FYS3150 Project 2

Viktor Braaten Gulbrandsen, Katchi Senthil Ganesh
(Dated: November 19, 2026)

Code is available at https://github.uio.no/viktorgu/fys3150, in the folder `project2/`. Each problem has its own folder `problemN/` with a `main.cpp` and a `makefile` where `make` compiles and runs.

BEAM BUCKLING SETUP

Consider a horizontal beam of length L, and let $x \in [0, L]$ be the horizontal position along it. We define $u(x)$ as the vertical displacement of the beam at x when a force F presses on the endpoint $x = L$, directed into the beam. The beam is pinned at both ends, so $u(0) = 0$ and $u(L) = 0$. It is given by:

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

where γ is a constant.

I. PROBLEM 1

We want to show that Eq. (1) can be written in the dimensionless form

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

with $\hat{x} \equiv x/L$ and $\lambda = FL^2/\gamma$.

Solution Let $f(x) = \hat{x} = x/L$. By definition ($f(x) = \hat{x}$), we write $u(\hat{x}) = u(f(x)) = u(\frac{x}{L})$. Rewriting this way, we have the equation

$$\gamma \frac{d^2u(\frac{x}{L})}{dx^2} = -Fu\left(\frac{x}{L}\right). \tag{3}$$

Let us now consider the left hand side. To first illustrate how the second derivative transforms, take a function ψ of $f(x)$ and differentiate once by the chain rule:

$$\frac{d\psi(f(x))}{dx} = \frac{d\psi(f(x))}{df(x)}\frac{df(x)}{dx} = \frac{d\psi(f(x))}{df(x)}\frac{1}{L}, \tag{4}$$

since $df/dx = 1/L$. Differentiating a second time, the factor $1/L$ is a constant and the chain rule applies again to what remains:

$$\frac{d^2\psi(f(x))}{dx^2} = \frac{d}{dx}\left(\frac{d\psi(f(x))}{df(x)}\frac{1}{L}\right) = \frac{d^2\psi(f(x))}{df(x)^2}\frac{df(x)}{dx}\frac{1}{L} = \frac{1}{L^2}\frac{d^2\psi(f(x))}{df(x)^2}. \tag{5}$$

Each derivative with respect to x thus gives one factor of $1/L$. Setting $\psi = u$ and $f(x) = \hat{x}$ in Eq. (1) gives

$$\frac{\gamma}{L^2}\frac{d^2u(\hat{x})}{d\hat{x}^2} = -Fu(\hat{x}), \tag{6}$$

and moving the constants to the right hand side,

$$\frac{d^2u(\hat{x})}{d\hat{x}^2} = -\frac{FL^2}{\gamma}u(\hat{x}). \tag{7}$$

With $\lambda = FL^2/\gamma$ this is Eq. (2). The variable $\hat{x}$ runs over $[0, 1]$, and the boundary conditions become $u(0) = u(1) = 0$.

II. PROBLEM 2

A. Set up

1. Discretization and Taylor

.

From the physics, the bending resistance of the beam has to balance the sideways push of the load, which for a beam with stiffness γ under a force F along its axis reads

$$\gamma \frac{d^2u(x)}{dx^2} = -Fu(x), \qquad x \in [0, L],$$

where $u(x)$ is the vertical displacement at position x. In our case this is used in the scaled form from Problem 1,

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

with $\hat{x} \equiv x/L$, $\hat{x} \in [0, 1]$, and $\lambda = FL^2/\gamma$, and the beam is pinned at both ends, so $u(0) = u(1) = 0$. Given the force, we want to use this equation to numerically approximate $u(\hat{x})$.

