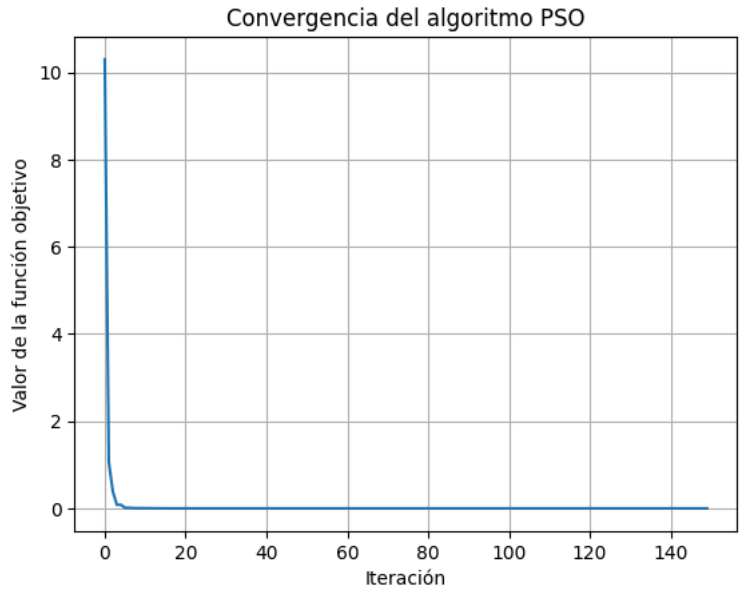
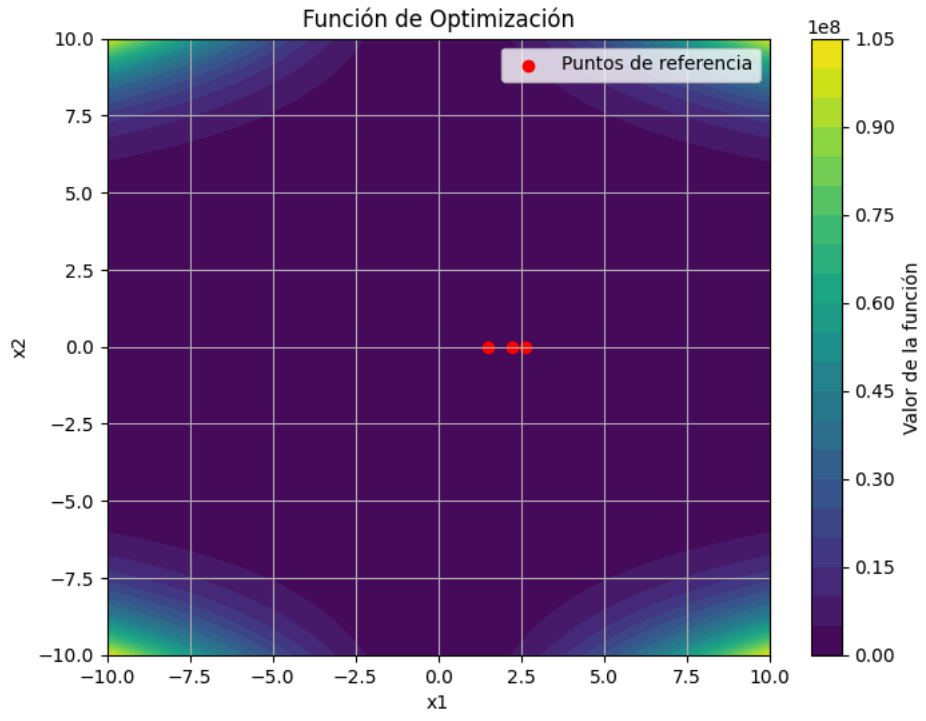
```
import numpy as np
import matplotlib.pyplot as plt

class Particle:
    def __init__(self, dim):
        self.position = np.random.uniform(-10, 10, dim)
        self.velocity = np.random.uniform(-1, 1, dim)
        self.best_position = np.copy(self.position)
        self.best_score = float('inf')

def objective_function(x):
    x1, x2 = x
    return (1.5 - x1 + x1*x2)**2 + (2.25 - x1 + x1*x2**2)**2 + (2.625 - x1 + x1*x2**3)**2

def update_velocity(particle, global_best_position, w=0.6, c1=0.8, c2=0.9):
    inertia = w * particle.velocity
    cognitive_component = c1 * np.random.random() * (particle.best_position - particle.position)
    social_component = c2 * np.random.random() * (global_best_position - particle.position)

    new_velocity = inertia + cognitive_component + social_component
    return new_velocity

def pso(dimensions=2, n_particles=30, n_iterations=150):
    particles = [Particle(dimensions) for _ in range(n_particles)]

    global_best_position = None
    global_best_score = float('inf')

    # Lista para almacenar los valores de la función objetivo
    convergence_values = []

    for _ in range(n_iterations):
        for particle in particles:
            score = objective_function(particle.position)
            # Crear una malla de puntos para graficar
            x1_vals = np.linspace(-10, 10, 100)
            x2_vals = np.linspace(-10, 10, 100)
            X1, X2 = np.meshgrid(x1_vals, x2_vals)
            Z = objective_function([X1, X2])
            if score < particle.best_score:
                particle.best_score = score
                particle.best_position = np.copy(particle.position)

            if score < global_best_score:
                global_best_score = score
                global_best_position = np.copy(particle.position)

        for particle in particles:
            particle.velocity = update_velocity(particle,
                                                global_best_position)
            particle.position += particle.velocity
            np.clip(particle.position, -10, 10, out=particle.position)

        print(f"Mejor posición hasta ahora: {global_best_position}, Valor en la mejor posición: {global_best_score}")
        # Guarda el mejor puntaje global en la lista de convergencia
        convergence_values.append(global_best_score)

    # Grafica la convergencia
    plt.plot(convergence_values)
    plt.xlabel('Iteración')
    plt.ylabel('Valor de la función objetivo')
    plt.title('Convergencia del algoritmo PSO')
    plt.grid(True)
    plt.show()
pso()


# Graficar la función
plt.figure(figsize=(8, 6))
plt.contourf(X1, X2, Z, levels=20, cmap='viridis')
plt.colorbar(label='Valor de la función')
plt.xlabel('x1')
plt.ylabel('x2')
plt.title('Función de Optimización')
plt.scatter([1.5, 2.25, 2.625], [0, 0, 0], color='red', marker='o', label='Puntos de referencia')
plt.legend()
plt.grid(True)
plt.show()
```

```
Mejor posición hasta ahora: [8.97018067 0.74719644], Valor en la mejor posición: 10.297130310236751
Mejor posición hasta ahora: [1.86484409 0.3180298 ], Valor en la mejor posición: 1.0539356804811477
Mejor posición hasta ahora: [2.97221289 0.34562319], Valor en la mejor posición: 0.3831893840167865
Mejor posición hasta ahora: [3.75809495 0.61335371], Valor en la mejor posición: 0.08181424567971
Mejor posición hasta ahora: [3.75809495 0.61335371], Valor en la mejor posición: 0.08181424567971
Mejor posición hasta ahora: [2.94790105 0.46246318], Valor en la mejor posición: 0.012686034373110585
Mejor posición hasta ahora: [2.94790105 0.46246318], Valor en la mejor posición: 0.012686034373110585
Mejor posición hasta ahora: [3.19668522 0.53758659], Valor en la mejor posición: 0.006628772460988024
Mejor posición hasta ahora: [2.91326434 0.49320324], Valor en la mejor posición: 0.0063660936093255585
Mejor posición hasta ahora: [2.91326434 0.49320324], Valor en la mejor posición: 0.0063660936093255585
Mejor posición hasta ahora: [2.88153641 0.47364923], Valor en la mejor posición: 0.0029671272561423395
Mejor posición hasta ahora: [2.88153641 0.47364923], Valor en la mejor posición: 0.0029671272561423395
Mejor posición hasta ahora: [3.05143166 0.52088173], Valor en la mejor posición: 0.0021683704876647837
Mejor posición hasta ahora: [3.00344449 0.49710339], Valor en la mejor posición: 0.0003256311909656131
Mejor posición hasta ahora: [3.04205144 0.51095568], Valor en la mejor posición: 0.00028248734556639223
Mejor posición hasta ahora: [3.00370906 0.50195862], Valor en la mejor posición: 2.732174796054398e-05
Mejor posición hasta ahora: [3.00370906 0.50195862], Valor en la mejor posición: 2.732174796054398e-05
Mejor posición hasta ahora: [2.99992141 0.49944899], Valor en la mejor posición: 6.509986037942572e-06
Mejor posición hasta ahora: [2.99992141 0.49944899], Valor en la mejor posición: 6.509986037942572e-06
Mejor posición hasta ahora: [2.99992141 0.49944899], Valor en la mejor posición: 6.509986037942572e-06
