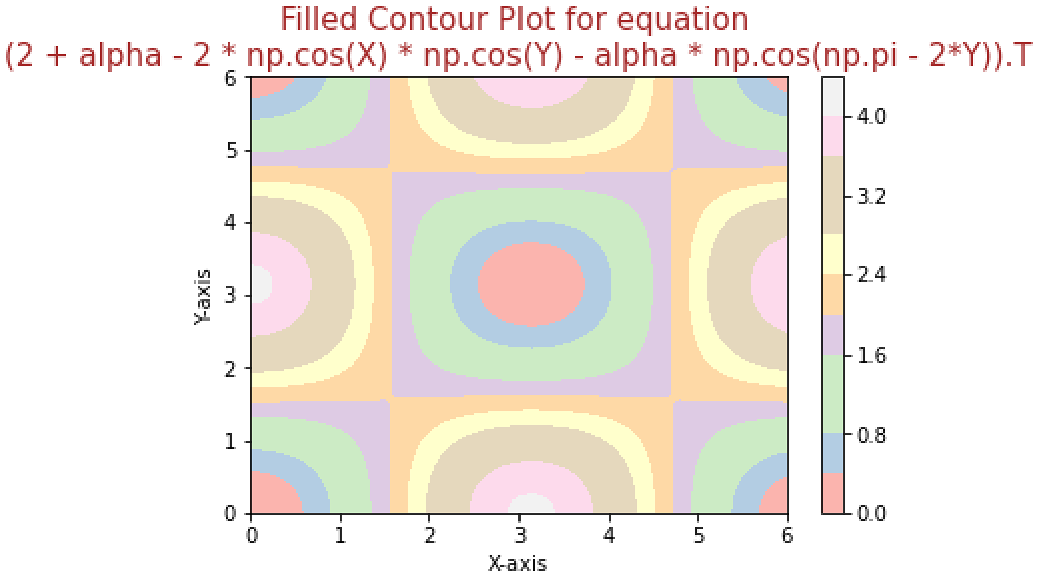
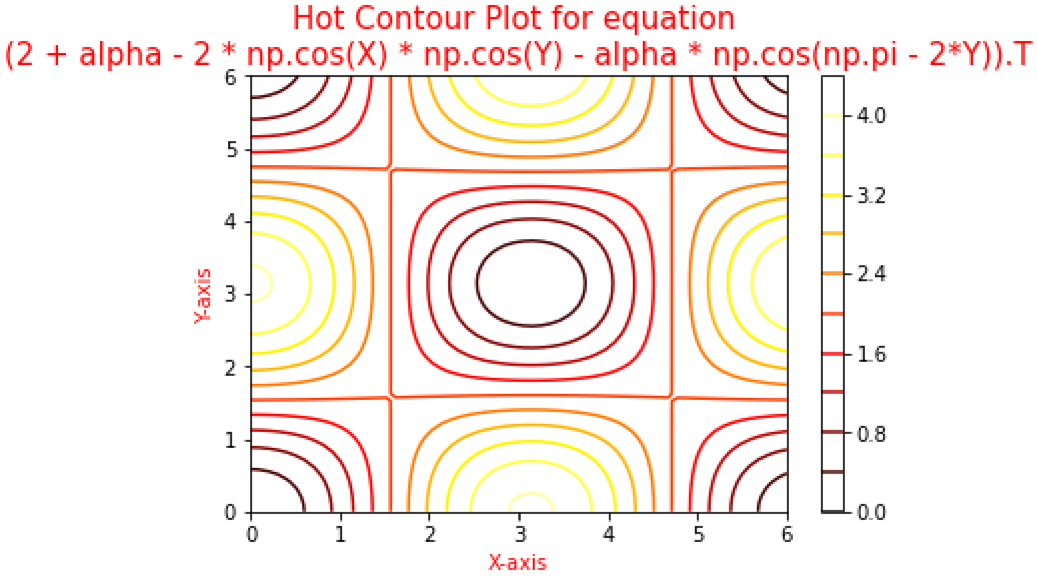
In [44]:

```
print('Assignment-3: Contour and Surface Plots')
print('\nIn your experiments you found that the phenomenon you are measuring is d
print('Z = (2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha * np.cos(np.pi - 2*Y)).
print('\n1.Contour plot')
print('\n1). Make a contour plot. Make sure to add labels in the plot or a legend
import numpy as np
import matplotlib.cm as cm
import matplotlib.pyplot as plt

#take alpha value as input and project the contour plots
alpha = float(input("\nEnter alpha value (any number between 0 to 1) : "))

x = np.linspace(0,6,60)
y = np.linspace(0,6,60)
X, Y = np.meshgrid(x, y)
Z = (2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha * np.cos(np.pi - 2*Y)).T

plt.contourf(X,Y,Z, levels=10, cmap='Pastel1')
plt.colorbar()
plt.title('Filled Contour Plot for equation \n(2 + alpha - 2 * np.cos(X) * np.cos
plt.xlabel('X-axis')
plt.ylabel('Y-axis')

plt.show()


print('2). Do this for an additional color mapping: e.g. hot/cold or black/white.
plt.title('Hot Contour Plot for equation \n(2 + alpha - 2 * np.cos(X) * np.cos(Y)
plt.xlabel('X-axis', color = 'red')
plt.ylabel('Y-axis', color = 'red')
cf = plt.contour(X,Y,Z, levels = 10,  cmap = 'hot')
plt.colorbar(cf)
plt.show()
```

```
Assignment-3: Contour and Surface Plots

In your experiments you found that the phenomenon you are measuring is describe
d by the following equation:
Z = (2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha * np.cos(np.pi - 2*Y)).T

1.Contour plot

1). Make a contour plot. Make sure to add labels in the plot or a legend for co
lors on the contours. You can choose either a filled contour plot or colored li
nes, your choice.

Enter alpha value (any number between 0 to 1) : 0.033
```



