FYS3150 Project 2

Andreas Isene

(Dated: September 27, 2021)

https://github.com/andris96/FYS3150

PROBLEM 1

The second order differential equation is given as:

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

To scale the equation we use: $\hat{x} = x/L$, in order to implement this we need to find out what $\frac{d^2}{dx^2}$ is in terms of $\hat{x}$. We start with the first derivative:

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

Then squaring this gives:

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

Putting this back into eq. **??** and defining $\lambda \equiv \frac{FL^2}{\gamma}$ we can finally arrive at the desired result:

$$\frac{\gamma}{L^2}\frac{d^2u}{d\hat{x}^2} = -Fu \rightarrow \frac{d^2u}{d\hat{x}^2} = -\lambda u$$

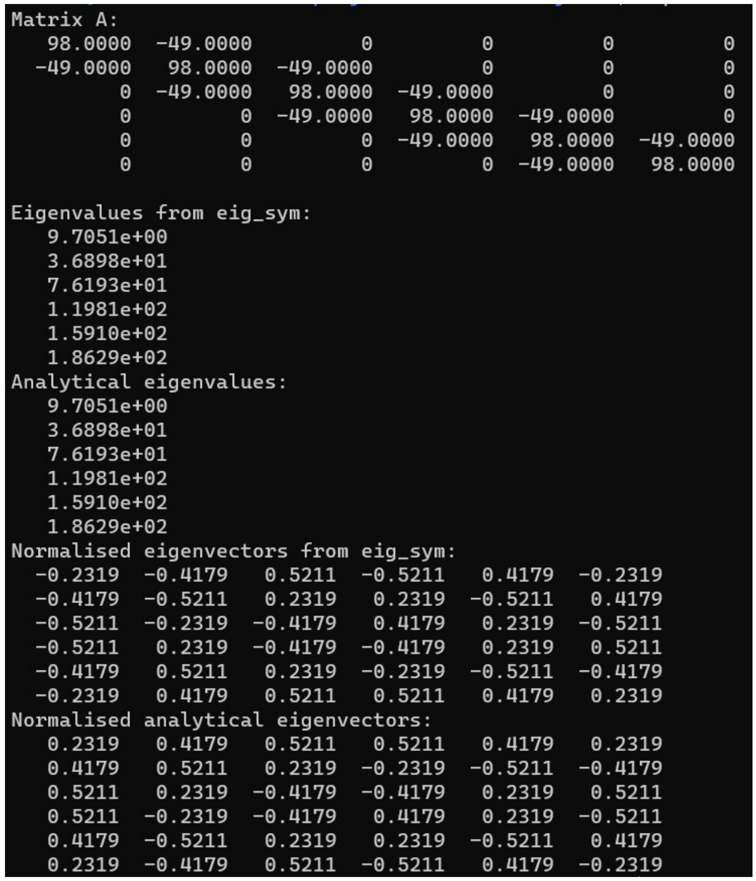
PROBLEM 2

To show that U preserves orthonormality we must show that $\vec{w}_i^T\vec{w}_j = \delta_{ij}$. This can easily be shown by using a property of transposition. We know that $(AB)^T = B^TA^T$ which gives us:

$$\vec{w}_i^T\vec{w}_j = (U\vec{v}_i)^T U\vec{v}_j = \vec{v}_i^T U^T U\vec{v}_j = \vec{v}_j^T I\vec{v}_i = \delta_{ij}$$

PROBLEM 3

Figure 1 is a screenshot from the output of the program which is called problem3_main.exe. The solutions from the analytical calculation seems to agree with the solutions given by eig_sym. The vectors are in opposite direction, but that does not matter as eigenvectors can be scaled with a constant c and still have the same properties.



```
Matrix A:
   98.0000  -49.0000         0         0         0         0
  -49.0000   98.0000  -49.0000         0         0         0
         0  -49.0000   98.0000  -49.0000         0         0
         0         0  -49.0000   98.0000  -49.0000         0
         0         0         0  -49.0000   98.0000  -49.0000
         0         0         0         0  -49.0000   98.0000

Eigenvalues from eig_sym:
   9.7051e+00
   3.6898e+01
   7.6193e+01
   1.1981e+02
   1.5910e+02
   1.8629e+02
Analytical eigenvalues:
   9.7051e+00
   3.6898e+01
   7.6193e+01
   1.1981e+02
   1.5910e+02
   1.8629e+02
Normalised eigenvectors from eig_sym:
  -0.2319  -0.4179   0.5211  -0.5211   0.4179  -0.2319
  -0.4179  -0.5211   0.2319   0.2319  -0.5211   0.4179
  -0.5211  -0.2319  -0.4179   0.4179   0.2319  -0.5211
  -0.5211   0.2319  -0.4179  -0.4179   0.2319   0.5211
  -0.4179   0.5211   0.2319  -0.2319  -0.5211  -0.4179
  -0.2319   0.4179   0.5211   0.5211   0.4179   0.2319
Normalised analytical eigenvectors:
   0.2319   0.4179   0.5211   0.5211   0.4179   0.2319
   0.4179   0.5211   0.2319  -0.2319  -0.5211  -0.4179
   0.5211   0.2319  -0.4179  -0.4179   0.2319   0.5211
   0.5211  -0.2319  -0.4179   0.4179   0.2319  -0.5211
   0.4179  -0.5211   0.2319   0.2319  -0.5211   0.4179
   0.2319  -0.4179   0.5211  -0.5211   0.4179  -0.2319
```

FIG. 1. Output from the program problem3_main.exe

PROBLEM 4

Figure 2 is a screenshot from the output of the program problem4_main.exe. It seems to correctly identify the element with the highest absolute value, and also gives the correct indices (starting from 0). Since it is a symmetric matrix we only need to consider the upper triangle.

```
Matrix A is given as:
   1.0000         0         0    0.5000
        0    1.0000   -0.7000         0
        0   -0.7000    1.0000         0
   0.5000         0         0    1.0000
The maximum off-diagonal value of A is: -0.7
Which is the element with indicies k = 1, l = 2
```

FIG. 2. Output from the program problem4_main.exe

PROBLEM 5

Figure 3 is a screenshot from the output of the program problem5_main.exe. As we can see, the result is the same as the previous results from problem 3.

```
convergence was reached before hitting maximum number of iterations.
Number of iterations: 35
eigenvectors:
   0.2319  -0.4179  -0.5211   0.5211   0.4179  -0.2319
   0.4179  -0.5211  -0.2319  -0.2319  -0.5211   0.4179
   0.5211  -0.2319   0.4179  -0.4179   0.2319  -0.5211
   0.5211   0.2319   0.4179   0.4179   0.2319   0.5211
   0.4179   0.5211  -0.2319   0.2319  -0.5211  -0.4179
   0.2319   0.4179  -0.5211  -0.5211   0.4179   0.2319
eigenvalues:
   9.7051e+00
   3.6898e+01
   7.6193e+01
   1.1981e+02
   1.5910e+02
   1.8629e+02
```

FIG. 3. Output from the program problem5_main.exe

PROBLEM 6

Figure 4 is a screenshot from the output of the program problem6_main.exe it shows the number of iterations as N goes from 10 to 70 with an increment of 5 each time. Figure 5 shows a plot of the values, with a x^2 curve for comparison. It seems like the scaling is of order N^2, but it could be important to let N increase even further to see what happens.

As for the case where A is a dense matrix I assume that there would not be much difference in the scaling behaviour, since every time the tridiagonal matrix is rotated other elements which are 0 becomes non-zero. Which is also why we need to have an epsilon to make sure they are close enough to zero. In other words, after quite few iterations the tridiagonal matrix will be transformed into a dense matrix.