The first step is defining the grid. We discretize this equation by evaluating it over a grid of $n + 1$ points,

$$\hat{x}_i = \hat{x}_{\text{start}} + i\,\frac{\hat{x}_{\text{end}} - \hat{x}_{\text{start}}}{n}, \qquad i = 0, 1, \ldots, n, \tag{8}$$

which for the beam gives $\hat{x}_i = ih$ with step $h = (1 - 0)/n = 1/n$. We write v_i for our approximation to $u(\hat{x}_i)$. At an interior point the second derivative is replaced by the three-point formula

$$\left.\frac{d^2u}{d\hat{x}^2}\right|_{\hat{x}_i} \approx \frac{u(\hat{x}_i - h) - 2u(\hat{x}_i) + u(\hat{x}_i + h)}{h^2} = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2}, \tag{9}$$

which follows from adding the Taylor expansions of $u(\hat{x}_i + h)$ and $u(\hat{x}_i - h)$ around $\hat{x}_i$. Evaluating the beam equation point by point over the grid then gives

$$\begin{aligned}
v_0 &= 0 && \text{pinned end, } \hat{x}_0 = 0 \\
\frac{v_0 - 2v_1 + v_2}{h^2} &= -\lambda v_1 && \text{the equation at } \hat{x}_1 = h \\
\frac{v_1 - 2v_2 + v_3}{h^2} &= -\lambda v_2 && \text{the equation at } \hat{x}_2 = 2h \\
&\vdots \\
\frac{v_{n-2} - 2v_{n-1} + v_n}{h^2} &= -\lambda v_{n-1} && \text{the equation at } \hat{x}_{n-1} = (n-1)h \\
v_n &= 0 && \text{pinned end, } \hat{x}_n = 1
\end{aligned}$$

The $n - 1$ interior lines stacked on top of each other form a column vector on each side,

$$\frac{1}{h^2}\begin{pmatrix} v_0 - 2v_1 + v_2 \\ v_1 - 2v_2 + v_3 \\ \vdots \\ v_{n-2} - 2v_{n-1} + v_n \end{pmatrix} = -\lambda \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \end{pmatrix}. \tag{10}$$

Aligning the indices in a matrix, so that every v_i from v_0 to v_n gets its own column, the left hand side becomes a matrix times the full grid vector,

$$\frac{1}{h^2}\begin{pmatrix} 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & -2 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{n-2} \\ v_{n-1} \\ v_n \end{pmatrix} = -\lambda \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \end{pmatrix}, \tag{11}$$

with $n-1$ rows and $n+1$ columns. The first column multiplies v_0 and the last multiplies v_n, and both are zero because the beam is pinned, so these two columns contribute nothing and are dropped. What remains is a square $(n-1)\times(n-1)$ matrix. Multiplying both sides by -1 moves the minus sign into the matrix,

$$\frac{1}{h^2}\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \end{pmatrix} = \lambda \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \end{pmatrix}. \tag{12}$$

2. Eigenvalues and Eigenvectors

If we let the factor $1/h^2$ be part of the matrix, i.e.

$$\mathbf{A} \equiv \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}, \qquad \vec{v} \equiv \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \end{pmatrix}, \tag{13}$$

we see that this can be written as

$$\mathbf{A}\vec{v} = \lambda\vec{v}, \tag{14}$$

The first step is solving for the λ that allow a $\vec{v} \neq \vec{0}$. Since $\lambda\vec{v} = \lambda\mathbf{I}\vec{v}$ with $\mathbf{I}$ the identity, moving it over gives

$$(\mathbf{A} - \lambda\mathbf{I})\,\vec{v} = \vec{0}. \tag{15}$$

If $\mathbf{A}-\lambda\mathbf{I}$ is invertible, its columns are linearly independent and the only solution of $(\mathbf{A}-\lambda\mathbf{I})\vec{v} = \vec{0}$ is $\vec{v} = \vec{0}$. We know, however, that the beam does bend under a large enough load, so a non-zero $\vec{v}$ has to exist for some λ. Those are the λ at which $\mathbf{A}-\lambda\mathbf{I}$ is singular, i.e. has no inverse. Subtracting $\lambda\mathbf{I}$ only changes the diagonal of $\mathbf{A}$, from $2/h^2$ to $2/h^2-\lambda$, and for most values of λ the resulting matrix can still be inverted. For exactly N values it cannot, which is the condition

