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

close to each other, 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: Begin with the Taylor series expansion for $f(x+h)$ and $f(x-h)$,

   $$f(x+h) = f(x) + h\partial_x f + \frac{h^2}{2}\partial_x^2 f + \frac{h^3}{3!}\partial_x^3 f + \frac{h^4}{4!}\partial_x^4 f$$
   $$f(x-h) = f(x) - h\partial_x f + \frac{h^2}{2}\partial_x^2 f - \frac{h^3}{3!}\partial_x^3 f + \frac{h^4}{4!}\partial_x^4 f.$$

   We add the two together to derive the central-difference method:

   $$f(x+h) + f(x-h) = 2f(x) + \frac{2h^2}{2!}\partial_x^2 f + 2\frac{h^4}{4!}\partial_x^4 f$$
   $$\implies \frac{f(x+h) + f(x-h) - 2f(x)}{h^2} = \partial_x^2 f + \frac{h^2}{12}\partial_x^4 f(x+h) \tag{1}$$

   where (1)'s last term is the truncated error.

(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?

Solution: The Lagrange form of the remainder is

$$R_3(x) = \frac{h^2}{12}\partial_x^4 f(x+h).$$

The order of accuracy is $O(h^2)$.

(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.

Solution: Intuitively, if $f \notin C^4$, then the $O(h^2)$ error from the Lagrange form of the remainder is not true. We can confirm this using the integral form of the remainder. To compute this, we once again add $f(x+h)$ and $f(x-h)$, except only up to the second order term because $f \in C^2$.

We arrive at

$$f(x+h) - 2f(x) + f(x-h) = \int_x^{x+h} f''(t)(x+h-t)\,dt + \int_x^{x-h} f''(t)(x-h-t)\,dt$$

$$f(x+h) - 2f(x) + f(x-h) = \int_0^h [f''(x+s) + f''(x-s)](h-s)\,ds,$$

where $t = x+s$ in the first integral and $t = x-s$ in the second integral. We can then divide by h^2 in order to achieve the desired form.

The error of the approximation is given by the difference between the difference approximation and the true value of the derivative. Thus,

$$E(h) = \frac{1}{h^2}\int_0^h \left[f''(x+s) + f''(x-s)\right](h-s)\,ds - f''(x).$$

Now, recall that for measuring error convergence, we measure the size of the error as $h \to 0$. The bracketed terms $[f''(x+s) + f''(x-s)]$ are approximately $[f''(x) + f''(x)] = 2f''(x)$ for $h \to 0$. So, the error now becomes

$$E(h) = \frac{1}{h^2}\int_0^h 2f''(x)(h-s)\,ds - f''(x).$$

We can now evaluate this integral directly.

$$\frac{1}{h^2}\int_0^h 2f''(x)(h-s)\,ds = \frac{1}{h^2}2f''(x)\frac{h^2}{2}\,ds,$$

which gives us

$$E(h) = \frac{2}{h^2}f''(x)\frac{h^2}{2}\,ds - f''(x) = 0.$$

So, somewhat miraculously, the error does converge to 0, and because the approximation is until second order, the order of accuracy is given by $O(h^2)$.

4. **Differentiability and Continuity**

(a) Is a function that is continuously differentiable (C^1) on a closed, bounded interval $[a,b]$ necessarily Lipschitz continuous on that interval? Justify your answer.

Solution: Yes. Note that necessarily, a C^1 function on a compact set obtains a max-min value for its derivative. So, $|f'(x)|$ is bounded. Now consider a function $f(x)$ over $[a,b]$. The fuction can be divided into subintervals s.t. by the Mean Value Theorem,

$$\frac{f(x)-f(y)}{x-y} = f'(\xi) \implies |f(x)-f(y)| = |f'(\xi)||x-y|$$

for $\xi \in (x,y)$. Because $f'(\xi)$ is bounded, we know that the Lipschitz constant must be the upper bound of the derivative.

(b) Provide an example of a function that is Lipschitz continuous on the interval $[-1, 1]$ but is not differentiable at every point in that interval. Explain why your example satisfies these conditions and state its Lipschitz constant.

Solution: Consider the function $f(x) = |x|$. This function is Lipschitz continuous with a Lipschitz constant of 1, computable by taking the maximum of $|f'(x)|$. However, it is not differentiable at $x = 0$.

5. **Bilinear Forms and the Inf-Sup Condition**

(a) The inf-sup (or LBB) condition for a bilinear form $B(u, v)$ states that its inf-sup constant, ρ_{min}, must be strictly positive. Define ρ_{min}. How does this condition relate to coercivity, and why is it a more general condition for the well-posedness of a variational problem $B(u, v) = \ell(v)$?

Solution: ρ_{min} is defined as the stretching factor of a bilinear operator on two possibly distinct linear spaces $\mathcal{U}, \mathcal{V}$. Mathematically,

$$\rho_{min} = \inf_{u \in \mathcal{U}, u \neq 0} \sup_{v \in \mathcal{V}, v \neq 0} \frac{|B(u, v)}{||u||_{\mathcal{U}} ||v||_{\mathcal{V}}}.$$

A bilinear form is called coercive if there exists $c > 0$ s.t. $A(x, x) \geq c||x||^2$ for a linear operator A and for all $x \in H$ a Hilbert space. One can rearrange this to obtain the inf-sup constant for exactly symmetric operators on a single Hilbert space,

$$\frac{|A(x, x)|}{||x^2||_H} \geq c,$$

where $c = \rho_{min}$.
The LBB condition states that for noncoercive operators, there exists a unique solution to the variational problem $A(u, v) = \ell(v)$ with the fulfillment of a strictly positive ρ_{min}.

