:a·
plotly | Graphing Libraries (https://plotly.com/)(/graphing-libraries/)
:utm_campaign=studio_cloud_launch&utm_content=sidebar)

Carpet Contour Plot in Python

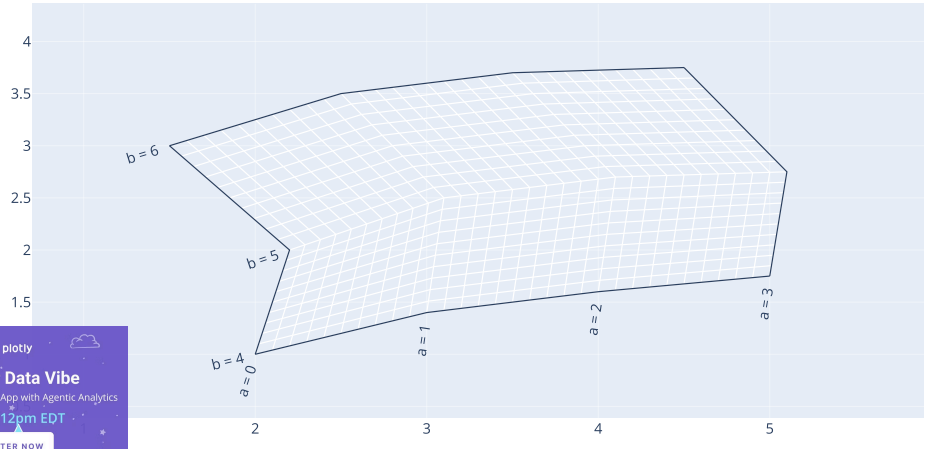
How to make carpet contour plots in Python with Plotly.

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)

Basic Carpet Plot

Set the `x` and `y` coordinates, using `x` and `y` attributes. If `x` coordinate values are omitted a cheater plot will be created. To save parameter values use `a` and `b` attributes. To make changes to the axes, use `aaxis` or `baxis` attributes. For a more detailed list of axes attributes refer to python reference (https://plotly.com/python/reference/carpet/#carpet-aaxis).

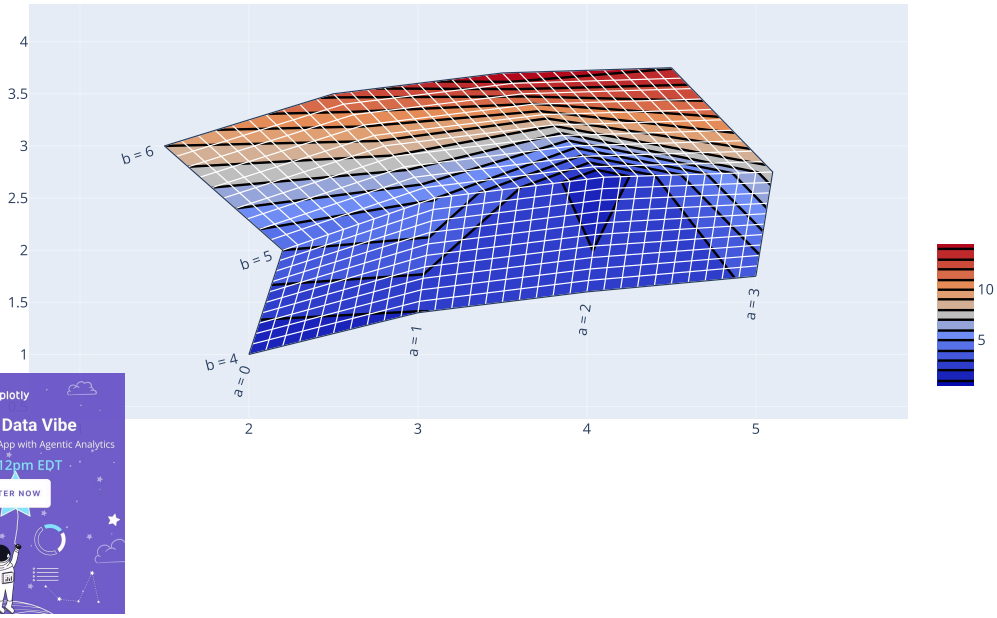
```
import plotly.graph_objects as go

fig = go.Figure(go.Carpet(
    a = [0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3],
    b = [4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6],
    x = [2, 3, 4, 5, 2.2, 3.1, 4.1, 5.1, 1.5, 2.5, 3.5, 4.5],
    y = [1, 1.4, 1.6, 1.75, 2, 2.5, 2.7, 2.75, 3, 3.5, 3.7, 3.75],
    aaxis = dict(
        tickprefix = 'a = ',
        smoothing = 0,
        minorgridcount = 9,
        type = 'linear'
    ),
    baxis = dict(
        tickprefix = 'b = ',
        smoothing = 0,
        minorgridcount = 9,
        type = 'linear'
    )
))

fig.show()
```

