 10$

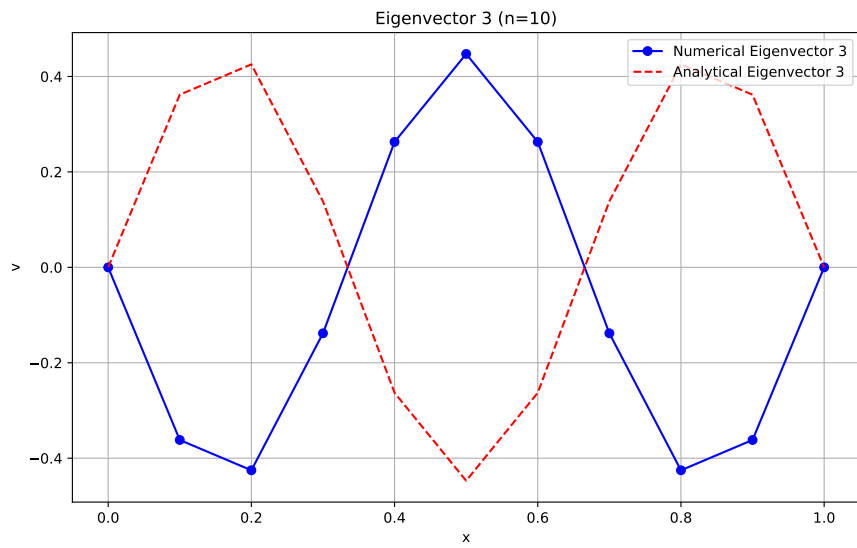


FIG. 5. Third eigenvector with $n = 10$

b) For the $n = 100$, here the plots.

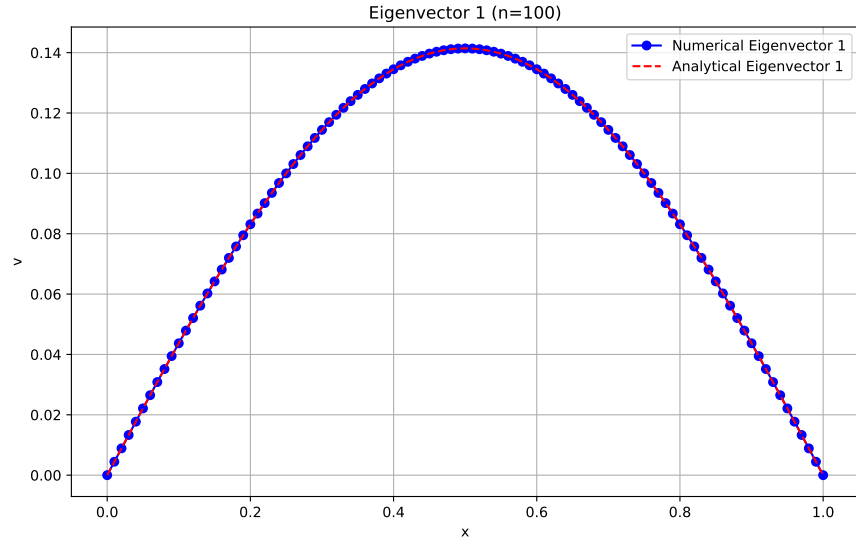


FIG. 6. First eigenvector with $n = 100$

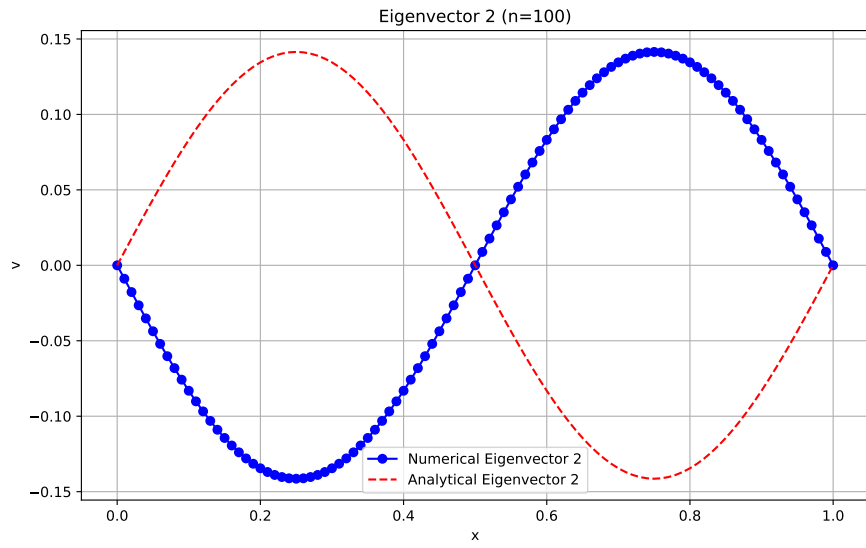


FIG. 7. Second eigenvector with $n = 100$

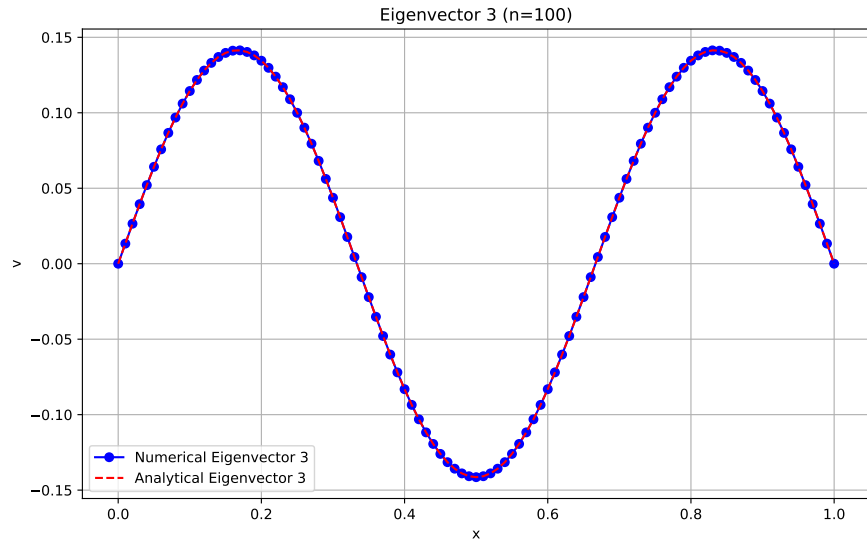


FIG. 8. Third eigenvector with $n = 100$