Homework 10 - Ilie Miruna Andreea

1) Find the tangent plane to the unit sphere $x^2+y^2+z^2=1$ at arbitrary point (x_0, y_0, z_0)

$$F(x,y,z) = x^2+y^2+z^2-1=0$$

$$\nabla F = \left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z}\right) = (2x, 2y, 2z)$$

$$\nabla F\ (x_0, y_0, z_0) = (2x_0, 2y_0, 2z_0)$$

$$\nabla F\ (x_0, y_0, z_0) \cdot \begin{bmatrix} x-x_0 \\ y-y_0 \\ z-z_0 \end{bmatrix} = 0$$

$$2x_0(x-x_0)+2y_0(y-y_0)+2z_0(z-z_0)=0$$

$$2x_0x-2x_0^2+2y_0y-2y_0^2+2z_0z-2z_0^2=0$$

$$2x_0x+2y_0y+2z_0z-2(x_0^2+y_0^2+z_0^2)=0$$

$$\Rightarrow 2x_0x+2y_0y+2z_0z-2=0$$

$$\boxed{x_0x+y_0y+z_0z=1}$$

2) Let $x, y, f: R^2 \to R^2$ and $f(x,y) = f(x(u,v), y(u,v))$. Prove:

$$a)\ \frac{df}{du} = \frac{df}{dx}\frac{dx}{du} + \frac{df}{dy}\frac{dy}{du} \quad \text{and}$$

b) $\frac{\partial f}{\partial v} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial v} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial v}$

Having $f(x,y) = f(x(u,v), y(u,v))$

where $x = x(u,v)$ and $y = y(u,v)$

a) $\frac{\partial f}{\partial u} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial u}$

$f(x,y) = f(x(u,v), y(u,v))$

Chain rule $\Longrightarrow$ $\frac{\partial f}{\partial u} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial u}$

b) $$\frac{\partial f}{\partial v} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial v} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial v}$$

$$f(x,y) = f(x(u,v), y(u,v))$$

chain rule $\Longrightarrow$ $$\frac{\partial f}{\partial v} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial v} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial v}$$

3) a) $f: R^2 \to R \quad f(x,y) = \frac{1}{2}(x^2 + \beta y^2)$

$$(x_{k+1}, y_{k+1}) = (x_k, y_k) - s_k \nabla f(x_k, y_k)$$

a) $\varphi(s_k) = f(x_{k+1}, y_{k+1}) =$

$$= f((x_k, y_k) - s_k \nabla f(x_k, y_k))$$

$$\nabla f(x, y) = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix} = \begin{bmatrix} x \\ by \end{bmatrix}$$

$$\nabla f(x_k, y_k) = \begin{bmatrix} x_k \\ b y_k \end{bmatrix}$$

$$x_{k+1} = x_k - s_k x_k$$

$$y_{k+1} = y_k - s_k b y_k$$

$$\varphi(s_k) = \frac{1}{2}\left((x_k - s_k x_k)^2 + b(y_k - s_k b y_k)^2\right)$$

$$\Rightarrow \varphi(s_k) = \frac{1}{2}\left(x_k^2 (1 - s_k)^2 + b y_k^2 (1 - s_k b)^2\right)$$

$$\varphi'(s_k) = \frac{d}{ds_k} \int \frac{1}{2}\left(x_k^2 (1 - s_k)^2 + b y_k^2 (1 - s_k b)^2\right)$$

$$\frac{d}{dS_K}\left[\frac{1}{2}x_K^2(1-S_K)^2\right] = x_K^2(1-S_K)(-1)$$

$$\frac{d}{dS_K}\left[\frac{1}{2}by_K^2(1-S_K b)^2\right] = by_K^2(1-S_K b)(-b)$$

$$\Rightarrow \varphi'(S_K) = -x_K^2(1-S_K) - b^2y_K^2(1-S_K b)$$

$$\varphi'(S_K) = 0$$

$$-x_K^2(1-S_K) - b^2y_K^2(1-S_K b) = 0$$

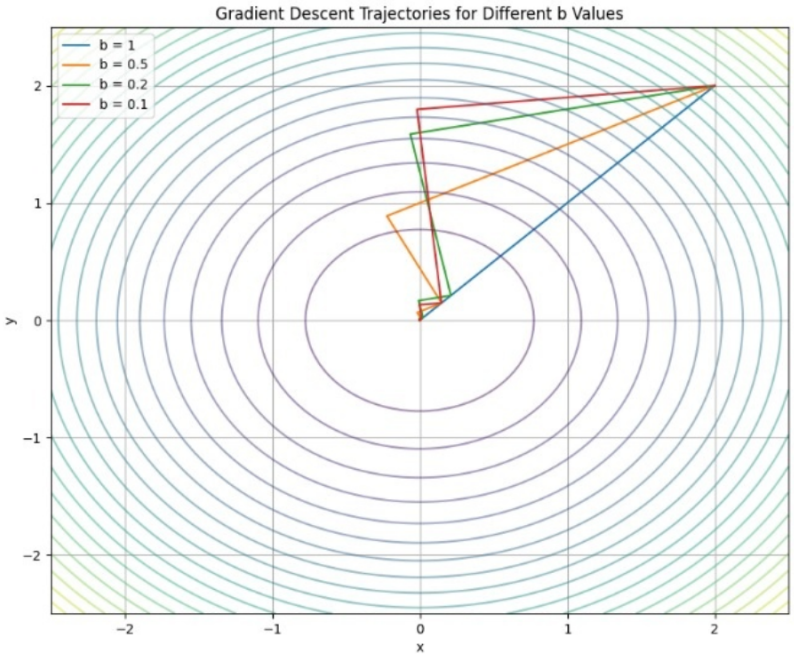
$$x_K^2 S_K + b^3 y_K^2 S_K = x_K^2 + b^2 y_K^2$$

$$S_K(x_K^2 + b^3 y_K^2) = x_K^2 + b^2 y_K^2$$

$$S_K = \frac{x_K^2 + b^2 y^2 K}{x_K^2 + b^3 y^2 K}$$

```python
# Define the quadratic function and gradient
def f(x, y, b):  1 usage
    return 0.5 * (x ** 2 + b * y ** 2)


def grad_f(x, y, b):  1 usage
    return np.array([x, b * y])


# Gradient descent algorithm
def gradient_descent(b, x0, y0, max_iter=50, tol=1e-6):  1 usage
    x, y = x0, y0
    trajectory = [(x, y)]

    for _ in range(max_iter):
        grad = grad_f(x, y, b)
        # Compute the step size s_k
        sk = (x ** 2 + (b ** 2) * y ** 2) / (x ** 2 + (b ** 3) * y ** 2)
        # Update the variables
        x, y = x - sk * grad[0], y - sk * grad[1]
        trajectory.append((x, y))
        # Check for convergence
        if np.linalg.norm(grad) < tol:
            break

    return np.array(trajectory)
```

```python
# Add contour plot of the function
x = np.linspace(-2.5,  stop: 2.5,  num: 100)
y = np.linspace(-2.5,  stop: 2.5,  num: 100)
X, Y = np.meshgrid( *xi: x, y)
Z = f(X, Y, b=1)  # Use b=1 for contour plot
plt.contour( *args: X, Y, Z, levels=20, alpha=0.5, cmap='viridis')

plt.title("Gradient Descent Trajectories for Different b Values")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.grid()
plt.show()
```


Gradient Descent Trajectories for Different b Values
b = 1
b = 0.5
b = 0.2
b = 0.1
y
x
2
1
0
−1
−2
−2
−1
0
1
2