Mejor posición hasta ahora: [2.99992141 0.49944899], Valor en la mejor posición: 6.509986037942572e-06
Mejor posición hasta ahora: [2.99936717 0.49980197], Valor en la mejor posición: 1.0299842844927709e-07
Mejor posición hasta ahora: [2.99936717 0.49980197], Valor en la mejor posición: 1.0299842844927709e-07
Mejor posición hasta ahora: [2.99936717 0.49980197], Valor en la mejor posición: 1.0299842844927709e-07
Mejor posición hasta ahora: [2.99936717 0.49980197], Valor en la mejor posición: 1.0299842844927709e-07
Mejor posición hasta ahora: [2.99936717 0.49980197], Valor en la mejor posición: 1.0299842844927709e-07
Mejor posición hasta ahora: [2.99936717 0.49980197], Valor en la mejor posición: 1.0299842844927709e-07
Mejor posición hasta ahora: [2.99936717 0.49980197], Valor en la mejor posición: 1.0299842844927709e-07
Mejor posición hasta ahora: [2.99936717 0.49980197], Valor en la mejor posición: 1.0299842844927709e-07
Mejor posición hasta ahora: [2.99936717 0.49980197], Valor en la mejor posición: 1.0299842844927709e-07
Mejor posición hasta ahora: [2.99943172 0.49985465], Valor en la mejor posición: 5.2168630259450027e-08
Mejor posición hasta ahora: [2.99943172 0.49985465], Valor en la mejor posición: 5.2168630259450027e-08
Mejor posición hasta ahora: [2.99943172 0.49985465], Valor en la mejor posición: 5.2168630259450027e-08
Mejor posición hasta ahora: [2.99943172 0.49985465], Valor en la mejor posición: 5.2168630259450027e-08
Mejor posición hasta ahora: [2.99943172 0.49985465], Valor en la mejor posición: 5.2168630259450027e-08
Mejor posición hasta ahora: [3.00006381 0.50001011], Valor en la mejor posición: 1.4055873557007206e-09
Mejor posición hasta ahora: [3.00006381 0.50001011], Valor en la mejor posición: 1.4055873557007206e-09
Mejor posición hasta ahora: [3.00006381 0.50001011], Valor en la mejor posición: 1.4055873557007206e-09
Mejor posición hasta ahora: [3.00006381 0.50001011], Valor en la mejor posición: 1.4055873557007206e-09
Mejor posición hasta ahora: [3.00004889 0.5000109 ], Valor en la mejor posición: 4.1720880731589427e-10
Mejor posición hasta ahora: [3.00004889 0.5000109 ], Valor en la mejor posición: 4.1720880731589427e-10
Mejor posición hasta ahora: [3.00004889 0.5000109 ], Valor en la mejor posición: 4.1720880731589427e-10
Mejor posición hasta ahora: [3.00004889 0.5000109 ], Valor en la mejor posición: 4.1720880731589427e-10
Mejor posición hasta ahora: [3.00004889 0.5000109 ], Valor en la mejor posición: 4.1720880731589427e-10
Mejor posición hasta ahora: [2.99999372 0.5000011 ], Valor en la mejor posición: 1.688281695215379e-10
Mejor posición hasta ahora: [2.99999372 0.5000011 ], Valor en la mejor posición: 1.688281695215379e-10
Mejor posición hasta ahora: [2.99999372 0.5000011 ], Valor en la mejor posición: 1.688281695215379e-10
Mejor posición hasta ahora: [2.99999372 0.5000011 ], Valor en la mejor posición: 1.688281695215379e-10
Mejor posición hasta ahora: [3.00000021 0.49999976], Valor en la mejor posición: 1.9310496872755176e-12
Mejor posición hasta ahora: [3.00000021 0.49999976], Valor en la mejor posición: 1.9310496872755176e-12
Mejor posición hasta ahora: [3.00000021 0.49999976], Valor en la mejor posición: 1.9310496872755176e-12
Mejor posición hasta ahora: [3.00000021 0.49999976], Valor en la mejor posición: 1.9310496872755176e-12
Mejor posición hasta ahora: [3.00000021 0.49999976], Valor en la mejor posición: 1.9310496872755176e-12
Mejor posición hasta ahora: [3.00000021 0.49999976], Valor en la mejor posición: 1.9310496872755176e-12
Mejor posición hasta ahora: [2.99999939 0.49999981], Valor en la mejor posición: 9.900757132892223e-14
Mejor posición hasta ahora: [2.99999939 0.49999981], Valor en la mejor posición: 9.900757132892223e-14
Mejor posición hasta ahora: [2.99999939 0.49999981], Valor en la mejor posición: 9.900757132892223e-14
Mejor posición hasta ahora: [2.99999939 0.49999981], Valor en la mejor posición: 9.900757132892223e-14
Mejor posición hasta ahora: [2.99999974 0.49999994], Valor en la mejor posición: 1.1268843662995316e-14
Mejor posición hasta ahora: [2.99999974 0.49999994], Valor en la mejor posición: 1.1268843662995316e-14
Mejor posición hasta ahora: [2.99999974 0.49999994], Valor en la mejor posición: 1.1268843662995316e-14
Mejor posición hasta ahora: [2.99999974 0.49999994], Valor en la mejor posición: 1.1268843662995316e-14
Mejor posición hasta ahora: [2.99999974 0.49999994], Valor en la mejor posición: 1.1268843662995316e-14
Mejor posición hasta ahora: [2.99999974 0.49999994], Valor en la mejor posición: 1.1268843662995316e-14
Mejor posición hasta ahora: [2.99999983 0.49999995], Valor en la mejor posición: 8.314367773229009e-15
Mejor posición hasta ahora: [2.99999978 0.49999994], Valor en la mejor posición: 7.690767070929637e-15
Mejor posición hasta ahora: [2.99999996 0.49999998], Valor en la mejor posición: 2.7224509029868066e-15
Mejor posición hasta ahora: [2.99999996 0.49999998], Valor en la mejor posición: 2.7224509029868066e-15