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0. \tag{16}$$

The determinant of an $N\times N$ matrix is a sum of products of N entries, one from each row, and here λ sits in N of the entries, so the left hand side is a polynomial of degree N in λ,

$$\det(\mathbf{A} - \lambda\mathbf{I}) = c_N\lambda^N + c_{N-1}\lambda^{N-1} + \cdots + c_1\lambda + c_0 = c_N\left(\lambda^{(1)} - \lambda\right)\left(\lambda^{(2)} - \lambda\right)\cdots\left(\lambda^{(N)} - \lambda\right), \tag{17}$$

so the eigenvalues $\lambda^{(1)},\ldots,\lambda^{(N)}$ are the N roots of this polynomial. so the eigenvalues $\lambda^{(1)},\ldots,\lambda^{(N)}$ are the N roots of this polynomial. Physically, these are the only N values of the dimensionless load $\lambda = FL^2/\gamma$ at which the pinned beam can hold a non-zero shape. For our $\mathbf{A}$, which is tridiagonal with constant diagonal $d = 2/h^2$ and constant off-diagonals $a = -1/h^2$, they are known in closed form [1]:

$$\lambda^{(j)} = d + 2a\cos\left(\frac{j\pi}{N+1}\right), \qquad j = 1,\ldots,N, \tag{18}$$

$$\vec{v}^{(j)} = \left(\sin\frac{j\pi}{N+1},\ \sin\frac{2j\pi}{N+1},\ \ldots,\ \sin\frac{Nj\pi}{N+1}\right)^T. \tag{19}$$

B. Solution

In this problem we wish to set up up `arma::eig_sym`*, verify it runs and check this solution in relation to the analytical one.*

The program `problem2/main.cpp` sets up $\mathbf{A}$ for $N = 6$ ($n = 7$, $h = 1/7$), solves the eigenvalue problem with `arma::eig_sym`, and computes the analytical eigenvalues and eigenvectors for the same N. The first and last row of $\mathbf{A}$ have only two non-zero entries, so they are set outside the loop that fills the remaining rows. The analytical eigenvectors are normalised with `arma::normalise`, and the sign of each column is flipped when its dot product with the corresponding numerical column is negative.

Table I gives the eigenvalues from the two methods and Table II the eigenvectors. The largest absolute difference is 5.7×10^{-14} for the eigenvalues and 8.9×10^{-16} for the eigenvector elements, i.e. at the level of double precision round-off. The matrix is thus set up correctly, and both the library routine and the analytical formulas describe the same problem.

TABLE I. Eigenvalues of $\mathbf{A}$ for $N = 6$, from `arma::eig_sym` and from the analytical formula, Eq. (18).

j	$\lambda^{(j)}$ (Armadillo)	$\lambda^{(j)}$ (analytical)
1	9.7051	9.7051
2	36.898	36.898
3	76.193	76.193
4	119.81	119.81
5	159.10	159.10
6	186.29	186.29

TABLE II. Unit-norm eigenvectors of $\mathbf{A}$ for $N = 6$, one per column, from `arma::eig_sym`. The normalised analytical eigenvectors, Eq. (19), agree element by element to 8.9×10^{-16}.

i	$\vec{v}^{(1)}$	$\vec{v}^{(2)}$	$\vec{v}^{(3)}$	$\vec{v}^{(4)}$	$\vec{v}^{(5)}$	$\vec{v}^{(6)}$
1	0.2319	−0.4179	0.5211	0.5211	0.4179	−0.2319
2	0.4179	−0.5211	0.2319	−0.2319	−0.5211	0.4179
3	0.5211	−0.2319	−0.4179	−0.4179	0.2319	−0.5211
4	0.5211	0.2319	−0.4179	0.4179	0.2319	0.5211
5	0.4179	0.5211	0.2319	0.2319	−0.5211	−0.4179
6	0.2319	0.4179	0.5211	−0.5211	0.4179	0.2319

III. PROBLEM 3

A. Set up

In Problem 2 we solved the eigenvalue problem with `arma::eig_sym`. We now wish to find a method solving it ourselves.

