Python (/python) > 3D Charts (/python/3d-charts) > 3D Isosurface Plots

Suggest an edit to this page (https://github.com/plotly/plotly.py/edit/doc-prod/doc/python/3d-isosurface-plots.md)

3D Isosurface Plots in Python

How to make 3D Isosurface Plots in Python with Plotly.

surfaces

s

> Plotly Studio: Transform any dataset into an interactive data application in minutes with AI. Sign up for early access now. (https://plotly.com/studio/?utm_medium=graphing_libraries&utm_campaign=studio_early_access&utm_content=sidebar)

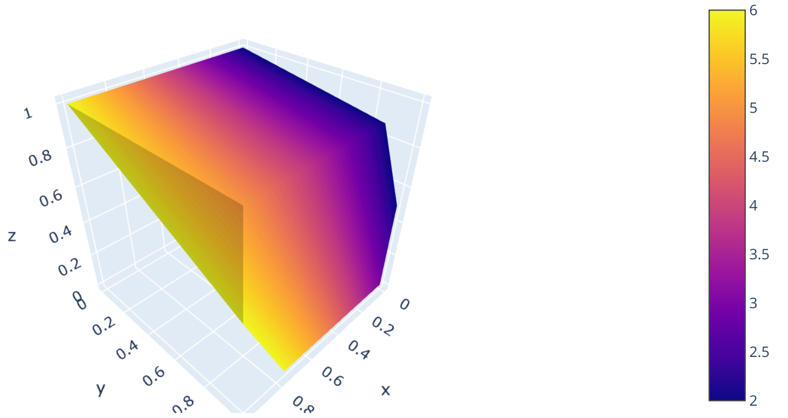
ance of

With go.Isosurface, you can plot isosurface contours (https://en.wikipedia.org/wiki/Isosurface) of a scalar field value, which is defined on x, y and z coordinates.

Basic Isosurface

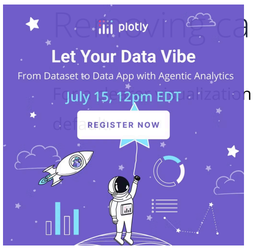
In this first example, we plot the isocontours of values isomin=2 and isomax=6. In addition, portions of the sides of the coordinate domains for which the value is between isomin and isomax (named the caps) are colored. Please rotate the figure to visualize both the internal surfaces and the caps surfaces on the sides.

```
import plotly.graph_objects as go

fig= go.Figure(data=go.Isosurface(
    x=[0,0,0,0,1,1,1,1],
    y=[1,0,1,0,1,0,1,0],
    z=[1,1,0,0,1,1,0,0],
    value=[1,2,3,4,5,6,7,8],
    isomin=2,
    isomax=6,
))

fig.show()
```



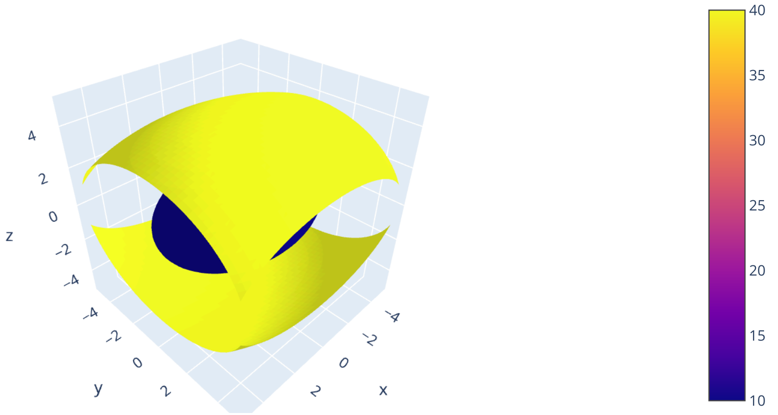
Removing caps when visualizing isosurfaces

