

# envolvente\_convexo

October 7, 2024

## 1 Envolverte Convexo

1.0.1 Trabajo realizado por: Jessica Naomi Millan Sánchez

1.0.2 Graficación Computacional

1.0.3 Profesora: Hazem Álvarez Rodríguez

1.0.4 Clase del 07 de octubre de 2024

```
[2]: import numpy as np
import random as rand
import matplotlib.pyplot as plt
```

Se implementa el algoritmo de Graham para identificar la envolvente convexa de un conjunto de puntos.

Utiliza el determinante de una matriz para determinar la orientación de 3 puntos consecutivos, y elimina el punto intermedio

```
[3]: def turn_right():
    array = [coord_points[0], coord_points[1]]
    for i in range(2, len(coord_points)):
        array.append(coord_points[i])
        while len(array) > 2 and np.linalg.
            det([array[-3], array[-2], array[-1]]) > 0:
            array.pop(-2)
    return array
```

Calcula la envolvente convexa de la parte inferior y superior, luego las combina para formar la envolvente completa

```
[4]: def convex_hull():
    coord_points.sort()
    l_upper = turn_right()
    coord_points.reverse()
    l_lower = turn_right()
    l = l_upper + l_lower
    return l
```

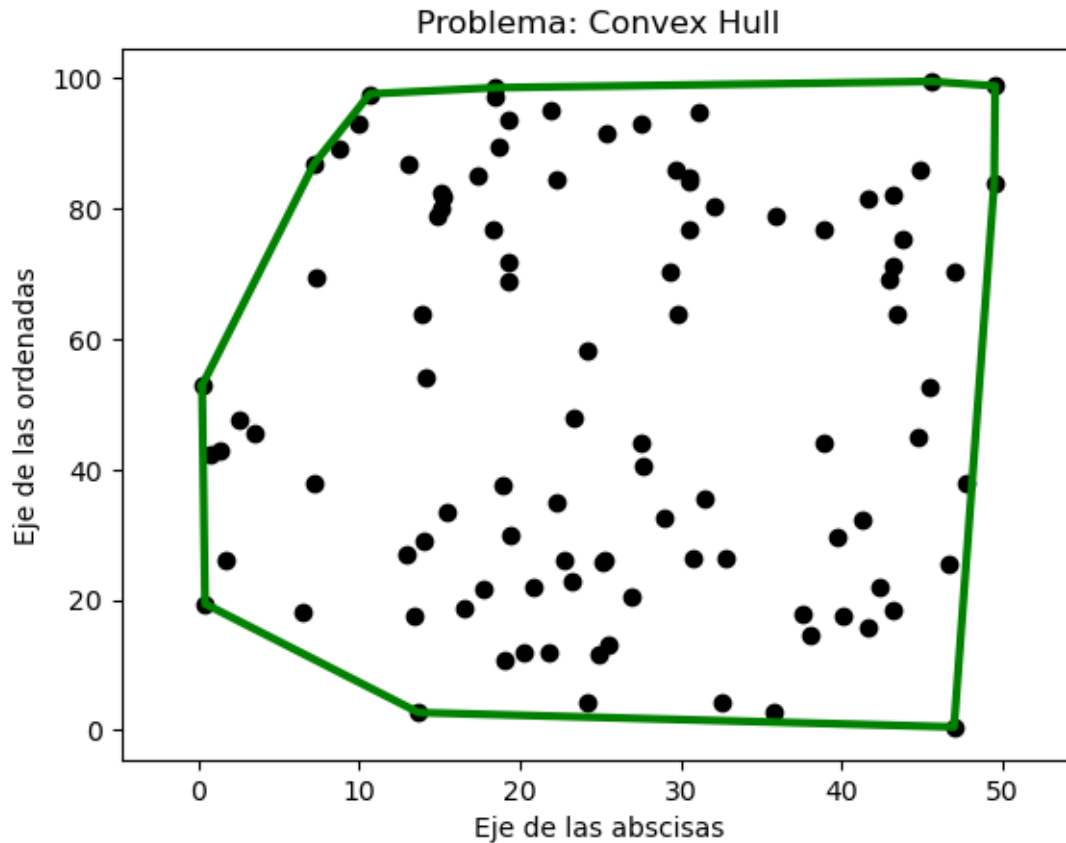
Función para asignar los límites del gráfico, también grafica los puntos y la envolvente convexa (línea verde).

```
[5]: def graph(convex_pol, coord_points):
    # Acomodando listas adecuadas para graficar en matplotlib
    x_points = [i[0] for i in coord_points]
    y_points = [i[1] for i in coord_points]
    x_polygon = [i[0] for i in convex_pol]
    y_polygon = [i[1] for i in convex_pol]
    # Definiendo límites extremos de la gráfica
    x_lim_der = max(x_points) + 5
    y_lim_sup = max(y_points) + 5
    x_lim_izq = min(x_points) - 5
    y_lim_inf = min(y_points) - 5
    # Asignación de los límites extremos
    plt.xlim(x_lim_izq, x_lim_der)
    plt.ylim(y_lim_inf, y_lim_sup)
    # Graficación
    plt.title('Problema: Convex Hull')
    plt.xlabel('Eje de las abscisas')
    plt.ylabel('Eje de las ordenadas')
    plt.plot(x_points, y_points, 'ko')
    plt.plot(x_polygon, y_polygon, 'g-', linewidth = 3.0)
    plt.show()
```

