Math 125 **Problem Set 2** **Due: Wednesday 10/5 at 11:59 p.m. EST**

Instruction: Read the assignment policy. For problems 1(d) and 5, include a printout your code with your homework submission. You should submit your assignment on Gradescope.

1. This question concerns the condition number of a problem.

(a) Let $f(x) = \ln(x)$. Find the condition number for $x > 0$. Discuss for what values of x the problem is ill-conditioned i.e. the condition number is very large.

(b) What is the condition number for $f(x) = \frac{x}{x-1}$ at x? Where is it ill-conditioned?

(c) Finding the roots of polynomials is typically a ill-conditioned problem. Consider a polynomial $P(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$ of degree n. Let x_j denote the j-th root of $P(x)$. Find the condition number of x_j with respect to perturbations of a single coefficient a_i.

(d) The Wilkinson polynomial is defined as $P(x) = (x-1)(x-2)\dots(x-19)(x-20)$. Consider the expanded version of the polynomial

$$\begin{aligned} P(x) = x^{20} &- 210x^{19} + 20615\,x^{18} - 1256850\,x^{17} + 53327946\,x^{16} - 1672280820x^{15} \\ &+ 40171771630\,x^{14} - 756111184500\,x^{13} + 11310276995381\,x^{12} \\ &- 135585182899530\,x^{11} + 1307535010540395\,x^{10} - 10142299865511450\,x^{9} \\ &+ 63030812099294896\,x^{8} - 311333643161390640\,x^{7} + 1206647803780373360\,x^{6} \\ &- 3599979517947607200\,x^{5} + 8037811822645051776\,x^{4} - 12870931245150988800\,x^{3} \\ &+ 13803759753640704000\,x^{2} - 8752948036761600000\,x^{1} + 2432902008176640000 \end{aligned}$$

Using a numerical solver of your choice, find all the roots after the perturbation of the coefficient x^{19} from -210 to $-210 - 2^{-23}$. Find the relative error of each computed root and present your result in a tabular form. Using your result in (c), explain why computing certain roots are more ill-conditioned than others.

Solution:

(a) The condition number is

$$\kappa(x) = \frac{|f'(x)||x|}{|f(x)|} = \frac{1}{|\ln(x)|}.$$

The problem is ill-conditioned for values of x near 1.

(b) The condition number is

$$\kappa(x) = \frac{|f'(x)||x|}{|f(x)|} = \frac{\left|\frac{1}{x-1} - \frac{x}{(x-1)^2}\right| |x|}{\left|\frac{x}{x-1}\right|} = \frac{1}{|x-1|}$$

The problem is ill-conditioned for values of x near 1.

(c) Since x_j is a root of $P(x)$, we have $a_0 + a_1 x_j + a_2 x_j^2 + ... + a_n x_j^n = 0$. We denote the relationship of x_j and a_i as $x_j = f(a_i)$. The condition number in the computed root x_j with respect to perturbation in a_i is

$$\lim_{\epsilon \to 0} \sup_{|\delta| \le \epsilon} \frac{\frac{|f(a_i+\delta) - f(a_i)|}{|f(a_i)|}}{\frac{|\delta|}{|a_i|}} = \lim_{\epsilon \to 0} \sup_{|\delta| \le \epsilon} \frac{\left|\frac{f(a_i+\delta) - f(a_i)}{\delta}\right|}{\frac{|f(a_i)|}{|a_i|}} = \frac{|f'(a_i)||a_i|}{|f(a_i)|}$$

Since x_j is a root of the polynomial is satisfies $a_0 + a_1 x_j + a_2 x_j^2 + ... + a_n x_j^n = 0$. We do implicit differentiation by taking the derivative with respect to a_i on both sides

$$a_1 + 2a_2 x_j + 3a_3 x_j + ... + \left(i a_i x_j^{i-1} + x_j^i \frac{dx_j}{da_i} \right) + ... + n a_n x_j = 0 \longrightarrow P'(x_j) + x_j^i \frac{dx_j}{da_i} = 0$$

Assuming $P'(x_j) \neq 0$, we obtain $\frac{da_i}{dx_j} = \frac{-x_j^i}{P'(x_j)}$. Therefore, the condition number is

$$\kappa(a_i) = \left| \frac{x_j^{i-1} a_i}{P'(x_j)} \right|$$

(d) The exact roots are $x_1 = 1$, $x_2 = 2$, ..., $x_{20} = 20$. The table below summarizes the computed roots and the relative error in the computed root. The relative error in the input data is 5.6766×10^{-10}. As it can be seen from the table, except for the last 6 roots, the relative error in the computed root is significantly much larger than the relative error in the input data. This is consistent with the closed form of the condition number in (c). This informs us that these roots are ill conditioned. Note that the largest perturbations occur to the roots at $x = 15$ and $x = 16$. In his own personal reaction to this discovery, Wilkinson stated "I regard it as the most traumatic experience in my career as a numerical analyst."