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.620911676656467e-16
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.620911676656467e-16
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.620911676656467e-16
Mejor posición hasta ahora: [2.99999997 0.49999999], Valor en la mejor posición: 1.2032486481751844e-16
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.5893036302801355e-17
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.5893036302801355e-17
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.5893036302801355e-17
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.3496830931480886e-17
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.3496830931480886e-17
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.670886850468231e-18
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.405066191137861e-19
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.405066191137861e-19
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.405066191137861e-19
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 9.347615824800562e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 9.347615824800562e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.0755583884666946e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.0755583884666946e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.0755583884666946e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.0755583884666946e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.0755583884666946e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.0755583884666946e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.0755583884666946e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.0755583884666946e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.0755583884666946e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.0755583884666946e-20
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 7.27503753078463e-21
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 7.27503753078463e-21
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 4.316876156423184e-21
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.3793476964005833e-21
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.3793476964005833e-21
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.3793476964005833e-21
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.3793476964005833e-21
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.584623279272998e-23
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.584623279272998e-23
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.584623279272998e-23
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.584623279272998e-23
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.584623279272998e-23
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.019456055784694e-23
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.019456055784694e-23
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 4.585254658709592e-25
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 4.585254658709592e-25
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 4.585254658709592e-25
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 4.585254658709592e-25
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 4.585254658709592e-25
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.124344270083304e-25
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 4.5744259712265994e-26
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 4.5744259712265994e-26
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.6045005341395705e-26
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.1314101947664277e-26
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 8.293531971157646e-27
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 8.293531971157646e-27
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 8.023627526519352e-27
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 8.023627526519352e-27
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.0835709968474266e-27
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.0835709968474266e-27
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.0835709968474266e-27
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 6.126275301018912e-28
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 5.540022225947437e-28
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.824368559724104e-28
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.824368559724104e-28
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.824368559724104e-28
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 2.824368559724104e-28
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 4.4730878516360185e-29
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.3450694731600455e-29
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.491325803185181e-30
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.491325803185181e-30
```

