Homework 5

Tiut Cristian

November 9, 2023

(a)

Let $f(x) = x^2$ be the convex function, so $f'(x) = 2x$.

So we get that $x_{n+1} = x_n - \eta \cdot 2x_n \implies x_{n+1} = x_n(1 - 2\eta) \implies \frac{x_{n+1}}{x_n} = 1 - 2\eta, \forall n \geq 1$.

Hence, $x_n = x_1(1 - 2\eta)^{n-1}, \forall n > 1$.

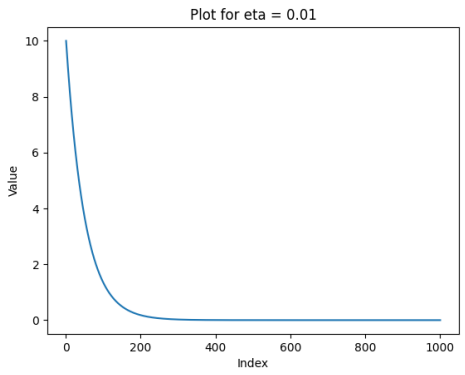
Since $Imf = [0, \infty)$, we need to show that the sequence is approaching 0.

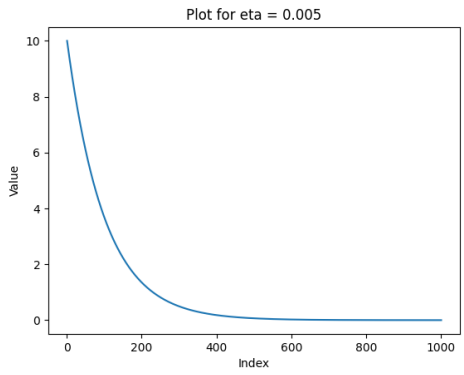
We will pick $x_1 = 10$.

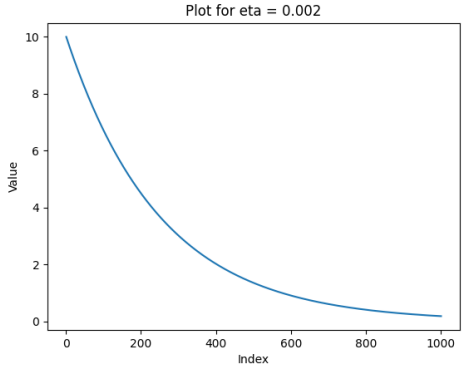
```python
import matplotlib.pyplot as plt
import numpy as np

x1 = 10  # first element of the (x_n) sequence
no_of_elements = 1000
eta_values = [0.01, 0.005, 0.002]

x = np.linspace( start: 1, no_of_elements, no_of_elements)  # take the first "no_of_elements" elements of the sequence
for eta in eta_values:
    y = x1 * (1 - 2 * eta) ** (x - 1)
    fig, ax = plt.subplots()
    ax.plot(x, y)
    plt.title(f"Plot for eta = {eta}")
    plt.xlabel("Index")
    plt.ylabel("Value")
    plt.show()
```







From the plots we get that the graph is eventually approaching 0 for small values of η.

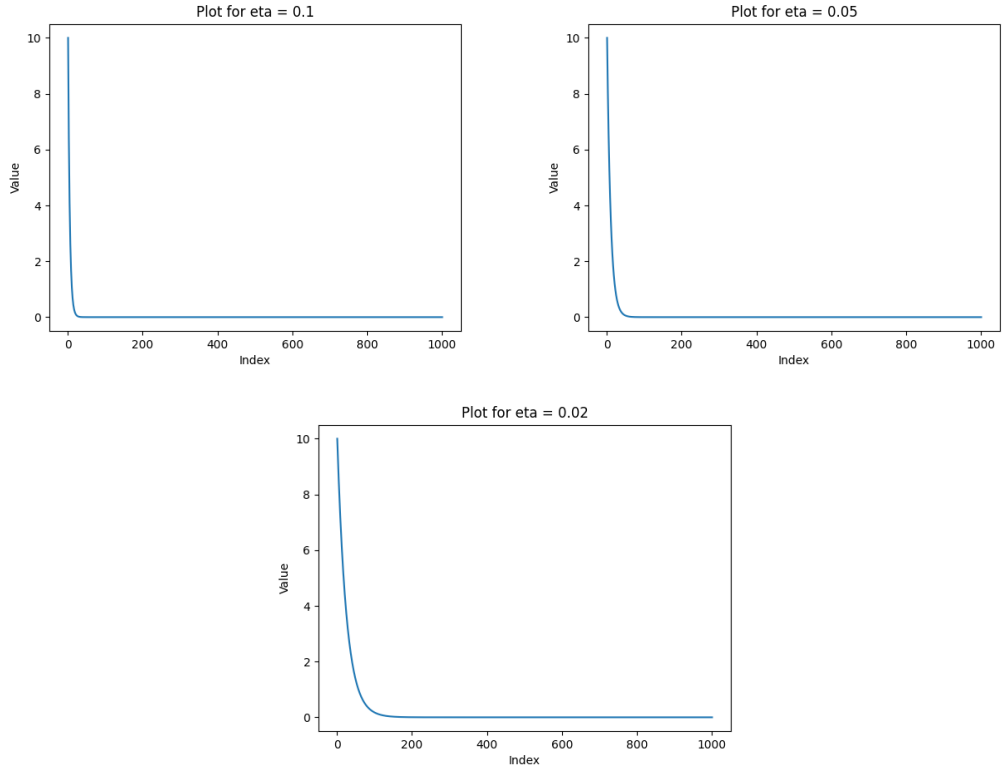
(b)