Exact roots	Perturbed roots	Relative error in computed root
20	20.8469	0.0423
19	19.5024 + 1.9403i	0.1055
18	19.5024 - 1.9403i	0.1363
17	16.7307 + 2.8126i	0.1662
16	16.7307 - 2.8126i	0.1816
15	13.9924 + 2.5188i	0.1809
14	13.9924 - 2.5188i	0.1799
13	11.7939 + 1.6525i	0.1574
12	11.7939 - 1.6525i	0.1388
11	10.0955 + 0.6449i	0.1010
10	10.0955 - 0.6449i	0.0652
9	8.9158	0.0094
8	8.0078	0.0010
7	6.9996	0.0001
6	6.0000	0.0000
5	5.0000	0.0000
4	4.0000	0.0000
3	3.0000	0.0000
2	2.0000	0.0000
1	1.0000	0.0000

You can find code that reproduces the above results in the `HW2_Soln` folder on Canvas. A copy of the code is also included at the end of this document.

2. Let $\tilde{f}$ be a backward stable algorithm to compute problem f. Prove that

$$\frac{|\tilde{f}(x) - f(x)|}{|f(x)|} = O(\kappa(x)\epsilon),$$

where $\kappa(x)$ denotes the relative condition number for computing $f(x)$ and ϵ denotes machine precision. Interpret the result in brief words.

Solution: Since $\tilde{f}$ be a backward stable, we have $\tilde{f}(x) = f(\tilde{x})$ with $\frac{|x-\tilde{x}|}{|x|} = O(\epsilon_{\text{mach}})$. Therefore,

$$\frac{|\tilde{f}(x) - f(x)|}{|f(x)|} = \frac{|f(x) - f(\tilde{x})|}{|f(x)|} = \left(\frac{\frac{|f(x)-f(\tilde{x})|}{|f(x)|}}{\frac{|x-\tilde{x}|}{|x|}} \right) \cdot \frac{|x-\tilde{x}|}{|x|} = O(\kappa(x)\epsilon_{\text{mach}}).$$

We now proceed to justify why the bracketed term is bounded above by the condition number $\kappa(x)$. We recall the definition of the condition number

$$\lim_{\epsilon \to 0} \sup_{|\delta| \leq \epsilon} \frac{\frac{|f(x+\delta)-f(x)|}{|f(x)|}}{\frac{|\delta|}{|x|}}$$

With that, note that

$$\frac{\frac{|f(x)-f(\tilde{x})|}{|f(x)|}}{\frac{|x-\tilde{x}|}{|x|}} \leq \sup_{|\delta| \leq \epsilon} \frac{\frac{|f(x+\delta)-f(x)|}{|f(x)|}}{\frac{|\delta|}{|x|}} \leq \kappa(x) + R,$$

for some suitably chosen ϵ and $R = O(1)$. The interpretation of the result is that the forward error of an algorithm depends on backward stability and conditioning.

3. Assume that $f(x)$ has two continuous derivatives, monotonically increasing, convex and has a root. A function is convex if $f''(x) > 0$ for all x. Prove that the root is unique and Newton's method will converge to it from any initial point.

Solution: Using convexity of f, we have $f''(x) > 0$ for all x. Since f is increasing, we also have that $f'(x) > 0$ for all x. Recall the error formula for Newton's method which is given by

$$e_{n+1} = \frac{1}{2} \frac{f''(\xi_n)}{f'(r)} e_n^2,$$

where ξ_n is a number between x_n and r (r is the root). From the two properties, we have that $e_{n+1} > 0$. By definition, $e_{n+1} = x_{n+1} - r$. Therefore, it follows that $x_{n+1} > r$. Since the function is monotonically increasing, we have that $f(x_{n+1}) > f(r) = 0$. This means the sequence x_n, for $n > 0$, is bounded below by 0. We now consider the Newton update as follows

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \to x_{n+1} - r = (x_n - r) - \frac{f(x_n)}{f'(x_n)} \to e_{n+1} = e_n - c$$

