

envolvente_Convexo

October 7, 2024

1 UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

2 CENTRO UNIVERSITARIO UAEM ZUMPANGO

2.1 GRAFICACION COMPUTACIONAL

2.1.1 PRESENTO: JESUS ENRIQUE LUGO RAMIREZ

2.1.2 ENVOLVENTE CONVEXO

07 DE OCTUBRE DEL 2024 IMPORTO LIBRERIAS

```
[10]: import random as rand # Importamos la librería random para generar puntos
      ↪ aleatorios
import numpy as np # Importamos numpy para realizar operaciones matemáticas,
      ↪ como la determinante
import matplotlib.pyplot as plt # Importamos matplotlib para la graficación

[11]: # Función que determina el envolvente convexo en una dirección (superior o
      ↪ inferior)
def turn_right():
    array = [coord_points[0], coord_points[1]] # Inicializamos el array con
    ↪ los primeros dos puntos
    for i in range(2, len(coord_points)): # Iteramos desde el tercer punto en
    ↪ adelante
        array.append(coord_points[i]) # Añadimos el punto actual al array
        # Mientras haya al menos 3 puntos y el giro sea hacia la derecha,
    ↪ eliminamos el punto intermedio
        while len(array) > 2 and np.linalg.det([array[-3], array[-2],
    ↪ array[-1]]) > 0:
            array.pop(-2) # Si el giro es a la derecha (determinante > 0),
    ↪ removemos el punto medio
        return array # Devolvemos la lista de puntos del envolvente convexo en una
    ↪ dirección
```

CONSTRUCCION DEL ENVOLVENTE

```
[12]: # Función principal que construye el envolvente convexo
def convex_hull():
```

```

    coord_points.sort() # Ordenamos los puntos por coordenada x (y en caso de
    ↪ empate)
    l_upper = turn_right() # Calculamos la parte superior del envoltente
    ↪ convexo
    coord_points.reverse() # Invertimos la lista para calcular la parte
    ↪ inferior
    l_lower = turn_right() # Calculamos la parte inferior del envoltente
    ↪ convexo
    l = l_upper + l_lower # Unimos ambas partes
    return l # Devolvemos el envoltente convexo completo

```

METODO PRINCIPAL PARA GRAFICAR

```

[13]: # Función para graficar el envoltente convexo y los puntos
def graph(convex_pol, coord_points):
    # Acomodando listas de coordenadas para graficar en matplotlib
    x_points = [i[0] for i in coord_points] # Extraemos las coordenadas x de
    ↪ los puntos
    y_points = [i[1] for i in coord_points] # Extraemos las coordenadas y de
    ↪ los puntos
    x_polygon = [i[0] for i in convex_pol] # Extraemos las coordenadas x del
    ↪ polígono convexo
    y_polygon = [i[1] for i in convex_pol] # Extraemos las coordenadas y del
    ↪ polígono convexo
    # Definimos los límites de la gráfica con un pequeño margen
    x_lim_der = max(x_points) + 5 # Límite derecho (máximo x + 5)
    y_lim_sup = max(y_points) + 5 # Límite superior (máximo y + 5)
    x_lim_izq = min(x_points) - 5 # Límite izquierdo (mínimo x - 5)
    y_lim_inf = min(y_points) - 5 # Límite inferior (mínimo y - 5)
    # Asignamos los límites de la gráfica
    plt.xlim(x_lim_izq, x_lim_der)
    plt.ylim(y_lim_inf, y_lim_sup)
    # Graficamos los puntos y el envoltente convexo
    plt.title("Problema: Convex Hull") # Título de la gráfica
    plt.xlabel("Eje de las abcisas") # Etiqueta del eje x
    plt.ylabel("Eje de las ordenadas") # Etiqueta del eje y
    plt.plot(x_points, y_points, "ko") # Dibujamos los puntos en negro
    plt.plot(x_polygon, y_polygon, 'g-', linewidth=3.0) # Dibujamos el
    ↪ envoltente convexo en verde
    plt.show() # Mostramos la gráfica