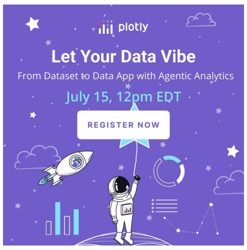
F[illegible] visualization of internal surfaces, it is possible to remove the caps (color-coded surfaces on the sides of the visualization domain). Caps are visible by de



surfaces
s
ance of

```
import plotly.graph_objects as go
import numpy as np

X, Y, Z = np.mgrid[-5:5:40j, -5:5:40j, -5:5:40j]

# ellipsoid
values = X * X * 0.5 + Y * Y + Z * Z * 2

fig = go.Figure(data=go.Isosurface(
    x=X.flatten(),
    y=Y.flatten(),
    z=Z.flatten(),
    value=values.flatten(),
    isomin=10,
    isomax=40,
    caps=dict(x_show=False, y_show=False)
    ))
fig.show()
```

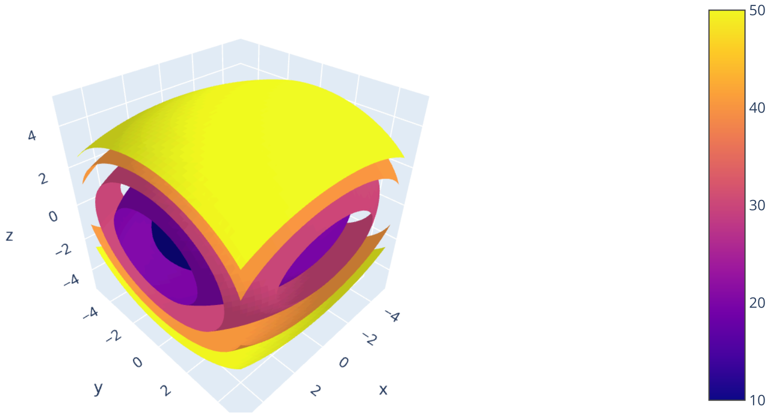


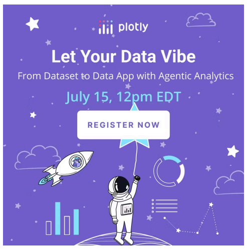


Modifying the number of isosurfaces

surfaces
s
ance of

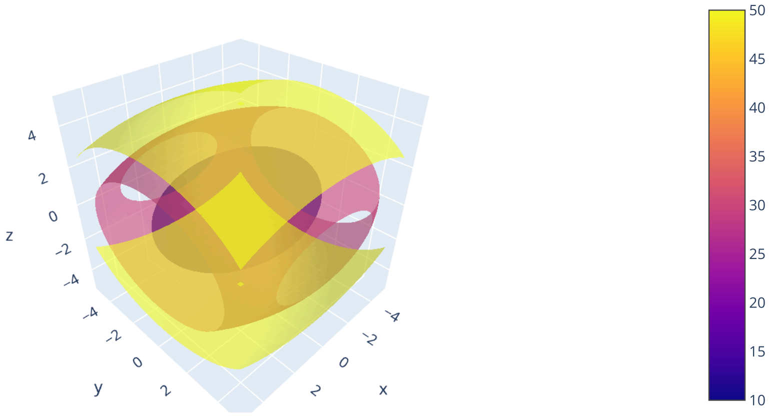
```
import plotly.graph_objects as go
import numpy as np

X, Y, Z = np.mgrid[-5:5:40j, -5:5:40j, -5:5:40j]

# ellipsoid
values = X * X * 0.5 + Y * Y + Z * Z * 2

fig = go.Figure(data=go.Isosurface(
    x=X.flatten(),
    y=Y.flatten(),
    z=Z.flatten(),
    value=values.flatten(),
    isomin=10,
    isomax=50,
    surface_count=5, # number of isosurfaces, 2 by default: only min and max
    colorbar_nticks=5, # colorbar ticks correspond to isosurface values
    caps=dict(x_show=False, y_show=False)
    ))
fig.show()
```





Changing the opacity of isosurfaces

```
import plotly.graph_objects as go
import numpy as np

X, Y, Z = np.mgrid[-5:5:40j, -5:5:40j, -5:5:40j]

# ellipsoid
values = X * X * 0.5 + Y * Y + Z * Z * 2

fig = go.Figure(data=go.Isosurface(
    x=X.flatten(),
    y=Y.flatten(),
    z=Z.flatten(),
    value=values.flatten(),
    opacity=0.6,
    isomin=10,
    isomax=50,
    surface_count=3,
    caps=dict(x_show=False, y_show=False)
    ))
fig.show()
```