For $n > 0$, $f(x_n) > 0$. In addition, since f is monotonically increasing, $f'(x_n) > 0$. All in all, we have that $e_{n+1} < e_n$ for $n > 0$. Therefore, the sequences e_n and x_n are decreasing and bounded below by 0 and r respectively. Using the monotonic convergence theorem, the limits $\lim_{n\to\infty} x_n = x_n^*$ and $\lim_{n\to\infty} e_n = e_n^*$ exist. If we take the limit on both sides of the above equation and use continuity of f, we get $e^* = e^* - \frac{f(x^*)}{f'(x^*)}$. This implies that $f(x^*) = 0$ and $x^* = r$. To show uniqueness of the root, assume that we have two distinct roots r_1 and r_2. Using Rolle's theorem, there is some $c \in [r_1, r_2]$ such that $f'(c) = 0$. This contradicts the fact that $f'(x) > 0$ for all x.

4. In this problem, we use Newton's method to do division.

(a) Show how the reciprocal $\frac{1}{x}$ can be computed iteratively using Newton's method. Find an iterative formula in a way that requires at most two multiplications, one addition or subtraction, and no divisions.

(b) Take x_k to be the estimate of $\frac{1}{x}$ during the k-th iteration of Newton's method. If we define $\epsilon_k \equiv ax_k - 1$, show that $\epsilon_{k+1} = -\epsilon_k^2$.

(c) Approximately how many iterations of Newton's method are needed to compute $\frac{1}{x}$ within d binary decimal points? Write your answer in terms of ϵ_0 and d, and assume $|\epsilon_0| < 1$.

(d) Is this method always convergent regardless of the initial guess of $\frac{1}{x}$?

Solution:

(a) We consider the function $f(a) = \frac{1}{a} - x$. The Newton update is given by

$$a_{n+1} = a_n - \frac{f(a_n)}{f'(a_n)} \rightarrow a_{n+1} = a_n - \frac{\frac{1}{a_n} - x}{-\frac{1}{a_n^2}} = a_n + (a_n - a_n^2 x) = a_n(2 - a_n \cdot x)$$

The update requires two multiplications and one subtraction.

(b) Using the definition of ϵ_k, we square it to obtain

$$\epsilon_k^2 = (aa_k - 1)^2 = a^2 a_k^2 + 1 - 2aa_k = -aa_k(2 - aa_k) + 1 = -aa_{k+1} + 1 = -(aa_{k+1} - 1) = -\epsilon_{k+1}$$

(c) Applying the definition of the error formula recursively, we have $\epsilon_n = -\epsilon_0^{2^n}$. We require that $|\epsilon_n| < 2^{-d}$. Applying logarithm in base 2 on both sides, we have $\log_2\left(|\epsilon_0|^{2^n}\right) < -d$. This leads to $-2^n \log_2 |\epsilon_0|^{-1} < -d$ which is equivalent to $2^n > \frac{d}{\log_2 |\epsilon_0|^{-1}}$. Applying logarithm in base 2 once again, the number of iterations must satisfy the following requirement

$$n > \log_2\left(\frac{d}{\log_2 |\epsilon_0|^{-1}}\right)$$

Note that the right hand side of the inequality is positive since it is assumed that $|\epsilon_0| < 1$.

(d) Note that $|\epsilon_{n+1}| = |\epsilon_n|^2 = |\epsilon_0|^{2^n}$. For this to convergence, we require that $|\epsilon_0| < 1$.

5. This question concerns root finding using the bisection, Newton and secant methods.

(a) Implement Newton's method. For each of the following equations, use your implementation to approximate the root to eight correct decimal places.

(1) $x^5 + x = 1$

(2) $\sin(x) = 6x + 5$

(3) $\ln(x) + x^2 = 3$

(b) Implement the secant Method. Use your implementation with starting guesses $x_0 = 0, x_1 = 1$ to find the root of $f(x) = x^3 + x - 1$.

(c) Implement the bisection method. Use your implementation to find the root of the equation $e^x = \sin(x)$ closest to 0.

Remark: For each of the above problems, your result should include a printout of the iterates.

Solution: You can find code that reproduces all the results for this problem in the `HW2_Soln` folder on Canvas.