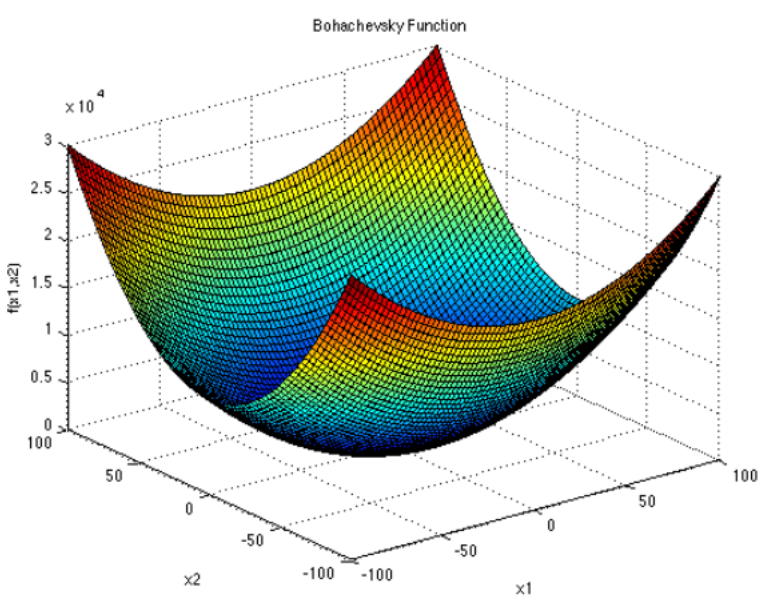
```
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.491325803185181e-30
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 3.491325803185181e-30
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.2449211160519093e-30
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.2449211160519093e-30
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.2634100435180267e-31
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.2634100435180267e-31
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.2634100435180267e-31
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 1.2634100435180267e-31
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 6.162975822039155e-32
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 6.162975822039155e-32
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 0.0
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 0.0
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 0.0
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 0.0
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 0.0
Mejor posición hasta ahora: [3.  0.5], Valor en la mejor posición: 0.0
```