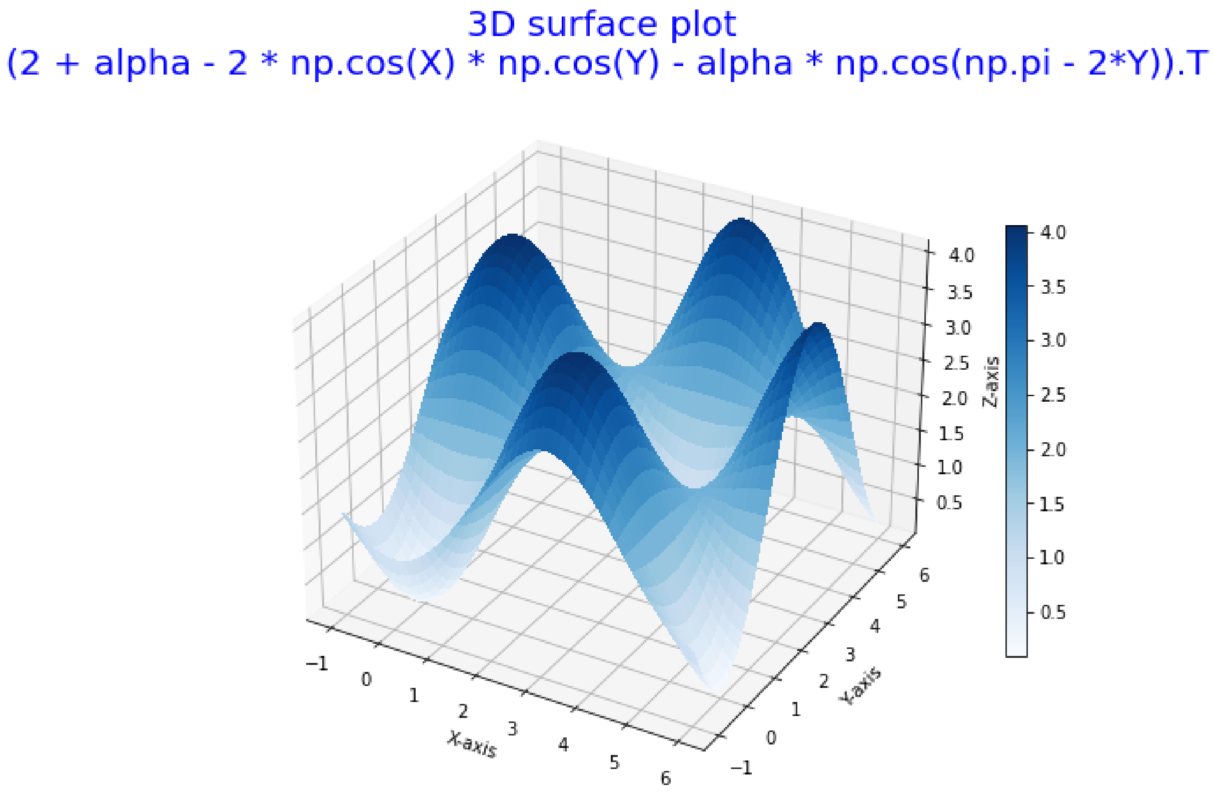
2). Do this for an additional color mapping: e.g. hot/cold or black/white.



