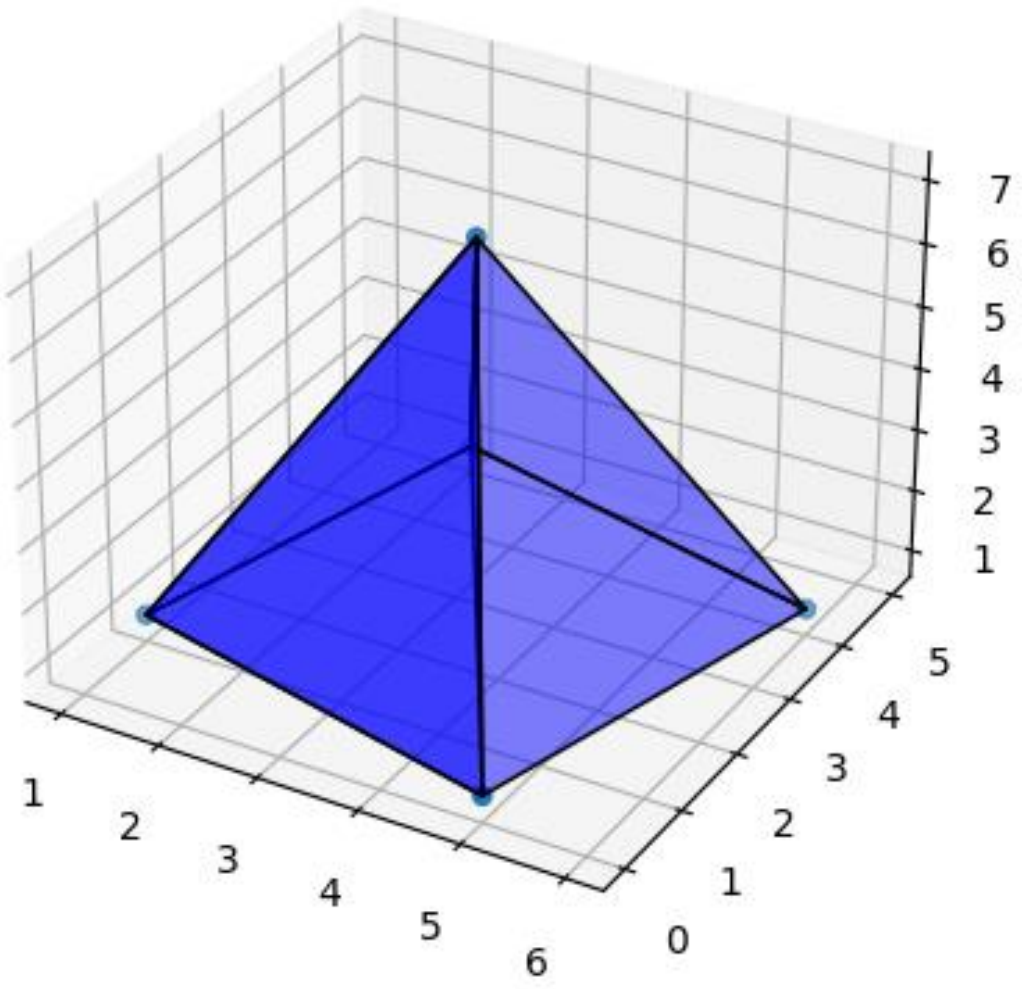
Exercício 5.1

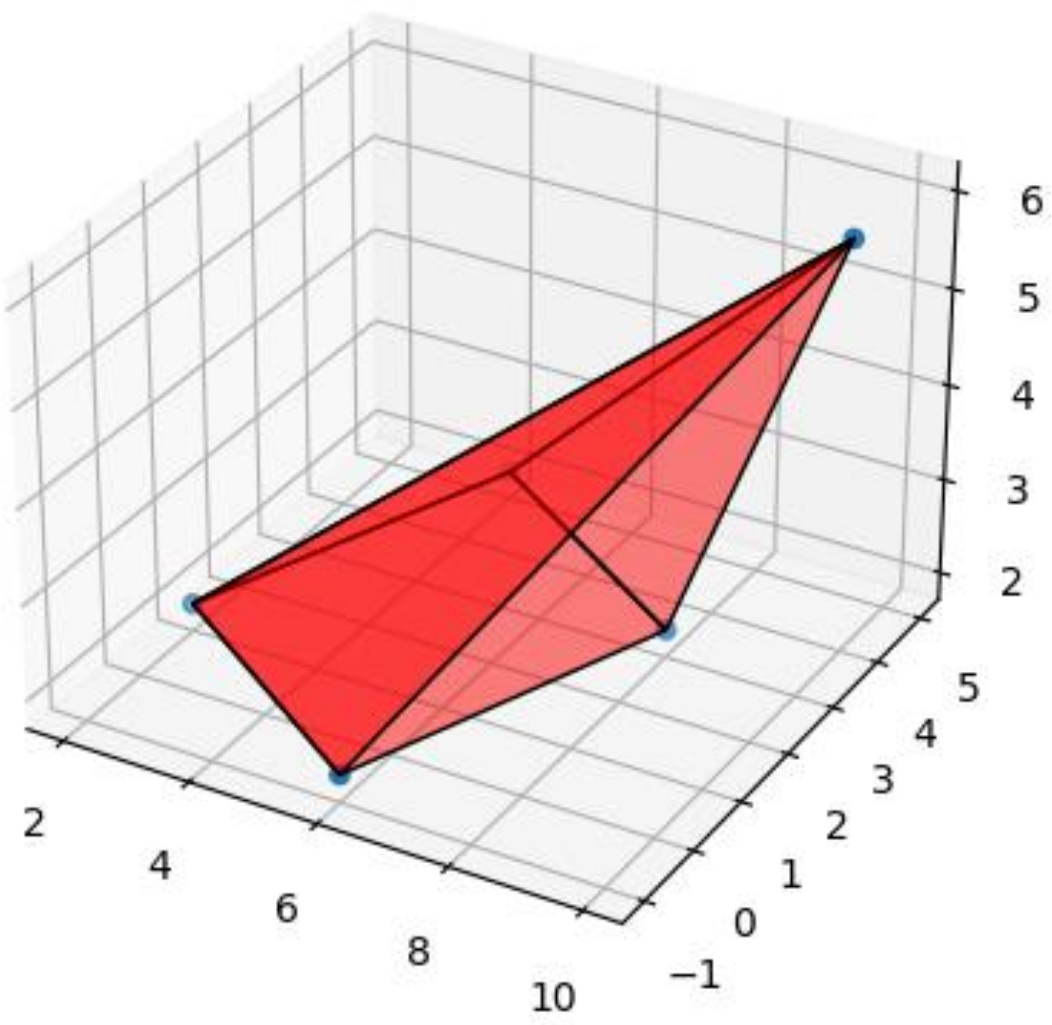
```
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d.art3d import Poly3DCollection
import numpy as np

fig = plt.figure()
ax = fig.add_subplot(projection = '3d')
v = np.array([[1, 1, 1], [2, 5, 1], [6, 4, 1], [5, 0, 1], [3.5, 2.5,
7]])
faces = [[v[0],v[1],v[2],v[3]], [v[0],v[1],v[4]], [v[0],v[3],v[4]],
[v[1],v[2],v[4]], [v[1],v[3],v[4]]]
ax.scatter3D(v[:, 0], v[:, 1], v[:, 2])
ax.add_collection3d(Poly3DCollection(faces, facecolors = 'blue',
edgecolors = 'black', alpha = 0.5))
plt.show()
```



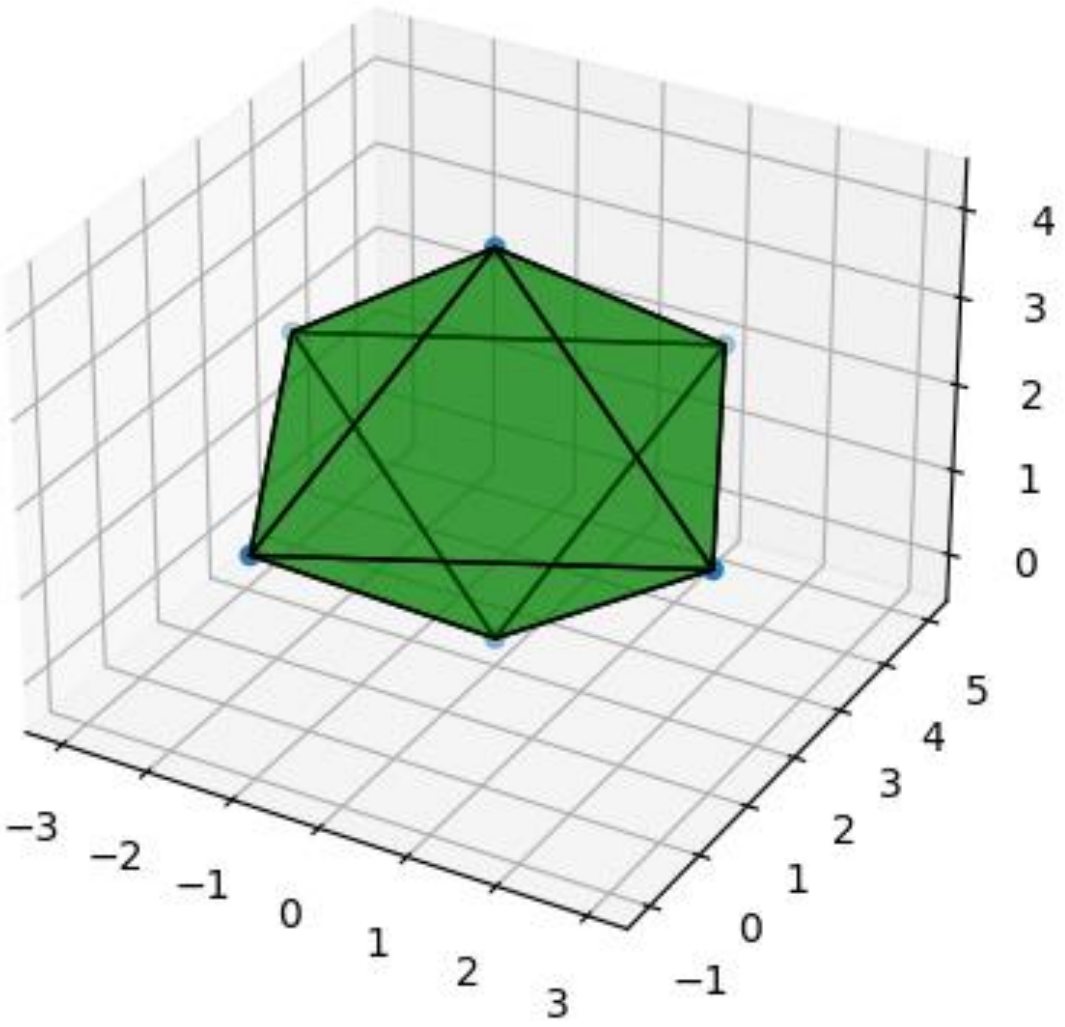
Exercício 5.2

```
fig = plt.figure()
ax = fig.add_subplot(projection = '3d')
v = np.array([[2, 1, 2], [4, 5, 2], [8, 3, 2], [6, -1, 2], [10, 4,
