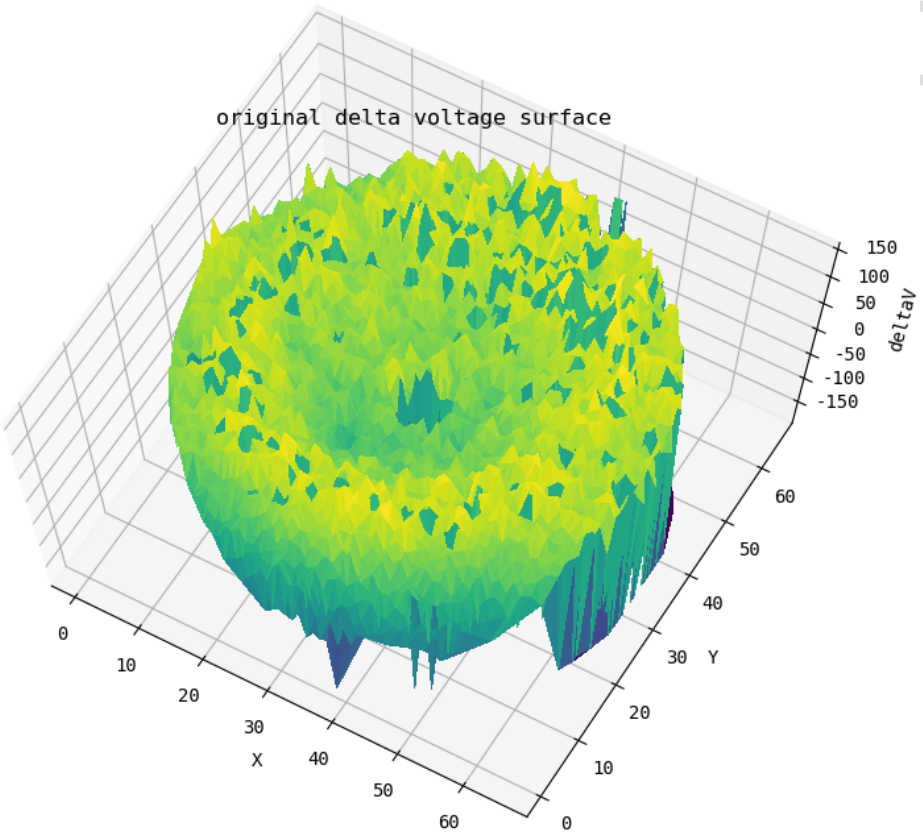
Sparse Vector to Mesh Grid

Description	Attachment
Wafer-level delta voltage (mV) convert from RO SPICE-to-silicon analysis.	deltaVRow.csv
SIDD uniformity	sidd_uniformity.csv

我們可能常需要將帶有 (x,y) 座標的稀疏向量資料點傳換成三維網格的特徵曲面並做後續的分析, 如圖, 我們將上面附件的原始資料作圖, 會發現有需多資料破洞, (可能是測試時發生錯誤, 或是資料轉換時發生資料範圍溢位等情況).

