

Matplotlib

1. Giới thiệu (Introduction):

- Có một cơ sở mã sử dụng khá rộng thích hợp dùng để vẽ, mô phỏng các điểm dữ liệu trên mặt phẳng, không gian,...
- Trong một đồ thị chúng có đầy đủ các yếu tố từ điểm dữ liệu, thông số các trục, tên các trục, màu sắc để phân biệt các điểm.
- Các bạn có thể đọc tại: <https://matplotlib.org/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py>. Với thông tin đầy đủ hơn

2. Hướng dẫn Sử dụng Matplotlib:

- Đầu tiên ta phải import thư viện dưới tên viết tắt thông dụng thường dùng dưới tên plt : **import matplotlib.pyplot as plt.**

2.1.1 plt.scatter(X, Y, s=None, c=None, marker=None).

- Ở đây mình sẽ chỉ đưa ra những tham số thường dùng còn nhiều tham số truyền vào khác bạn có thể tham khảo.
- **X,Y**: là giá trị tương ứng từng với tọa độ trên trục.
- **s**: Kích thước của điểm dữ liệu trên đồ thị.
- **c**: Màu sắc là điểm dữ liệu trên đồ thị. Thường có các màu cơ bản green('g'), blue('b'), red('r'),...
- marker: Hình đại diện cho điểm dữ liệu. ví dụ marker='x' thì dữ liệu trên đồ thị sẽ được hiển thị dưới biểu tượng x. Bạn có thể tra thêm những biểu tượng hiển thị cho điểm dữ liệu tại đây:

<https://bitly.vn/41z8>

Ví dụ: Vẽ 2 điểm dữ liệu từ một mảng X và Y.

2.1.2 plt.show()

- Hiển thị đồ thị với những thông số đã setup ra màn hình.

2.1.3 plt.xlabel(xlabel) ↔ plt.ylabel(ylabel)

- Hai hàm này mục đích như nhau để thiết lập tên của các trục

2.1.4 plt.xlim(left,right) ↔ plt.ylim(left,right)

- Ta giới hạn hàm số từ giá trị left → right.

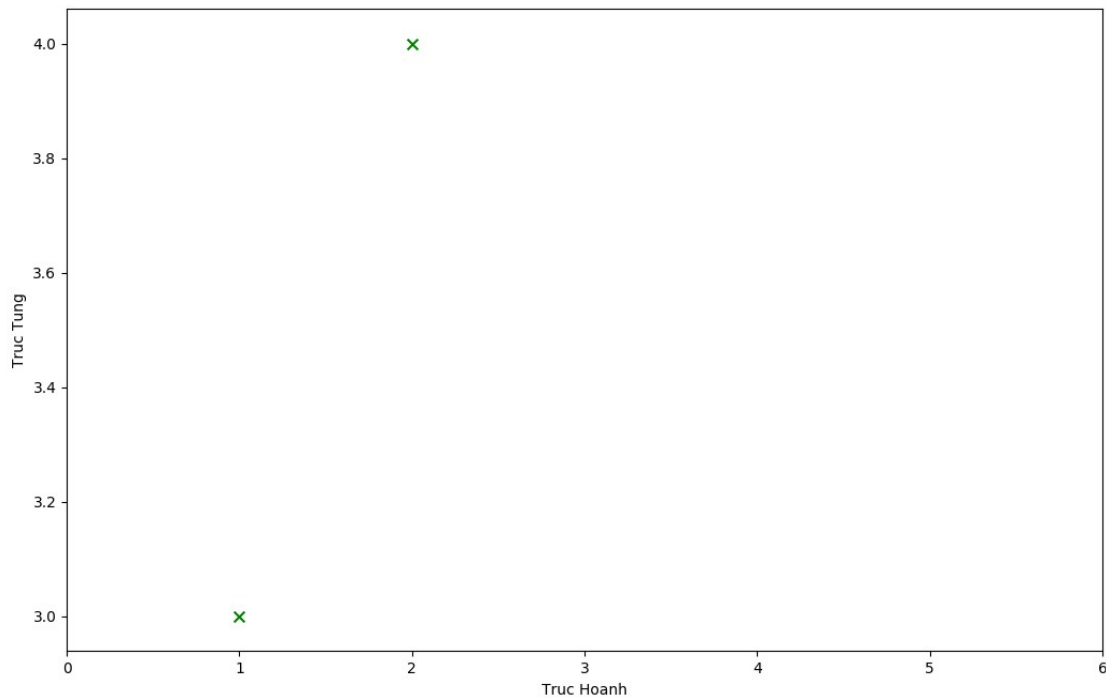
Ví dụ: Vẽ những điểm dữ liệu đơn giản trên đồ thị từ hai ma trận X và Y.

```
X = [1,2]
Y = [3,4]

plt.scatter(X,Y,s=50,c='g',marker='x')
plt.xlim(0,6)
plt.ylabel("Trục Tung")
plt.xlabel("Trục Hoành")
plt.show()
```