```

CREO 100 PUNTOS ALEATORIOS ENTRE UN RANGO 0 A 50 PARA EL EJE x Y DE 0 A 100 PARA EL EJE Y. TAMBIEN REALIZO LA GRAFICA DEL POLIGONO CONVEXO

```

[14]: # Generamos 100 puntos aleatorios dentro de un rango
num_points = 100 # Número de puntos a generar
coord_points = [] # Lista que contendrá los puntos generados

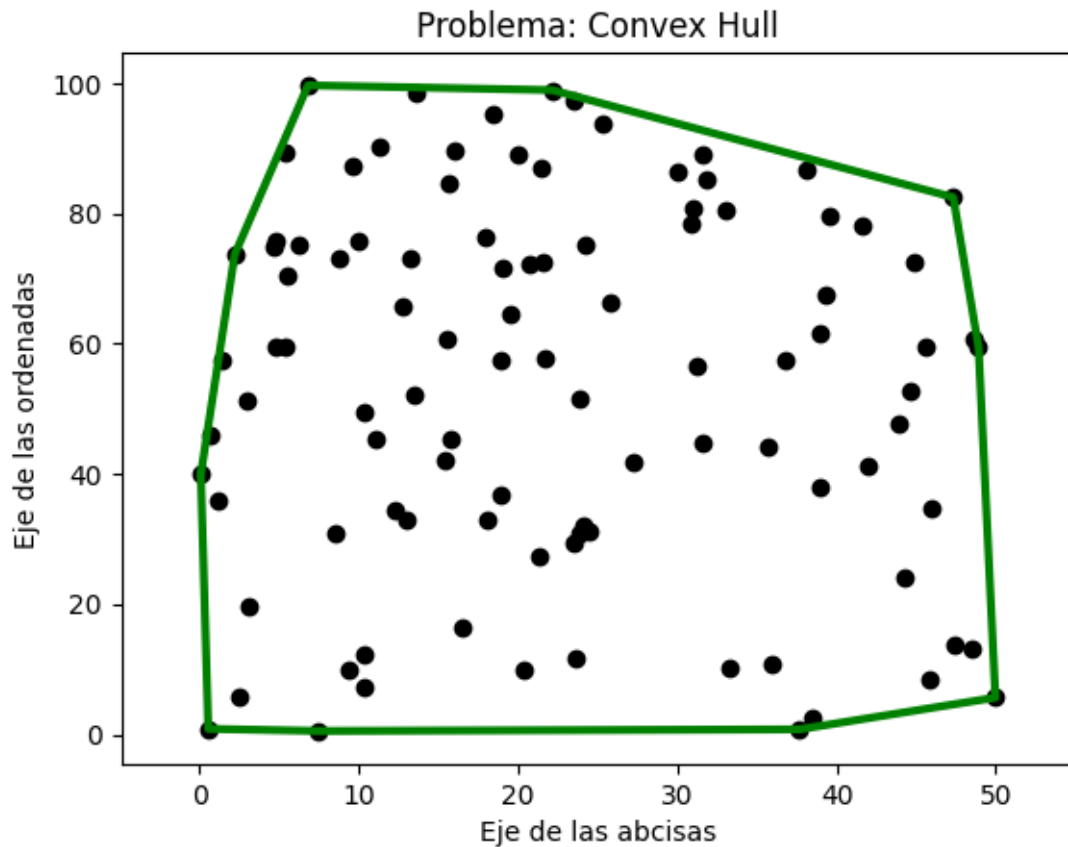
```

```

for i in range(num_points):
    # Añadimos puntos con coordenadas x entre 0 y 50, y entre 0 y 100 en y
    coord_points.append([rand.uniform(0, 50), rand.uniform(0, 100), 1.0])

# Creación y graficación del polígono convexo
convex_pol = convex_hull() # Calculamos el envolvente convexo
graph(convex_pol, coord_points) # Graficamos el envolvente convexo junto con
    ↪ los puntos

