

Course

Progress

Dates

Discussion

MO Index

Visualizing additional dimensions?

m_powers 4d

This video shows an example of visualizing an objective f based on two independent variables (x, y) .

My posts

All posts

Topics

Learners

Search all posts

Add a post

All posts sorted by recent activity

Welcome to CSE.0002x! Please introduce yourself!

He ...

darmofal Staff

21

4w

PSET 3: Assertion Errors in Contour Plot For PSET 3

contou ...

sandipan_dey

5

5d

Visualizing additional dimensions? No preview available

m_powers

5

4d

Quadratic programming(?) for fun and profit

No preview available

m_powers

6

4d

Hi Hi i am interested in Electrical & Computer Engineering. I ...

j-jakubczyk1975

1d

step size times J-prime

No preview available

m_powers

4

5d

Stiffness issue?

No preview available

JavierM0401

3

4d

inconsistent plotting label why is it that in the loop, the (XX) is ...

pkchong79

3

4d

Exam 1 code

Hi, will solutions for the coding problems in Exam 1 ...

dyaeger

2

4d

Solutions to the problem sets

Are the solutions to the problem s ...

Vasiliki_Ts

2

3d

Load more posts

Showing 3 responses

sandipan_dey 2d

Could think of the following visualizations for an arbitrary function of 3 variables, may be useful.

3D scatterplot with color representing the 4th variable

```
from matplotlib import pyplot as plt
import numpy as np
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
w = np.linspace(-1,1,20)
x = np.linspace(-1,1,20)
y = np.linspace(-1,1,20)
W, X, Y = np.meshgrid(w, x, y)
Z = np.sin(W**2 + X**2) + np.sin(X**2 + Y**2)
ax.scatter3D(W, X, Y, c=Z, cmap='jet', alpha=1)
plt.show()
```

https://discussions.edx.org/course-v1:MITx+CSE.0002x+2T2023/posts/64ef93860b881604f9e3b1ef?_gl=1*mwdzll*_ga*MTMwNTUzNTM2OS4xNjkwNDA3NTky*_ga_D3KS4KMDT0*MTY5MzY3ODE5OC44MC4xLjE... 1/3

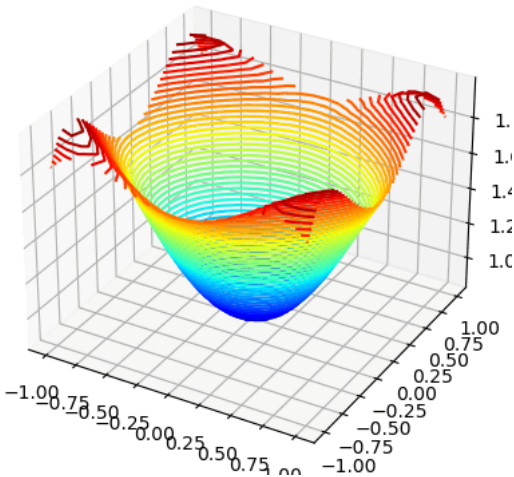
3D contourplot

Fix Y and contour plot

```
W, X = np.meshgrid(w, x)
fig = plt.figure(figsize=(20,20))
i = 0
for Y in np.linspace(-1,1,64):
    ax = fig.add_subplot(8, 8, i+1, projection='3d')
    Z = np.sin(W**2 + X**2) + np.sin(X**2 + Y**2)
    ax.contour3D(W, X, Z, 50, cmap='jet')
    ax.set_title(f'Y={Y:.3f}')
    i += 1
plt.show()
```



Y=-1.000

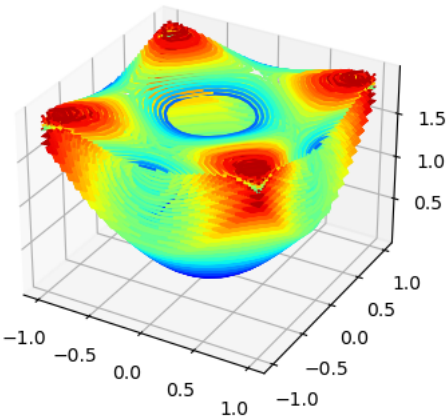


Contour plot for all Y values together

```
from matplotlib.colors import LinearSegmentedColormap

colormap = LinearSegmentedColormap.from_list('custom',
                                             [(0, '#00ff00'),
                                              (1, '#ff0000')], N=256)

W, X = np.meshgrid(w, x)
ax = plt.axes(projection='3d')
for Y in np.linspace(-1,1,64):
    Z = np.sin(W**2 + X**2) + np.sin(X**2 + Y**2)
    ax.contour3D(W, X, Z, 20, cmap='jet')
plt.show()
```



edX

- [About](#)
- [Affiliates](#)
- [edX for Business](#)
- [Open edX](#)
- [Careers](#)

[News](#)

Legal

- [Terms of Service & Honor Code](#)
- [Privacy Policy](#)
- [Accessibility Policy](#)
- [Trademark Policy](#)
- [Sitemap](#)
- [Cookie Policy](#)
- [Your Privacy Choices](#)

Connect

- [Idea Hub](#)
- [Contact Us](#)
- [Help Center](#)
- [Security](#)
- [Media Kit](#)



© 2023 edX LLC. All rights reserved.
深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)



m_powers 1h ⋮

Thanks again @sandipan_dey! The 3d meshgrid plot is what I have in mind currently but I am considering whether there is a way to omit all points below a certain threshold or make them "transparent". That way the hotspots will be more visible -- I don't really need to understand the contour of the entire space, just find the maxima.



m_powers 4d

Thanks for this! This helps me visualize a possible approach: 3D scatter plot, but only display values with z above some threshold. Should be able to see hotspots pretty well in that case.

In the example you suggested, it would be nice to be able to slide those cross-section planes around to quickly find coordinates of intersections.



wangaj_mit Staff 4d

I don't have a direct solution, but this came to mind. While you can't see all planes at once, this shows contour/density plots along the three principle orthogonal planes.

<https://en.wikipedia.org/wiki/File:Microwaveoventransient.webm>

You could also look up how they present CT scans in medicine, where they image a bunch of slices along an axis. If you step through / animate those slices for your function, you could get a sense of where your maxima lie.

Add a response