(a) (1) We plot the graph and use $x = 1$ as an estimate. The iterates are

```
1.00000000
0.80000000
0.75811024
0.75500175
0.75488211
0.75487782
```

(2) We plot the graph and use $x = -2$ as an estimate. The iterates are

```
-1.05072270
-0.97139339
-0.97089894
```

(3) We plot the graph and use $x = 3$ as an estimate. The iterates are

```
1.87916648
1.60831241
1.59219780
1.59214294
```

(b) The iterates are

```
0.50000000
0.63636364
0.69005236
0.68202042
0.68232578
```

(c) We plot the graph and use the initial interval to be $[-4, -3]$. The iterates are

```
-3.50000000
-3.25000000
-3.12500000
-3.18750000
-3.15625000
-3.17187500
-3.17968750
-3.18359375
-3.18164063
-3.18261719
-3.18310547
-3.18286133
-3.18298340
-3.18304443
-3.18307495
-3.18305969
-3.18306732
```

6. Determine whether a fixed point iteration of $f(x)$ is locally convergent to the given fixed point r.

(a) $f(x) = (2x-1)^{\frac{1}{3}}, r = 1$

(b) $f(x) = \dfrac{x^3+1}{2}, r = 1$

(c) $f(x) = \sin(x) + x, r = 0$

Solution: For all of these problems, we consider $x \to g(x)$ and decide convergence based on $|g'(r)|$ being strictly less than 1.

(a) $g'(x) = \dfrac{2}{3}(2x-1)^{\frac{-2}{3}}$. Note that $g'(1) = \frac{2}{3}$. Since $|g'(r)| < 1$, it is locally convergent to $r = 1$.

(b) $g'(x) = \dfrac{3}{2}x^2$. Note that $g'(1) = \frac{3}{2}$. Since $|g'(r)| > 1$, it is divergent.

(c) $g'(x) = \cos(x) + 1$. Note that $g'(0) = 2$. Since $|g'(r)| > 1$, it is divergent.

7. Which of the following three fixed point iterations converge to $\sqrt{5}$? Rank the ones that converge from fastest to slowest.

(a) $x \to \dfrac{4}{5}x + \dfrac{1}{x}$

(b) $x \to \dfrac{x}{2} + \dfrac{5}{2x}$

(c) $x \to \dfrac{x+5}{x+1}$

Solution: For all of these problems, we consider $x \to g(x)$ and decide convergence based on $|g'(\sqrt{5})|$ being strictly less than 1.

(a) $g'(x) = \dfrac{4}{5} - \dfrac{1}{x^2}$. We note that $g'(\sqrt{5}) = \frac{3}{5} < 1$. Therefore, it is convergent.

(b) $g'(x) = \dfrac{1}{2} - \dfrac{5}{2x^2}$. We note that $g'(\sqrt{5}) = 0 < 1$. Therefore, it is convergent.

(c) $g'(x) = \dfrac{1}{x+1} - \dfrac{x+5}{(x+1)^2} = -\dfrac{4}{(x+1)^2}$. We note that $g'(\sqrt{5}) = -\frac{4}{(\sqrt{5}+1)^2} = -0.3820$. Since $|g'(\sqrt{5})| < 1$, it is convergent.

The fastest is the iteration in (b). The second fastest is the iteration in (c). The iteration in (a) is the slowest.

8. Extra Credit: Assume that $f(x)$ is twice continuously differentiable for all x in some interval I. Assume that $f(r) = 0$ and x_0, x_1 are sufficiently close to the root r. Prove that the secant method converges to the root. What is the order of convergence?

Solution: Following the same error analysis as in Newton method, we obtain

$$e_{n+1} = \frac{1}{2}\frac{f''(r)}{f'(r)}e_n e_{n-1} = Ce_n e_{n-1}$$

To find C, assume $|e_{n+1}| \approx A|e_n|^{\alpha}$. Using this in the above equation and some algebra yields $\alpha = \frac{1+\sqrt{5}}{2}$ and $A = \frac{f''(r)}{2f'(r)}$. The order of convergence is superlinear.

Code for problem 1

```
% ---------------------------------------------------------------------------
% This code considers the conditioning of the Wilkinson polynomial
% ---------------------------------------------------------------------------
% The polynomial coefficients of the Wilkinson polynomial.
% x^20-210x^19+20615x^18+....
coeff = [1; -210; 20615; -1256850; 53327946; -1672280820;
40171771630;   -756111184500; 11310276995381; -135585182899530;
1307535010540395;  -10142299865511450; 63030812099294896; -311333643161390640;
  1206647803780373360; -3599979517947607200; 8037811822645051776;
  -12870931245150988800; 13803759753640704000;
  -8752948036761600000;   2432902008176640000];
% Exact roots
exact_roots = [20:-1:1]';
% Numerically computed roots after pertubation of
% the coefficient of x^19  by -2^{-23}
coeff(2) = coeff(2)-2^-23;
perturbed_roots = roots(coeff);
% Compute approximate relative condition number
rel_error = abs(perturbed_roots-exact_roots)./(abs(exact_roots));
```