Cài đặt đồ thị ra màn hình hai điểm dữ liệu của X và Y

Figure 1



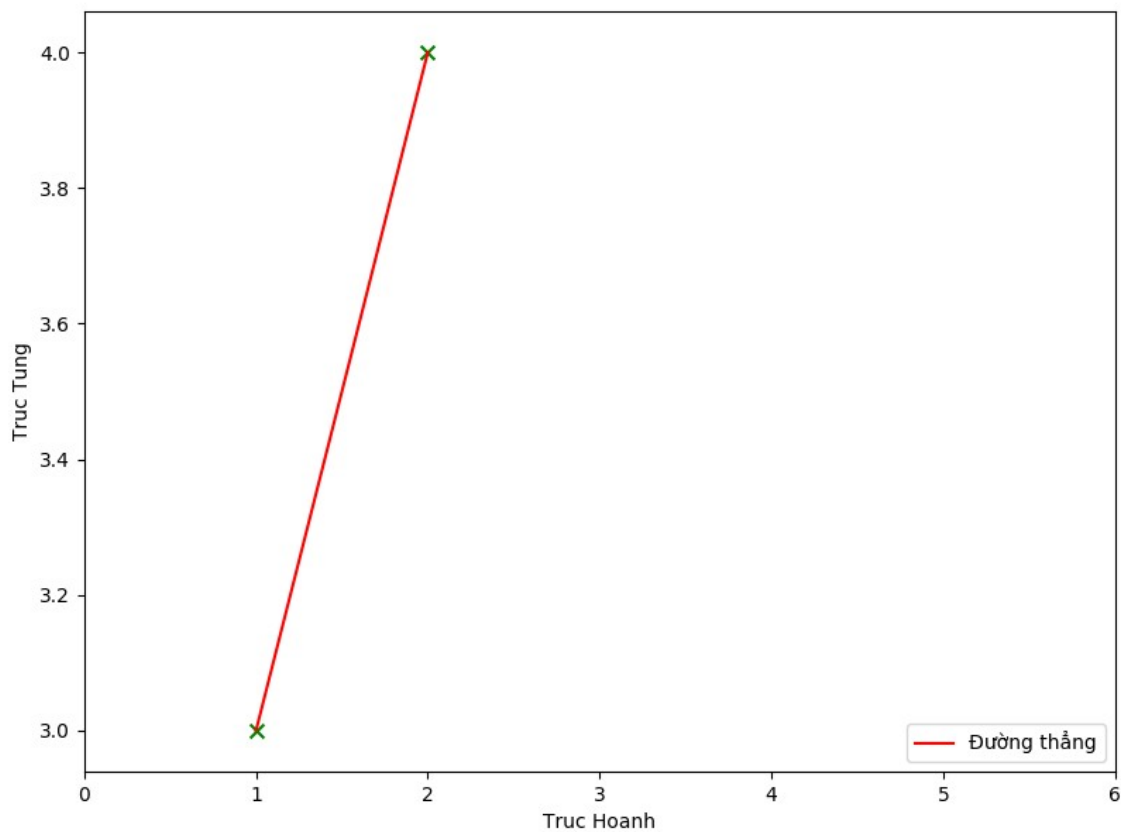
Sau khi chạy file main.py nhận được đồ thị biểu diễn hai điểm dữ liệu

2.1.5 `plt.plot(X, Y, c, label)`

- Vẽ đường thẳng đi qua các điểm X và Y.
- **X,Y** : hai m trận chứa điểm dữ liệu.
- **c** : màu của đường.
- **label** : Tên chú thích của đường.

Ví dụ: Ta xử dụng tiếp ví dụ vừa này và vẽ đường thẳng qua 2 điểm

```
X = [1, 2]
Y = [3, 4]
plt.scatter(X, Y, s=50, c='g', marker='x')
plt.plot(X, Y, c='r', label="Đường thẳng")
plt.xlim(0, 6)
plt.ylabel("Trục Tung")
plt.xlabel("Trục Hoành")
plt.legend(loc=4)
plt.show()
```