```



CREO 50,000 PUNTOS ALEATORIOS ENTRE UN RANGO 0 A 50 PARA EL EJE x Y DE 0 A 100 PARA EL EJE y . TAMBIEN REALIZO LA GRAFICA DEL POLIGONO CONVEXO. ESTO CON EL FIN DE ABARCAR CASI TODO EL PLANO PARA GRAFICAR UN CUADRADO

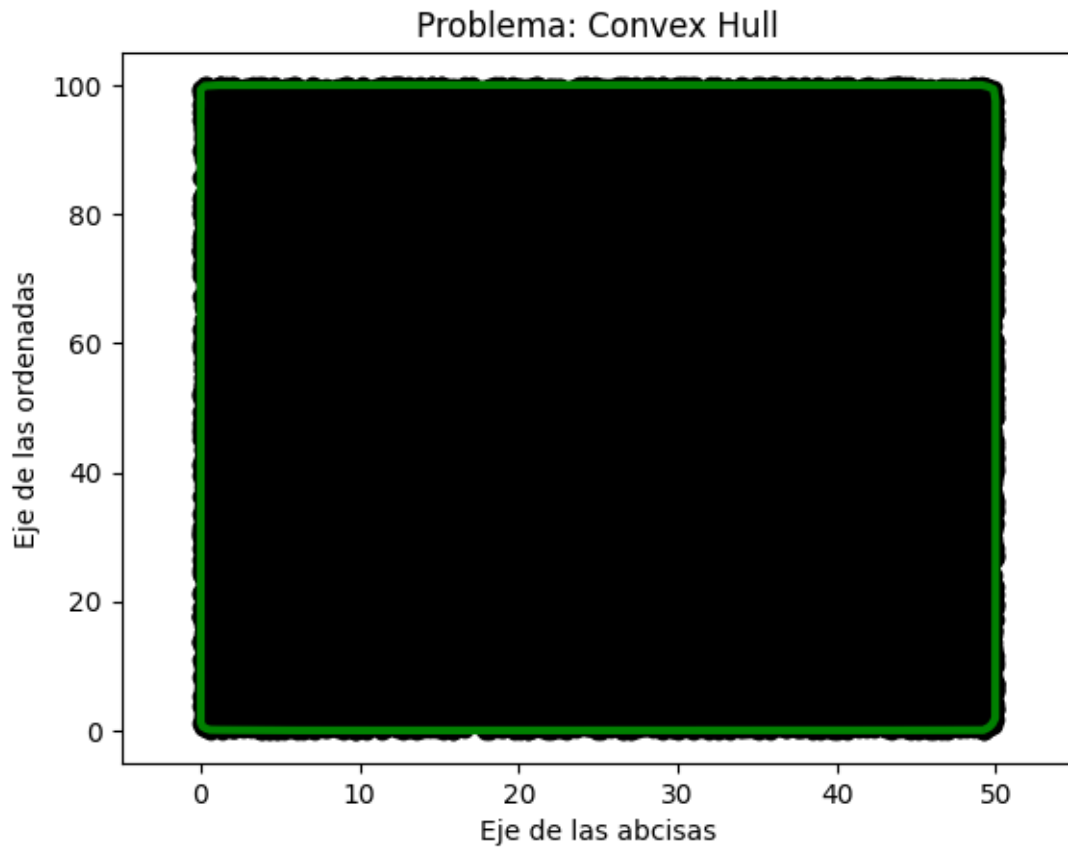
```

[15]: num_points = 50000
      coord_points = []
      for i in range(num_points):
          coord_points.append([rand.uniform(0,50), rand.uniform(0,100), 1.0])

      #creacion y graficacion del poligoino convexo
      convex_pol = convex_hull()

