



Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

Licenciatura en Ingeniería en Computación

Materia: Seminario de Solución de Problemas de Inteligencia Artificial I. Clave: I7039.

Profesor: Sencion Echauri Felipe

Estudiante: Silva Moya José Alejandro. Código: 213546894.

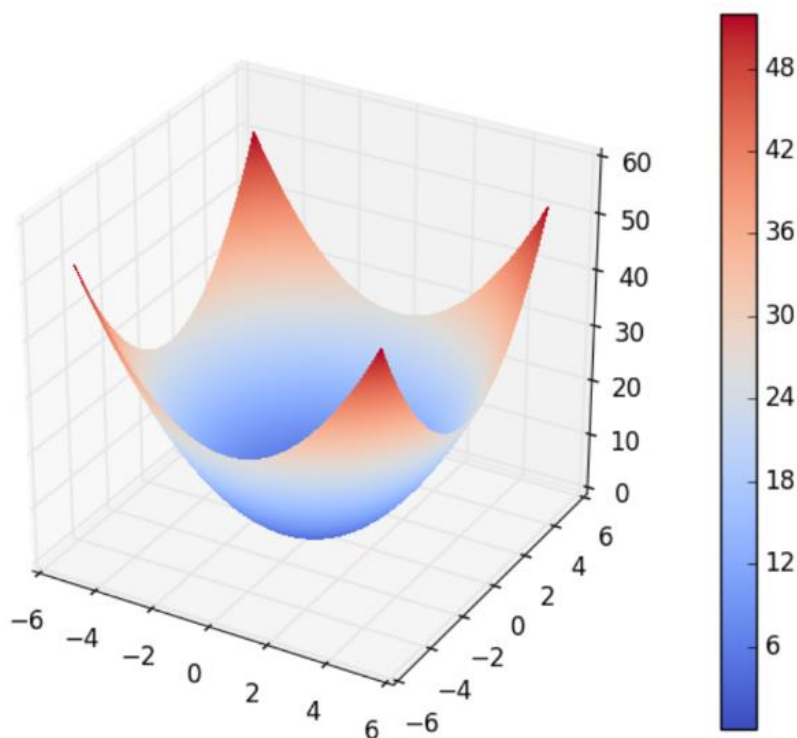
Actividad 4: Funciones para optimizar.



Instrucciones: Graficar las funciones descritas en el archivo adjunto usando Python.

Ejercicio 1

Sphere

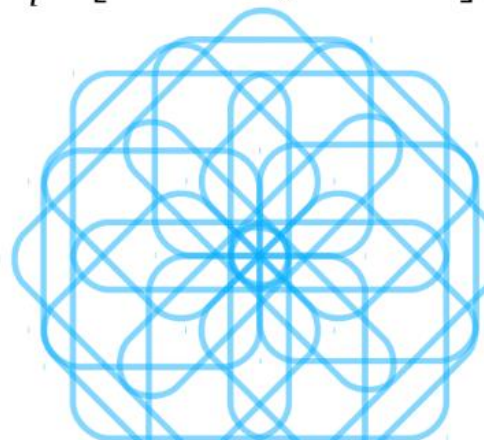


$$f(x) = \sum_{i=1}^n x_i^2$$

$$x^* = 0$$

$$f(x^*) = 0$$

$$x_i \in [-5.12, +5.12]$$



```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
from matplotlib import cm #Para modelos de color
import numpy as np
```

```
#####EJERCICIO 1#####
```

```
fig = plt.figure()
```

```
ax = fig.gca(projection = '3d')
```

```

x = np.arange(-5.12,5.12,0.01) #Puntos de -5.12 hasta 5.12 con aumentos de 0.01 en eje X.
y = np.arange(-5.12,5.12,0.01) #Puntos de -5.12 hasta 5.12 con aumentos de 0.01 en eje Y.
x,y = np.meshgrid(x,y)
z = x**2 + y**2 #Nuestra funcion generada en terminos de Z.

