1

September 30, 2023

0.0.1 Importing necessary libraries

[1]:
```
import numpy as np
```

0.0.2 Problem 1 (a)

$f(x) = x^T Ax + b$
$\Rightarrow \nabla f(x) = 2Ax$
where $A = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 1 \end{bmatrix}$ By setting the gradient to zero $\Rightarrow 2Ax = 0$
$\Rightarrow x = 0$

[2]:
```
A_a = np.matrix([[2, -1, -1], [-1, 2, 0], [-1, 0, 1]])
b_a = np.matrix([[1]])

def f_a(x):
    return x.T @ A_a @ x + b_a

def gradient_f_a(x):
    return 2* A_a @ x

def gradient_descent(x0, learning_rate, num_iterations):
    x = x0
    for i in range(num_iterations):
        x = x - learning_rate * gradient_f_a(x)
    return x

N = 1000
t = 0.02

x0 = np.random.rand(3,1)
result_gradient_descent = gradient_descent(x0, t, N)
print("Minimizer using gradient descent:", result_gradient_descent)
print("Value of f_a at minimizer:", f_a(result_gradient_descent))
result_gradient_zero = (1/2) * A_a.I @ np.matrix([[0],[0],[0]])
print("Minimizer using gradient zero:", result_gradient_zero)
```

```
print("Value of f_a at minimizer using gradient zero:",␣
 ↪f_a(result_gradient_zero))
```

```
Minimizer using gradient descent: [[1.25703112e-04]
 [6.97599643e-05]
 [1.56749217e-04]]
Value of f_a at minimizer: [[1.00000001]]
Minimizer using gradient zero: [[0.]
 [0.]
 [0.]]
Value of f_a at minimizer using gradient zero: [[1.]]
```

0.0.3 Problem 1 (b)

$f(x) = ||Ax - b||^2$
$\Rightarrow \nabla f(x) = 2A^T(Ax - b)$
where $A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}$ By setting the gradient to zero $\Rightarrow 2A^T(Ax - b) = 0$
$\Rightarrow A^T(Ax - b) = 0$
$\Rightarrow A^TAx = A^Tb$
$\Rightarrow x = (A^TA)^{-1}A^Tb$

[3]:
```
A_b = np.matrix([[1,2],[2,4],[3,1]])
b_b = np.matrix([[1],[3],[1]])

def f_b(x):
    return x.T @ A_b.T @ A_b @ x - b_b.T @ A_b @ x - x.T @ A_b.T @ b_b + b_b.T␣
 ↪@ b_b

def gradient_f_b(x):
    return 2* A_b.T @ (A_b @ x - b_b)

def gradient_descent(x0, learning_rate, num_iterations):
    x = x0
    for i in range(num_iterations):
        x = x - learning_rate * gradient_f_b(x)
    return x

N = 1000
t = 0.02
x0 = np.random.rand(2,1)
result_gradient_descent = gradient_descent(x0, t, N)
print("Minimizer using gradient descent:", result_gradient_descent)
print("Value of f_b at minimizer:", f_b(result_gradient_descent))
result_gradient_zero = (A_b.T @ A_b).I @ A_b.T @ b_b
print("Minimizer using gradient zero:", result_gradient_zero)
```

```python
print("Value of f_b at minimizer using gradient zero:",
 ↪f_b(result_gradient_zero))
```

```
Minimizer using gradient descent: [[0.12]
 [0.64]]
Value of f_b at minimizer: [[0.2]]
Minimizer using gradient zero: [[0.12]
 [0.64]]
Value of f_b at minimizer using gradient zero: [[0.2]]
```

0.0.4 Problem 1 (c)

$f(x) = ||Ax - b||^2$
$\Rightarrow \nabla f(x) = 2A^T(Ax - b)$
where $A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 3 & 1 & 9 \\ 4 & 1 & 0 \\ 2 & 1 & 4 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 3 \\ 1 \\ 0 \\ 9 \end{bmatrix}$ By setting the gradient to zero $\Rightarrow 2A^T(Ax - b) = 0$
$\Rightarrow A^T(Ax - b) = 0$
$\Rightarrow A^TAx = A^Tb$
$\Rightarrow x = (A^TA)^{-1}A^Tb$

[4]:
```python
A_c = np.matrix([[1,2,1],[2,4,2],[3,1,9],[4,1,0],[2,1,4]])
b_c = np.matrix([[1],[3],[1],[0],[9]])

def f_c(x):
    return x.T @ A_c.T @ A_c @ x - b_c.T @ A_c @ x - x.T @ A_c.T @ b_c + b_c.T
 ↪@ b_c

def gradient_f_c(x):
    return 2* A_c.T @ (A_c @ x - b_c)

def gradient_descent(x0, learning_rate, num_iterations):
    x = x0
    for i in range(num_iterations):
        x = x - learning_rate * gradient_f_c(x)
    return x

N = 1000
t = 0.02
x0 = np.random.rand(3,1)
# result_gradient_descent = gradient_descent(x0, t, N)
# print("Minimizer using gradient descent:", result_gradient_descent)
# print("Value of f_c at minimizer:", f_c(result_gradient_descent))
result_gradient_zero = (A_c.T @ A_c).I @ A_c.T @ b_c
print("Minimizer using gradient zero:", result_gradient_zero)
```

```python
print("Value of f_c at minimizer using gradient zero:", f_c(result_gradient_zero))
```

```
Minimizer using gradient zero: [[0.05744939]
 [0.65686105]
 [0.33915902]]
Value of f_c at minimizer using gradient zero: [[56.99048278]]
```