Optimization Test Problems

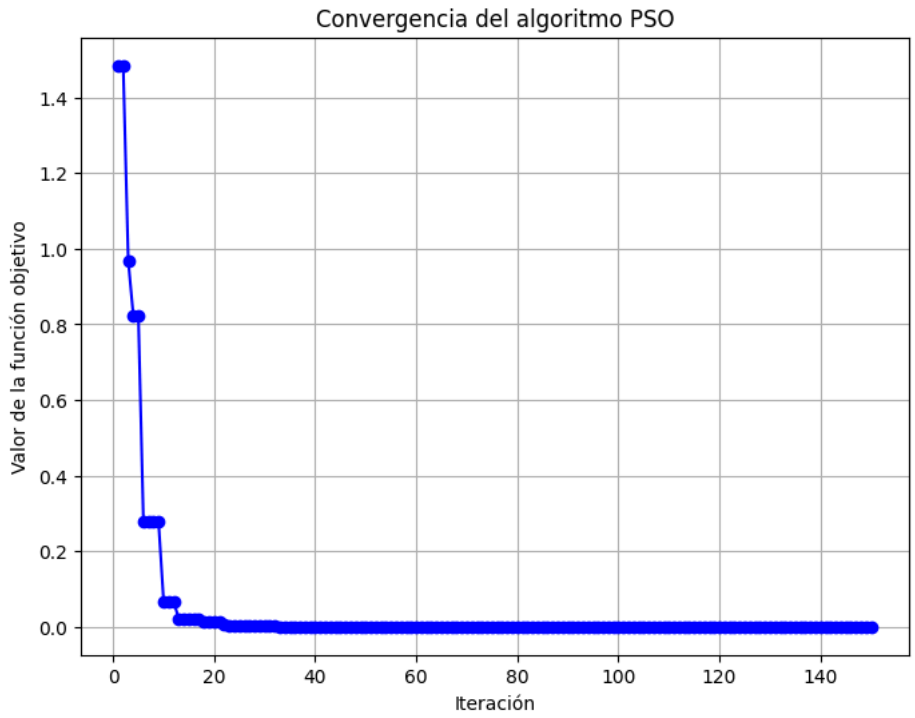
BOHACHEVSKY FUNCTIONS



Funcion utilizada.

