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

3D Streamtube Plots in Python

How to make 3D streamtube plots in Python with Plotly.

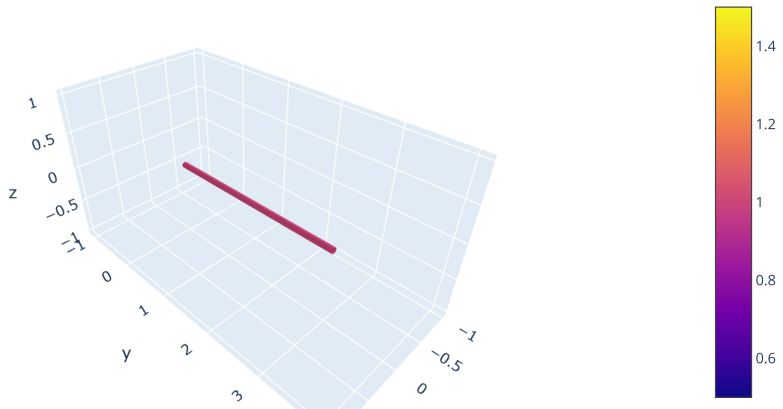
Plotly Studio: Transform any dataset into an interactive data application in minutes with AI. Sign up for early access now. (https://plotly.com/studio/?utm_medium=graphing_libraries&utm_campaign=studio_early_access&utm_content=sidebar)

Introduction

In streamtube plots, attributes include `x`, `y`, and `z`, which set the coordinates of the vector field, and `u`, `v`, and `w`, which set the x, y, and z components of the vector field. Additionally, you can use `starts` to determine the streamtube's starting position.

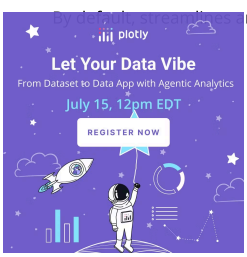
Basic Streamtube Plot

```
import plotly.graph_objects as go

fig = go.Figure(data=go.Streamtube(x=[0, 0, 0], y=[0, 1, 2], z=[0, 0, 0],
                                   u=[0, 0, 0], v=[1, 1, 1], w=[0, 0, 0]))
fig.show()
```

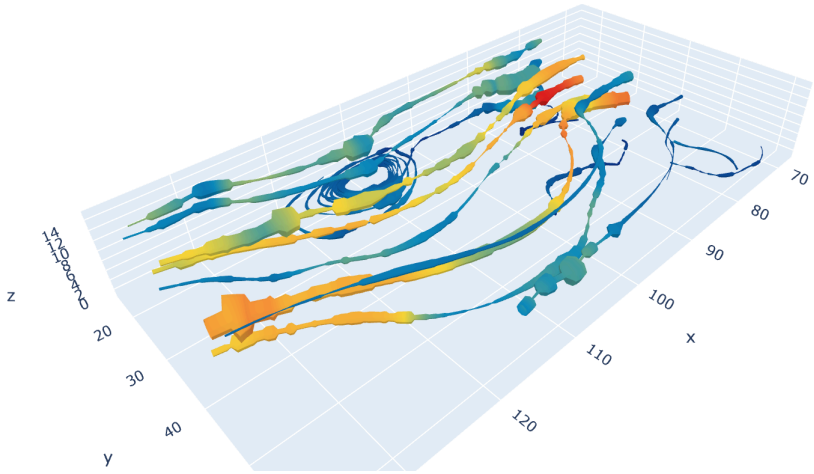


Starting Position and Segments

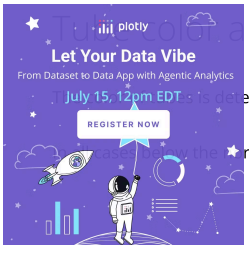
By default, streamtubes are initialized in the x-z plane of minimal y value. You can change this behaviour by providing directly the starting points of streamtubes.



```
import plotly.graph_objects as go

import pandas as pd

df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/streamtube-wind.csv').drop(['Unnamed: 0'],axis=1)

fig = go.Figure(data=go.Streamtube(
    x = df['x'],
    y = df['y'],
    z = df['z'],
    u = df['u'],
    v = df['v'],
    w = df['w'],
    starts = dict(
        x = [80] * 16,
        y = [20,30,40,50] * 4,
        z = [0,0,0,0,5,5,5,5,10,10,10,10,15,15,15,15]
    ),
    sizeref = 0.3,
    colorscale = 'Portland',
    showscale = False,
    maxdisplayed = 3000
))

fig.update_layout(
    scene = dict(
        aspectratio = dict(
            x = 2,
            y = 1,
            z = 0.3
        )
    ),
    margin = dict(
        t = 20,
        b = 20,
        l = 20,
        r = 20
    )
)

fig.show()
```