```

```

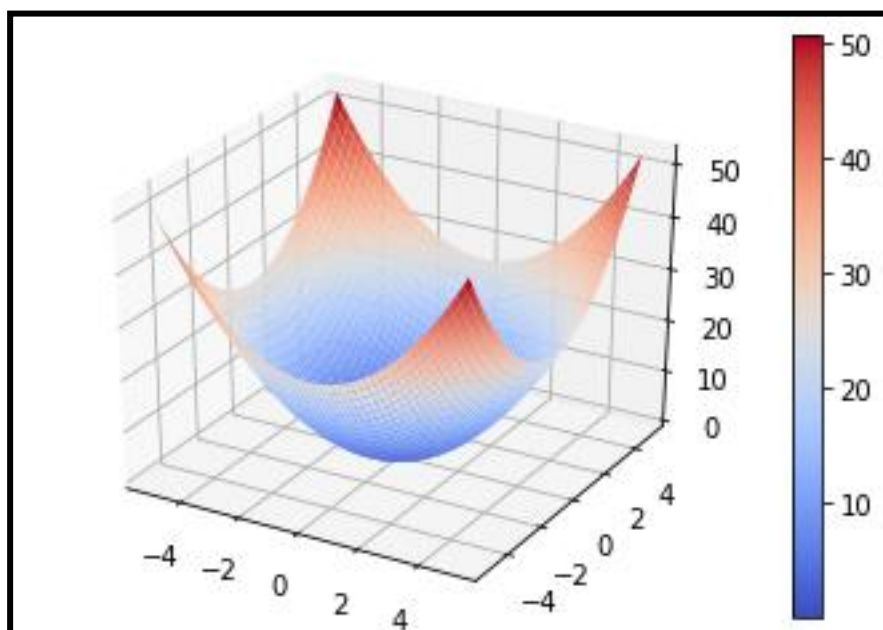
surf = ax.plot_surface(x, y, z, cmap = cm.coolwarm)
fig.colorbar(surf)
fig.show()

```

```

1 from mpl_toolkits.mplot3d import Axes3D
2 import matplotlib.pyplot as plt
3 from matplotlib import cm #Para modelos de color
4 import numpy as np
5
6 #####EJERCICIO 1#####
7 fig = plt.figure()
8 ax = fig.gca(projection = '3d')
9
10 x = np.arange(-5.12,5.12,0.01) #Puntos de -5.12 hasta 5.12 con aumentos de 0.01 en eje X.
11 y = np.arange(-5.12,5.12,0.01) #Puntos de -5.12 hasta 5.12 con aumentos de 0.01 en eje Y.
12 x,y = np.meshgrid(x,y)
13 z = x**2 + y**2 #Nuestra funcion generada en terminos de Z.
14
15 surf = ax.plot_surface(x, y, z, cmap = cm.coolwarm)
16 fig.colorbar(surf)
17 fig.show()
18 #####