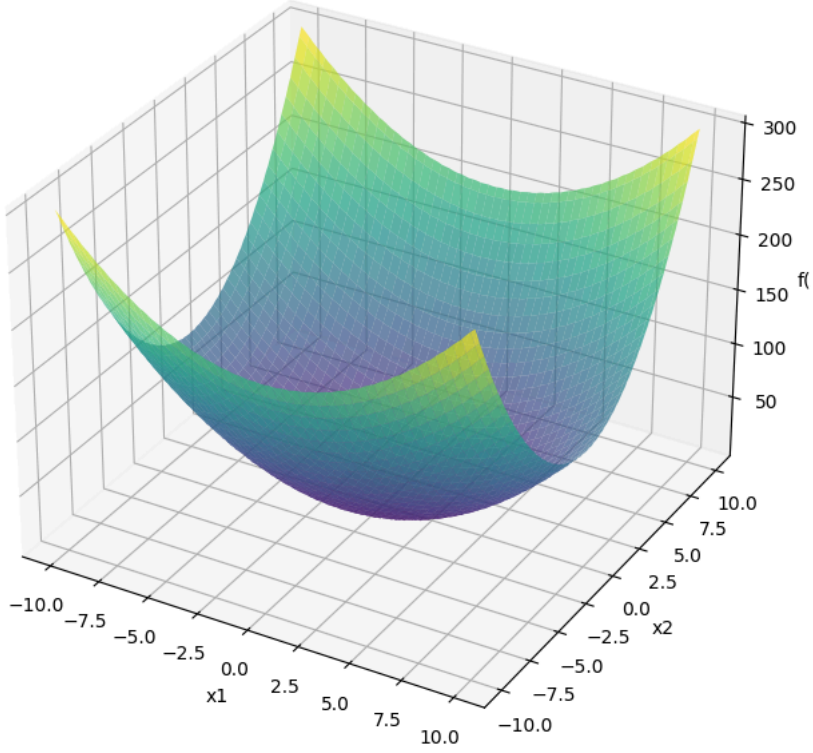
$$f_1(\mathbf{x}) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Función a minimizar
def f(x):
    return x[0]**2 + 2*x[1]**2 - 0.3*np.cos(3*np.pi*x[0]) - 0.4*np.cos(4*np.pi*x[1]) + 0.7

# Algoritmo PSO
class Particle:
    def __init__(self, dim):
        self.position = np.random.uniform(-10, 10, dim)
        self.velocity = np.random.uniform(-1, 1, dim)
        self.best_position = np.copy(self.position)
        self.best_score = f(self.position)

def pso(n_particles, n_iterations, dim):
    particles = [Particle(dim) for _ in range(n_particles)]
    global_best = min(particles, key=lambda x: x.best_score)

    # Listas para almacenar los valores de la función objetivo en cada iteración
    iteration_values = []
    best_values = []

    for _ in range(n_iterations):
        for particle in particles:
           # Actualizar velocidad y posición
            w = 0.7   # Inercia
            c1 = 1.5  # Coeficiente cognitivo
            c2 = 1.5  # Coeficiente social

            r1, r2 = np.random.rand(dim), np.random.rand(dim)

            particle.velocity = (w * particle.velocity +
                                 c1 * r1 * (particle.best_position - particle.position) +
                                 c2 * r2 * (global_best.best_position - particle.position))

            particle.position += particle.velocity

            # Evaluar la nueva posición
            score = f(particle.position)

            if score < particle.best_score:
                particle.best_score = score
                particle.best_position = np.copy(particle.position)

                if score < global_best.best_score:
                    global_best = particle

        # Registrar los valores de la función objetivo
        iteration_values.append(global_best.best_score)
        best_values.append(global_best.best_position)