```

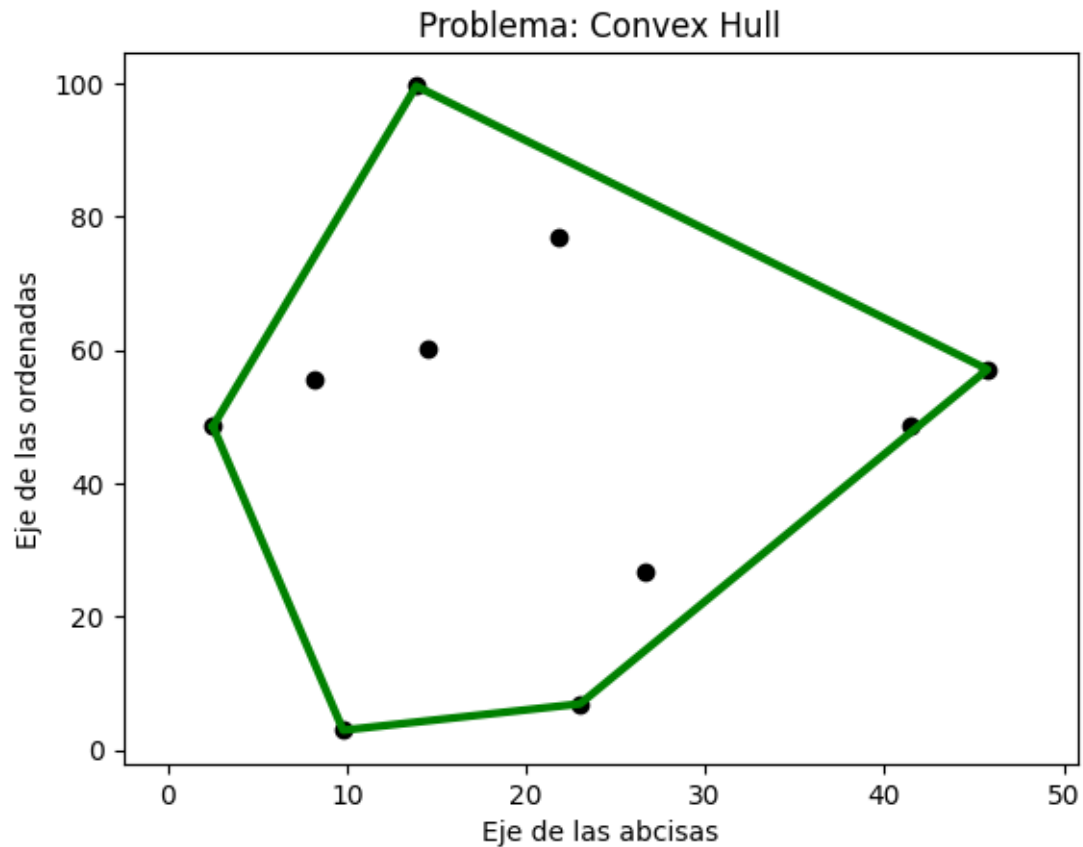
```
graph(convex_pol, coord_points)
```



CREO 10 PUNTOS ALEATORIOS ENTRE UN RANGO 0 A 50 PARA EL EJE x Y DE 0 A 100 PARA EL EJE y . TAMBIEN REALIZO LA GRAFICA DEL POLIGONO CONVEXO.

```
[16]: num_points = 10
      coord_points = []
      for i in range(num_points):
          coord_points.append([rand.uniform(0,50), rand.uniform(0,100), 1.0])

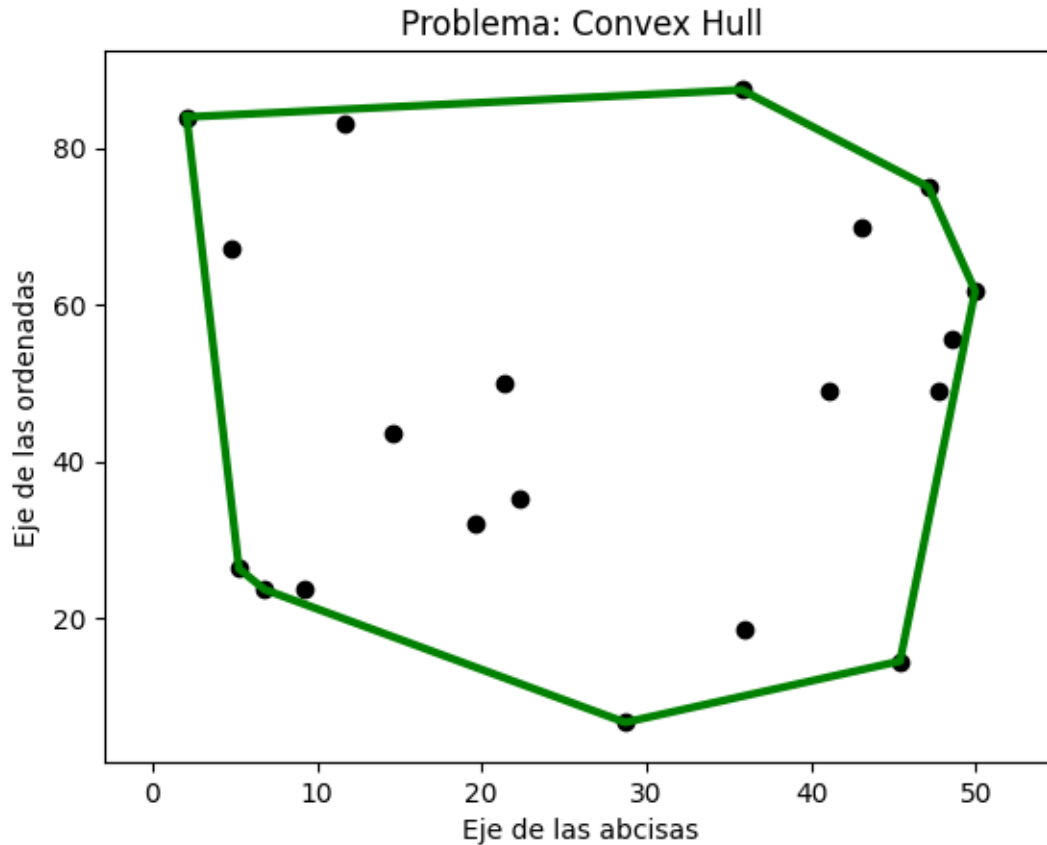
      #creacion y graficacion del poligoino convexo
      convex_pol = convex_hull()
      graph(convex_pol, coord_points)
```



CREO 20 PUNTOS ALEATORIOS ENTRE UN RANGO 0 A 50 PARA EL EJE x Y DE 0 A 100 PARA EL EJE Y. TAMBIEN REALIZO LA GRAFICA DEL POLIGONO CONVEXO.

```
[17]: num_points = 20
      coord_points = []
      for i in range(num_points):
          coord_points.append([rand.uniform(0,50), rand.uniform(0,100), 1.0])

      #creacion y graficacion del poligoino convexo
      convex_pol = convex_hull()
      graph(convex_pol, coord_points)
```



CREO 30 PUNTOS ALEATORIOS ENTRE UN RANGO 0 A 50 PARA EL EJE x Y DE 0 A 100 PARA EL EJE Y. TAMBIEN REALIZO LA GRAFICA DEL POLIGONO CONVEXO.

```
[18]: num_points = 30
      coord_points = []
      for i in range(num_points):
          coord_points.append([rand.uniform(0,50), rand.uniform(0,100), 1.0])

      #creacion y graficacion del poligoino convexo
      convex_pol = convex_hull()
      graph(convex_pol, coord_points)
```