(b) Consider the bilinear form

$$\mathcal{B}(y, x) = y^T A x$$

with the rank-1 matrix

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}.$$

Show that if the domain and codomain are the full space $\mathbb{R}^2$, the inf-sup constant is zero. Then, identify the correct subspaces

$$\mathcal{U} = \mathcal{R}(A^T) \quad \text{and} \quad \mathcal{V} = \mathcal{R}(A)$$

and compute the non-trivial inf-sup constant on these subspaces using the Euclidean norm $\|\cdot\|_2$.

Solution: Note that A has linearly dependent columns, so it has a nontrivial nullspace. Therefore, using $\mathbb{R}^2$ as the full domain and codomain to calculate the inf-sup constant means that we may select $y \in \mathcal{N}(A^T)$ or $x \in \mathcal{N}(A)$ to minimize $|A(x, y)|$. Thus, we must use the new linear spaces of $\mathcal{U} = \mathcal{R}(A^T)$ and $\mathcal{V} = \mathcal{R}(A)$, defined as $\mathcal{U} = \mathcal{V} = \text{span}\left([1, 2]^T\right)$.

$\rho_{min} = \sigma_{min}(A)$ for matrices. Thus, we calculate the singular values of AA^T.

$$AA^T = \begin{bmatrix} 5 & 10 \\ 10 & 20 \end{bmatrix},$$

and then calculate the eigenvalues, which are computed to be $\lambda_1 = 0, \lambda_2 = 25$. We then take the square root of all eigenvalues to find the minimum singular value. $\sigma_1 = 0, \sigma_2 = 5$. Since we require a strictly positive inf-sup constant for our problem to be well-posed, we select $\rho_{min} = \sigma_2 = 5$.

For the following problems, please reference the code as provided in the GitHub link provided here: https://github.com/ericzyhan/AMS-527. Follow the instructions in the README to run the code.

6. **Rounding Errors in Finite Differences**

   (a) The minimum value of the magnitude of error on this plot is 4.36×10^{-10}. The value of h at which this occurs $h = 1 \times 10^{-8}$. The machine epsilon for double-precision is $\sqrt{\epsilon_{mach}} \approx 2.22 \times 10^{-8}$, which is of the same magnitude as the optimal h we calculated.

   (b) For double-precision, the optimal h is 1×10^{-5}. The theoretical optimal value which minimizes error is $(\epsilon_{mach})^{\frac{1}{3}} \approx 2.22 \times 10^{-5.3}$, which is approximately of the same magnitude as the optimal h calculated.

   For single-precision, the optimal h is 1×10^{-2}. The theoretical optimal value is $(\epsilon_{mach})^{\frac{1}{3}} \approx 0.099$, which is almost exactly the optimal h.

7. **Rounding Errors in the Quadratic Formula**

   (a) The standard quadratic equation suffers from catastrophic cancellation in the case where the sign of b is opposite that of the sign preceding the square root of the discriminant. This is because for large b, the b^2 term underneath the square root dominates the $-4ac$ term, and so $\sqrt{b^2 - 4ac}$ is almost equal to b. When the sign of b is opposite that of the sign preceding the square root of the discriminant, you end up subtracting (or adding, in the negative case) two extremely close numbers, resulting in catastrophic cancellation.

   This problem is avoided with the other root because you end up adding (or subtracting, in the negative case) two close numbers, which does not result in catastrophic cancellation.

   To calculate the other root, select the form of the alternative formula s.t. the sign of b matches the sign preceding the discriminant.

   (b) Below is the table generated by the naive algorithm.

Coefficients			Reference Solutions		Approximate Solutions	
a	b	c	Root 1	Root 2	Root 1	Root 2
1	4	3	-1.0	-3.0	-1.0	-3.0
1e200	4e200	3e200	-1.0	-3.0	Error	Error
0	2	8	-4.0	N/A	nan	-inf
1	1e8	1	-1.00000000e-8	-9.99999990e7	-7.45	-100000000
1	-8	15.99999999	4.00001	3.99999	nan	-inf
1e-200	-1e200	1e-200	$\approx$ 1e400	$\approx$ 1e-400	Error	Error

   (c) Below is the table generated by the robust algorithm.

Coefficients			Reference Solutions		Approximate Solutions	
a	b	c	Root 1	Root 2	Root 1	Root 2
1	4	3	-1.0	-3.0	-1.0	-3.0
1e200	4e200	3e200	-1.0	-3.0	-1.0	-3.0
0	2	8	-4.0	N/A	-4.0	N/A
1	1e8	1	-1.00000000e-8	-9.99999990e7	-1.00000000e-8	-9.99999990e7
1	-8	15.99999999	4.00001	3.99999	4.00031	3.99968
1e-200	-1e200	1e-200	$\approx$ 1e400	$\approx$ 1e-400	$\approx$ 1e-400	$\approx$ 1e-400

   I believe there is a slight discrepancy in the $(a, b, c) = (1, -8, 15.99999999)$ case because of discrepancies in the initially defined coefficients. The reference solutions more accurately calculate the roots to a polynomial defined by $(a, b, c) = (1, -8, 16 + \varepsilon)$ where $O(\varepsilon) < O(10^{-8})$.