Spring 2022: Numerical Analysis
Assignment 4 (due April 6, 2022 at 11:59pm ET)

1. **[Rounding errors, 1+2+3pt]** Computers use finite precision to represent real numbers, which leads to rounding. You can see the size of the rounding error for real numbers around 1 using the MATLAB command `eps(1)` or the numpy command `np.spacing(1)`. This number, also called the *machine epsilon*, is $\epsilon = 2.22 \times 10^{-16}\ldots$ for the standard (double precision) representation of numbers in a computer. Try the following experiments in MATLAB (or Python/Octave/Julia).[1]

   (a) Report the analytical/exact result, and the result you get when using your computer for:

   $$a = (1-1) + 10^{-16}, \quad b = 1 - (1 + 10^{-16}).$$

   What do you think is happening?

   (b) The n-th Hilbert matrix $H_n \in \mathbb{R}^{n\times n}$ has the entries $h_{ij} = (i+j-1)^{-1}$ for $i,j = 1,\ldots,n$.[2] It is known that solving systems with the Hilbert matrix increases rounding errors since the matrix is poorly conditioned. Let $\boldsymbol{e}_n$ be the column vector of length n that contains all 1's. Report the exact and the numerically computed values for

   $$\|H_n(H_n^{-1}\boldsymbol{e}_n) - \boldsymbol{e}_n\|, \text{ for } n = 5, 10, 20.$$

   Here, $\|\cdot\|$ is the usual Euclidean norm. Also report the condition numbers of H_n (with respect to either norm) for $n = 5, 10, 20$.

   (c) As you know, for differentiable functions f holds

   $$f'(x_0) = \lim_{h\to 0} \frac{f(x_0+h) - f(x_0)}{h}.$$

   Thus, to numerically approximate a derivative of a function for which the derivative is hard to derive analytically, one can use, for a small h, the approximation

   $$f'(x_0) \approx f'_{\mathsf{num}}(x_0) := \frac{f(x_0+h) - f(x_0)}{h}.$$

   In this approximation, one subtracts very similar numbers in the numerator of the fraction from each other, and then multiplies with a large number (namely h^{-1}), which can lead to errors that are much larger than the machine epsilon. Compute approximations of the derivative of the function

   $$f(x) = \frac{\exp(x)}{\cos(x)^3 + \sin(x)^3}$$

   at $x_0 = \pi/4$. In order to do so, use progressively smaller perturbations $h = 10^{-k}$ for $k = 1,\ldots 16$. Present the errors in the resulting approximation in a log-log plot, i.e., use a logarithmic scale to plot the values of h on the x-axis, and a logarithmic scale to plot the errors between the finite difference approximation $f'_{\mathsf{num}}(x_0)$ and the exact value, which is $f'(x_0) = 3.101766393836051$, on the y-axis.[3] What do you observe as h becomes smaller, and for which h do you get the best approximation to the derivative?

[1] You can use the command `format long` to get 15 digits output from your computer. If you need more digits of a number a, you can use `fprintf('%2.20f\n',a)` to see 20 digits.

[2] You can get H_n by using the MATLAB command `hilb(n)`.

[3] MATLAB offers a `loglog` function to do that.

2. **[Orthongonalization and least squares, 2+3pt]**.

   (a) Given any two nonzero vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}^n$, construct a Householder matrix H, such that $H\boldsymbol{x}$ is a scalar multiple of $\boldsymbol{y}$. Is the matrix H unique?

   (b) Use Householder matrices to compute the QR-factorization of the matrix:

   $$\begin{bmatrix} 9 & -6 \\ 12 & -8 \\ 0 & 20 \end{bmatrix}.$$

   Write down both formulations we discussed in class, i.e., $A = \hat{Q}\hat{R}$ with $\hat{Q} \in \mathbb{R}^{m\times n}$, $\hat{R} \in \mathbb{R}^{n\times n}$ as well as $A = QR$ with $Q \in \mathbb{R}^{m\times m}$, $\hat{R} \in \mathbb{R}^{m\times n}$.