Let us multiply each value of η from (a) by 10 and visualize the results.

```python
import matplotlib.pyplot as plt
import numpy as np

x1 = 10  # first element of the (x_n) sequence
no_of_elements = 1000
eta_values = [0.1, 0.05, 0.02]

x = np.linspace( start: 1, no_of_elements, no_of_elements)  # take the first "no_of_elements" elements of the sequence
for eta in eta_values:
    y = x1 * (1 - 2 * eta) ** (x - 1)
    fig, ax = plt.subplots()
    ax.plot(x, y)
    plt.title(f"Plot for eta = {eta}")
    plt.xlabel("Index")
    plt.ylabel("Value")
    plt.show()
```



From the plots we get that by increasing η, the sequence converges faster to 0 (the minimum of f).

(c)

From (a) we get that $x_n = x_1(1-2\eta)^{n-1}, \forall n > 1$.

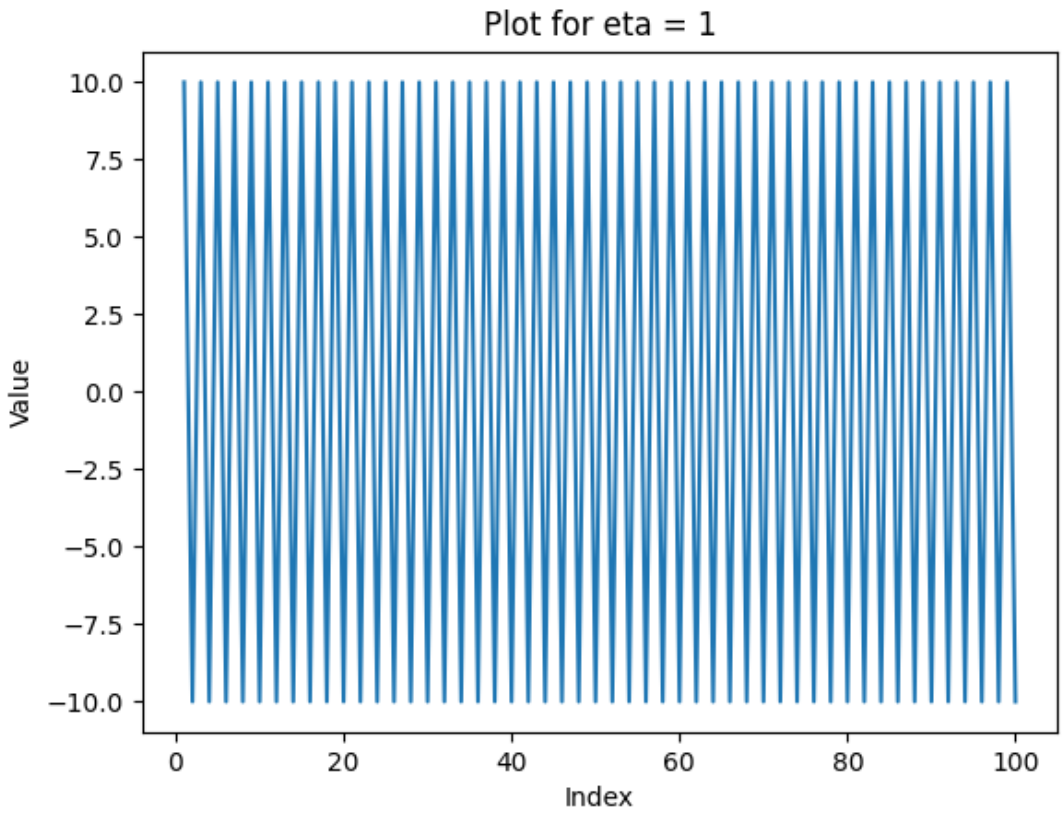
So if we take $\eta = 1$, the general term of the sequence becomes $x_n = x_1(-1)^{n-1}$, which diverges since it has 2 subsequences converging to different values.