6]])
faces = [[v[0],v[1],v[2],v[3]], [v[0],v[1],v[4]], [v[0],v[3],v[4]],
[v[1],v[2],v[4]]]
ax.scatter3D(v[:, 0], v[:, 1], v[:, 2])
ax.add_collection3d(Poly3DCollection(faces, facecolors = 'red',
edgecolors = 'black', alpha = 0.5))
plt.show()
```



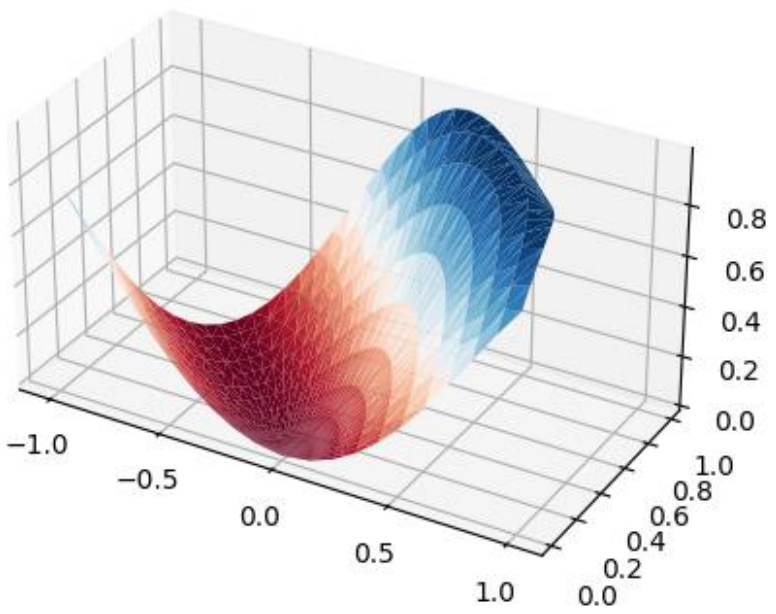
Exercício 5.3

```
fig = plt.figure()
ax = fig.add_subplot(projection = '3d')
v = np.array([[3, 1, 2], [1, 5, 2], [-3, 3, 2], [-1, -1, 2], [0, 2,
2+np.sqrt(5)], [0, 2, 2-np.sqrt(5)]])
faces = [[v[0],v[1],v[5]], [v[0],v[3],v[5]], [v[2],v[1],v[5]],
[v[2],v[3],v[5]],
        [v[0],v[1],v[4]], [v[0],v[3],v[4]], [v[2],v[1],v[4]],
[v[2],v[3],v[4]]]
ax.scatter3D(v[:, 0], v[:, 1], v[:, 2])
ax.add_collection3d(Poly3DCollection(faces, facecolors = 'green',
edgecolors = 'black', alpha = 0.5))
plt.show()