Isosurface with Additional Slices

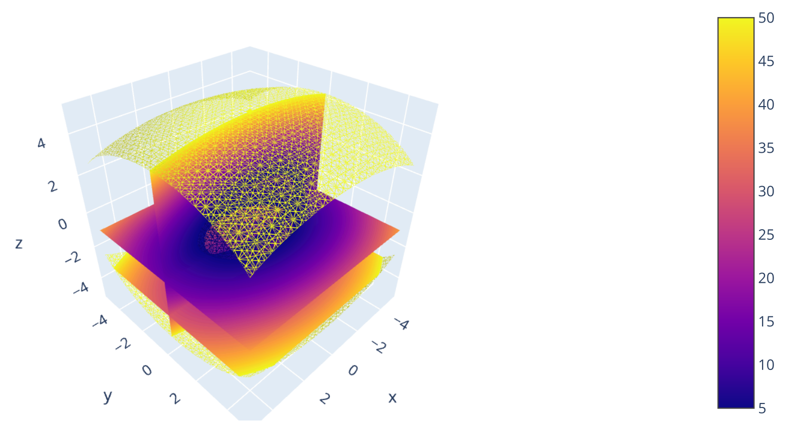
Here we visualize slices parallel to the axes on top of isosurfaces. For a clearer visualization, the fill ratio of isosurfaces is decreased below 1 (completely filled).

surfaces

s

ance of

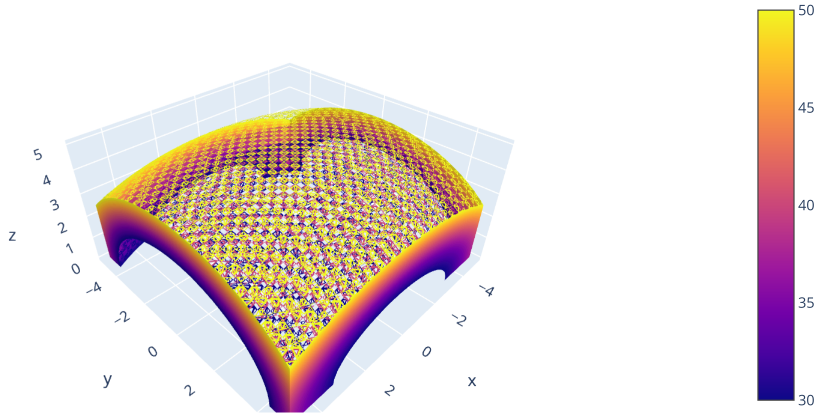
```
import plotly.graph_objects as go
import numpy as np

X, Y, Z = np.mgrid[-5:5:40j, -5:5:40j, -5:5:40j]

# ellipsoid
values = X * X * 0.5 + Y * Y + Z * Z * 2

fig = go.Figure(data=go.Isosurface(
    x=X.flatten(),
    y=Y.flatten(),
    z=Z.flatten(),
    value=values.flatten(),
    isomin=5,
    isomax=50,
    surface_fill=0.4,
    caps=dict(x_show=False, y_show=False),
    slices_z=dict(show=True, locations=[-1, -3,]),
    slices_y=dict(show=True, locations=[0]),
    ))
fig.show()
```



