Instructions

In this project, you will study the use of Newton's method for finding an eigenvalue/eigenvector pair of a matrix.

The deliverable for the project should be a written report in which you develop a narrative about the project where you describe the problem and present your code and results. The organization need not specifically follow the exact order of the questions below. Full sentences and paragraphs should be used where appropriate and you should describe your mathematical approach taken. Your paper should look like a journal article *not* homework solutions.

Software developed for the project can be in any language including, but not limited to C, C++, Java, Fortran, MATLAB, Python, or Julia. Feel free to use this as an opportunity to expand your programming skillset.

If you use any resources to help with the problem, *be sure to both cite the resource while also rewriting the details in your own words*. Use of tools such as ChatGPT are expressly forbidden. This is an opportunity to practice and develop your own writing skills.

Finally, you should decide upon and answer a question *not* specifically asked for. Some possible extra questions are given below, but you can make one up yourself, as well, and feel free to discuss this choice with me.

The grade will be based on

- 70% for simply answering the questions,
- 20% for answering this new question of your own,
- 10% for the general quality (although it need not be typeset for full credit here).

Question

Consider an $n \times n$ ***symmetric*** matrix A. We seek an eigenvalue $\lambda \in \mathbb{R}$ and eigenvector $\mathbf{v} \in \mathbb{R}^n$ pair which satisfy

$$A\mathbf{v} = \lambda \mathbf{v}.$$

This is only n nonlinear equations for $n+1$ unknowns (λ and all n components of $\mathbf{v}$). This fact is evidenced by the result that eigenvectors are only unique up to scaling. So, in order to add another equation and to make the eigenvector (almost) unique, we will add the equation

$$1 = \frac{1}{2}\|\mathbf{v}\|_2^2.$$

In order to convert this into the form of a root finding problem, define a new variable $\mathbf{x}$ which is an $n+1$ dimensional vector whose first n entries are $\mathbf{v}$ and whose last entry is λ. Then rewrite the above equations in the form $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ by moving all terms to the left.

1. Determine the full definition of $\mathbf{f}$.
2. Determine the Jacobian of $\mathbf{f}$. It may be helpful to write it in block form.
3. Use the information you found to write some software which runs Newton's method to (approximately) find a solution to $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. This will be much easier with a programming language with built in linear algebra routines such as MATLAB, Python with ScyPy & NumPy, or Julia.
4. Apply your method to finding eigenvalues and eigenvectors of the matrix

   $$A = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}$$

   whose eigenvalues are $\lambda = 2, 4$ and eigenvectors are $\mathbf{v} = [1 \quad 1]^T, [1 \quad -1]^T$, respectively. You may want to choose random starting vectors in order to converge to different solutions. Discuss your results, including convergence rates.
5. Apply your method to at least one other symmetric matrix which is at least 3×3 in size.

Some possible "extra questions" are

1. Write software which uses Broyden's method instead. Apply it to all the same problems.
2. Discuss why we limit to symmetric matrices here and test what happens for various non-symmetric matrices. Perhaps try complex starting vectors to see what happens.
3. If you used a built-in linear system solver (like backslash in MATLAB), implement any method discussed this semester yourself and use it instead.
4. Pick a different auxiliary equation (i.e. not $1 = \frac{1}{2}\|\mathbf{v}\|_2^2$) to add to the system and solve using it. Discuss any consequences on convergence or results.