[Q1] In your experiments you found that the phenomenon you are measuring is described by the following equation:

Z = (2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha * np.cos(np.pi - 2*Y)).T

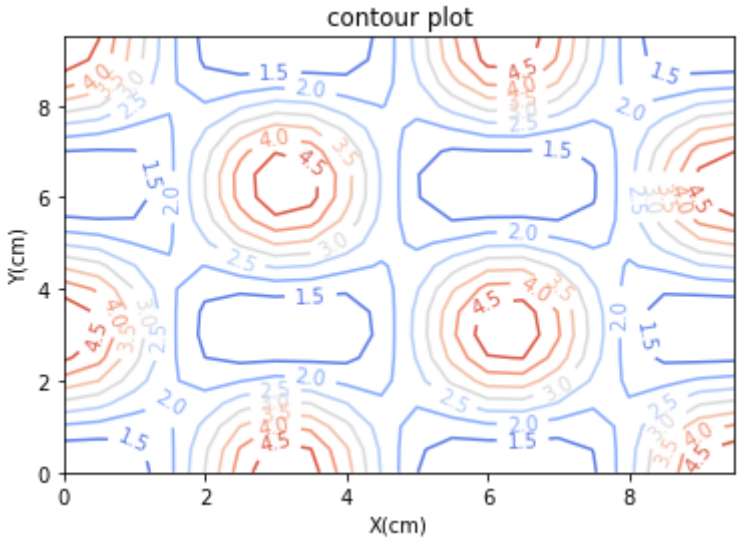
To be complete, be sure to show the contour plot for the entire range for which this function has interesting features to observe.

Contour plot 1). Make a contour plot. Make sure to add labels in the plot or a legend for colors on the contours. You can choose either a filled contour plot or colored lines, your choice. 2). Do this for an additional color mapping: e.g. hot/cold or black/white.

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

In [33]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm

delta = 0.5
x = np.arange(0,10.0,delta)
y = np.arange(0,10.0,delta)
X, Y = np.meshgrid(x,y)
alpha = 0.5
Z = (2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha * np.cos(np.pi - 2*Y)).T
plt.figure()
cp = plt.contour(X,Y,Z,cmap = cm.coolwarm)
plt.clabel(cp)
plt.title('contour plot')
plt.xlabel('X(cm)')
plt.ylabel('Y(cm)')
```

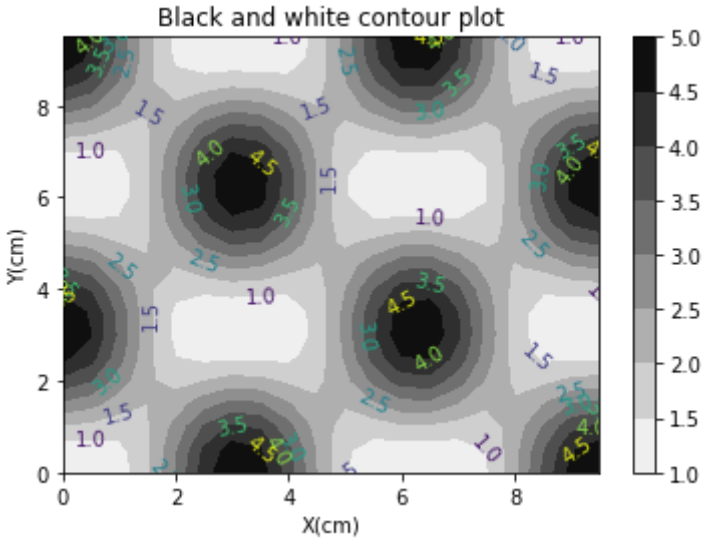
Out[33]: Text(0, 0.5, 'Y(cm)')



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

In [30]:
```
delta = 0.5
x = np.arange(0,10.0,delta)
y = np.arange(0,10.0,delta)
X, Y = np.meshgrid(x,y)
alpha =0.5
Z = (2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha * np.cos(np.pi - 2*Y)).T
plt.figure()
cp = plt.contourf(X,Y,Z)
plt.clabel(cp)
contour_filled = plt.contourf(X,Y,Z,cmap = cm.binary)
plt.colorbar(contour_filled)
plt.title('Black and white contour plot')
plt.xlabel('X(cm)')
plt.ylabel('Y(cm)')
```

Out[30]:
```
Text(0, 0.5, 'Y(cm)')
```



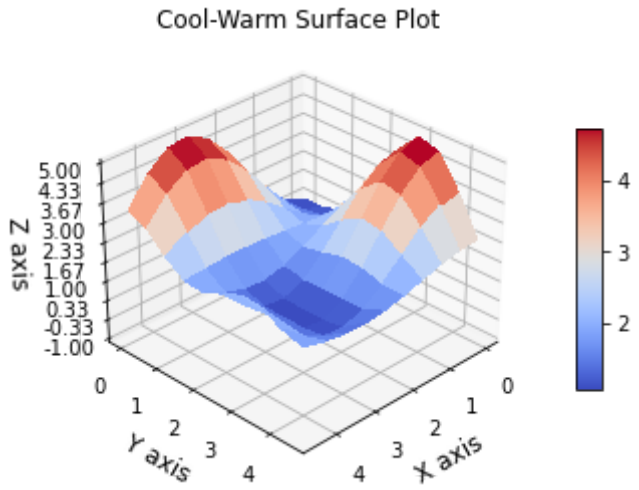
[Q2] Surface plots (or mesh plots)

1). Using the same data set as before, create a surfaceplot. Also be sure to choose an appropriate color mappingto help in interpretation. If you can't make a surface plot, amesh plot (where the surface is not filled in) willsuffice.

2). Generate at least one additional viewpoint of thesurface that may also be helpful in providinginsights.

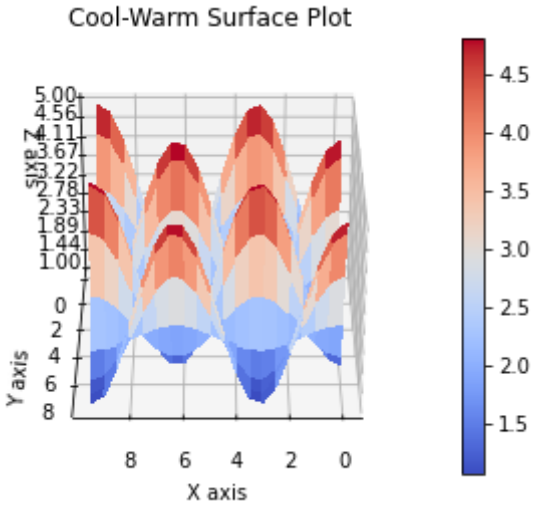
In [18]:
```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib.ticker import LinearLocator, FormatStrFormatter

