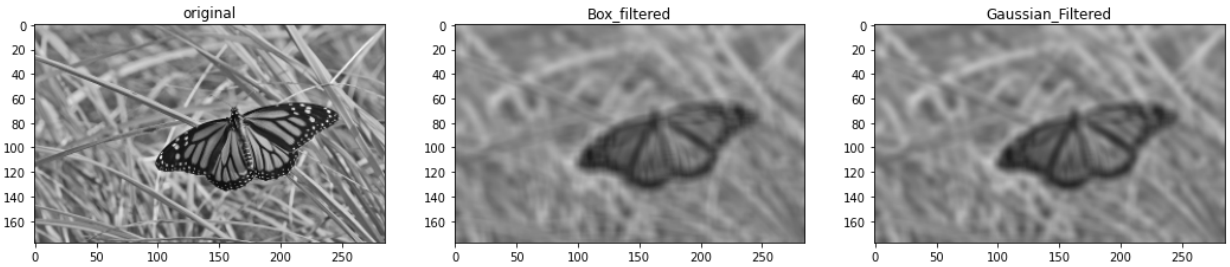
190432J

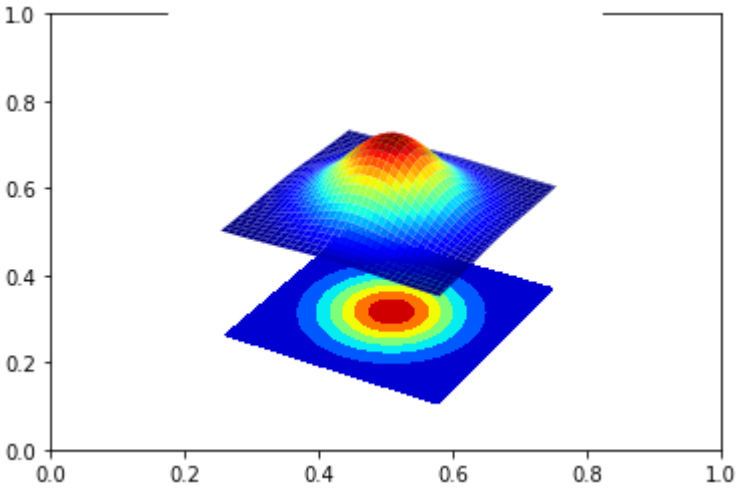
Pathirana R.P.U.A.

In [ ]:
```
#Q1
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

im = cv.imread('butterfly.jpg', cv.IMREAD_REDUCED_GRAYSCALE_8)
assert im is not None

k_size = 9
sigma = 4
box_kernal = 1./81*np.ones((9,9))
im_avg = cv.filter2D(im, -1 ,box_kernal )
im_gaussian = cv.GaussianBlur(im, (k_size,k_size), sigma)

fig, ax = plt.subplots(1,3, figsize=(18,16))

ax[0].imshow(im,cmap='gray', vmin = 0, vmax=255)
ax[0].set_title("original")
ax[1].imshow(im_avg ,cmap='gray', vmin = 0, vmax=255)
ax[1].set_title("Box_filtered")
ax[2].imshow(im_gaussian,cmap='gray', vmin = 0, vmax=255)
ax[2].set_title("Gaussian_Filtered")


plt.show()
```



In [ ]:
```
#Q2

import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm

fig, ax = plt.subplots()
ax = fig.add_subplot(111, projection= '3d')

step = 0.1
sigma =2
x = np.arange(-5,5+step,step)
y = np.arange(-5,5+step,step)
```

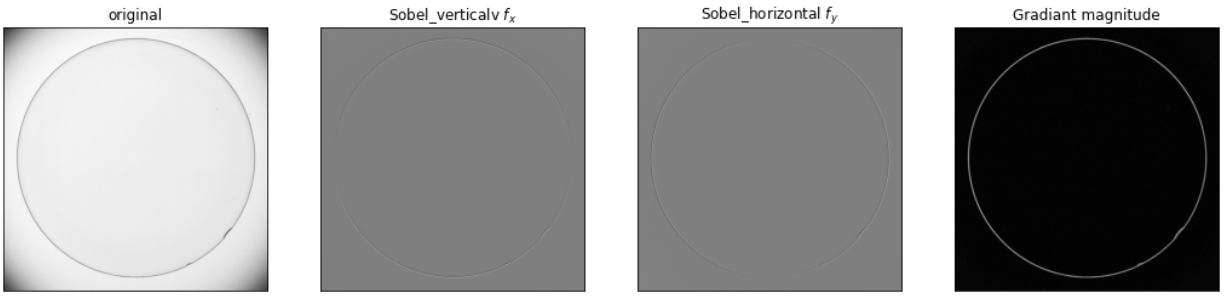
```
xx, yy = np.meshgrid(x,y)
g = np.exp(-(xx**2+yy**2)/(2*sigma**2))

surf = ax.plot_surface(xx,yy,g,cmap=cm.jet)

cset = ax.contourf(xx,yy, g, zdir='z', offset=np.min(g)-1.5, cmap=cm.jet)

plt.axis('off')
ax.set_zlim(np.min(g)-1.5,np.max(g))
plt.show()
```



In [ ]:

```
#Q3

import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

im = cv.imread('contact_lens.tif', cv.IMREAD_GRAYSCALE).astype(np.float32)
assert im is not None

sobel_v = np.array([(-1, -2, -1), (0, 0, 0), (1, 2, 1)] , dtype=np.float32)
im_x = cv.filter2D(im,-1, sobel_v )
sobel_h = np.array([(-1, 0, 1), (-2, 0, 2), (-1, 0, 1)], dtype=np.float32)
im_y = cv.filter2D(im,-1, sobel_h )

grad_mag = np.sqrt ((im_x**2)+(im_y**2))

fig, ax = plt.subplots(1,4, figsize=(18,16))

ax[0].imshow(im,cmap='gray', vmin = 0, vmax=255)
ax[0].set_title("original")
ax[1].imshow(im_x ,cmap='gray', vmin = -1020, vmax=1020)
ax[1].set_title("Sobel_verticalv $f_x$")
ax[2].imshow(im_y,cmap='gray', vmin = -1020, vmax=1020)
ax[2].set_title("Sobel_horizontal $f_y$")
ax[3].imshow(grad_mag,cmap='gray', )
ax[3].set_title("Gradiant magnitude")


for i in range (4):
    ax[i].set_xticks([]), ax[i].set_yticks([])
```

```
plt.show()
```



In [ ]:

```
#Q4

import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

im = cv.imread('tom.jpg', cv.IMREAD_GRAYSCALE).astype(np.float32)
assert im is not None

sigma = 2
gaussinan_1d = cv.getGaussianKernel(5,sigma)
im_lp = cv.sepFilter2D(im, -1, gaussinan_1d,gaussinan_1d)

im_hp = im - im_lp
im_sharpend =  cv.addWeighted(im, 1.0 , im_hp , 2.0, 0 )
ig, ax = plt.subplots(1,4, figsize=(18,16))

ax[0].imshow(im,cmap='gray')
ax[0].set_title("original")
ax[1].imshow(im_lp ,cmap='gray')
ax[1].set_title("F_lp")
ax[2].imshow(im_hp,cmap='gray')
ax[2].set_title("F_hp")
ax[3].imshow(im_sharpend,cmap='gray', )
ax[3].set_title("Sharpened")


for i in range (4):
    ax[i].set_xticks([]), ax[i].set_yticks([])

plt.show()
```