Project 2

Malin Eriksen
(Dated: October 9, 2022)

https://github.com/malineri/fys3150/tree/main/project_2

PROBLEM 1

In this project we will be looking at a horizontal beam. The beam will be of length L, and a force F will be applied in the endpoint of this beam. This will be described by the second-order differential equation (1).

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

We begin with scaling this formula into a dimensionless equation, by changing the x to a unitless variable $\hat{x} = \frac{x}{L}$. The derivation term then becomes,

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

We have a second order differential equation, so we have to do this operation twice. Adding this to equation(1) we get the formula,

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

Which we rearrange slightly,

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

Now we introduce a new variable lambda $\lambda = \frac{FL^2}{\gamma}$ then we find that the dimensionless equation of our first equation can be written as,

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

PROBLEM 2

To solve this problem we will be using matrices. We write a short program to set up a tridiagonal NxN matrix **A**, when N = 6. We want this matrix to solve the classic eigenvector, eigenvalue problem $\mathbf{A}\vec{v} = \lambda\vec{v}$ in the same way we formulated our dimensionless equation(2).[**?**] We being with creating an algorithm with a general function that determines a tridiagonal matrix. This algorithm we see in Algorithm 1.

Algorithm 1 Creating function that takes values from diagonal and makes a tridiagonal matrix.

```
arma::mat create_tridiagonal(int n, double a, double d, double e)
arma::mat A = arma::mat(n, n, arma::fill::eye);    ▷ n x n identity matrix
int N = 6    ▷ We begin with setting the length of the matrix N = 6
A(0, 0) = d; A(1, 0) = e; A(0, 1) = a;    ▷ Manually filling in values of A, Could potentially be changed to a for loop for bigger matrices.
return A;
```

We now want to put in values of N, a, d and e, to determine our exact values. We want those to be $a, e = -1/h^2$ and $d = -2/h^2$. And we also want the program to print the eigenvalues and eigenvectors. A program returnng the matrix, its eigenvalue and its corresponding eigenvectors is written in Algorithm 2.

Algorithm 2 Our main containing the values of our matrix, and printing the matrix, its eigenvector and eigenvalues.

```
int main()
int N = 6.; float h = 1.; float a = (-1.)/(h*h); float d = (2.)/(h*h);    ▷ Setting values we use in the matrix
arma::mat A = create_tridiagonal(N, a, d, a);    ▷ create matrix A from function
arma::vec eigval;
arma::mat eigvec;
arma::eig_sym(eigval, eigvec, A);
int width = 18; int prec = 10;    ▷ Parameters for output formatting
std::cout << "#" << std::setw(width-1) << A
<< std::endl;
std::cout << "#" << std::setw(width-1) << eigvec
<< std::endl;
std::cout << "#" << std::setw(width-1) << eigval
<< std::endl;
return 0;
```

Now that we have found the eigenvalues and corresponding eigenvectors we print the results, to make it easier to compare the analytical and numerical solutions we remove the $/h^2$ part of the a, d and e values, so that we use the simple tridiagonal matrix with 2 on the diagonal and -1 on the upper and lower diagonal. This gives us the results we can see in table(1).

TABLE I. Eigenvalues and corresponding eigenvectors of a tridiagonal matrix A

Eigenvalues λ	Eigenvectors $\vec{v}$
0.1981	[0.2319, 0.4179, 0.5211, 0.5211, 0.4179, 0.2319]
0.7530	[-0.4179, -0.5211, -0.2319, 0.2319, 0.5211, 0.4179]
1.5550	[0.5211, 0.2319, -0.4179, -0.4179, 0.2319, 0.5211]
2.4450	[0.5211, -0.2319, -0.4179, 0.4179, 0.2319, -0.5211]
3.2470	[0.4179, -0.5211, 0.2319, 0.2319, -0.5211, 0.4179]
3.8019	[-0.2319, 0.4179, -0.5211, 0.5211, -0.4179, 0.2319]

We can check if these values are correct by comparing the results to the analytical results with the formulas

$$\lambda^i = d + 2a\cos\left(\frac{i\pi}{N+1}\right)$$

$$\vec{v}^i = \left[\sin\left(\frac{i\pi}{N+1}\right), \sin\left(\frac{2i\pi}{N+1}\right), ..., \sin\left(\frac{Ni\pi}{N+1}\right)\right]^T$$

Where i = 1, ... , N. We make a program that solves this algorithm for our N = 6 case. This program is called *analytical_2.py*, and the algorithm is added to the github repository. We get the same values on the eigenvalues, but we get different eigenvectors, something that tells us there must be a small mistake in one of the programs. If time I will go back and look at potential mistakes in the program. In table(2) we see the values we get from the analytical solution.

TABLE II. Eigenvalues and corresponding eigenvectors of a tridiagonal matrix A

Eigenvalues λ	Eigenvectors $\vec{v}$
0.1981	[0.43388374, -0.78183148, 0.97492791, -0.97492791,0.78183148, -0.43388374]
0.7530	[0.78183148, -0.97492791, 0.43388374, 0.43388374, -0.97492791, 0.78183148]
1.5550	[0.97492791, -0.43388374, -0.78183148, 0.78183148, 0.43388374, -0.97492791]
2.4450	[0.97492791, 0.43388374, -0.78183148, -0.78183148, 0.43388374, 0.97492791]
3.2470	[0.78183148, 0.97492791, 0.43388374, -0.43388374, -0.97492791, -0.78183148]
3.8019	[0.43388374, 0.78183148, 0.97492791, 0.97492791, 0.78183148, 0.43388374]

PROBLEM 3

-