fig = plt.figure()
ax = fig.gca(projection='3d')
ax.view_init(azim=45)
X = np.arange(0,5,0.5)
Y = np.arange(0,5,0.5)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X**2 + Y**2)
alpha = 0.5
Z = (2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha * np.cos(np.pi - 2*Y))
surf = ax.plot_surface(X, Y, Z, cmap =cm.coolwarm ,linewidth=0, antialiased
ax.set_zlim(-1,5)
ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))
fig.colorbar(surf,shrink=0.6,aspect =10,pad=0.09)
ax.set_xlabel('X axis',fontsize=12)
ax.set_ylabel('Y axis',fontsize=12)
ax.set_zlabel('Z axis',fontsize=12)
ax.set_title('Cool-Warm Surface Plot', fontsize=12)
plt.show()
```

```
/var/folders/ty/4py3xkds2xd231nxqy3v22j40000gp/T/ipykernel_26030/16405341
39.py:7: MatplotlibDeprecationWarning: Calling gca() with keyword argumen
ts was deprecated in Matplotlib 3.4. Starting two minor releases later, g
ca() will take no keyword arguments. The gca() function should only be us
ed to get the current axes, or if no axes exist, create new axes with def
ault keyword arguments. To create a new axes with non-default arguments,
use plt.axes() or plt.subplot().
  ax = fig.gca(projection='3d')
```



In [34]:
```
fig = plt.figure()
ax = fig.gca( projection='3d')
ax.view_init(azim=90 )
X = np.arange(0,10,0.5)
Y = np.arange(0,10,0.5)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X**2 + Y**2)
alpha = 0.5
Z = (2 + alpha - 2 * np.cos(X) * np.cos(Y) - alpha * np.cos(np.pi - 2*Y))
surf = ax.plot_surface(X, Y, Z, cmap =cm.coolwarm ,linewidth=0, antialiased
ax.set_zlim(1, 5)
ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))
fig.colorbar(surf)
ax.set_xlabel('X axis')
ax.set_ylabel('Y axis')
ax.set_zlabel('Z axis')
ax.set_title('Cool-Warm Surface Plot')
plt.show()
```

```
/var/folders/ty/4py3xkds2xd231nxqy3v22j40000gp/T/ipykernel_26030/12054236
29.py:2: MatplotlibDeprecationWarning: Calling gca() with keyword argumen
ts was deprecated in Matplotlib 3.4. Starting two minor releases later, g
ca() will take no keyword arguments. The gca() function should only be us
ed to get the current axes, or if no axes exist, create new axes with def
ault keyword arguments. To create a new axes with non-default arguments,
use plt.axes() or plt.subplot().
  ax = fig.gca( projection='3d')
```



In [ ]: