Homework 10 Nihai-Dan Cirjan 313

1. Tangent plane to the unit sphere

$x^2 + y^2 + z^2 = 1$ at (x_0, y_0, z_0)

$\nabla f(x,y,z) \perp$ Tangent plane

$\Rightarrow \nabla f(x_0,y_0,z_0) \cdot (x-x_0, y-y_0, z-z_0) = 0$

$\nabla f(x_0,y_0,z_0) = \left(\frac{df}{dx_0}, \frac{df}{dy_0}, \frac{df}{dz_0}\right) = (2x_0, 2y_0, 2z_0)$

$\Rightarrow \nabla f(x_0,y_0,z_0) \cdot (x-x_0, y-y_0, z-z_0) = 0$

$(2x_0, 2y_0, 2z_0) \cdot (x-x_0, y-y_0, z-z_0) = 0 \quad | :2$

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

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

$x_0x + y_0y + z_0z - \underbrace{(x_0^2 + y_0^2 + z_0^2)}_{f(x_0,y_0,z_0)=1} = 0$

$x_0x + y_0y + z_0z = 1$ (Tangent plane at (x_0,y_0,z_0))

2. $f: \mathbb{R}^2 \to \mathbb{R}, \quad f(x,y) = \frac{1}{2}(x^2 + by^2), \quad b > 0$

Gradient descent for finding its minimum

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

$$\nabla f(x,y) = \left(\frac{df}{dx}, \frac{df}{dy}\right) = \left(\frac{2}{2}x, \frac{2b}{2}y\right) = (x, by)$$

$$\Rightarrow (x_{k+1}, y_{k+1}) = (x_k, y_k) - \alpha_k (x_k, by_k)$$

$$(x_{k+1}, y_{k+1}) = ((1-\alpha_k)x_k, (1-b\alpha_k)y_k)$$

The step size is determined using exact line search.
We look for the optimal step size (learning rate) $\alpha_k \geq 0$ by minimizing the function

$$\varphi(\alpha_k) = f(x_{k+1}, y_{k+1}) = f((1-\alpha_k)x_k, (1-b\alpha_k)y_k)$$

$$\varphi(\alpha_k) = \frac{1}{2}\left((1-\alpha_k)^2 x_k^2 + b\cdot(1-b\alpha_k)^2 y_k^2\right) \to \min$$

In order for $\varphi(\alpha_k)$ to be min we want $\varphi'(\alpha_k) = 0$

$$\varphi'(\alpha_k) = \frac{1}{2}\left[2\cdot(1-\alpha_k)x_k^2 + 2b\cdot(1-b\alpha_k)y_k^2\cdot(-b)\right] = 0$$

$$\frac{1}{2}\left[2\cdot(1-\alpha_k)x_k^2 - 2b^2(1-b\alpha_k)y_k^2\right] = 0$$

$$\frac{1}{2}\left[2x_k^2 - 2\alpha_k x_k^2 - 2b^2 y_k^2 + \alpha_k 2b^3 y_k^2\right] = 0$$

