Problem 1 (16pts)
Consider the following scalar-valued function
\[ f(x, y, z) = x^2 y + \sin(z + 6y), \]
where \( x, y, z \in \mathbb{R} \).

(A) Compute partial derivatives with respect to \( x, y, \) and \( z \).

(B) We can consider \( f \) to take a vector \( \theta \in \mathbb{R}^3 \) as input where \( \theta = [x, y, z]^T \). Compute the gradient \( \nabla_\theta f \).
Evaluate \( \nabla_\theta f \) at \( \theta = [3, \frac{\pi}{2}, 0]^T \).
Problem 2 (16pts)
In this problem, you will demonstrate Clairaut's Theorem, which states that in general, the order in which one computes partial differentiates does not matter. Consider the following scalar-valued function,

\[ f(x, y) = x \sin(xy), \]

where \( x, y \in \mathbb{R} \).

(A) Compute \( \frac{\partial}{\partial x} \frac{\partial}{\partial y} f(x, y) \). This means we first compute the partial derivative of \( f \) with respect to \( y \), then compute the partial derivative of the resulting function with respect to \( x \). This is sometimes denoted \( \partial_{xy} f \).

(B) Compute \( \frac{\partial}{\partial y} \frac{\partial}{\partial x} f(x, y) \).

The correct answers for parts (A) and (B) should be the same, which demonstrates Clairaut's Theorem. This theorem holds more generally for functions \( f(x_1, x_2, \ldots, x_n) \) of \( n \) variables. This theorem is useful since it's sometimes more convenient/efficient to computation partial derivatives in a specific order.
Problem 3 (16pts)

In gradient descent, we attempt to minimize some function $f(x)$ by iteratively updating the parameter $x \in \mathbb{R}^n$ according to the following formula:

$$x_{t+1} = x_t - \lambda (\nabla_x f(x_t))^T,$$

where $\lambda \geq 0$ is a small value known as the learning rate or step size. This formula says to update $x$ so as to move in a direction proportional to the negative gradient.

Consider the function $f(x) = x^T Ax$ where $A \in \mathbb{R}^{n \times n}$ is a matrix.

(A) Implement a function $f(A, x)$ that takes as input an $n \times n$ numpy array $A$ and a 1D array $x$ of length $n$ and returns the output $x^T Ax$.

(B) Implement a function $\text{grad}_f(A, x)$ that takes the same two arguments as above but returns $\nabla_x f(x)$ evaluated at $x$.

(C) Now implement a third and final function $\text{grad\_descent}(A, x, \lambda, \text{num\_iters})$ that takes the additional arguments $\lambda$, representing the learning rate $\lambda$ above, and $\text{num\_iters}$ indicating the total number of iterations of gradient descent to perform. The function should output, via either printing or plotting, the values of $x_t$ and $f(x_t)$ at each iteration of gradient descent.

(D) Use the function you wrote in part (C) to perform gradient descent on $f$ with $A = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$. Set each element of the initial $x_0$ to any value with magnitude between 10 and 100 of your choosing. Run gradient descent for 50 iterations with learning rates $\lambda = 1, 0.25, 0.1, \text{and } 0.01$. What do you notice? Does $x_t$ always converge to the same value? Does our gradient descent algorithm work every time?
Problem 4 (18pts)
Consider the following vector function from \( \mathbb{R}^3 \) to \( \mathbb{R}^3 \):

\[
f(x) = \begin{bmatrix}
\sin(x_1 x_2 x_3) \\
\cos(x_2 + x_3) \\
\exp\left(-\frac{1}{2}(x_3^2)\right)
\end{bmatrix}
\]

(A) Compute the Jacobian matrix of \( f(x) \)?

(B) Write the determinant of this Jacobian matrix as a function of \( x \).

(C) Is the Jacobian a full rank matrix for all of \( x \in \mathbb{R}^3 \)? Explain your reasoning.
Problem 5 (16pts)
Compute the gradients for the following expressions. (You can use identities, but show your work.)

(A) $\nabla_x \text{trace}(xx^T + \sigma^2 I)$  Assume $x \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}$.

(B) $\nabla_x \frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)$  Assume $x, \mu \in \mathbb{R}^n$ and invertible symmetric $\Sigma \in \mathbb{R}^{n\times n}$.

(C) $\nabla_x (c - Ax)^T (c - Ax)$  Assume $x, c \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$.

(D) $\nabla_x (c + Ax)^T (c - Bx)$  Assume $x \in \mathbb{R}^n$, $c \in \mathbb{R}^m$ and $A, B \in \mathbb{R}^{m\times n}$. 
Problem 6 (16pts)  (A) The sigmoid function $f : \mathbb{R} \to \mathbb{R}$ (also called the logistic function) is defined to be:

$$f(z) = \frac{1}{1 + e^{-z}}$$  \hspace{1cm} (1)

Compute the derivative of the sigmoid function, i.e., $f'(z)$. Verify that $f'(z) = f(z) (1 - f(z))$

(B) The cost function of logistic regression, a very popular machine learning model, has the following form:

$$c(\theta, x, y) = -y \log\left(\frac{1}{1 + e^{-\theta^\top x}}\right) - (1 - y) \log\left(1 - \frac{1}{1 + e^{-\theta^\top x}}\right)$$  \hspace{1cm} (2)

where $\theta \in \mathbb{R}^d, x \in \mathbb{R}^d, y \in \mathbb{R}$. Compute $\frac{\partial c(\theta, x, y)}{\partial \theta}$, the partial derivative with regards to $\theta$. Verify that $\frac{\partial c(\theta, x, y)}{\partial \theta} = (f(\theta^\top x) - y) x^\top$. 
Problem 7 (2pts)
Approximately how many hours did this assignment take you to complete?

My notebook URL: https://colab.research.google.com/xxxxxxxxxxxxxxxxxxxxxxxxxxxx