Numerical Analysis HW 1

Eric Han

Fall 2025

1. **Errors and Conditioning**

   (a) Explain the distinction between the *condition number* of a mathematical problem and the *stability* of a numerical algorithm. Can a stable algorithm produce a solution with a large relative error for an ill-conditioned problem? Justify your answer with a brief example.

   Solution: Generally, the condition number of a mathematical problem measures the sensitivity of the the problem's output to the perturbation of its inputs. The stability of a numerical algorithm measures the ability of a numerical algorithm to resist and control the amplification of numerical errors. These two are intimately related, as a function with a large condition number will have an increased sensitivity to small changes in its inputs, meaning that small numerical errors can drastically effect the step-wise output of its associated algorithm.

   While a stable algorithm guarantees control for errors introduced by computational arithmetic, such as rounding errors, an ill-conditioned problem will produce large errors in its output for comparatively small errors in its input. So, it is possible that a stable algorithm can produce a solution with a large relative error for an ill conditioned problem. For instance, consider the dynamical system defined by
$$f^{(n)}(x) = x + a.$$
   The relative condition number is given by
$$\left|\frac{xf'(x)}{f(x)}\right| = \frac{x}{x+a}.$$
   It becomes clear that the problem is ill-conditioned for $x \approx -a$, because the relative condition number blows up for $x \to -a$. We can see this ill-conditioning reveal itself in the numerical solving of this function as well, where direct evaluation with a value of x s.t. $x + a \approx 0$ causes catastrophic cancellation of two nearby numbers. A small error in the initial input will propagate consistently throughout further iterations of the system, giving us a large error relative to the initial input.

   (b) Consider the problem of evaluating the function $f(x) = \frac{1-\cos(x)}{x^2}$ for values of x near zero. Is this problem well-conditioned or ill-conditioned for $x \to 0$? Explain why direct evaluation of this formula on a computer using floating-point arithmetic is a numerically unstable algorithm.

   Solution: Let us first calculate the relative condition number.
$$\left|\frac{xf'(x)}{f(x)}\right| = \frac{\frac{x^2\sin(x)-(1-\cos(x))2x}{x^3}}{\frac{1-\cos(x)}{x^2}} = \frac{x^2(\sin(x)) - (1-cos(x))2x}{(1-\cos(x))x} = \frac{x\sin(x)}{1-\cos(x)} - 2.$$
   Because we are interested in the conditioning of the function as $x \to 0$, we take the limit of the relative condition number as $x \to 0$. Using the Taylor expansions for $\sin(x)$ and $\cos(x)$, we obtain that
$$\lim_{x\to 0} \kappa_f(x) = \lim_{x\to 0} \frac{x\sin(x)}{1-\cos(x)} - 2 = \lim_{x\to 0} \frac{x^2\left(1 - \frac{x^2}{3!} + \dots\right)}{1 - x^2\left(\frac{1}{2!} - \frac{x^2}{4!} + \dots\right)} - 2 = \frac{1}{\frac{1}{2}} - 2 = 0.$$
   So, the problem is well-conditioned. The reason why direct evaluation fails near 0 is because of catastrophic cancellation. In the case of a division algorithm that uses subtraction, we run into this issue. to As $x \to 0$, both the numerator and denominator approach values very near

to 0, and so no matter the robustness of our approximation, the catastrophic cancellation principle will result in large relative errors.

2. **Norms and Their Properties**

   (a) In a finite-dimensional vector space like $\mathbb{R}^n$, all norms are equivalent. Explain what this mathematical equivalence means. Despite this, why does the choice of norm (e.g., ℓ^1, ℓ^2, or ℓ^∞) still matter significantly in practical applications like machine learning and optimization?

   Solution: Within finite-dimensional spaces, we say that different ℓ^p norms are equivalent because for any two orders of the discrete norm, $||\cdot||_p$ and $||\cdot||_q$, there exist 2 constants, c_1 and c_2, s.t. one can tightly bound the measure of one norm in terms of a constant factor of the other. That is,

   $$c_1||\cdot||_p \leq ||\cdot||_q \leq c_2||\cdot||_p.$$

   From an analysis perspective, it is (mostly) sufficient to say that all norms on a finite linear space are equivalent. In applied settings however, different ℓ^p norms influence the problem at hand. For instance, consider some kind of optimization problem where you are attempting to minimize a weighting vector, $\overline{\sigma}$. Under the ℓ^1 norm, defined as $\sum_{i=1}^n \overline{\sigma}$, the norm is smaller the sparser $\overline{\sigma}$ is. Under the ℓ^2 norm, the norm is minimized when there are smaller overall entries in the vector.

   (b) Explain the concept of *completeness* in a normed space (i.e., a Banach space). Why is this property crucial for numerical methods that generate sequences of approximations? Use the example of the space of polynomials on $[0,1]$ with the L^2 norm to illustrate what can go wrong in an incomplete space.

   Solution: Let $\mathcal{V}$ be a normed space. $\mathcal{V}$ is complete if every Cauchy sequence within it converges to a limit in $\mathcal{V}$. That is, for every convergent sequence a_n within $\mathcal{V}$, in which its terms get progressively closer to each other for $n > N$, the limit of these terms must exist within $\mathcal{V}$. For numerical methods that generate sequences of approximations, it is important that these sequences converge to a limit in order to provide a true approximation of the answer. If these methods do not exist within a complete space, then it is possible that the sequence will never deliver a "true" answer within the context of the problem.

   A good example is the space of polynomials from $[0,1]$ under the L^2 norm. Consider the sequence of partial sums given by the Taylor series approximation of $\sin(x)$,

   $$\sum_{i=0}^{n} \frac{(-1)^n x^{2n+1}}{(2n+1)!},$$

   for some finite n, which is a convergent Cauchy series. For $n \to \infty$, this series converges to $\sin(x)$, which does not exist within the defined space of polynomials.

3. Taylor Series and Error Analysis

   (a) Using Taylor series expansions, derive the second-order centered difference formula for the second derivative:

   $$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

   Solution:

   (b) Use the Lagrange form of the Taylor remainder to derive the leading term of the truncation error for this formula. What is the order of accuracy?

   (c) The error formula derived in part (b) requires f to be sufficiently smooth (e.g., C^4). What happens to the error if f is only C^2? Use the integral remainder to argue about the convergence and, if possible, the order of accuracy.