```

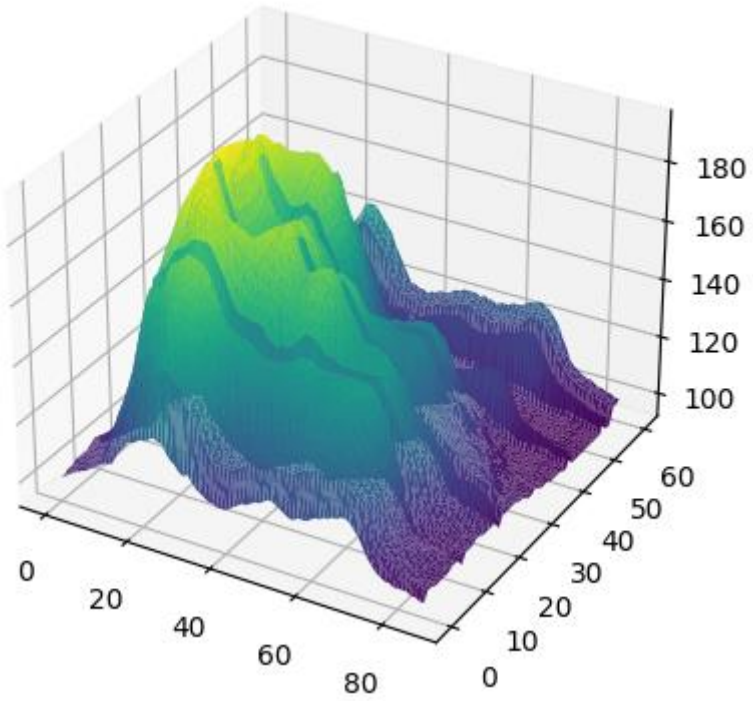


Exercício 5.4

```
n_raio = 10
n_angulos = 48
raio = np.linspace(0.125, 1.0, n_raio)
angulo = np.linspace(0, np.pi, n_angulos, endpoint = False)[...,
np.newaxis]
x = np.append(0, (raio*np.cos(angulo)).flatten())
y = np.append(0, (raio*np.sin(angulo)).flatten())
z = np.sin((x+y)**2)
ax = plt.figure().add_subplot(projection = '3d')
ax.plot_trisurf(x, y, z, linewidth = 0.2, cmap = 'RdBu')
ax.set_box_aspect((np.ptp(x), np.ptp(y), np.ptp(z)))
plt.show()
```

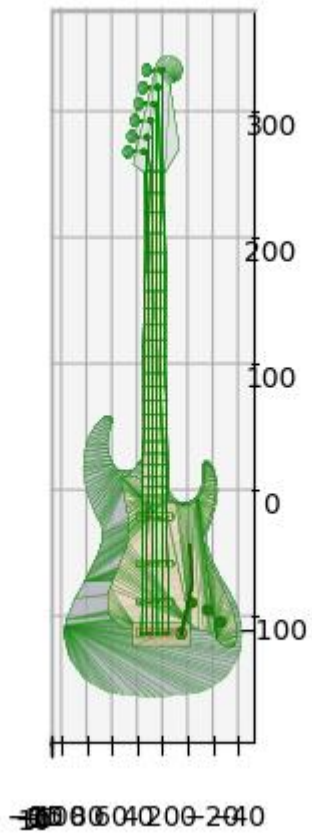



```
import pandas as pd
url = 'https://raw.githubusercontent.com/holtzy/The-Python-Graph-
Gallery/master/static/data/volcano.csv'
data = pd.read_csv(url)
df=data.unstack().reset_index()
df.columns=["X","Y","Z"]
df['X']=pd.Categorical(df['X'])