```



Ejercicio 2

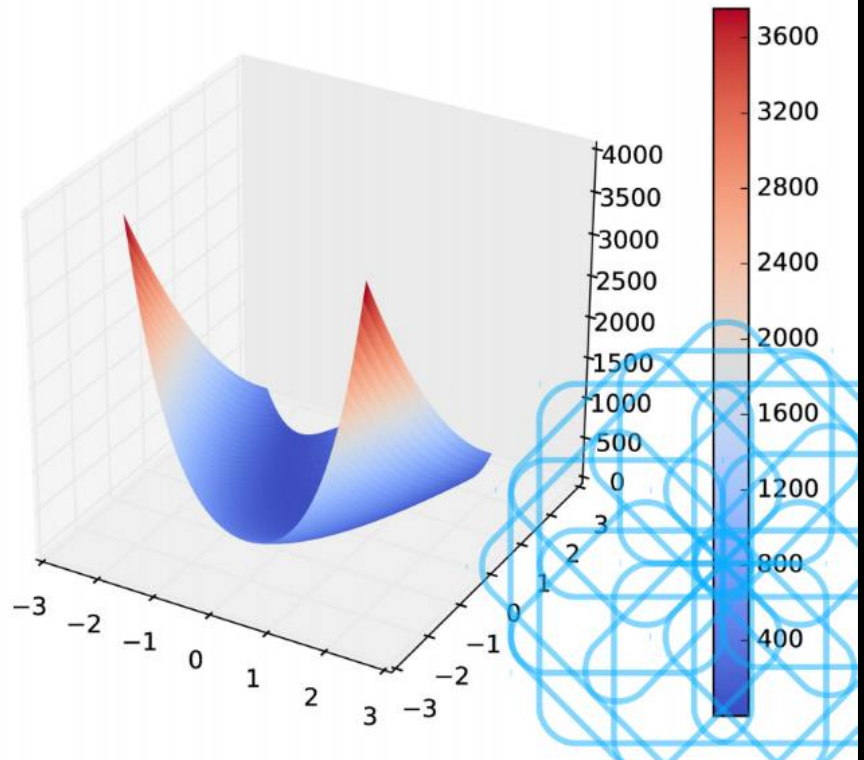
Rosenbrock

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

$$x^* = [1, \dots, 1]$$

$$f(x^*) = 0$$

$$x_i \in [-2.048, +2.048]$$



#####EJERCICIO 2#####

```
fig = plt.figure()
```

```
ax = fig.gca(projection = '3d')
```

```
x = np.arange(-2.048,2.048,0.01) #Puntos de -2.048 hasta 2.048 con aumentos de 0.01 en eje X.
```

```
y = np.arange(-2.048,2.048,0.01) #Puntos de -2.048 hasta 2.048 con aumentos de 0.01 en eje Y.
```

```
x,y = np.meshgrid(x,y)
```

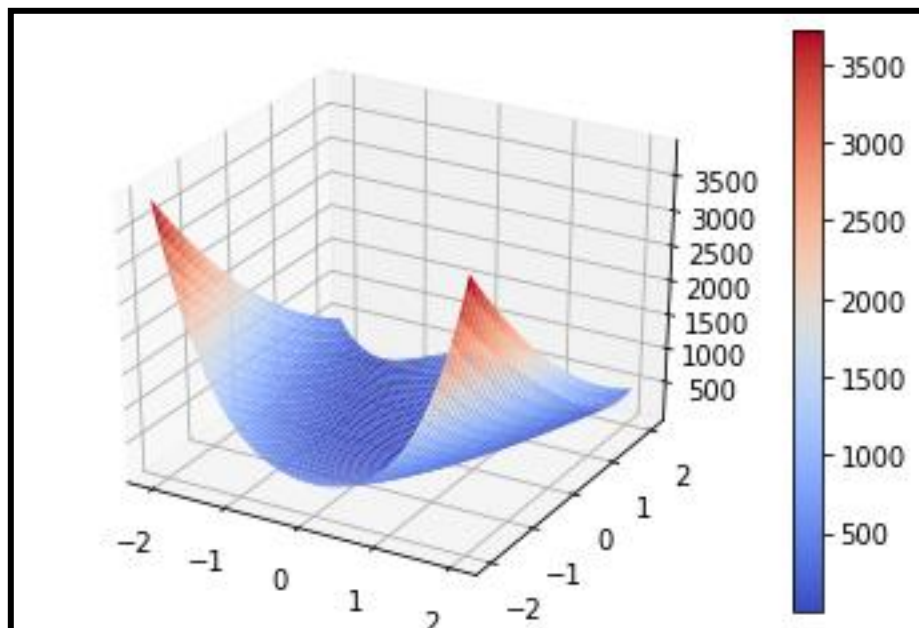
```
z = 100*(y - (x**2))**2 + (x - 1)**2 #Nuestra funcion generada en terminos de Z.
```

```
surf = ax.plot_surface(x, y, z, cmap = cm.coolwarm)
```

```
fig.colorbar(surf)
```

```
fig.show()
```

```
20 #####EJERCICIO 2#####
21 fig = plt.figure()
22 ax = fig.gca(projection = '3d')
23
24 x = np.arange(-2.048,2.048,0.01) #Puntos de -2.048 hasta 2.048 con aumentos de 0.01 en eje X.
25 y = np.arange(-2.048,2.048,0.01) #Puntos de -2.048 hasta 2.048 con aumentos de 0.01 en eje Y.
26 x,y = np.meshgrid(x,y)
27 z = 100*(y - (x**2))**2 + (x - 1)**2 #Nuestra funcion generada en terminos de Z.
28
29 surf = ax.plot_surface(x, y, z, cmap = cm.coolwarm)
30 fig.colorbar(surf)
31 fig.show()
32 #####
```