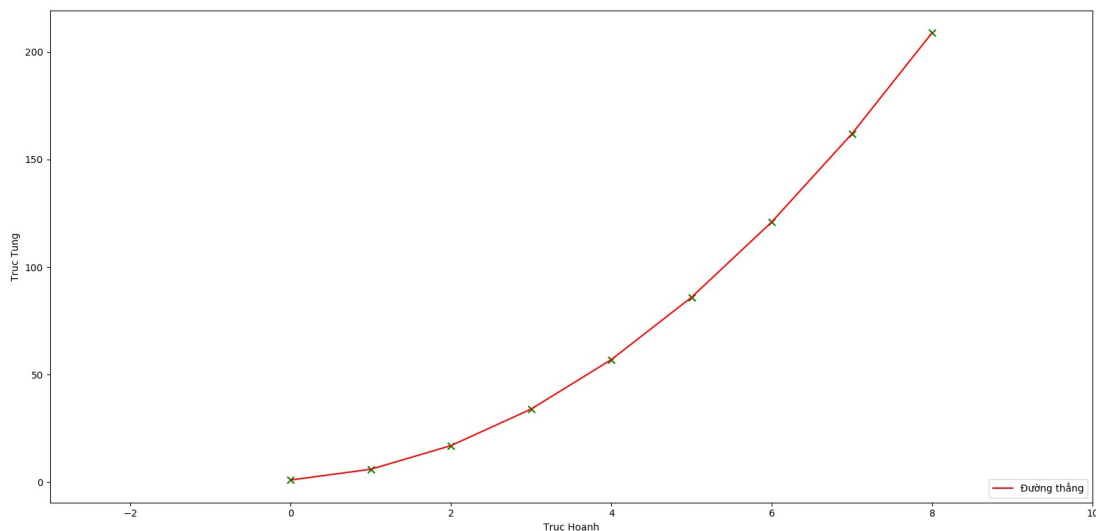


Ví dụ: Ta thử vẽ một parabol bằng plot

```

w = np.array([[1], [2], [3]], dtype=int)
X = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8], dtype=int)
Y = w[0]+w[1]*X+w[2]*X*X
plt.scatter(X,Y,s=50,c='g',marker='x')
plt.plot(X,Y, c='r', label="Đường thẳng")
plt.xlim(-3, 10)
plt.ylabel("Trục Tung")
plt.xlabel("Trục Hoành")
plt.legend(loc=4)
plt.show()

```



- Ở đây w sẽ đóng vai trò là a,b,c trong phương trình ax^2+bx+c .

Tiếp đến ta sẽ tìm hiểu cách biểu diễn dưới mặt phẳng 3D. Tuy nhiên tôi sẽ sử dụng chủ yếu vẽ các điểm trong mặt phẳng 3 chiều . Đầu tiên ta sẽ đi vào cách vẽ các điểm dữ liệu trên một mặt phẳng. Trước hết ta cần phải thêm module Axes3D .

-\$ from mpl_toolkits.mplot3d import Axes3D

2.2.1. Vẽ điểm dữ liệu trong không gian không khác nhiều so với vẽ trong mặt phẳng hai chiều tuy nhiên nó có một số thay đổi sau đây.

- Bạn phải tạo ra một figure bằng lệnh **plt.figure()** và gán nó bằng bất cứ biến nào. Ở đây tôi sẽ dùng là biến **fig**