Multiple Isosurfaces with Caps

surfaces
s
ance of

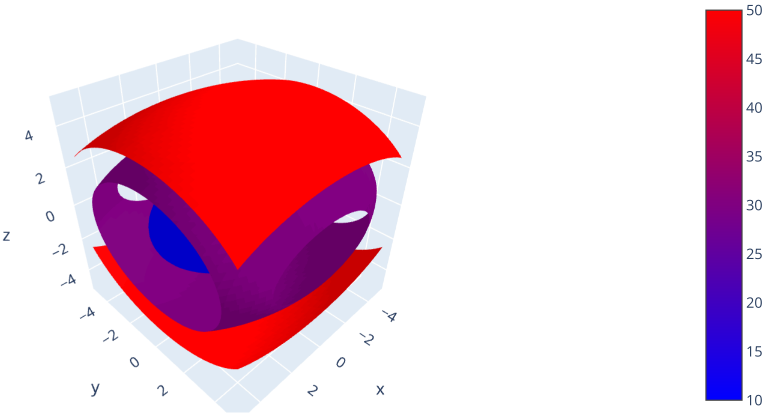
```
import plotly.graph_objects as go
import numpy as np

X, Y, Z = np.mgrid[-5:5:40j, -5:5:40j, 0:5:20j]

values = X * X * 0.5 + Y * Y + Z * Z * 2

fig = go.Figure(data=go.Isosurface(
    x=X.flatten(),
    y=Y.flatten(),
    z=Z.flatten(),
    value=values.flatten(),
    isomin=30,
    isomax=50,
    surface=dict(count=3, fill=0.7, pattern='odd'),
    caps=dict(x_show=True, y_show=True),
    ))
fig.show()
```



surfaces
s
ance of

Changing the default colorscale of isosurfaces

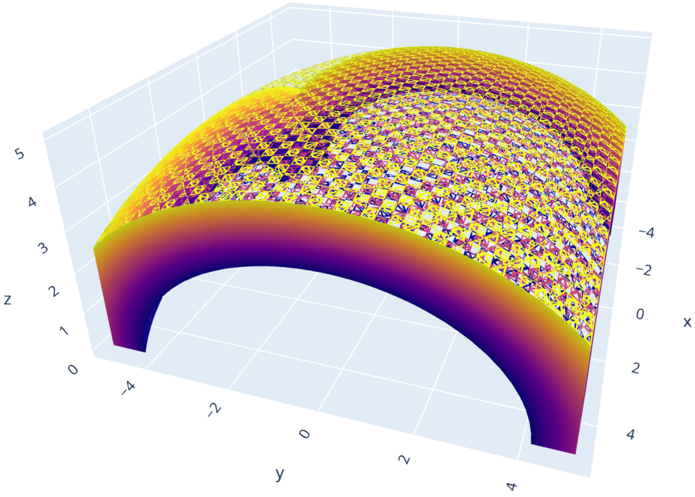
```
import plotly.graph_objects as go
import numpy as np

X, Y, Z = np.mgrid[-5:5:40j, -5:5:40j, -5:5:40j]

# ellipsoid
values = X * X * 0.5 + Y * Y + Z * Z * 2

fig = go.Figure(data=go.Isosurface(
    x=X.flatten(),
    y=Y.flatten(),
    z=Z.flatten(),
    value=values.flatten(),
    colorscale='BlueRed',
    isomin=10,
    isomax=50,
    surface_count=3,
    caps=dict(x_show=False, y_show=False)
    ))
fig.show()
```



Customizing the layout and appearance of isosurface plots

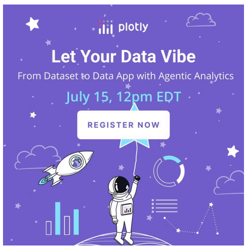
surfaces
s
ance of

```
import plotly.graph_objects as go
import numpy as np

X, Y, Z = np.mgrid[-5:5:40j, -5:5:40j, 0:5:20j]

values = X * X * 0.5 + Y * Y + Z * Z * 2

fig = go.Figure(data=go.Isosurface(
    x=X.flatten(),
    y=Y.flatten(),
    z=Z.flatten(),
    value=values.flatten(),
    isomin=30,
    isomax=50,
    surface=dict(count=3, fill=0.7, pattern='odd'),
    showscale=False, # remove colorbar
    caps=dict(x_show=True, y_show=True),
    ))

fig.update_layout(
    margin=dict(t=0, l=0, b=0), # tight layout
    scene_camera_eye=dict(x=1.86, y=0.61, z=0.98))
fig.show()
```



Reference

See https://plotly.com/python/reference/isosurface/ (https://plotly.com/python/reference/isosurface/) for more information and chart attribute options!



What About Dash?

