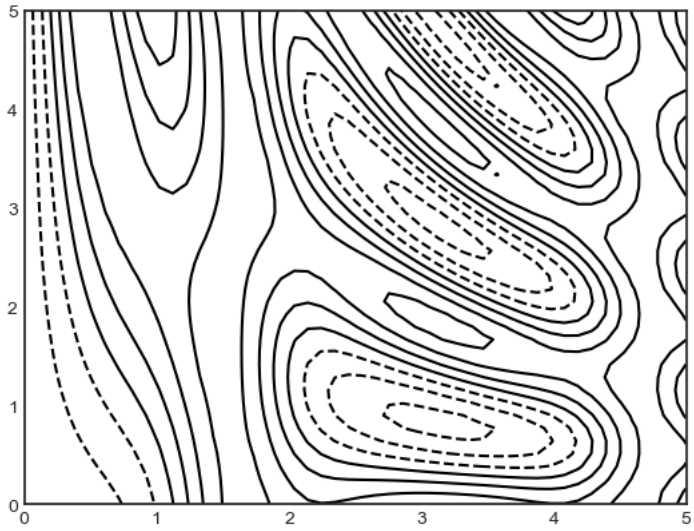
In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('seaborn-white')
import numpy as np
```

In [33]:
```
def f(x,y):#three dimennsional grid
    return np.sin(x)**10 +np.cos(10+y*x)*np.cos(x)
```

In [34]:
```
x=np.linspace(0,5,50)
y=np.linspace(0,5,40)
X,Y=np.meshgrid(x,y)#meshgrid is used for intepreting 2 dimensional data as grid
Z=f(X,Y)
```

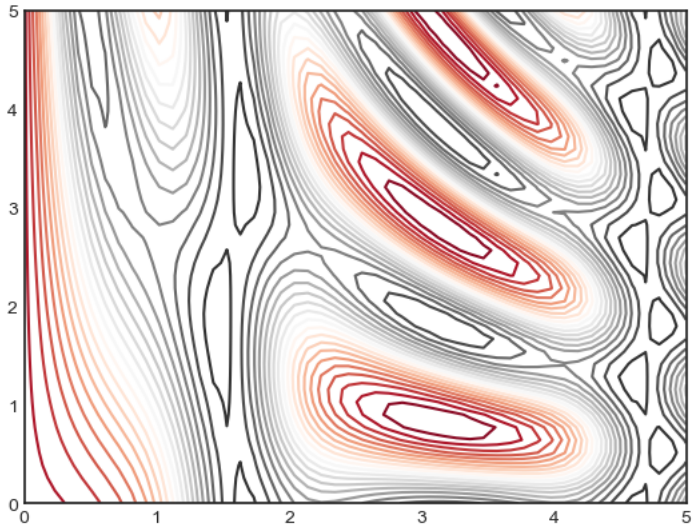
In [35]:
```
plt.contour(X,Y,Z,colors="black")
```

Out[35]: <matplotlib.contour.QuadContourSet at 0x211da6b63a0>



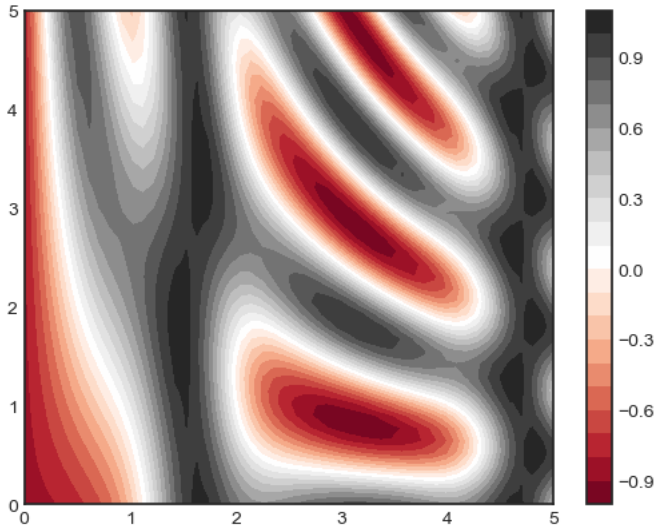
In [36]:
```
#we can sepcify number of lines and color coding as well
plt.contour(X,Y,Z,20,cmap="RdGy")#red gray color map
```

Out[36]: <matplotlib.contour.QuadContourSet at 0x211dab752e0>



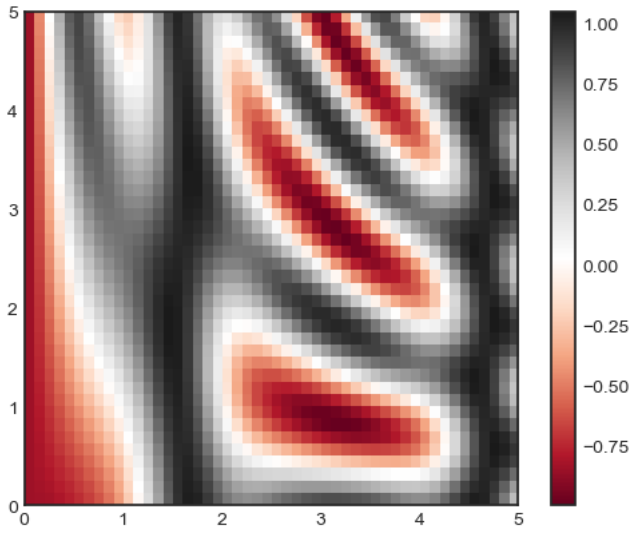
In [37]:
```
#contourf fills the space between lines with colors
plt.contourf(X,Y,Z,20,cmap="RdGy")
#using colorbar to represent color scheme
plt.colorbar()
```

Out[37]: <matplotlib.colorbar.Colorbar at 0x211dac91940>



In [38]:
```
#imshow intepresents 2 dimensional grid as image
plt.imshow(Z,extent=[0, 5, 0, 5],origin='lower',cmap='RdGy',aspect="equal")
plt.colorbar()
#imshow doesnt take x and y axes,by exntent we have to specify [xmin,xmax,ymin,ymax]
#imshow has orgin at upper left by default and hence we have bring it to lower by specifying the orgin argument
#by default imshow will try to make aspect ratio equal that is make x and y units equal
```

Out[38]: <matplotlib.colorbar.Colorbar at 0x211dadfa9a0>



In [39]:
```
contours=plt.contour(X,Y,Z,3,colors='black')#storing contour plot in a variable
plt.clabel(contours)#labelling contours
plt.imshow(Z,extent=[0, 5, 0, 5],origin='lower',cmap='RdGy',aspect="equal",alpha=0.3)#alpha represents transparency
plt.colorbar()
```

Out[39]:
```
<matplotlib.colorbar.Colorbar at 0x211dc0027f0>
```