Homework Assignment #1

Due: End of Day Sept 12 (Friday), 2025

Each problem is 10 points unless otherwise noted.

Problem 1 (Taylor Expansions)

(1) Use the Taylor expansions of e^x, $\sin x$, and $\cos x$ to prove the $e^{j\theta} = \cos\theta + j\sin\theta$.

(Hint, it is in the published lecture notes already. Please do not cut paste and go through the process. It's important to understand Taylor expansion and the use of polynomials and derivatives to approximate functions.)

(2) Construct the Taylor expansions for the following functions at $x_0 = 0$:

$$\frac{1}{1-x}$$

And

$$\ln(1+x)$$

Show the steps of your constructions.

Problem 2 (The Numerical Square Root Finding Algorithm):

The goal of this problem is to give you a taste of iterative numerical conversion.

The square root $\sqrt{a}$ for most positive real numbers a are irrational numbers and is not easy to calculate. Since 1,500 BC, ancient Babylonians have been using the following iterative method to find square roots. The method works as follows:

Assume $x^{(0)}$ is the initial solution, say some random positive number.

We then iteratively calculate the following sequence of solutions $x^{(1)}, x^{(2)}, \dots$, where each solution $x^{(j+1)} = \frac{1}{2}\left(x^{(j)} + \frac{a}{x^{(j)}}\right)$ for $j = 0, 1, \dots$

Answer the following questions:

(1) Please implement the above simple root finding algorithm using your favorite programming language (e.g., python, MATLAB, C, Java, etc.) and see if the algorithm works with some examples and compare the results from a calculator.

(2) Can you explain why the algorithm can find the square root?

Hint: For the algorithm to find a square it needs to converge to $\sqrt{a}$, i.e.,

$$\lim_{j\to\infty}\left|x^{(j)} - \sqrt{a}\right| = 0$$

(3) It is said that the Babylonian root finding algorithm has a quadratic convergence. Can you explain why?

Hint: The **rate of convergence** is usually defined as follows.

Assume that a sequence $\{x^{(j)}\}$ converges to x^*, i.e., $\lim_{j\to\infty} x^{(j)} = x^*$.

The sequence is said to have an order of convergence q ($q \geq 1$) at a rate of μ if $\lim_{j\to\infty} \frac{|x^{(j+1)}-x^*|}{|x^{(j)}-x^*|^q} = \mu$.

When $q = 1$, the convergence is referred to as linear convergence.

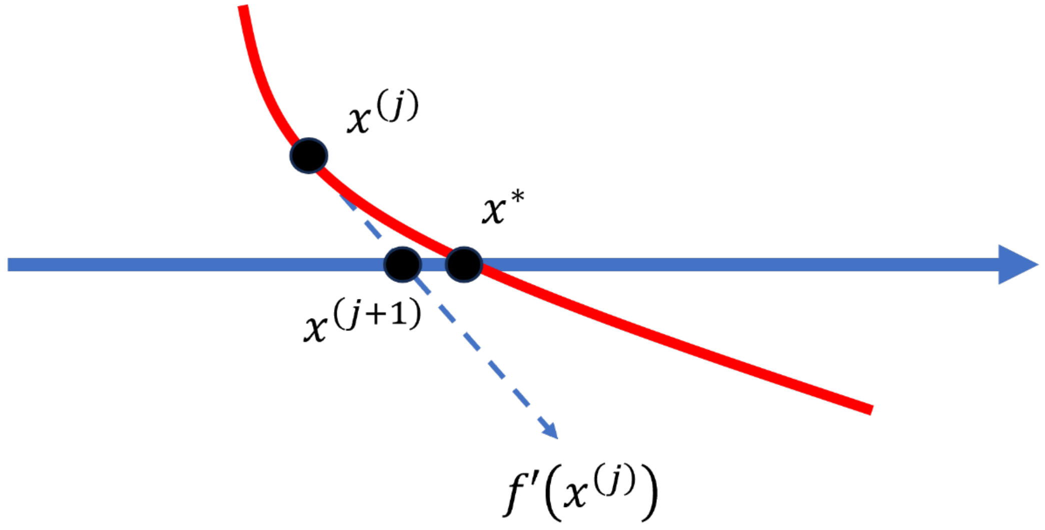
When $q = 2$, the convergence is referred to as quadratic convergence.

(4) Can you extend this approach to find cubic root or more generally n −th root?

Hint: The Babylonian square root finding algorithm is in fact a special case of the Newton's method for solving equations.