Any real matrix has a singular value decomposition, $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, with $\mathbf{U}$ and $\mathbf{V}$ orthogonal and $\mathbf{\Sigma}$ diagonal with non-negative entries. For our tridiagonal matrix this reads

$$\frac{1}{h^2}\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{20}$$

Now, since the tridiagonal matrix is symmetric, $\mathbf{A}^T = \mathbf{A}$, and positive definite (all its eigenvalues are positive by Eq. (18)), the two rotations coincide, $\mathbf{U} = \mathbf{V}$. The SVD thus reduces to the spectral theorem, i.e. there exists one orthogonal matrix $\mathbf{S}$, with $\mathbf{S}^T\mathbf{S} = \mathbf{I}$, such that

$$\mathbf{S}^T\mathbf{A}\mathbf{S} = \mathbf{D}, \qquad \mathbf{D} = \mathrm{diag}\left(\lambda^{(1)}, \ldots, \lambda^{(N)}\right). \tag{21}$$

Similar to $\mathbf{U}$ and $\mathbf{V}$ in the SVD, $\mathbf{S}$ represents a rotation. Since $\mathbf{S}^T = \mathbf{S}^{-1}$, multiplying Eq. (21) by $\mathbf{S}$ from the left gives

$$\mathbf{A}\mathbf{S} = \mathbf{S}\mathbf{D}. \tag{22}$$

so the columns of $\mathbf{S}$ are the eigenvectors and the diagonal of $\mathbf{D}$ the eigenvalues. Jacobi's rotation method applies a series of rotations $\mathbf{S}_1, \mathbf{S}_2, \ldots$ to $\mathbf{A}$,

$$\mathbf{A}^{(1)} \equiv \mathbf{A}, \qquad \mathbf{A}^{(m+1)} = \mathbf{S}_m^T \mathbf{A}^{(m)} \mathbf{S}_m, \tag{23}$$

B. Solution

We implement `max_offdiag_symmetric` in `problem3/main.cpp`. The matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & -0.7 & 0 \\ 0 & -0.7 & 1 & 0 \\ 0.5 & 0 & 0 & 1 \end{pmatrix} \tag{24}$$

returns -0.7 at $(k, l) = (1, 2)$.

IV. PROBLEM 4

A. Solution

We implement `jacobi_rotate` and `jacobi_eigensolver` in `problem4/main.cpp`, with the rotation formulas from [1]. The solver rotates away the largest off-diagonal element until it is below $\epsilon = 10^{-8}$, then reads the eigenvalues from the diagonal and the eigenvectors from the columns of the accumulated rotation matrix, sorted ascending. For $N = 6$ it converges after 35 rotations. The largest difference from the analytical solution is 1.7×10^{-13} for the eigenvalues and 5.9×10^{-11} for the eigenvector elements, and the values reproduce Tables I and II.

V. PROBLEM 5

A. Solution

The same Jacobi algorithm as in Problem 4 is run for $N = 5, 10, 20, 40, 80$ in `problem5/main.cpp`. Table III gives the rotations needed for each N, and Figures 1 and 2 show them against N.

TABLE III. Rotations needed for the largest off-diagonal element to fall below $\epsilon = 10^{-8}$.

N	rotations	time (s)
5	31	2.7×10^{-6}
10	153	1.6×10^{-5}
20	660	2.0×10^{-4}
40	2731	3.0×10^{-3}
80	11260	4.9×10^{-2}

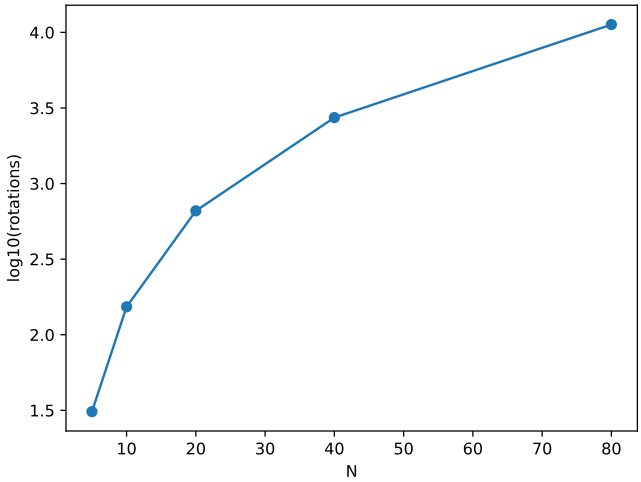

FIG. 1. Rotations to convergence against N, log scale on the y-axis.