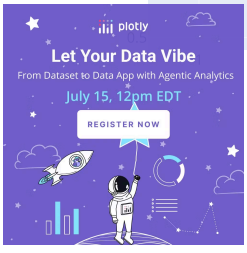


plotly
Let Your Data Vibe
From Dataset to Data App with Agentic Analytics
July 15, 12pm EDT
REGISTER NOW

Add Contours

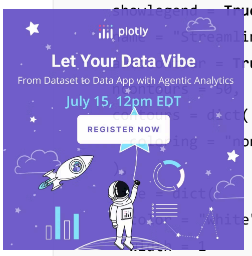
```
import plotly.graph_objects as go

fig = go.Figure()

fig.add_trace(go.Contourcarpet(
    a = [0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3],
    b = [4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6],
    z = [1, 1.96, 2.56, 3.0625, 4, 5.0625, 1, 7.5625, 9, 12.25, 15.21, 14.0625],
    autocontour = False,
    contours = dict(
        start = 1,
        end = 14,
        size = 1
    ),
    line = dict(
        width = 2,
        smoothing = 0
    ),
    colorbar = dict(
       len = 0.4,
        y = 0.25
    )
))

fig.add_trace(go.Carpet(
    a = [0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3],
    b = [4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6],
    x = [2, 3, 4, 5, 2.2, 3.1, 4.1, 5.1, 1.5, 2.5, 3.5, 4.5],
    y = [1, 1.4, 1.6, 1.75, 2, 2.5, 2.7, 2.75, 3, 3.5, 3.7, 3.75],
    aaxis = dict(
        tickprefix = 'a = ',
        smoothing = 0,
        minorgridcount = 9,
        type = 'linear'
    ),
    baxis = dict(
        tickprefix = 'b = ',
        smoothing = 0,
        minorgridcount = 9,
        type = 'linear'
    )
))

fig.show()
```



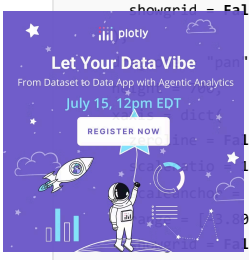


Add Multiple Traces

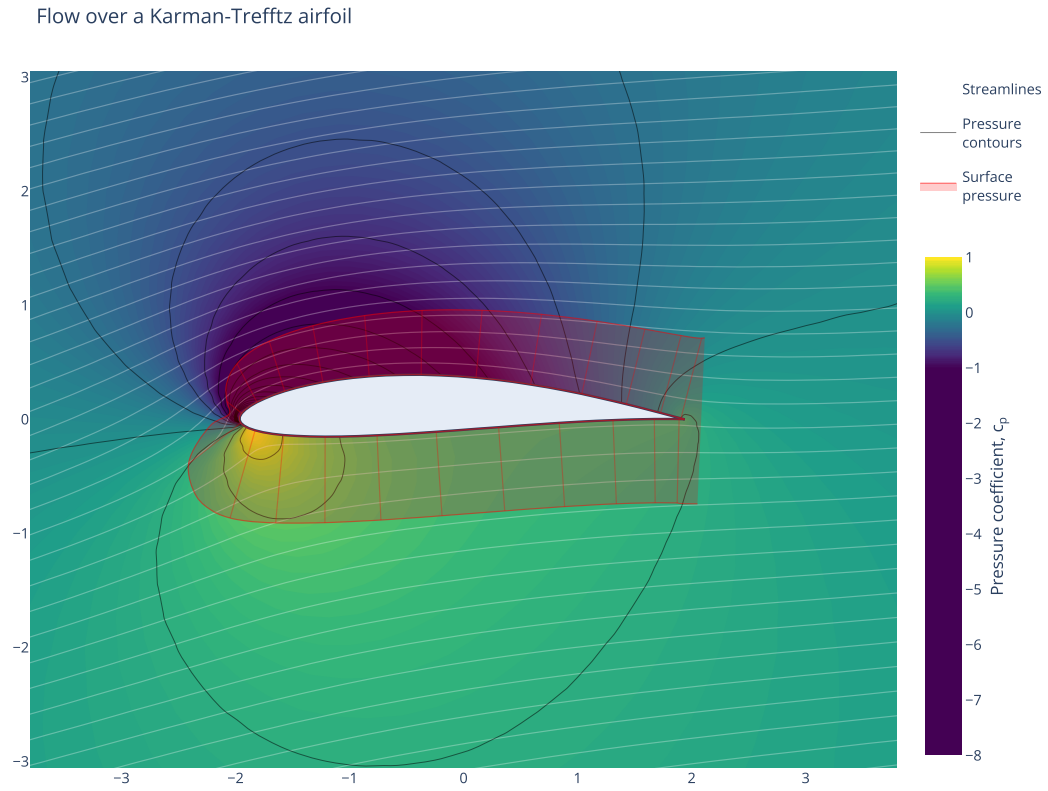
```
import plotly.graph_objects as go
import json
from urllib.request import urlopen

url = "https://raw.githubusercontent.com/bcdunbar/datasets/master/airfoil_data.json"
data = json.load(urlopen(url))


fig=go.Figure()

fig.add_trace(go.Carpet(
    a = data[0]['a'],
    b = data[0]['b'],
    x = data[0]['x'],
    y = data[0]['y'],
    baxis = dict(
      startline = False,
      endline = False,
      showticklabels = "none",
      smoothing = 0,
      showgrid = False
    ),
    aaxis = dict(
      startlinewidth = 2,
      startline = True,
      showticklabels = "none",
      endline = True,
      showgrid = False,
      endlinewidth = 2,
      smoothing = 0
    )
))

fig.add_trace(go.Contourcarpet(
    z = data[1]['z'],
    autocolorscale = False,
    zmax = 1,
    name = "Pressure",
    colorscale = "Viridis",
    zmin = -8,
    colorbar = dict(
      y = 0,
      yanchor = "bottom",
      len = 0.75,
      title = dict(
        text="Pressure coefficient, c<sub>p</sub>",
        side="right")
    ),
    contours = dict(
      start = -1,
      size = 0.025,
      end = 1.000,
      showlines = False
    ),
    line = dict(
      smoothing = 0
    ),
    autocontour = False,
    zauto = False
))

fig.add_trace(go.Contourcarpet(
    z = data[2]['z'],
    opacity = 0.300,
    showlegend = True,
    name = "Streamlines",
    autocontour = True,
    ncontours = 50,
    contours = dict(
      coloring = "none"
    ),
    line = dict(
      color = "white",
      width = 1
```



```
    )
))

fig.add_trace(go.Contourcarpet(
    z = data[3]['z'],
    showlegend = True,
    name = "Pressure<br>contours",
    autocontour = False,
    line = dict(
        color = "rgba(0, 0, 0, 0.5)",
        smoothing = 1
    ),
    contours = dict(
        size = 0.250,
        start = -4,
        coloring = "none",
        end = 1.000,
        showlines = True
      )
))

fig.add_trace(go.Scatter(
    x = data[4]['x'],
    y = data[4]['y'],
    legendgroup = "g1",
    name = "Surface<br>pressure",
    mode = "lines",
    hoverinfo = "skip",
    line = dict(
      color = "rgba(255, 0, 0, 0.5)",
      width = 1,
      shape = "spline",
      smoothing = 1
    ),
    fill = "toself",
    fillcolor = "rgba(255, 0, 0, 0.2)"
))

fig.add_trace(go.Scatter(
    x = data[5]['x'],
    y = data[5]['y'],
    showlegend = False,
    legendgroup = "g1",
    mode = "lines",
    hoverinfo = "skip",
    line = dict(
      color = "rgba(255, 0, 0, 0.3)",
      width = 1
    )
))

fig.add_trace(go.Scatter(
    x = data[6]['x'],
    y = data[6]['y'],
    showlegend = False,
    legendgroup = "g1",
    name = "cp",
    text = data[6]['text'],
    hoverinfo = "text",
    mode = "lines",
    line = dict(
      color = "rgba(255, 0, 0, 0.2)",
      width = 0
    )
))

fig.update_layout(
    yaxis = dict(
      zeroline = False,
      range = [-1.800,1.800],
      showgrid = False
```