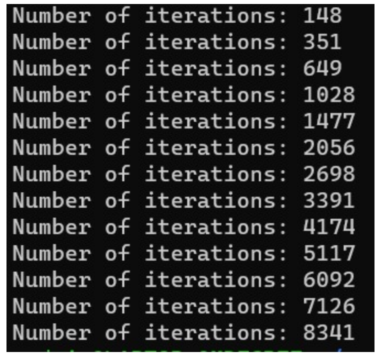


FIG. 4. Output from the program problem6_main.exe

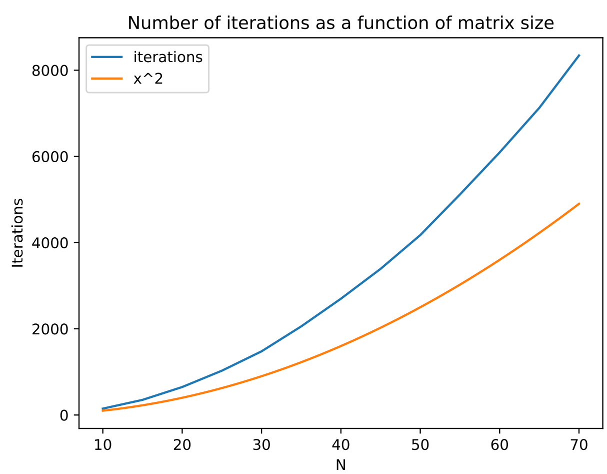


FIG. 5. Plot shows how many iterations and therefore how many rotation transformations are needed to solve the differential equation. The orange line is a plot of x^2 to compare.

PROBLEM 7

Figure 6 and 7 show the eigenvectors with n = 10 and n = 100 respectively. We can see that the third eigenvector is showing opposite values as the analytical one, which is just due to the fact that the normalization does not account for the sign of the vector. As we can se, with increasing n the values from the Jacobi's rotations becomes more similar to the analytical ones.

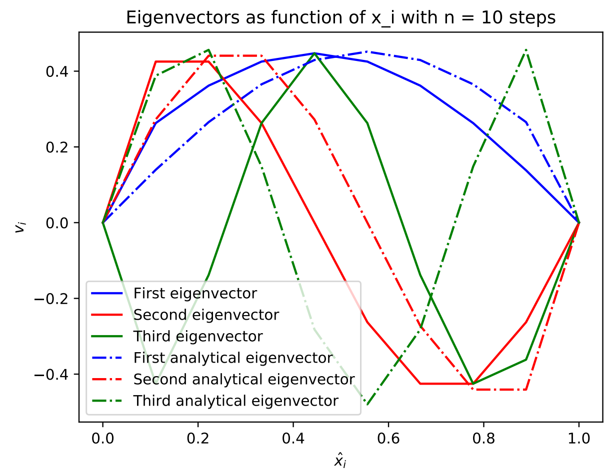


FIG. 6. Plot shows the three lowest eigenvectors as a function of x_i with n = 10

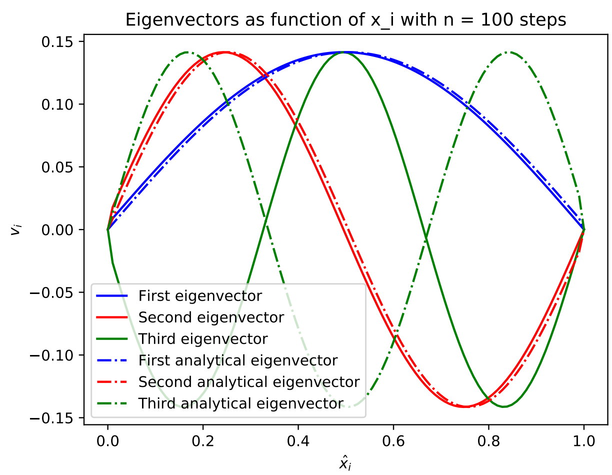


FIG. 7. Plot shows the three lowest eigenvectors as a function of x_i with n = 100