df['X']=df['X'].cat.codes
fig = plt.figure()
ax = fig.add_subplot(projection = '3d')
ax.plot_trisurf(df['Y'], df['X'], df['Z'], cmap=plt.cm.viridis,
linewidth=0.2)
plt.show()
```

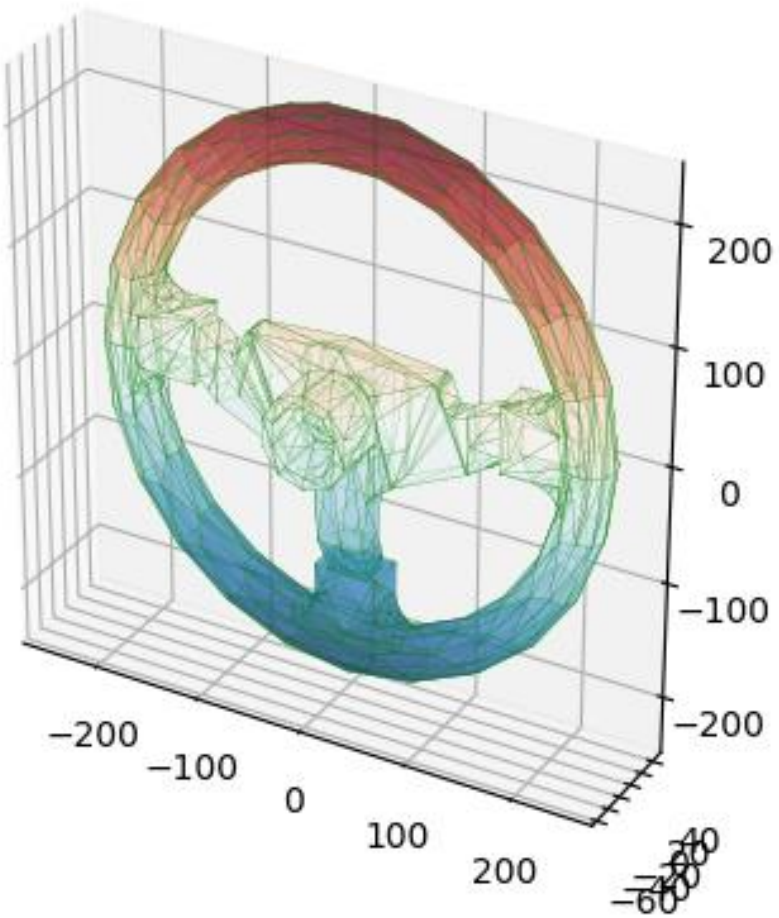

180
160
140
120
100
60
50
40
30
20
10
0
0
20
40
60
80

Exercício 5.5

```
from plyfile import PlyData
import numpy as np
import matplotlib.pyplot as plt
plydata = PlyData.read('/stratocaster.ply')
with open('/stratocaster.ply', 'rb') as f:
    plydata = PlyData.read(f)
plydata.elements[0].name
plydata.elements[0].data[0]
nr_vertices = plydata.elements[0].count
nr_faces = plydata.elements[1].count
vertices = np.array([plydata['vertex'][k] for k in
range(nr_vertices)])
x, y, z = zip(*vertices)
faces = [plydata['face'][k][0] for k in range(nr_faces)]