   (c) Use it to find the least squares solution to the system of linear equations

   $$\begin{aligned} 9x - 6y &= 300 \\ 12x - 8y &= 600 \\ 20y &= 900\ . \end{aligned}$$

   Plot the three lines above and indicate the location of the least squares solution.

3. **[Newton's method for systems, 1+1+2+2pt]** Let $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$ defined by $f(x, y) = (f_1(x, y), f_2(x, y))^T$, where

   $$f_1(x, y) = x^2 + 4y^2 - 4, \quad f_2(x, y) = 2y - \sqrt{3}x^2.$$

   We want to find the roots of f, i.e., all pairs $(x, y) \in \mathbb{R}^2$ such that $f(x, y) = (0, 0)^T$.

   (a) Sketch or plot the sets $\mathcal{S}_i = \{(x, y) \in \mathbb{R}^2 : f_i(x, y) = 0\}$, $i = 1, 2$, i.e., the set of all zeros of f_1 and f_2. What geometrical shapes do these sets have?

   (b) Calculate analytically the roots of f, i.e., the intersection of the sets $\mathcal{S}_1$ and $\mathcal{S}_2$.

   (c) Calculate the Jacobian of f, defined by

   $$J_f(x, y) = \begin{pmatrix} \partial_x f_1(x, y) & \partial_y f_1(x, y) \\ \partial_x f_2(x, y) & \partial_y f_2(x, y) \end{pmatrix} \in \mathbb{R}^{2\times 2}.$$

   Here, $\partial_x f_i(x, y)$ and $\partial_y f_i(x, y)$, $i = 1, 2$ denote the partial derivatives of f_i with respect to x and y, respectively.

   (d) The Newton method in 2D is as follows: Starting from an initial value $(x_0, y_0)^T \in \mathbb{R}^2$, compute the iterates

   $$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - [J_f(x_k, y_k)]^{-1} f(x_k, y_k), \text{ for } k = 0, 1, \dots,$$

   where $[J_f(x_k, y_k)]^{-1}$ is the inverse of the Jacobi matrix of f evaluated at (x_k, y_k). Implement the Newton method in 2D and use it to calculate the first 5 iterates for the starting values $(x_0, y_0) = (2, 3)$ and $(x_0, y_0) = (-1.5, 2)$. Plot these iterates in the xy-plane together with the curves $\mathcal{S}_1$ and $\mathcal{S}_2$. Please also hand in your code.[4]

[4] Some useful syntax: The MATLAB commands `b=[1;2]` and `A=[1, 2; 3, 4]` create the column vector $\boldsymbol{b} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and the 2-by-2 matrix $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$. Moreover, `A*b` is a simple matrix multiplication and to obtain $A^{-1}\boldsymbol{b}$, you can use either `inv(A)*b`, which inverts the matrix A, or (much better!) the command `A\b`, which solves the linear system $A\boldsymbol{x} = \boldsymbol{b}$. You can use the command `surf` to make surface plots.

4. **[Eigenvalue/vector properties, 8pts]** Prove the following statements, using the basic definition of eigenvalues and eigenvectors, or give a counterexample showing the statement is not true. Assume $A \in \mathbb{R}^{n\times n}$, $n \geq 1$.

   (a) If λ is an eigenvalue of A and $\alpha \in \mathbb{R}$, then $\lambda + \alpha$ is an eigenvalue of $A + \alpha I$, where I is the identity matrix.

   (b) If λ is an eigenvalue of A and $\alpha \in \mathbb{R}$, then $\alpha\lambda$ is an eigenvalue of αA.

   (c) If λ is an eigenvalue of A, then for any positive integer k, λ^k is an eigenvalue of A^k.

   (d) If B is "similar" to A, which means that there is a nonsingular matrix S such that $B = SAS^{-1}$, then if λ is an eigenvalue of A, it is also an eigenvalue of B. How do the eigenvectors of B relate to the eigenvectors of A?

   (e) Every matrix with $n \geq 2$ has at least two distinct eigenvalues, say λ and μ, with $\lambda \neq \mu$.

   (f) Every real matrix has a real eigenvalue.

   (g) If A is singular, then it has an eigenvalue equal to zero.

   (h) If all the eigenvalues of a matrix A are zero, then $A = 0$.