    # Graficar la convergencia
    plt.figure(figsize=(8, 6))
    plt.plot(range(1, n_iterations + 1), iteration_values, marker='o', linestyle='-', color='b')
    plt.xlabel('Iteración')
    plt.ylabel('Valor de la función objetivo')
    plt.title('Convergencia del algoritmo PSO')
    plt.grid(True)
    plt.show()

    print("Mejor posición encontrada:", global_best.best_position)
    print("Valor mínimo de la función:", global_best.best_score)

pso(n_particles=30, n_iterations=150, dim=2)
```



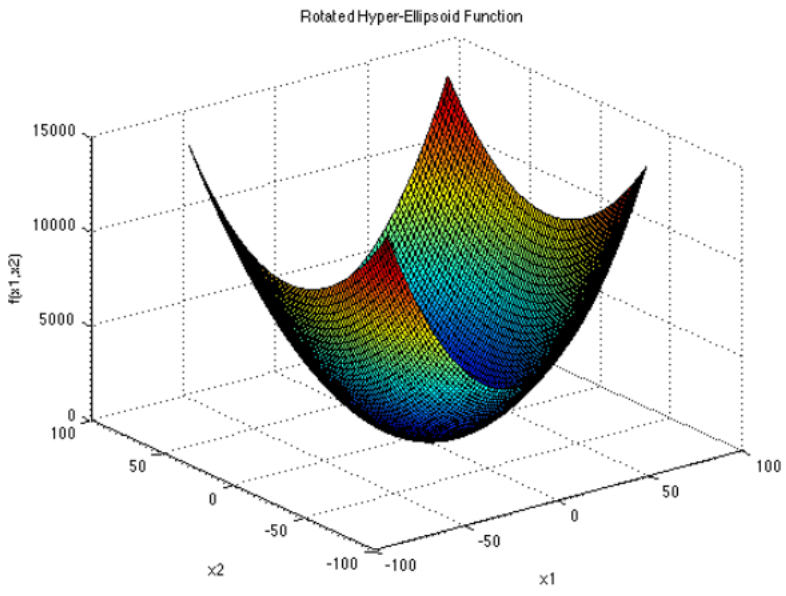

```
Mejor posición encontrada: [ 1.76581426e-09 -1.84728551e-10]
Valor mínimo de la función: 0.0
```

Mínimo global de la función



Otra funcion que parece la misma pero no lo es.

ROTATED HYPER-ELLIPSOID FUNCTION



$$f(\mathbf{x}) = \sum_{i=1}^{d} \sum_{j=1}^{i} x_j^2$$

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Función a minimizar
def f(x):
    return sum(sum(x[j]**2 for j in range(len(x))) for i in range(len(x)))

# Algoritmo PSO
class Particle:
    def __init__(self, dim):
        self.position = np.random.rand(dim)
        self.velocity = np.random.rand(dim)
        self.best_position = np.copy(self.position)
        self.best_score = f(self.position)

def pso(n_particles, n_iterations, dim):
    particles = [Particle(dim) for _ in range(n_particles)]
```

```
    global_best = min(particles, key=lambda x: x.best_score)

    # Listas para almacenar los valores de la función objetivo en cada iteración
    iteration_values = []
    best_values = []

    for _ in range(n_iterations):
        for particle in particles:
           # Actualizar velocidad y posición
            w = 0.7   # Inercia
            c1 = 1.5  # Coeficiente cognitivo
            c2 = 1.5  # Coeficiente social

            r1, r2 = np.random.rand(dim), np.random.rand(dim)

            particle.velocity = (w * particle.velocity +
                                 c1 * r1 * (particle.best_position - particle.position) +
                                 c2 * r2 * (global_best.best_position - particle.position))

            particle.position += particle.velocity

            # Evaluar la nueva posición
            score = f(particle.position)

            if score < particle.best_score:
                particle.best_score = score
                particle.best_position = np.copy(particle.position)

                if score < global_best.best_score:
                    global_best = particle

        # Registrar los valores de la función objetivo
        iteration_values.append(global_best.best_score)
        best_values.append(global_best.best_position)

    # Graficar la convergencia
    plt.figure(figsize=(8, 6))
    plt.plot(range(1, n_iterations + 1), iteration_values, marker='o', linestyle='-', color='b')
    plt.xlabel('Iteración')
    plt.ylabel('Valor de la función objetivo')
    plt.title('Convergencia del algoritmo PSO')
    plt.grid(True)
    plt.show()

    print("Mejor posición encontrada:", global_best.best_position)
    print("Valor mínimo de la función:", global_best.best_score)

pso(n_particles=30, n_iterations=100, dim=2)
```