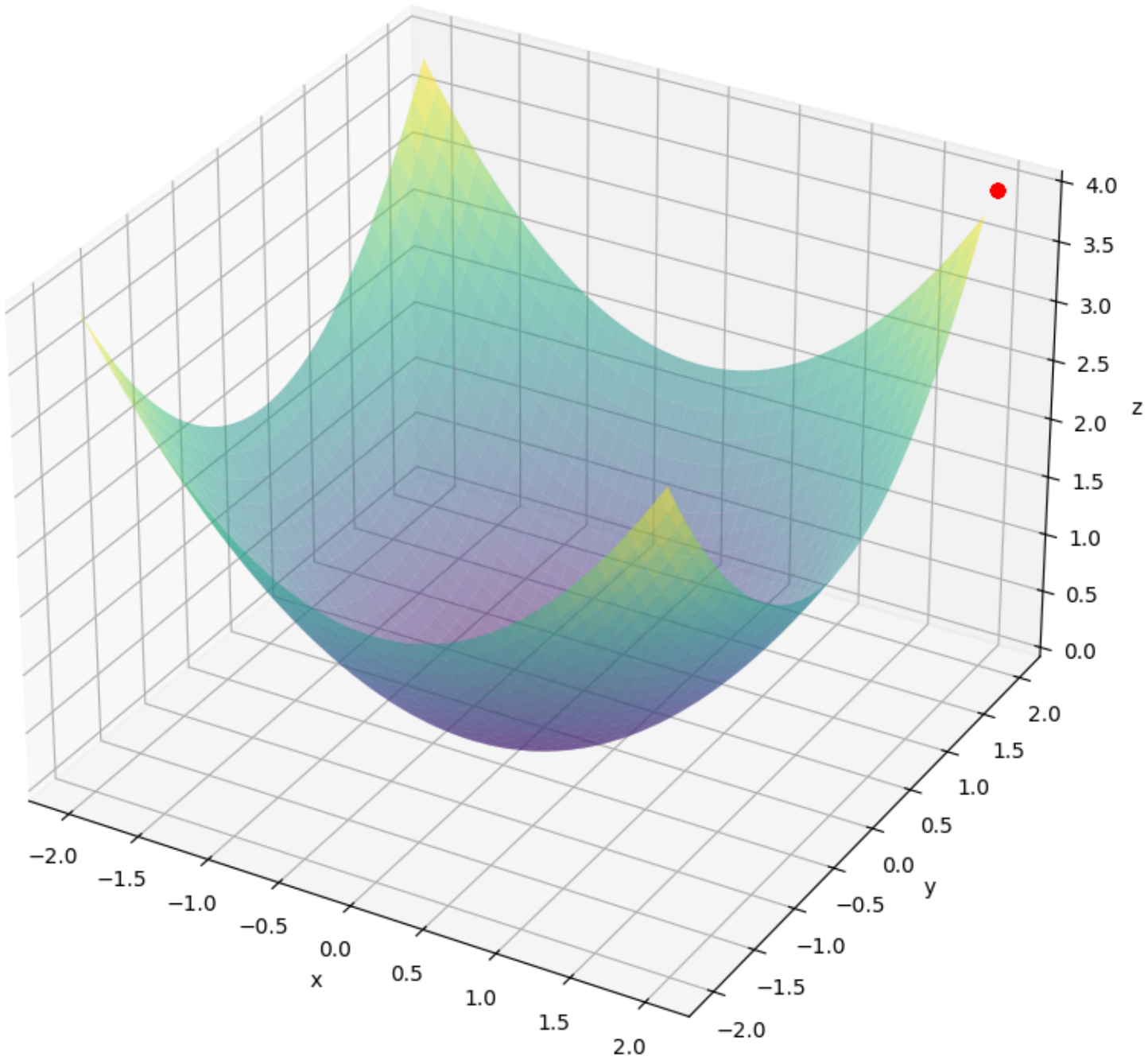
$$\frac{1}{2}\left[\alpha_k(-2x_k^2 + 2b^3 y_k^2) + 2x_k^2 - 2b^2 y_k^2\right] = 0 \quad /\cdot 2$$