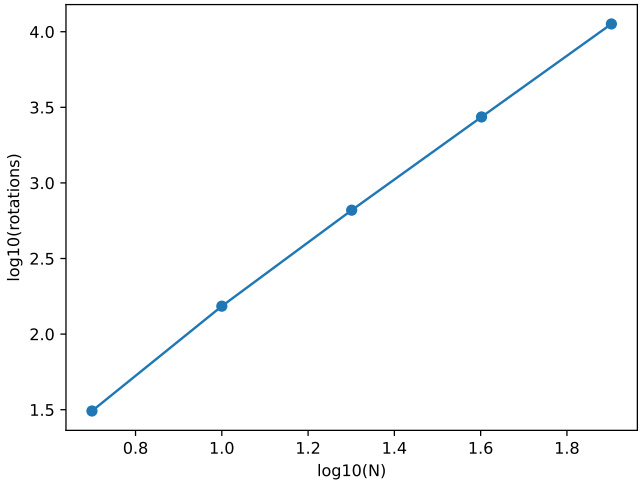

FIG. 2. Rotations to convergence against N, log scale on both axes.

As we see from the plots, the number of rotations scales as N^2. This is we also see by the log-log plot being linear with slope 2. For a dense matrix we expect the same scaling, since the first rotation already fills in elements outside the three diagonals.

VI. PROBLEM 6

A. Solution

The same Jacobi algorithm is run for $n = 10$ and $n = 100$ steps in `problem6/main.cpp`. The three eigenvectors with the lowest eigenvalues, extended with the boundary points $v_0 = v_n = 0$, are plotted against $\hat{x}_i$ in Figures 3 and 4 together with the analytical eigenvectors, Eq. (19).

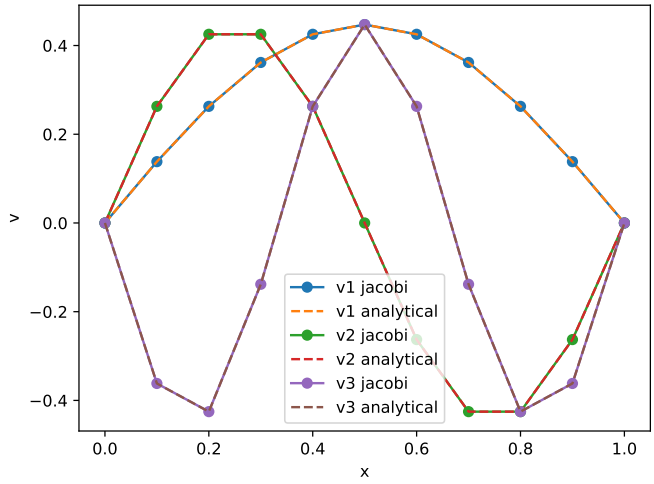

FIG. 3. The three lowest eigenvectors for $n = 10$, Jacobi (markers) and analytical (dashed).

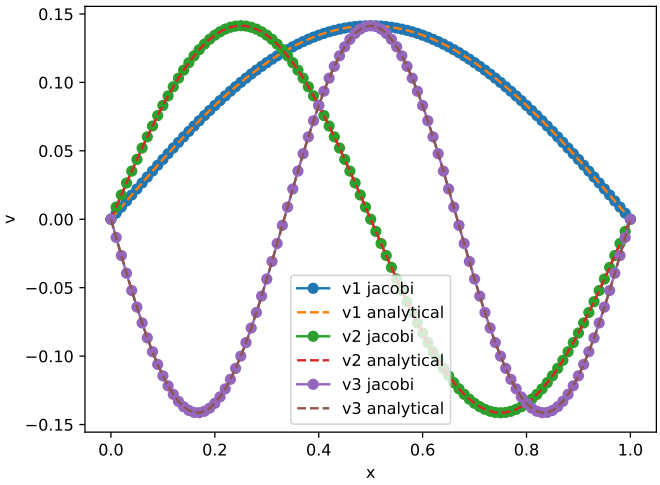

FIG. 4. The three lowest eigenvectors for $n = 100$, Jacobi (markers) and analytical (dashed).

As we see, the numerical and analytical eigenvectors lie on top of each other for both n. The eigenvectors are the first three buckling modes of the beam, $\sin(j\pi\hat{x})$ sampled at the grid points.

---

[1] A. Kvellestad, *FYS3150 Lecture Notes*, v1.20.1 (2026), https://github.com/anderkve/FYS3150/blob/master/lecture_notes/2026/lecture_notes.pdf