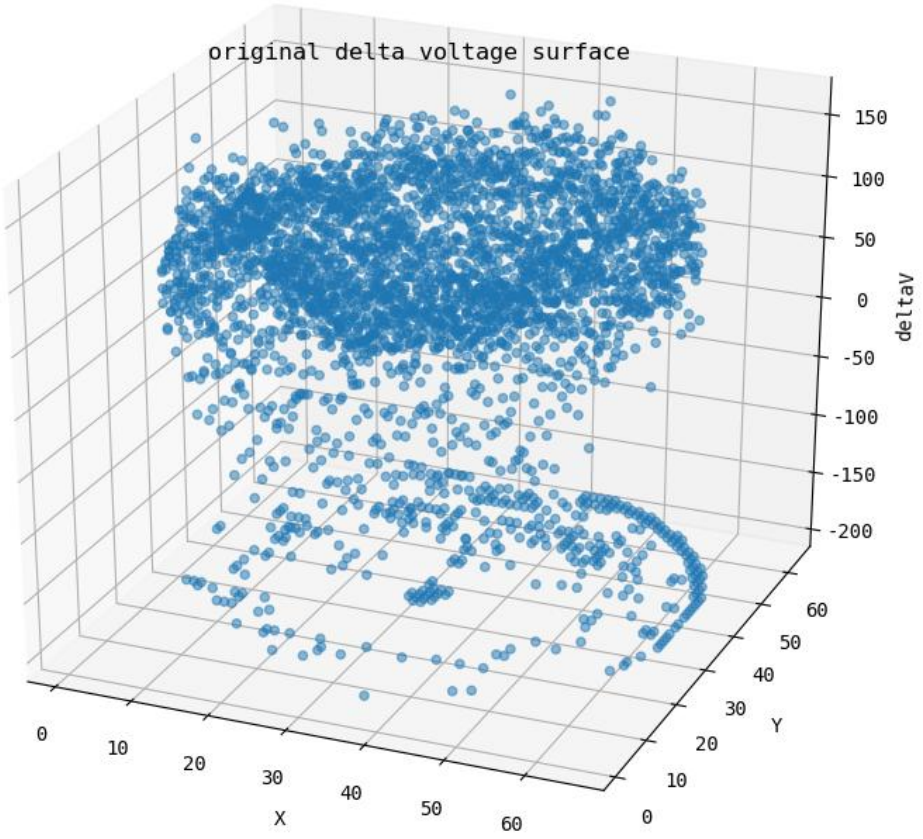


很多有密集恐懼症的人應該都受 不了, 這時候我們可以先快速檢視一下資料分佈的形狀與範圍, 是否有不合理或會干擾分析的數據? 一般可以先從剃除 2-sigma 邊界外的資料點開始.

```
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

```
df = pd.read_csv('deltaVRow.csv')
x,y,z = map(np.array,zip(*df.values))

fig = plt.figure(figsize=(8,7))
ax = fig.add_subplot(111,projection='3d',title=f'original delta voltage
surface')
#ax.plot_trisurf(x,y,z,cmap=plt.cm.viridis,alpha=1,antialiased=False,linewidth
=0,edgecolors='none')
ax.scatter(x,y,z,alpha=0.5)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('deltaV')
ax.view_init(70,300)
ax.dist = 8
plt.show()
```



我們將圖旋轉一些視角觀察, 發現晶圓周圍與中心位置有非常大的資料斷層, 我們決定只保留 2-sigma 資料並將中心處資料剃除後才進行後須分析. 不用擔心, 這些動作只是讓原本資料點變得更稀疏而已 (而且這些資料點會嚴重干擾分析, 若真的存在應該直接篩掉).

```
dm = df.describe(percentiles=[0.025,0.975]) # delta voltage statistics
```

```
b1,b2 = dm['deltaV(mV)']['2.5%'],dm['deltaV(mV)']['97.5%'] # 2 sigma boundary
dt = df[(b1<df['deltaV(mV)'])&(df['deltaV(mV)']<b2)]
idx = (dt[(30<dt['X'])&(dt['X']<40)&(25<dt['Y'])&(dt['Y']<35)]['deltaV(mV)']<-
100).index
dt = dt.drop(index=idx) # drop invalid center data
print(f'filter data: {dt.shape[0]/df.shape[0]*100:.1f}%')
```

之後調用 scipy.interpolate.griddata 來做二圍網格的內插補點, 只需一行程式.

```
from scipy.interpolate import griddata
import numpy as np

x,y,z = map(np.array,zip(*dt.values))
gx,gy = np.mgrid[x.min():x.max():200j,y.min():y.max():200j]
pz = griddata(np.array([x,y]).T,z,(gx,gy),method='linear')

# drop nan with Dataframe
d =
pd.DataFrame(np.array([gx.ravel(),gy.ravel(),pz.ravel()]).T,columns=['X','Y','
Z']).dropna()
px,py,pz = zip(*d.values)

fig = plt.figure(figsize=(8,7))
ax = fig.add_subplot(111,projection='3d',title=f'original delta voltage
surface')
ax.plot_trisurf(px,py,pz,cmap=plt.cm.viridis,alpha=1,antialiased=False,linewid
th=0,edgecolors='none')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('deltaV')
ax.view_init(70,300)
ax.dist = 8
plt.show()
```

如此, 原本稀疏的資料點就順利提升解析度到二維網格資料點了!