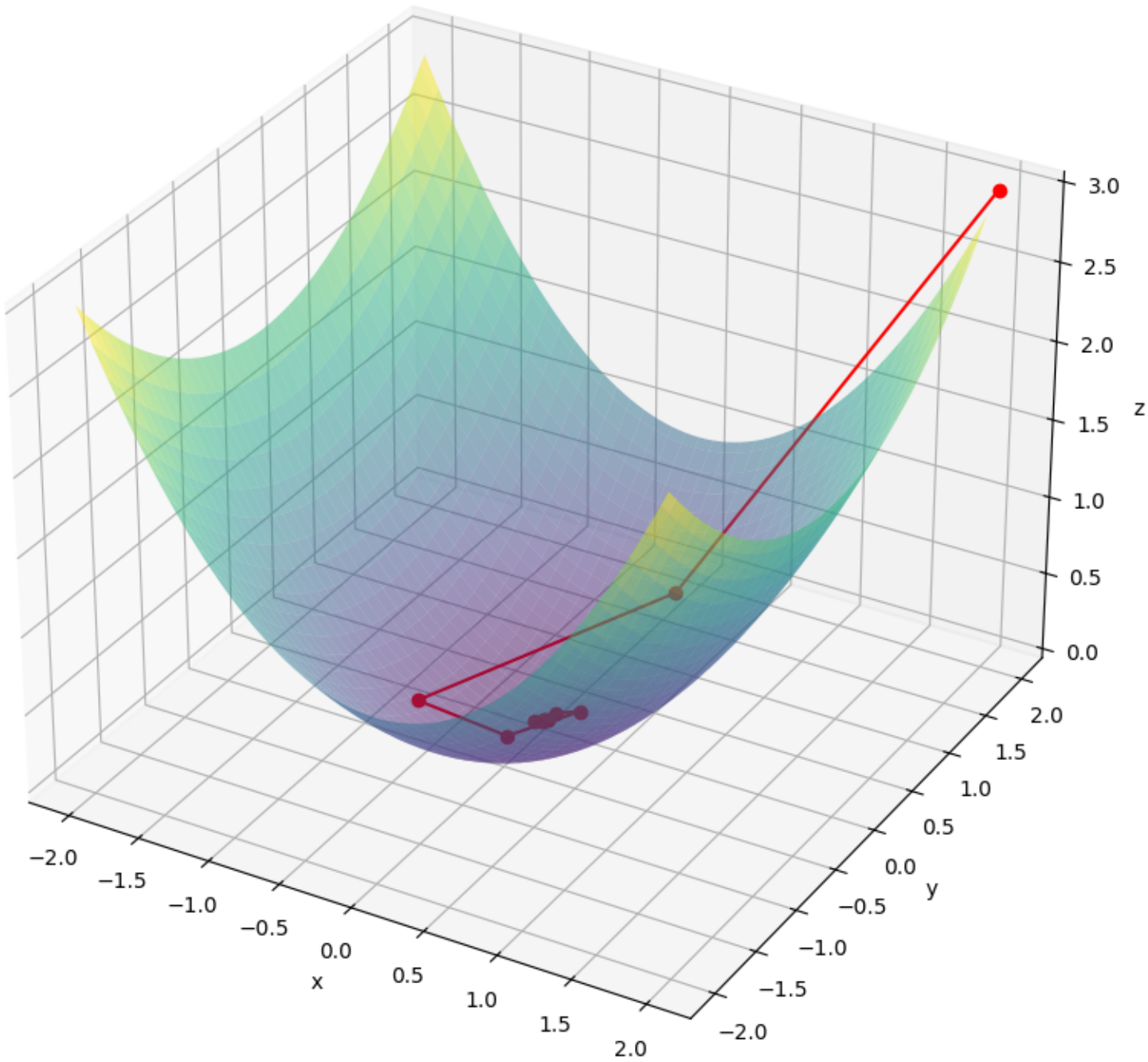
$$\alpha_k(-x_k^2 + b^3 y_k^2) + x_k^2 - b^2 y_k^2 = 0$$