Genera los puntos aleatorios (100) y genera la envolvente convexa y los puntos que conforman el perímetro con los métodos definidos anteriormente.

```
[6]: num_points = 100
coord_points = []
for i in range(num_points): coord_points.append([rand.uniform(0,50), rand.
    ↪uniform(0,100), 1.0])

convex_pol = convex_hull()
graph(convex_pol, coord_points)
```



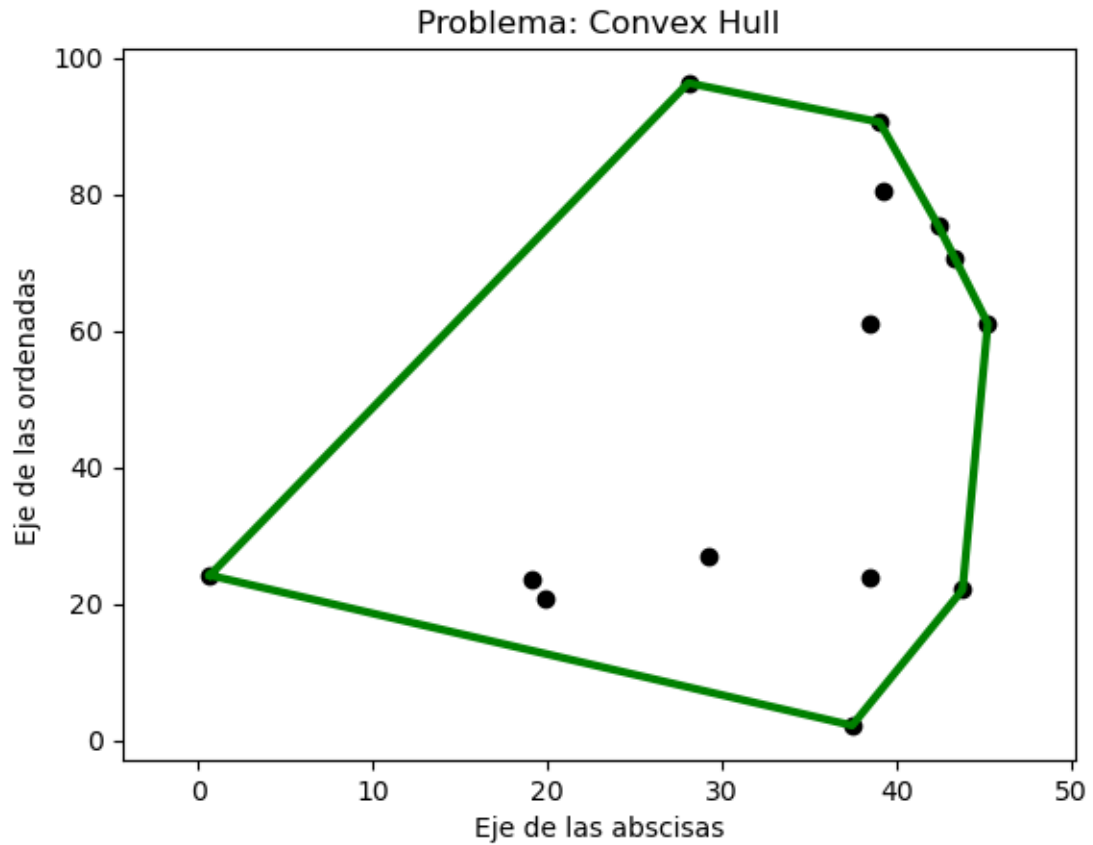
## 1.1 Actividad

1.1.1 Generar 2 figuras distintas junto con el envolvente convexo correspondiente

1.1.2 Figura 1:

```
[9]: num_points = 14
      coord_points = []
      for i in range(num_points): coord_points.append([rand.uniform(0,50), rand.
        ↪uniform(0,100), 1.0])

      convex_pol = convex_hull()
      graph(convex_pol, coord_points)
```



1.1.3 Figura 2:

```
[11]: num_points = 1000
      coord_points = []
      for i in range(num_points): coord_points.append([rand.uniform(0,50), rand.
        ↪uniform(0,100), 1.0])

      convex_pol = convex_hull()
      graph(convex_pol, coord_points)
```