Consider an equation $f(x) = 0$. Let x^* be one of its solutions. The Newton's method will start with an initial solution $x^{(0)}$ and iteratively construction a sequence of solutions $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ converging to $x^{(*)}$, where giving the solution $x^{(j)}$, $x^{(j+1)} = x^{(j)} - \frac{f(x^{(j)})}{f'(x^{(j)})}$.

To see this, consider the following illustration.



Note that $x^{(j+1)}$ is the solution to the line equation $L(x)$: $\frac{L(x)-f(x^{(j)})}{x-x^{(j)}} = f'(x^{(j)})$, i.e., any point $(x, L(x))$ on the line L must have the slope $f'(x^{(j)})$ with respect to $\left(x^{(j)}, f(x^{(j)})\right)$.

In particular, we are looking for the $x^{(j+1)}$ such that $L(x^{(j+1)}) = 0$.

This yields the equation: $\frac{0-f(x^{(j)})}{x^{(j+1)}-x^{(j)}} = f'(x^{(j)})$.

Solving this equation for $x^{(j+1)}$, we have $x^{(j+1)} = x^{(j)} - \frac{f(x^{(j)})}{f'(x^{(j)})}$.

Can you explain why the Babylonian Root Finding algorithm is a special case of Newton's method for solving equations? What equation is it solving?

Problem 3 (Orthogonality implies linear independence):

Let $u_1, u_2, \cdots, u_N$ be an orthogonal set of non-zero vectors. Prove that the set is also linear independent.

Problem 4 (Change of Basis)

Consider the vector space $\mathbb{E}^2$, i.e., a Euclidean plane. The usual basis that we use is the $x-$ and $y-$axis's, i.e., $U = \left\{\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\right\}$. Now, assume that we rotate the coordinate system by 45°. Now the new axis become $W = \left\{\begin{bmatrix}\frac{\sqrt{2}}{2}\\ \frac{\sqrt{2}}{2}\end{bmatrix}, \begin{bmatrix}-\frac{\sqrt{2}}{2}\\ \frac{\sqrt{2}}{2}\end{bmatrix}\right\}$. (Note that we have normalized these base vectors, i.e., their norms are 1.)

Now consider a vector $v = \begin{bmatrix}4\\5\end{bmatrix}$, i.e., $v = 4\begin{bmatrix}1\\0\end{bmatrix} + 5\begin{bmatrix}0\\1\end{bmatrix}$, i.e., the coefficients of v under U is $\begin{bmatrix}4\\5\end{bmatrix}$. What is the representation of v in $W = \left\{\begin{bmatrix}\frac{\sqrt{2}}{2}\\ \frac{\sqrt{2}}{2}\end{bmatrix}, \begin{bmatrix}-\frac{\sqrt{2}}{2}\\ \frac{\sqrt{2}}{2}\end{bmatrix}\right\}$?

Problem 5 (SVM via Matlab):

Consider the following simple labeled data sets in 2D space,

$$\begin{array}{ll} (5,0) & +1 \\ (5,2) & +1 \\ (7,7) & -1 \\ (9,6) & -1 \\ (2,0) & -1 \end{array}$$

Find the optimal linear SVM for the datasets using Matlab and plot the data points and the classifier you find.

Problem 6 (Least square solvers via Matlab):

Use Matlab to the solve the following optimization problems:

$$x = \begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}$$

$$A = \begin{bmatrix}1 & 2 & 3 & 4\\5 & 6 & 7 & 8\\9 & 10 & 11 & 12\end{bmatrix}$$

$$b = \begin{bmatrix}13\\14\\15\end{bmatrix}$$

(1) $\min(Ax-b)^T(Ax-b)$ or $\min|Ax-b|^2$

(2) $$\begin{array}{ll} min & (Ax-b)^T(Ax-b) \\ s.t. & \quad x \geq 0 \end{array}$$

Problem 7 (Linear programming):

UNM CS Lab orders computers from two vendors Apple and Dell. Each semester, at least 10 computers must be ordered. A computer from Dell costs 2300 USD each, and a computer from Apple costs 600 USD each. (Yep, Apple is cheaper because it is using its own silicon. Dell is more expensive because it must equip with a discrete graphics card.) Costs must be kept to less than 10,000 USD (because it comes from the course fees). Moreover, UNM requires that the number of computers from each Vendor can't exceed twice the number from another. How many computers UNM CS must order from each vendor to minimize the total cost subject to the above constraints? Show that how this problem can be modeled using linear programming. Solve your linear program either via Matlab or Python. Did you notice any issues? Can you explain what causes the issue?