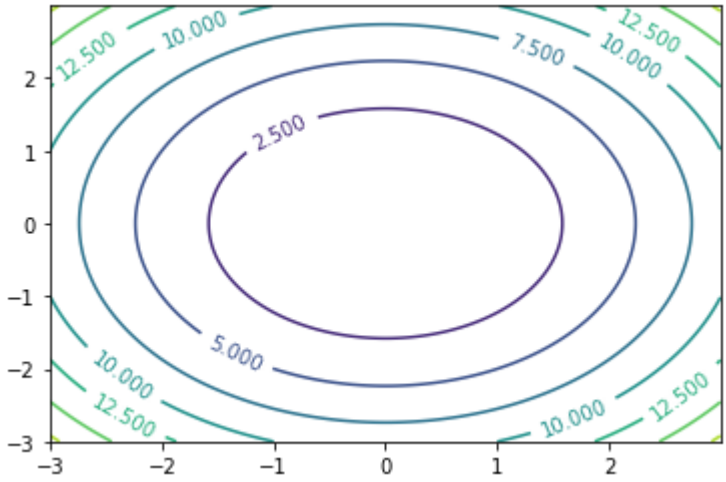
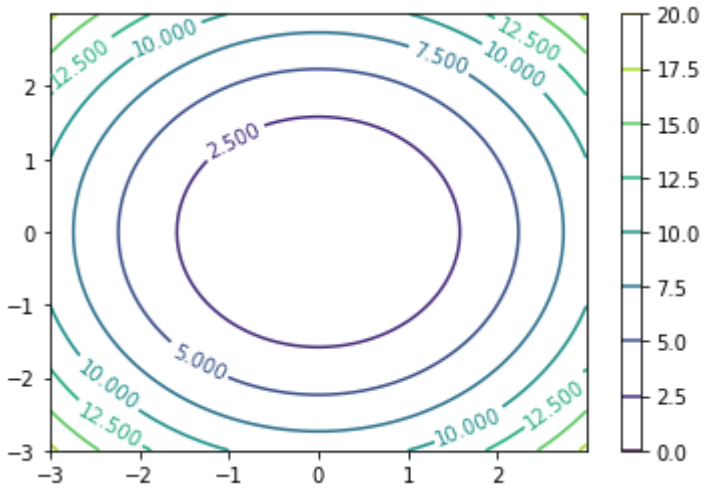
In [1]:

```
# Contours represent the outline of an object. Contours are continuous
# lines highlighting the shape of objects. Contours are useful in the area
# of cartography, which means map-making. On maps, a contour joins points of equal
# height. So, all the points on a contour line are at an equal elevation from the sea le
# In other applications where we use contours, all the points on the same contour line
# have the same values (or magnitude).
# Let’s draw a simple contour. We will create and visualize our own data by creating
# circular contour as follows:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(-3, 3, 0.005)
y = np.arange(-3, 3, 0.005)
X, Y = np.meshgrid(x, y)
Z = (X**2 + Y**2)
out = plt.contour(X, Y, Z)
plt.clabel(out, inline=True,fontsize=10)
plt.show()
```



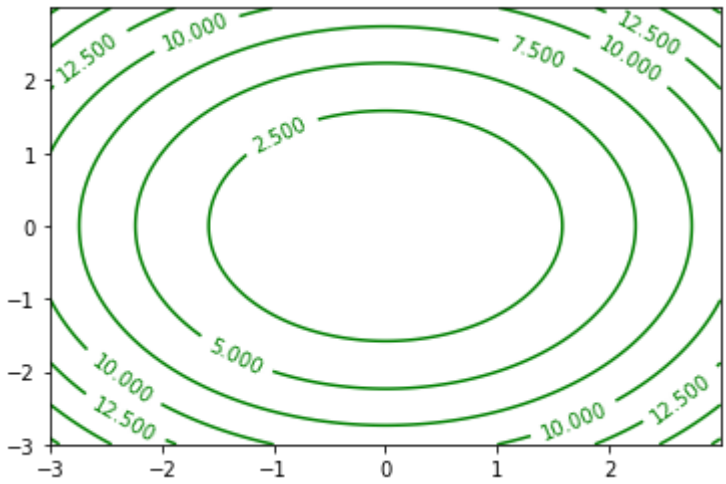
In [2]:

```
# You can also add a color bar to the output as follows:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(-3, 3, 0.005)
y = np.arange(-3, 3, 0.005)
X, Y = np.meshgrid(x, y)
Z = (X**2 + Y**2)
out = plt.contour(X, Y, Z)
plt.clabel(out, inline=True,fontsize=10)
plt.colorbar(out)
plt.show()
```