Dash (https://dash.plot.ly/) is an open-source framework for building analytical applications, with no Javascript required, and it is tightly integrated with the Plotly graphing library.

Learn about how to install Dash at https://dash.plot.ly/installation (https://dash.plot.ly/installation).

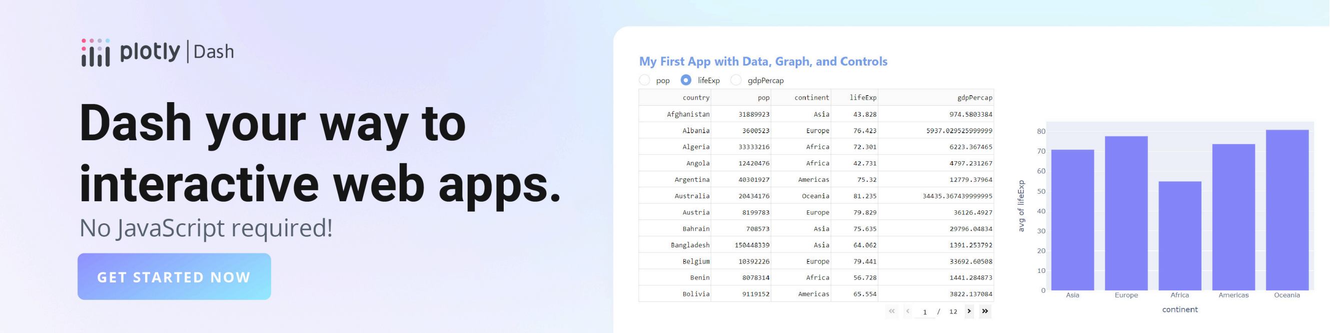
Everywhere in this page that you see fig.show(), you can display the same figure in a Dash application by passing it to the figure argument of the Graph component (https://dash.plot.ly/dash-core-components/graph) from the built-in dash_core_components package like this:

surfaces

s

```
import plotly.graph_objects as go # or plotly.express as px
fig = go.Figure() # or any Plotly Express function e.g. px.bar(...)
# fig.add_trace( ... )
# fig.update_layout( ... )

from dash import Dash, dcc, html

app = Dash()
app.layout = html.Div([
    dcc.Graph(figure=fig)
])

app.run(debug=True, use_reloader=False)  # Turn off reloader if inside Jupyter
```

ance of



My First App with Data, Graph, and Controls

○ pop ● lifeExp ○ gdpPercap

country	pop	continent	lifeExp	gdpPercap
Afghanistan	31889923	Asia	43.828	974.5803384
Albania	3600523	Europe	76.423	5937.029525999999
Algeria	33333216	Africa	72.301	6223.367465
Angola	12420476	Africa	42.731	4797.231267
Argentina	40301927	Americas	75.32	12779.37964
Australia	20434176	Oceania	81.235	34435.367439999995
Austria	8199783	Europe	79.829	36126.4927
Bahrain	708573	Asia	75.635	29796.04834
Bangladesh	150448339	Asia	64.062	1391.253792
Belgium	10392226	Europe	79.441	33692.60508
Benin	8078314	Africa	56.728	1441.284873
Bolivia	9119152	Americas	65.554	3822.137084

« ‹ 1 / 12 › »

(https://dash.plotly.com/tutorial?utm_medium=graphing_libraries&utm_content=python_footer)

JOIN OUR MAILING LIST

Sign up to stay in the loop with all things Plotly — from Dash Club to product updates, webinars, and more!

SUBSCRIBE (HTTPS://GO.PLOT.LY/SUBSCRIPTION)

Products

Dash (https://plotly.com/dash/)
Consulting and Training (https://plotly.com/consulting-and-oem/)

Pricing

Enterprise Pricing (https://plotly.com/get-pricing/)

About Us

Careers (https://plotly.com/careers)
Resources (https://plotly.com/resources/)
Blog (https://medium.com/@plotlygraphs)

Support

Community Support (https://community.plot.ly/)
Documentation (https://plotly.com/graphing-libraries)

Terms of Service (https://community.plotly.com/tos) Privacy Policy (https://plotly.com/privacy/)