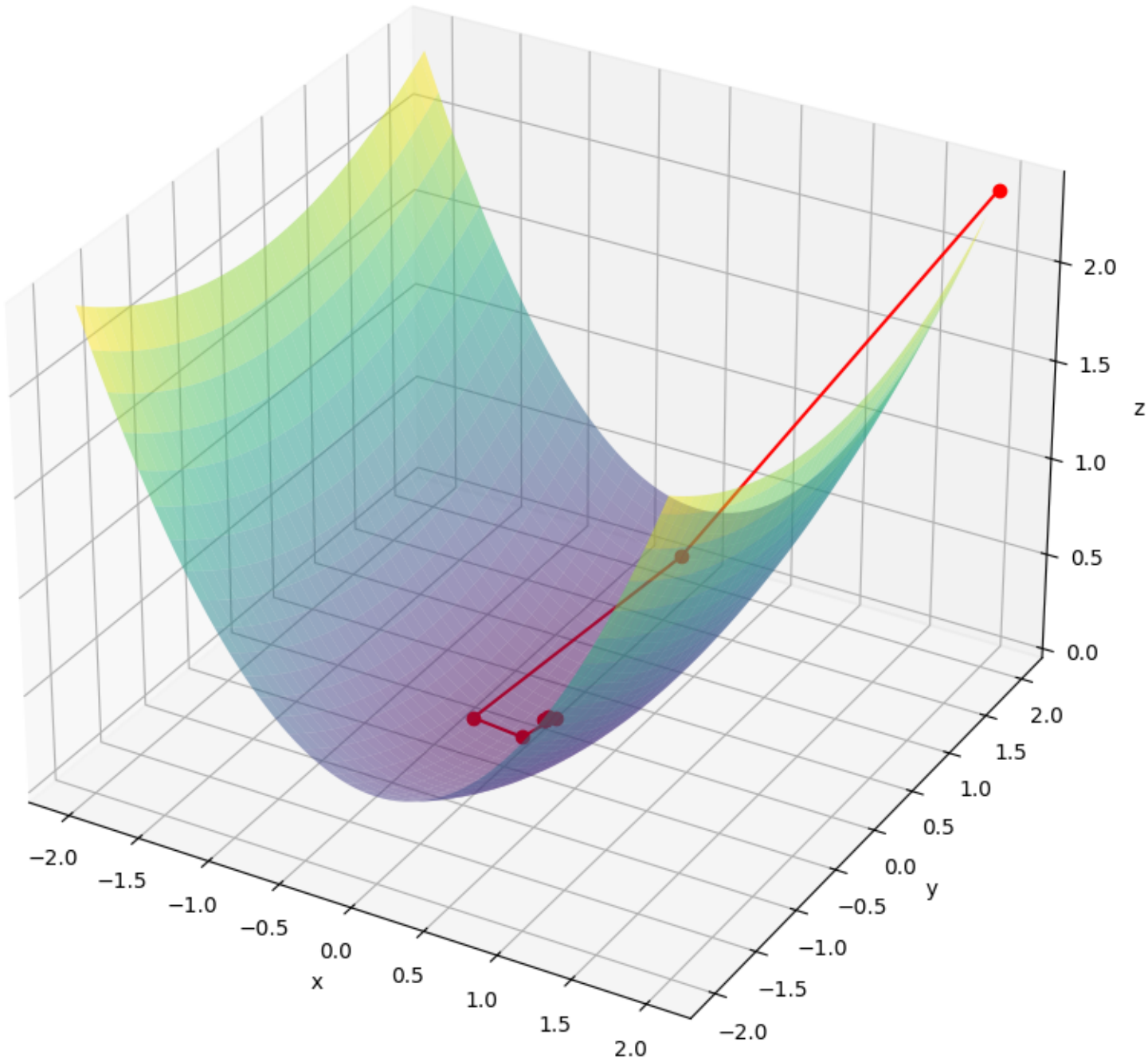
$$\alpha_k(-x_k^2 + b^3 y_k^2) = b^2 y_k^2 - x_k^2$$