Ejercicio 3

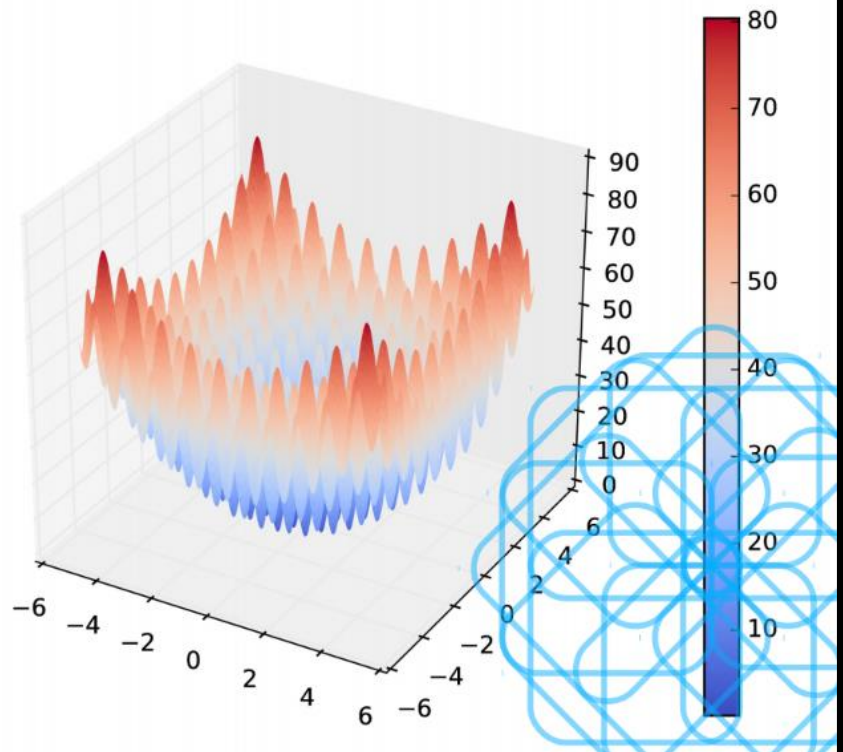
Rastrigin

$$f(x) = 10n + \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i)$$

$$x^* = 0$$

$$f(x^*) = 0$$

$$x_i \in [-5.12, +5.12]$$



#####EJERCICIO 3#####

```
fig = plt.figure()
```

```
ax = fig.gca(projection = '3d')
```

```
x = np.arange(-5.12,5.12,0.01) #Puntos de -5.12 hasta 5.12 con aumentos de 0.01 en eje X.
```

```
y = np.arange(-5.12,5.12,0.01) #Puntos de -5.12 hasta 5.12 con aumentos de 0.01 en eje Y.
```

```
x,y = np.meshgrid(x,y)
```

```
z = 10*2 + (x**2 - 10*(np.cos(2*np.pi*x))) + (y**2 - 10*(np.cos(2*np.pi*y)))
```

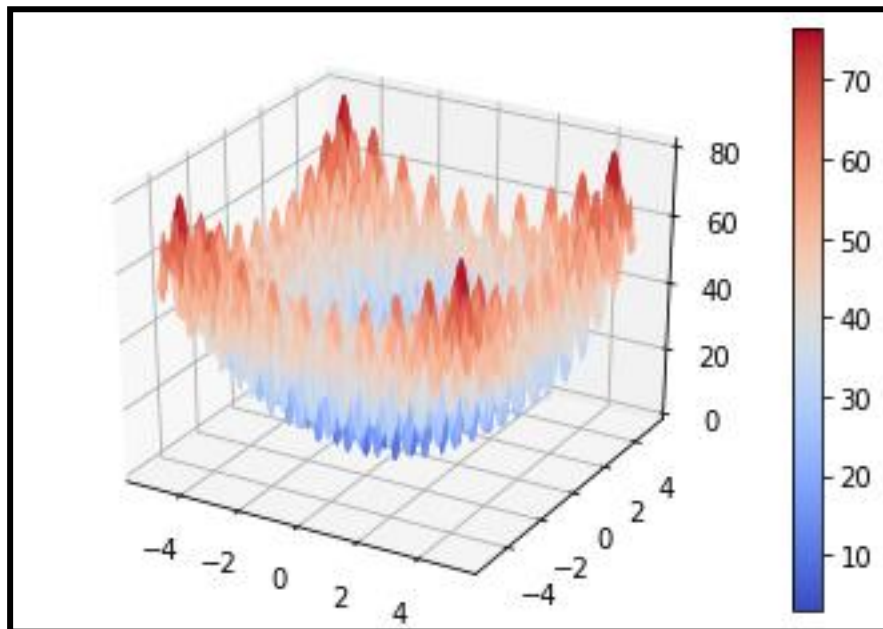
```
#Nuestra funcion generada en terminos de Z.
```

```
surf = ax.plot_surface(x, y, z, cmap = cm.coolwarm)
```

```
fig.colorbar(surf)
```