This can be seen in the below plot:

```python
import matplotlib.pyplot as plt
import numpy as np

x1 = 10  # first element of the (x_n) sequence
no_of_elements = 100
eta_values = [1]

x = np.linspace( start: 1, no_of_elements, no_of_elements)  # take the first "no_of_elements" elements of the sequence
for eta in eta_values:
    y = x1 * (1 - 2 * eta) ** (x - 1)
    fig, ax = plt.subplots()
    ax.plot(x, y)
    plt.title(f"Plot for eta = {eta}")
    plt.xlabel("Index")
    plt.ylabel("Value")
    plt.show()
```



(d)

Let $f(x) = x^4 - 4x^2$ be the nonconvex function. $\implies f'(x) = 4x^3 - 8x$.

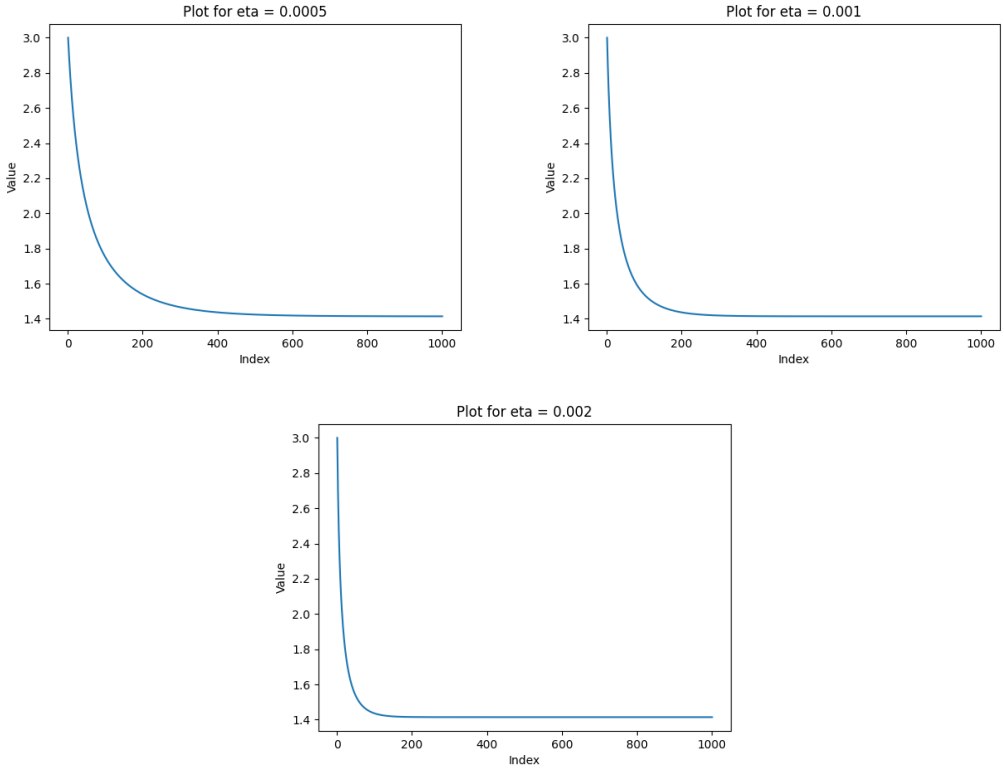
Therefore, $x_{n+1} = x_n - \eta(4x_n^3 - 8x_n), \forall n \geq 1$.

$\eta > 0 \implies 1 + 2\eta > 1$

```python
import matplotlib.pyplot as plt

x1 = 3  # first element of the (x_n) sequence
no_of_elements = 1000
eta_values = [0.0005, 0.001, 0.002]

for eta in eta_values:
    tmp = [x1]
    for i in range(1, no_of_elements):
        x = tmp[i-1]
        tmp.append(x - eta*(4 * x**3 - 8*x))
    plt.plot( *args: range(1, no_of_elements+1), tmp)
    plt.title(f"Plot for eta = {eta}")
    plt.xlabel("Index")
    plt.ylabel("Value")
    plt.show()
```



From the plots above we can see that the function gets stuck in a local minimum, which in this case is around 1.4.