ax = plt.figure().add_subplot(projection = '3d')
ax.plot_trisurf(x, y, z, triangles = faces, cmap = 'RdBu_r', edgecolor
= 'green',
    linewidth = 0.1, alpha = 0.5)
ax.view_init(elev=90, azim=180)
ax.set_box_aspect((np.ptp(x), np.ptp(y), np.ptp(z)))
plt.show()
```



```
plydata = PlyData.read('/steeringweel.ply')
with open('/steeringweel.ply', 'rb') as f:
    plydata = PlyData.read(f)
plydata.elements[0].name
plydata.elements[0].data[0]
nr_vertices = plydata.elements[0].count
nr_faces = plydata.elements[1].count
vertices = np.array([plydata['vertex'][k] for k in
range(nr_vertices)])
x, y, z = zip(*vertices)
faces = [plydata['face'][k][0] for k in range(nr_faces)]
ax = plt.figure().add_subplot(projection = '3d')
ax.plot_trisurf(x, y, z, triangles = faces, cmap = 'RdBu_r', edgecolor
= 'green',
    linewidth = 0.1, alpha = 0.5)
ax.set_box_aspect((np.ptp(x), np.ptp(y), np.ptp(z)))
plt.show()
```



```
plydata = PlyData.read('/big_porsche.ply')
with open('/big_porsche.ply', 'rb') as f:
    plydata = PlyData.read(f)
plydata.elements[0].name
plydata.elements[0].data[0]
nr_vertices = plydata.elements[0].count
nr_faces = plydata.elements[1].count
vertices = np.array([plydata['vertex'][k] for k in
range(nr_vertices)])
x, y, z = zip(*vertices)
faces = [plydata['face'][k][0] for k in range(nr_faces)]
ax = plt.figure().add_subplot(projection = '3d')
ax.plot_trisurf(x, y, z, triangles = faces, cmap = 'RdBu_r', edgecolor
= 'green',
    linewidth = 0.1, alpha = 0.5)
ax.view_init(elev=90, azim=-90)
ax.set_box_aspect((np.ptp(x), np.ptp(y), np.ptp(z)))
plt.show()
```