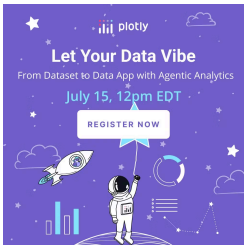


```
    ),
    title = "Flow over a Karman-Trefftz airfoil",
    hovermode = "closest",
    margin = dict(
      r = 60,
      b = 40,
      l = 40,
      t = 80
    ),
    width = 900
)

fig.show()
```



Reference

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



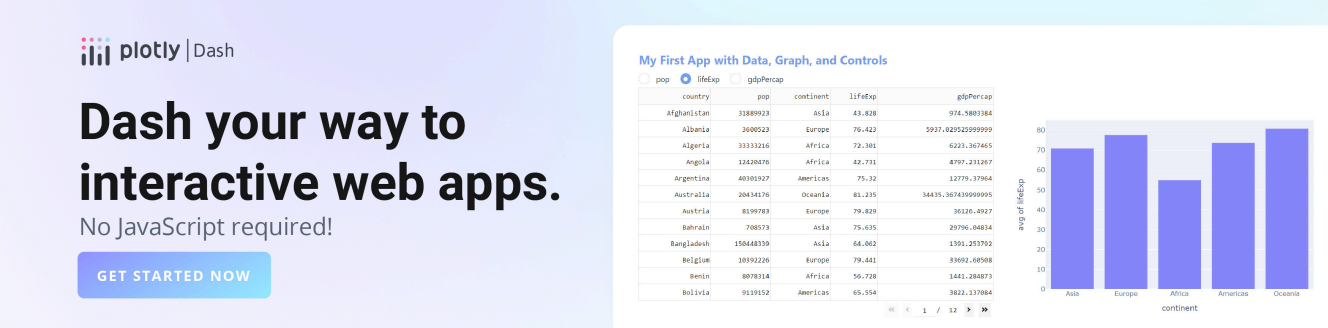
What About Dash?

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:

```
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
```



(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)

Copyright © 2025 Plotly. All rights reserved.

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