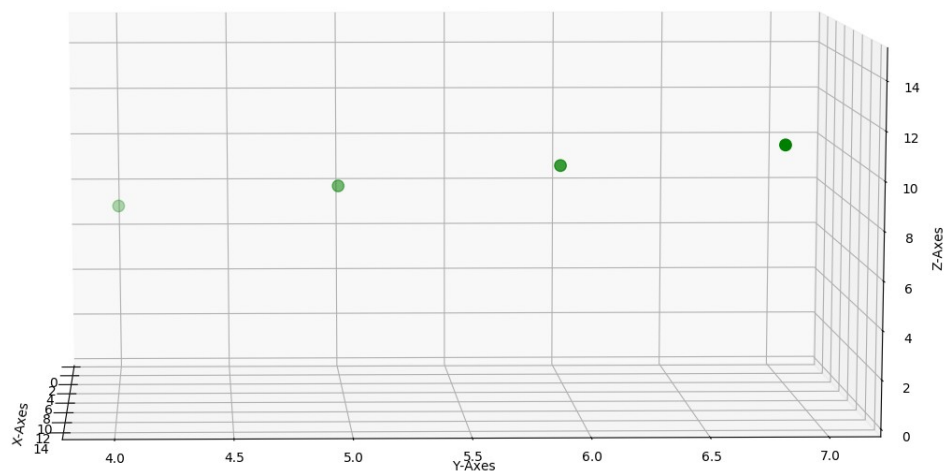
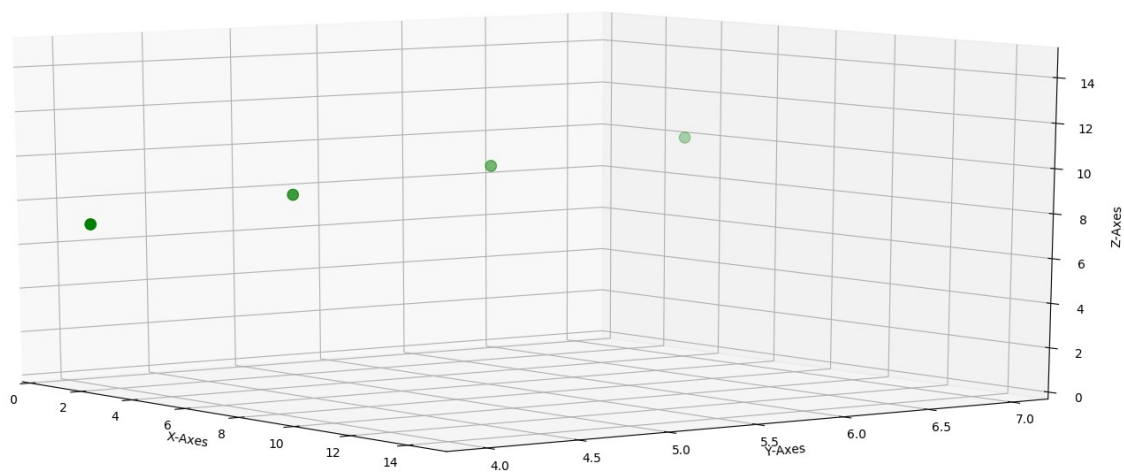
- **fig = plt.figure()**

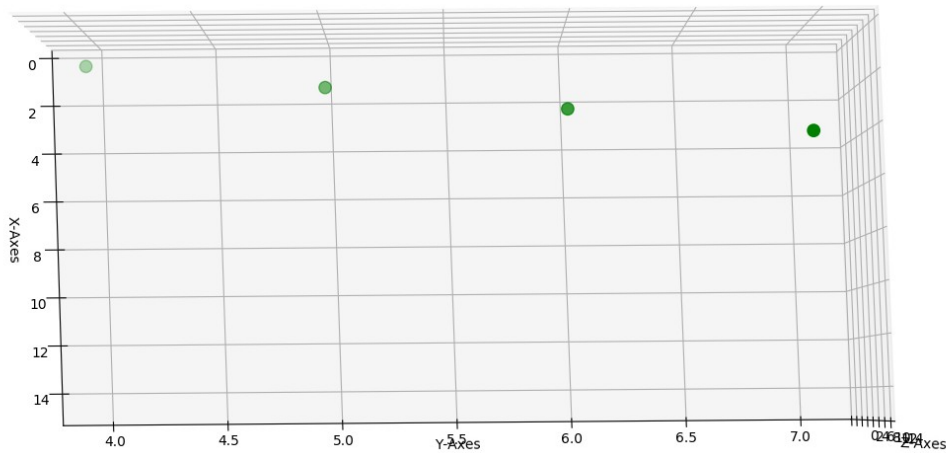
- Tiếp đến tạo ra 3 trục của đồ thị. Ở đây tôi sẽ dùng biến **ax** để tạo ra đồ thị.

- **ax = fig.add_subplot(111, projection='3d')**

- Còn lại các hàm khá giống trong mặt phẳng hai chiều, sau đây là một ví dụ đơn giản về vẽ các điểm dữ liệu trong không gian.

```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
x=[1, 2, 3, 4]
y=[4, 5, 6, 7]
z=[7, 8, 9, 10]
ax.scatter(x,y,z, zdir='z', marker='o', c='g', s=80)
ax.set_xlim(0, 15)
ax.set_ylim(0, 15)
ax.set_zlim(0, 15)
ax.set_xlabel("X-Axes")
ax.set_ylabel("Y-Axes")
ax.set_zlabel("Z-Axes")
plt.show()
```





Hướng nhìn từ ba phía của đồ thị không gian và những điểm dữ liệu trong nó

Mình sẽ có bản update thêm và sửa đổi về cách vẽ của các đồ thị mong thêm sự góp ý của các bạn

Bạn có thể đọc thêm các cách vẽ khác tại trang chủ của matplotlib từ vẽ 2D → 3D

3. Nguồn tham khảo:

- Trang của matplotlib:

https://matplotlib.org/api/_as_gen/matplotlib.pyplot.html#module-matplotlib.pyplot