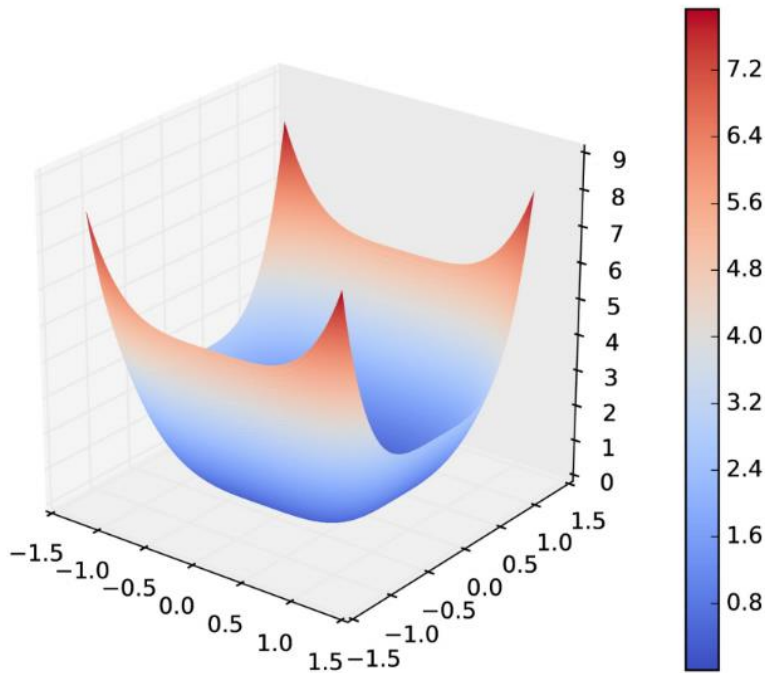
```
fig.show()
```

```
34 #####EJERCICIO 3#####
35 fig = plt.figure()
36 ax = fig.gca(projection = '3d')
37
38 x = np.arange(-5.12,5.12,0.01) #Puntos de -5.12 hasta 5.12 con aumentos de 0.01 en eje X.
39 y = np.arange(-5.12,5.12,0.01) #Puntos de -5.12 hasta 5.12 con aumentos de 0.01 en eje Y.
40 x,y = np.meshgrid(x,y)
41 z = 10*2 + (x**2 - 10*(np.cos(2*np.pi*x))) + (y**2 - 10*(np.cos(2*np.pi*y)))
42 #Nuestra funcion generada en terminos de Z.
43
44 surf = ax.plot_surface(x, y, z, cmap = cm.coolwarm)
45 fig.colorbar(surf)
46 fig.show()
```

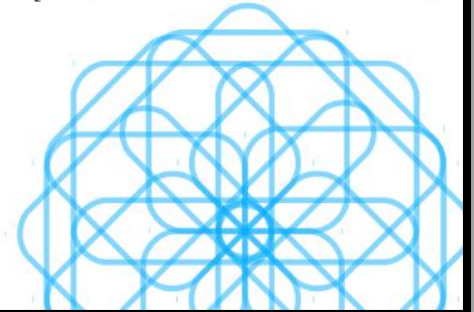


Ejercicio 4

Quartic



$$f(x) = \sum_{i=1}^n ix_i^4$$
$$x^* = 0$$
$$f(x^*) = 0$$
$$x_i \in [-1.28, +1.28]$$



#####EJERCICIO 4#####

```
fig = plt.figure()
```

```
ax = fig.gca(projection = '3d')
```

```
x = np.arange(-1.28,1.28,0.01) #Puntos de -1.28 hasta 1.28 con aumentos de 0.01 en eje X.
```

```
y = np.arange(-1.28,1.28,0.01) #Puntos de -1.28 hasta 1.28 con aumentos de 0.01 en eje Y.
```

```
x,y = np.meshgrid(x,y)
```

```
z = 1*(x**4) + 2*(y**4)
```

```
surf = ax.plot_surface(x, y, z, cmap = cm.coolwarm)
```

```
fig.colorbar(surf)
```

```
fig.show()
```



```

49 #####EJERCICIO 4#####
50 fig = plt.figure()
51 ax = fig.gca(projection = '3d')
52
53 x = np.arange(-1.28,1.28,0.01) #Puntos de -1.28 hasta 1.28 con aumentos de 0.01 en eje X.
54 y = np.arange(-1.28,1.28,0.01) #Puntos de -1.28 hasta 1.28 con aumentos de 0.01 en eje Y.
55 x,y = np.meshgrid(x,y)
56 z = 1*(x**4) + 2*(y**4)
57
58 surf = ax.plot_surface(x, y, z, cmap = cm.coolwarm)
59 fig.colorbar(surf)
60 fig.show()

```