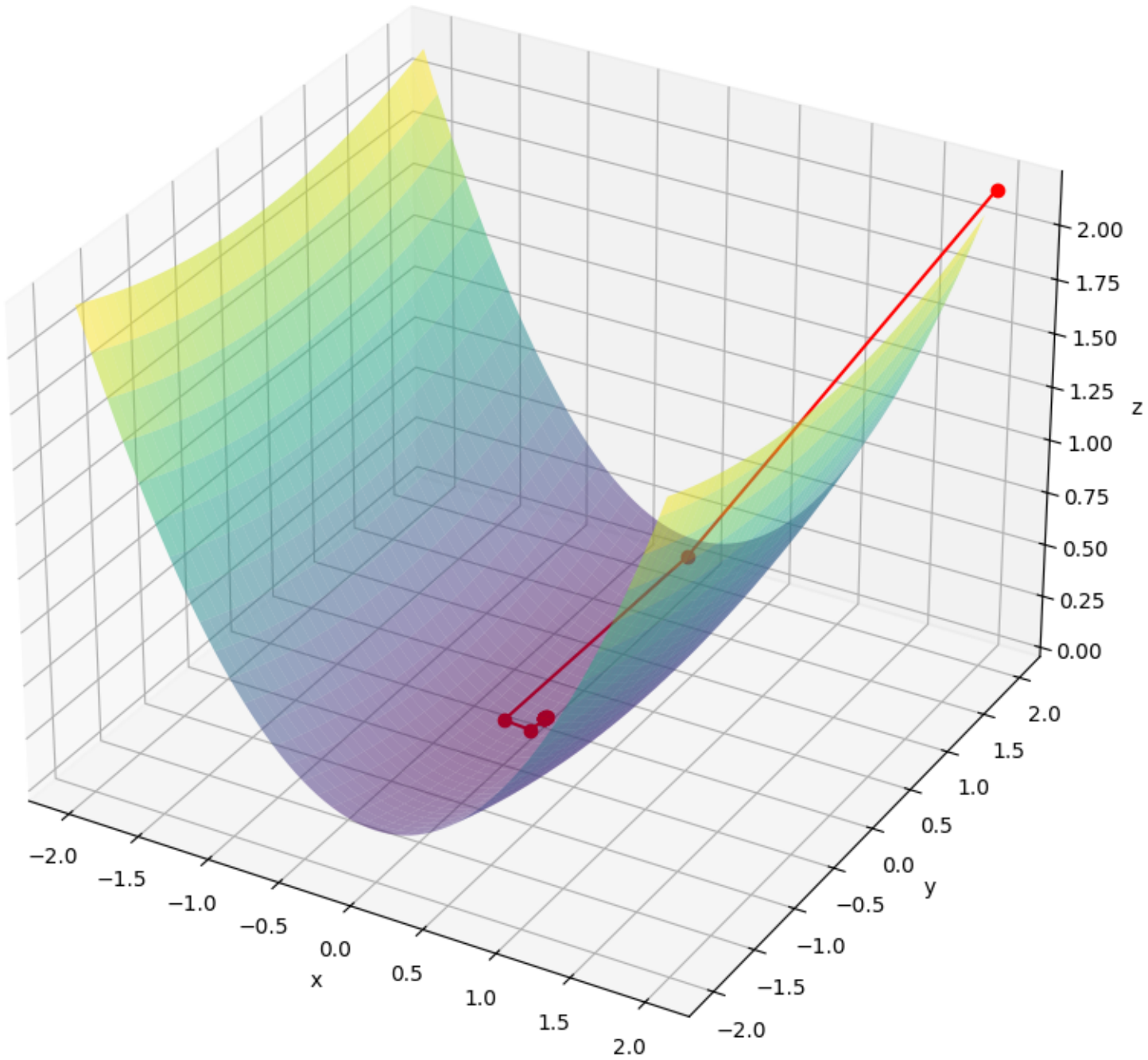
$$\alpha_k = \frac{b^2 y_k^2 - x_k^2}{-x_k^2 + b^3 y_k^2}$$

