ME 594 - Numerical Methods: Homework 5

October 4, 2021

Four Problems: (10 pts each) Note: A * next to a problem number indicates you must electronically submit a ".m" file.

1*. An important eigenvalue problem related to mechanical engineering has to do with the buckling of a rod under an applied load. The differential equation and associated boundary conditions governing the problem are

$$EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + Py = 0, \qquad y(0) = 0, \quad y(L) = 0,$$

assuming the rod is pinned at both ends. Here E is the modulus of elasticity, I is the area moment of inertia of the cross section, and P is the applied load. Here we will take the length $L = 1$ in the x-direction, and y represents lateral buckling deflections. Later in the course, we will discuss the process of discretizing this equation, i.e. turning the differential equation into a system of algebraic equations. It can be shown that the discretized problem can be written like an eigenvalue problem:

$$Ay = \lambda y,$$

where A is the tridiagonal matrix given on Canvas. The vector y represents the interior deflections, i.e. all points other than the boundary points that have prescribed boundary conditions. It can be shown that the column will first buckle when the applied load reaches the value $P = -\lambda EIh^2$, where λ (a negative number) is the <u>smallest</u> eigenvalue of the matrix A, and $h = L/(n+1)$ is the spacing between discretized nodes; the matrix A has size $n \times n$. The file on Canvas indicates that $n = 25$ was chosen for this problem.

Use the Inverse Power Method to compute the smallest eigenvalue of this matrix, and compare the result to the theoretical value of $\lambda_{\text{theoretical}} = -\pi^2/(n+1)^2$. The eigenvector corresponding to this eigenvalue illustrates the shape of this first buckling mode. Plot the buckling mode for $0 \leq x \leq L = 1$. Remember that you have to add a zero to the front and back of the computed eigenvector to include the boundary conditions, i.e. $y(x=0) = 0$ and $y(x=1) = 0$.

As discussed in the lecture, do not calculate A^{-1} directly as part of your solution, as that is unstable. Use a tolerance of 10^{-6} for the convergence criterion of the Inverse Power Method.

2*. Write a program to calculate all of the eigenvalues of the matrix

$$\begin{bmatrix} 2.5 & -2.0 & 2.5 & 0.5 \\ 0.5 & 5.0 & -2.5 & -0.5 \\ -1.5 & 1.0 & 3.5 & -2.5 \\ 2.0 & 3.0 & -5.0 & 3.0 \end{bmatrix}$$

using Givens Rotations and QR-factorization and iteration. Note, you do not have to use this program to find the eigenvectors. You can confirm the answer using the Matlab function `eig`. Also, the book and other references do not indicate a well defined convergence criterion so just run a large enough number of iterations to assume convergence; I ran 40 iterations for this problem.

3*. Write a program to calculate all of the eigenvalues of the matrix

$$\begin{bmatrix} 7 & 6 & -3 \\ -12 & -20 & 24 \\ -6 & -12 & 16 \end{bmatrix}$$

using Householder Reflections and QR-factorization and iteration. Note, you do not have to use this program to find the eigenvectors. You can confirm the answer using the Matlab function `eig`. Also, the book and other references do not indicate a well defined convergence criterion so just run a large enough number of iterations to assume convergence; I ran 20 iterations for this problem.

4*. Write a program using Jacobi's method to determine all the eigenpairs (both eigenvalues and corresponding eigenvectors) of the real, symmetric matrix

$$\begin{bmatrix} -8 & 16 & 23 & -13 \\ 16 & 9 & 2 & 3 \\ 23 & 2 & 1 & -23 \\ -13 & 3 & -23 & -7 \end{bmatrix}.$$

Continue to run sweeps until $\mathrm{off}(A) < \epsilon$, where $\mathrm{off}(A)$ was the norm defined in class and $\epsilon = 10^{-6}$. Report the number of sweeps needed to achieve this tolerance in addition to the eigenpairs.

*** Do not use the intrinsic functions `eig` or `qr` to write the programs. ***