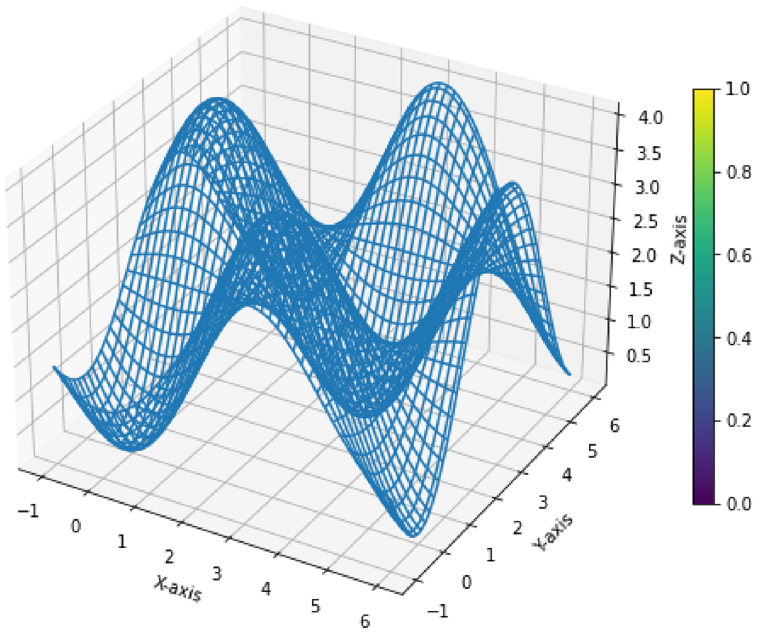
In [8]:

```
print('\n2.Surface plots (or mesh plots)')
print('\n1). Using the same data set as before, create a surface plot. Also be su
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

#take alpha value as input and project the surface plots
alpha = float(input("\nEnter alpha value (any number between 0 to 1) : "))

#linespace indicates the x and y axis range
x = np.linspace(-1,6,90)
y = np.linspace(-1,6,90)
X, Y = np.meshgrid(x, y)
Z = (2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha * np.cos(np.pi - 2*Y)).T

#surface plot

fig = plt.figure(figsize=(9,9))
ax = plt.axes(projection='3d')

#used antialiased to smoothen the surface
s = ax.plot_surface(X, Y, Z , cmap = 'Blues', antialiased = False, linewidth = 0)
#shrink reduces the size of the colorbar
fig.colorbar(s, ax = ax, shrink = 0.5)

ax.set_title('3D surface plot \n(2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha *
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')

plt.show()

#mesh plot
fig = plt.figure(figsize=(9,9))
ax = plt.axes(projection='3d')
m = ax.plot_wireframe(X,Y,Z)
fig.colorbar(m, shrink = 0.5)

ax.set_title('3D Mesh Plot for the above surface plot \n(2 + alpha - 2 * np.cos(X
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')

plt.show()

print('\n2). Generate at least one additional viewpoint of the surface that may a

fig = plt.figure(figsize=(11,11))
ax = fig.add_subplot(projection='3d')

s1 =  ax.plot_surface(X, Y, Z , cmap = 'summer', antialiased = False, linewidth =

fig.colorbar(s1, shrink = 0.5)

ax.set_title('3D surface plot - Additional viewpoint of surface \n(2 + alpha - 2
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')

#azimuth and elevation values are passed to view_init()
ax.view_init(30, 90)

plt.show()
```

```
2.Surface plots (or mesh plots)

1). Using the same data set as before, create a surface plot. Also be sure to c
hoose an appropriate color mapping to help in interpretation. If you can’t make
a surface plot, a mesh plot (where the surface is not filled in) will suffice.

Enter alpha value (any number between 0 to 1) : 0.039
```

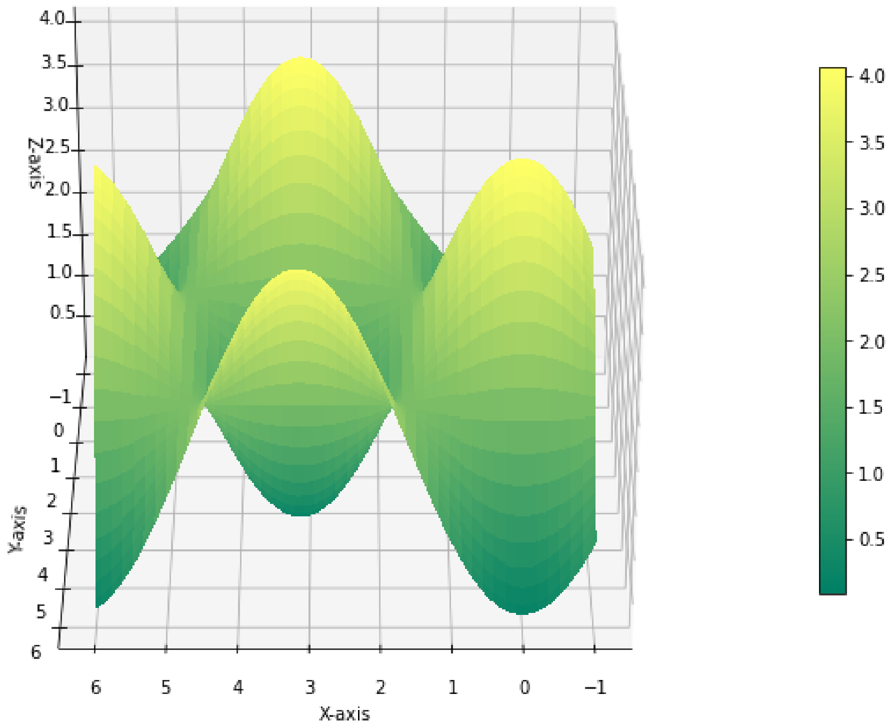


3D Mesh Plot for the above surface plot
(2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha * np.cos(np.pi - 2*Y)).T



2). Generate at least one additional viewpoint of the surface that may also be helpful in providing insights.

3D surface plot - Additional viewpoint of surface
(2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha * np.cos(np.pi - 2*Y)).T


Z-axis
Y-axis
X-axis