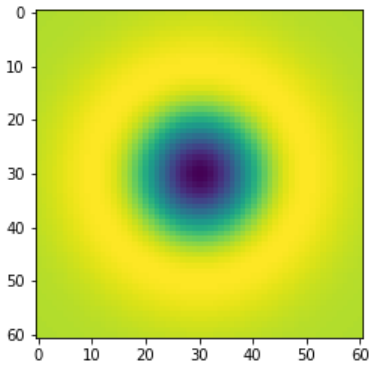
Index No - 180601T

Name - M.A.L Shyamal

In [2]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
```

In [3]:
```
sigma = 10
hw = 3*sigma
X , Y = np.meshgrid(np.arange(-hw, hw+1,1),np.arange(-hw, hw+1,1))
log = 1/(2*np.pi*sigma**2)*(X**2 / (sigma**2) +Y**2/(sigma**2) -2 )*np.exp(-(X**2 + Y**2)/(2*sigma**2))

plt.imshow(log)
plt.show()
```

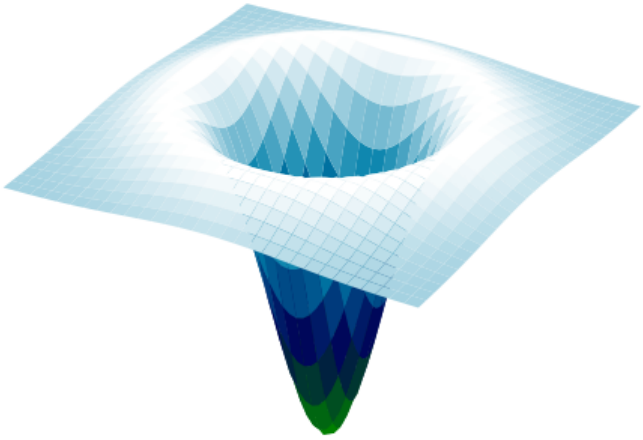


In [4]:
```
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
from matplotlib.ticker import LinearLocator , FormatStrFormatter
```

In [5]:
```
fig = plt.figure(figsize=(10,10))
ax = fig.add_subplot(111,projection = '3d')

surf = ax.plot_surface(X,Y,log,cmap=cm.ocean,linewidth=0,antialiased = True)

ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))
plt.axis('off')
plt.show()
```

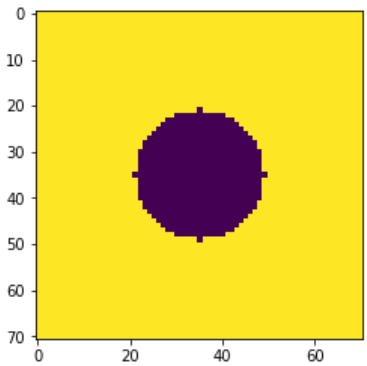


In [6]:
```
w , h = 71,71
hw = w//2
hh = h//2

f = np.ones((h,w), dtype=np.float32)*255
X , Y = np.meshgrid(np.arange(-hh, hh+1,1),np.arange(-hw, hw+1,1))
r = w//5 #14

f*= X**2 + Y**2 > r**2

plt.imshow(f)
plt.show()
```

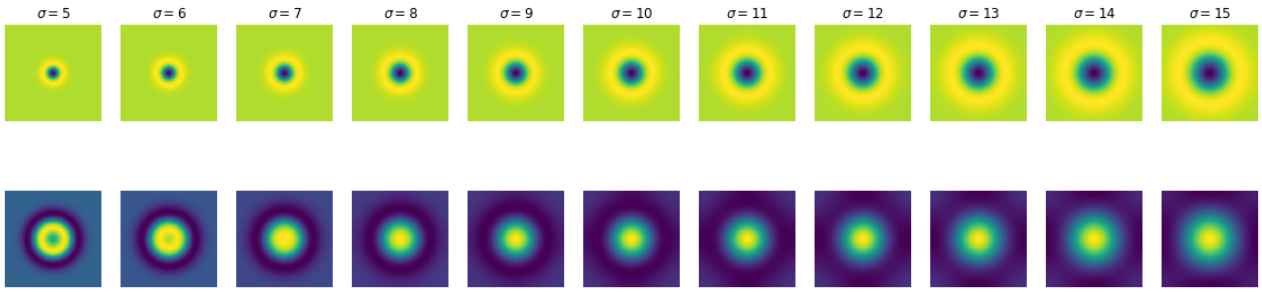


In [7]:
```
s = 11
fig, ax = plt.subplots(2,s, figsize=(20,5))
scale_space = np.empty((h,w,s), dtype=np.float32)
sigmas = np.arange(5,16,1)

for i,sigma in enumerate(sigmas):
    log_hw = 3*np.max(sigmas)
    X , Y = np.meshgrid(np.arange(-log_hw, log_hw+1,1),np.arange(-log_hw, log_hw+1,1))
    log = 1/(2*np.pi*sigma**2)*(X**2 / (sigma**2) +Y**2/(sigma**2) -2 )*np.exp(-(X**2 + Y**2)/(2*sigma**2))
    f_log = cv.filter2D(f,-1,log)
    scale_space[:,:,i] = f_log
    ax[0,i].axis('off')
    ax[0,i].set_title('$\sigma = {}$'.format(sigma))
    ax[0,i].imshow(log)
    ax[1,i].imshow(f_log)
    ax[1,i].axis('off')

indices = np.unravel_index(np.argmax(scale_space,axis=None),scale_space.shape)
print(indices) # r = \sqrt{2}*sigma
print(sigmas[indices[2]])
```

```
(35, 35, 5)
10
```



In [11]:
```
img1 = cv.imread('img1.ppm')
img2 = cv.imread('img4.ppm')

img1 = cv.cvtColor(img1, cv.COLOR_BGR2GRAY)
img2 = cv.cvtColor(img2, cv.COLOR_BGR2GRAY)

sift = cv.SIFT_create()

keypoints_1, descriptors_1 = sift.detectAndCompute(img1,None)
keypoints_2, descriptors_2 = sift.detectAndCompute(img2,None)

bf = cv.BFMatcher(cv.NORM_L1, crossCheck=True)

matches = bf.match(descriptors_1,descriptors_2)
matches = sorted(matches, key = lambda x:x.distance)

img3 = cv.drawMatches(img1, keypoints_1, img2, keypoints_2, matches[:50], img2, flags=2)
plt.figure(figsize=(15,15))
plt.imshow(img3)
plt.xticks([]), plt.yticks([])
plt.show()
```