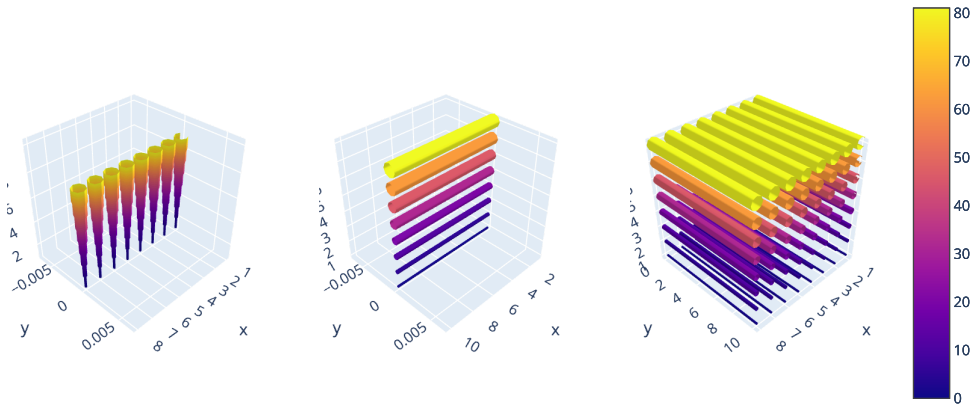
Tube color and diameter



ermined by their local norm, and the diameter of the field by the local divergence (https://en.wikipedia.org/wiki/Divergence) of the vector field.

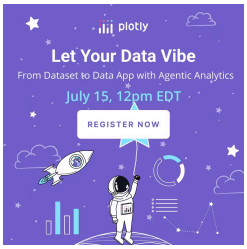
rm is proportional to `z**2` but the direction of the vector is different, resulting in a different divergence field.

```
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import numpy as np

x, y, z = np.mgrid[0:10, 0:10, 0:10]
x = x.flatten()
y = y.flatten()
z = z.flatten()

u = np.zeros_like(x)
v = np.zeros_like(y)
w = z**2

fig = make_subplots(rows=1, cols=3, specs=[[{'is_3d': True}, {'is_3d': True}, {'is_3d':True}]])

fig.add_trace(go.Streamtube(x=x, y=y, z=z, u=u, v=v, w=w), 1, 1)
fig.add_trace(go.Streamtube(x=x, y=y, z=z, u=w, v=v, w=u), 1, 2)
fig.add_trace(go.Streamtube(x=x, y=y, z=z, u=u, v=w, w=v), 1, 3)

fig.update_layout(scene_camera_eye=dict(x=2, y=2, z=2),
                  scene2_camera_eye=dict(x=2, y=2, z=2),
                  scene3_camera_eye=dict(x=2, y=2, z=2))
fig.show()
```



Reference

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



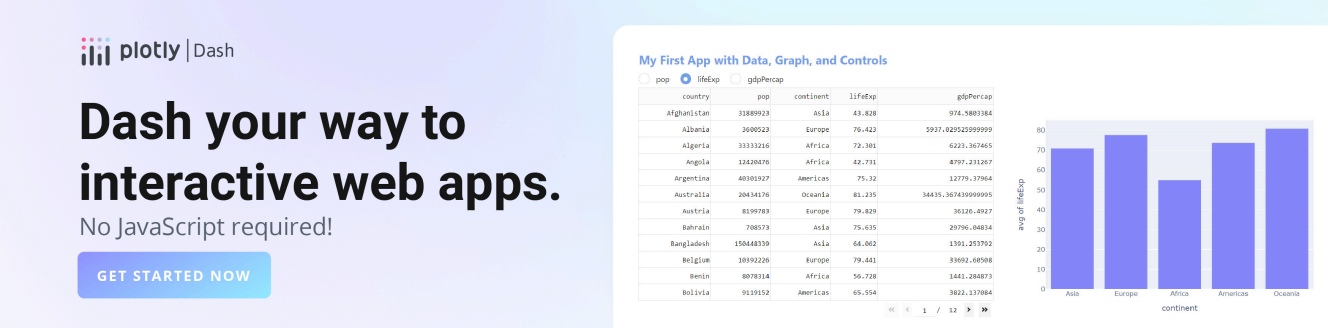
What About Dash?

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/)