

```

import os, sys, os.path, shutil
import matplotlib as mpl

import xlrd, openpyxl
import hashlib
import pweave

mode_serv = False
for param in sys.argv[1:]:
    if param[:len("serv")]=="serv" and param[len("serv")]=="=":
        mode_serv = param[len("serv")+1:] in "1 y Y yes Yes YES"
    #if not mode_serv:
    #    mpl.use('TkAgg')
    #else:
    #    mpl.use('Agg')
if mode_serv:
    mpl.use('Agg')
try:
    import appnope
    appnope.nope()    # stop the apple power nap
except ImportError:
    pass

import pandas
from math import pi
import matplotlib.pyplot as plt
import matplotlib.colors as colors
import matplotlib.patches as mpatches
from matplotlib.collections import PatchCollection
import numpy as np
from numpy.random import rand
from pylab import pcolor, show, colorbar, xticks, yticks, pcolormesh, imshow
from random import gauss
from scipy.interpolate import interp2d
import threading, multiprocessing, time
from decimal import Decimal

from scipy.optimize import curve_fit
mpl.rc('text', usetex=True)
#plt.rc('text', usetex=True)
mpl.rc('text.latex', preamble=r"\usepackage{amsmath} \usepackage{graphicx} \usepackage{ni

import scipy as sp
import subprocess
import datetime
from matplotlib.ticker import PercentFormatter
from mpl_toolkits.mplot3d import Axes3D

from mpl_toolkits.mplot3d.art3d import Poly3DCollection, Line3DCollection
import itertools

#vector of position P1 +1/-1 for right and left
v1=[0,0,0]
v2=[-1,0,0]
v3=[0,-1,0]
v4=[0,0,-1]

```

```

fig = plt.figure()
ax = fig.add_subplot(131, projection='3d')

# vertices of a pyramid
# v = np.array([[-1, -1, -1], [1, -1, -1], [1, 1, -1], [-1, 1, -1], [0, 0, 1]])
v = np.array([v1,v2,v3,v4])
ax.scatter3D(v[:, 0], v[:, 1], v[:, 2])

# generate list of sides' polygons of our pyramid
# verts = [ [v[0],v[1],v[2]], [v[0],v[2],v[3]], [v[1],v[2],v[3]]]
verts = list(itertools.combinations(v,3))

# plot sides
ax.add_collection3d(Poly3DCollection(verts,
    facecolors=['r','g','b','y'], linewidths=1, edgecolors='r', alpha=.25))
ax.set_xlim(-1,1)
ax.set_ylim(-1,1)
ax.set_zlim(-1,1)

#P2
#vector of position P1 +1/-1 for right and left
v1=[1,-1,0]
v2=[-1,0,1]
v3=[0,1,-1]
v4=[-1,-1,-1]

ax = fig.add_subplot(132, projection='3d')

# vertices of a pyramid
# v = np.array([[-1, -1, -1], [1, -1, -1], [1, 1, -1], [-1, 1, -1], [0, 0, 1]])
v = np.array([v1,v2,v3,v4])
ax.scatter3D(v[:, 0], v[:, 1], v[:, 2])

# generate list of sides' polygons of our pyramid
verts = list(itertools.combinations(v,3))

# plot sides
ax.add_collection3d(Poly3DCollection(verts,
    facecolors=['r','g','b','y'], linewidths=1, edgecolors='r', alpha=.25))
ax.set_xlim(-1,1)
ax.set_ylim(-1,1)
ax.set_zlim(-1,1)

#P2
#vector of position P1 +1/-1 for right and left
v1=[-1,1,1]
v2=[1,1,-1]
v3=[1,-1,1]
v4=[-1,-1,-1]

ax = fig.add_subplot(133, projection='3d')

```

```

# vertices of a pyramid
# v = np.array([[-1, -1, -1], [1, -1, -1], [1, 1, -1], [-1, 1, -1], [0, 0, 1]])
v = np.array([v1,v2,v3,v4])
ax.scatter3D(v[:, 0], v[:, 1], v[:, 2])

# generate list of sides' polygons of our pyramid (all combinaisons of three point
verts = list(itertools.combinations(v,3))

# plot sides
ax.add_collection3d(Poly3DCollection(verts,
    facecolors=['r','g','b','y'], linewidths=1, edgecolors='r', alpha=.25))
ax.set_xlim(-1,1)
ax.set_ylim(-1,1)
ax.set_zlim(-1,1)

plt.show()

```