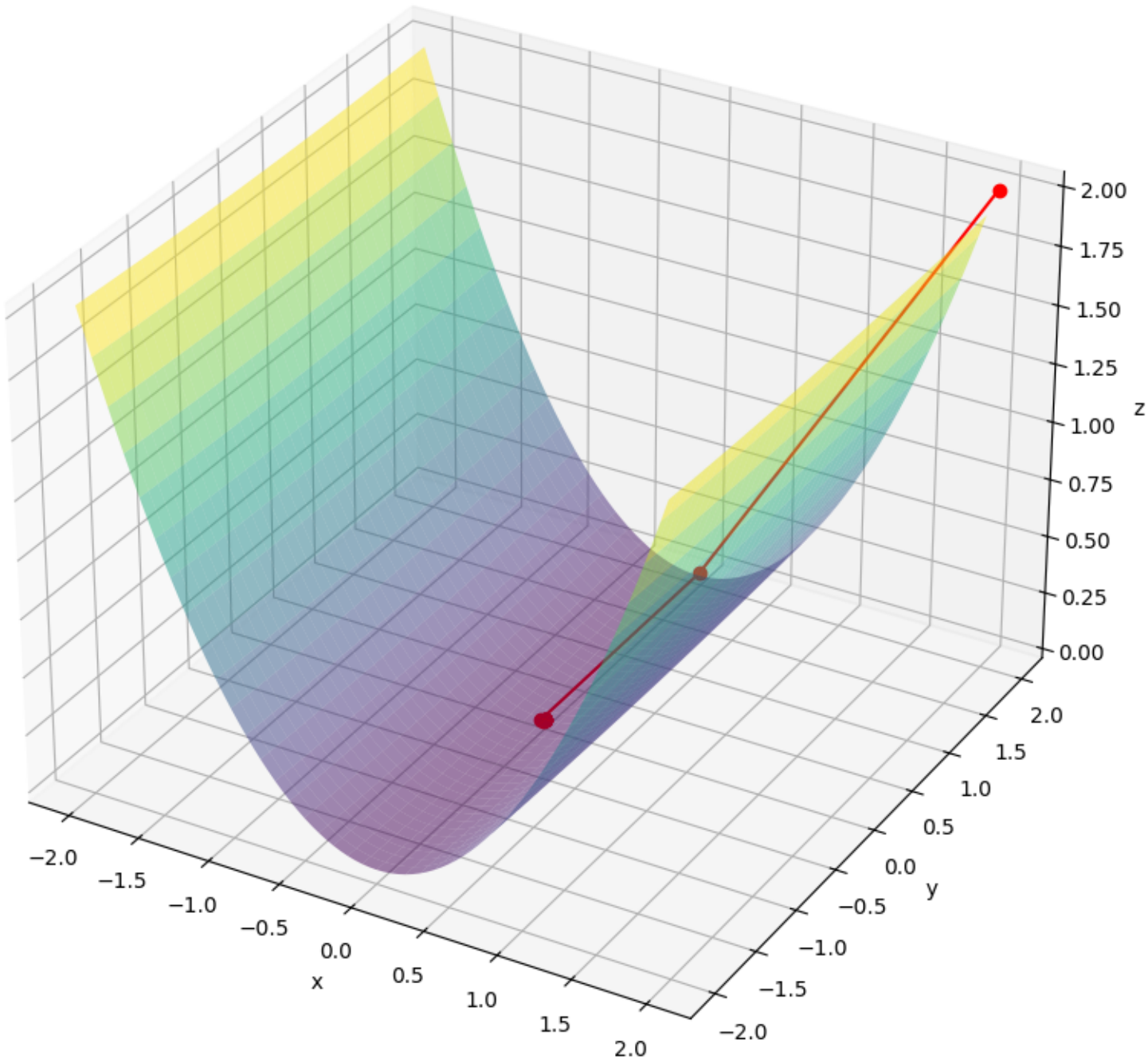
```python
from matplotlib import pyplot as plt
import numpy as np
# Author: Mihai-Dan Crisan 913

def f(x, y, b):
    return 1/2 * (x**2 + b * y**2)

def grad_f(x, y, b):
    return [x, b * y]

def learning_rate(x, y, b): # sk using the formula from the written homework
    if (b**3 * y**2 - x**2) == 0:
        return (b**2 * y**2 - x**2) / 0.0000000001
    return (b**2 * y**2 - x**2) / (b**3 * y**2 - x**2)

def grad_descent(x0, y0, b, iterations):
    x = x0
    y = y0
    x_list = [x0]
    y_list = [y0]
    grad = grad_f(x, y, b)

    for i in range(iterations):
        sk = learning_rate(x, y, b)
        x = x * (1 - sk)
        y = y * (1 - b * sk)
        x_list.append(x)
        y_list.append(y)
        grad = grad_f(x, y, b)
    return x_list, y_list

def main():
    b_values = [1, 1/2, 1/5, 1/10, 1/100]
    for b in b_values:
        fig = plt.figure(figsize=(10, 10))
        ax = fig.add_subplot(111, projection='3d')

        x = np.arange(-2, 2, 0.05)
        y = np.arange(-2, 2, 0.05)
        X, Y = np.meshgrid(x, y)
        Z = f(X, Y, b)
        ax.plot_surface(X, Y, Z, cmap='viridis', alpha=0.5)

        x0 = 2
        y0 = 2

        iterations = 10
        x_list, y_list = grad_descent(x0, y0, b, iterations)
        z_list = [f(x, y, b) for x, y in zip(x_list, y_list)]
        ax.plot(x_list, y_list, z_list, 'ro-')

        ax.set_xlabel('x')
        ax.set_ylabel('y')
        ax.set_zlabel('z')
        plt.show()

"""
for b = 1 it gets stuck in the first iteration because the learning rate is 0
As b gets smaller, the learning rate gets bigger and the algorithm converges faster
Graphically it looks like the algorithm is going straight down the slope of the function
and then jumps to the global minimum.
If b were to be 0 the algorithm should jump straight to the global minimum which will be
the entire x axis.
"""

if __name__ == "__main__":
    main()
```


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
z
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
y


3.0
2.5
2.0
1.5
1.0
0.5
0.0
z
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
y
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x


x
y
z
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


2.00
1.75
1.50
1.25
z
1.00
0.75
0.50
0.25
0.00
2.0
1.5
1.0
0.5
0.0
y
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
x
0.5
1.0
1.5
2.0


2.00
1.75
1.50
1.25
z
1.00
0.75
0.50
0.25
0.00
2.0
1.5
1.0
0.5
0.0
y
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
x
0.5
1.0
1.5
2.0