COL 726 Homework 1

Due: Thursday, Jan 17

Note: In some questions, earlier parts ask you to prove results that may be useful in later parts. If you are unable to prove the earlier results, you are still allowed to use them to complete the derivation in the later parts.

1. Given the fixed matrix $\mathbf{A} = \begin{bmatrix} 1 & 2.01 \\ 1.01 & 2.03 \end{bmatrix}$, consider the problem $f : \mathbb{R}^2 \to \mathbb{R}^2$ of finding the vector $\mathbf{x}$ such that $\mathbf{Ax} = \mathbf{b}$ for an input vector $\mathbf{b}$.

   (a) Compute $f(\mathbf{b})$ for the input vector $\mathbf{b} = [1.01 \quad 1.02]^T$, using any method you have studied before. Then compute $f(\mathbf{b} + \Delta\mathbf{b})$ for the slightly perturbed input $\mathbf{b} + \Delta\mathbf{b} = [1.01 \quad 1.0201]^T$.

   (b) Using only these input-output pairs, what can you say about the condition number of f in the ∞-norm? Give as quantitative an answer as possible.

2. Consider the standard formula for finding the roots of a quadratic equation $ax^2 + bx + c = 0$,

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

   Assume that all the coefficients a, b, c are positive.

   (a) Identify <u>two</u> situations in which computing this formula in floating-point arithmetic can lead to loss of accuracy. Are both the roots x_1 and x_2 affected by accuracy loss?

   (b) Derive an alternative formula for the roots by multiplying both numerator and denominator by $-b \mp \sqrt{b^2 - 4ac}$ and simplifying as much as possible. How does this formula compare to the previous one in terms of accuracy?

3. Let $f(x) = (x - 1)^2$ for an arbitrary <u>real</u> number x. Suppose this formula is computed using floating-point arithmetic. Using the floating-point axioms discussed in class, give a bound on the absolute forward error. Express your answer in the form $|\tilde{f}(x) - f(x)| \leq c\epsilon_{\text{machine}} + O(\epsilon_{\text{machine}}^2)$ for some c which may depend on x.

4. Let $\mathbf{A}$ be a "tall" $m \times n$ matrix with $m > n$, and $\mathbf{B}$ be a "wide" $n \times p$ matrix with $n < p$. Let $\mathbf{C} = \mathbf{AB}$.

   (a) Using the columnwise interpretation of matrix multiplication, explain why $\mathbf{C}$ has rank at most n.

(b) Using Trefethen & Bau's Theorem 1.2 ("*A matrix* $\mathbf{A}$ *with* $m \geq n$ *has full rank if and only if it maps no two distinct vectors to the same vector*"), prove that $\mathbf{A}$ maps nonzero vectors to nonzero vectors. Consequently, prove that it maps linearly independent vectors to linearly independent vectors.

(c) If $\mathbf{A}$ and $\mathbf{B}$ are both full rank, is it true that $\text{rank}(\mathbf{C}) = n$? Justify your answer.

5. (a) Using the triangle inequality, derive a lower bound for $\|\mathbf{x} + \mathbf{y}\|$ in terms of $\|\mathbf{x}\|$ and $\|\mathbf{y}\|$.

(b) Suppose $\mathbf{A}$ is a square matrix, and you are given $\|\mathbf{A}\|$ for an arbitrary induced norm $\|\cdot\|$. Based only on the value of $\|\mathbf{A}\|$, find a real number δ such that the matrix $\mathbf{I} + \epsilon\mathbf{A}$ is invertible for all $\epsilon < \delta$. (Hint: Consider the action of $\mathbf{I} + \epsilon\mathbf{A}$ on an arbitrary vector $\mathbf{x}$.)

6. The variance of a collection of real numbers $X = (x_1, x_2, \ldots, x_n)$ can be written in two mathematically equivalent ways,

$$\text{Var}(X) = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2 \tag{1}$$

$$= \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2, \tag{2}$$

where $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$.

(a) Write a Python program `hw1.py` containing two functions, `var1(X)` and `var2(X)`, that compute the variance of an array of numbers in the two ways above.

(b) Demonstrate that `var1` is less numerically stable than `var2`, by comparing them on input data whose variance you know exactly. Construct the input in such a way that the error observed in `var1` is at least 10^3 times that in `var2`. Describe your results in the report, and include in your program a function `data1()` that returns your chosen data.

(c) Challenge question: The typical way of computing the sums for μ and Var is by adding the summands one at a time in order. What numerical difficulty can this suffer from, especially when n is large? Try to think of a different summation strategy which will improve the result, then implement it in a function `var3(X)` and discuss it in your report. Demonstrate the benefit on some input `data2()`. (Note: Don't worry about computation time, only numerical accuracy.)

Submission: Upload a zip file containing (i) a PDF of your answers to Questions 1–5 and 6(b),(c), and (ii) the Python file `hw1.py` containing the functions `var1`, `var2`, `data1`, `var3`, and `data2`.