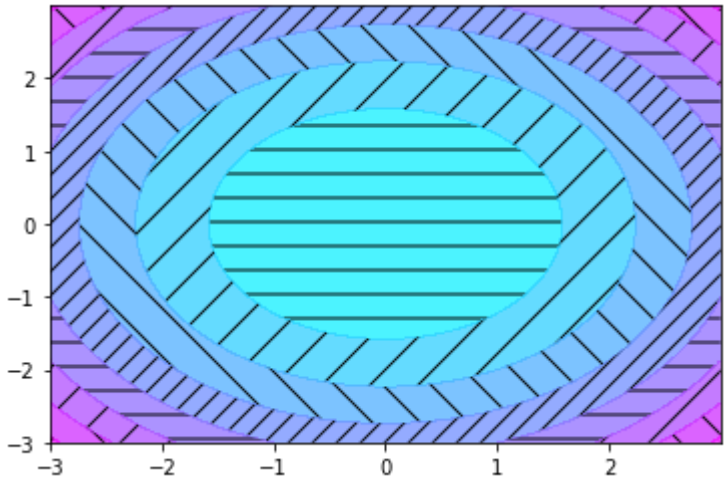
In [3]:

```
# You can also set the colors of the contour as follows:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(-3, 3, 0.005)
y = np.arange(-3, 3, 0.005)
X, Y = np.meshgrid(x, y)
Z = (X**2 + Y**2)
out = plt.contour(X, Y, Z, colors='g')
plt.clabel(out, inline=True, fontsize=10)
plt.show()
```



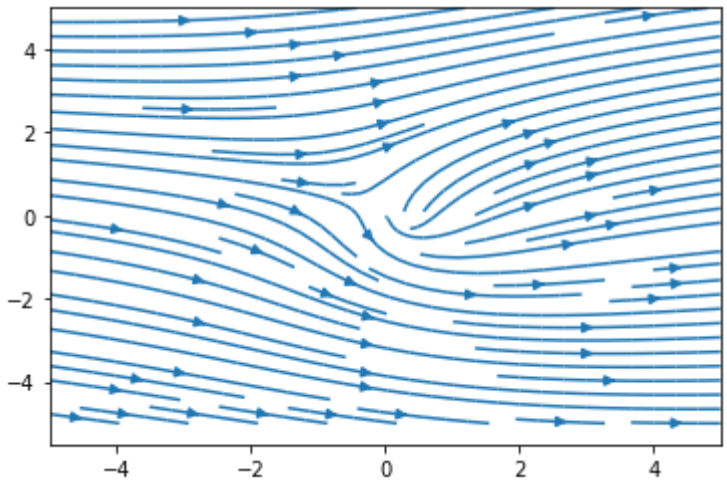
In [4]:

```
# You can also have a filled contour. The styles are used to highlight the various areas
# in the contour visualization. Let's visualize filled contours as follows:

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(-3, 3, 0.005)
y = np.arange(-3, 3, 0.005)
X, Y = np.meshgrid(x, y)
Z = (X**2 + Y**2)
plt.contourf(X, Y, Z, hatches=['-', '/', '\\', '//'],
             cmap='cool', alpha=0.75)
plt.show()
```



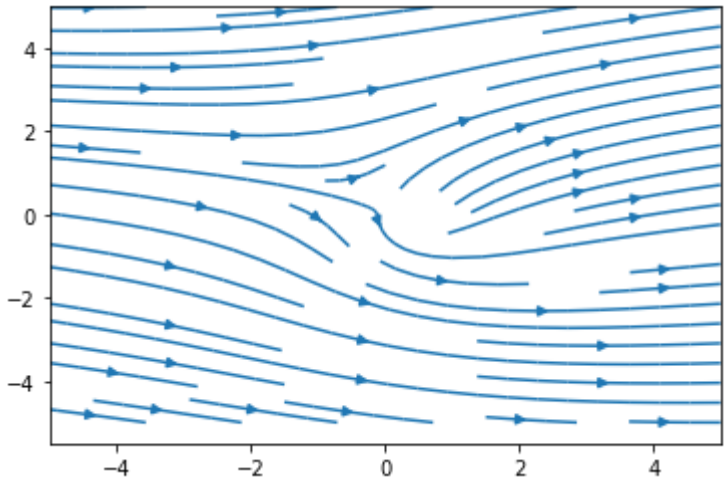
In [5]:

```
# Vectors, by contrast, are entities that have magnitude and direction.
# For example, force has a magnitude and a direction. A specific example
# is a magnetic force field. You can visualize vectors with stream plots.
# Let's create our own dataset to visualize this. We will create a
# mesh with X and Y. Then we will create U and V to show the magnitude.

import numpy as np
import matplotlib.pyplot as plt
Y, X = np.mgrid[-5:5:200j, -5:5:300j]
U = X**2 + Y**2
V = X + Y
# You can create a simple stream plot as follows:
plt.streamplot(X, Y, U, V)
plt.show()
```



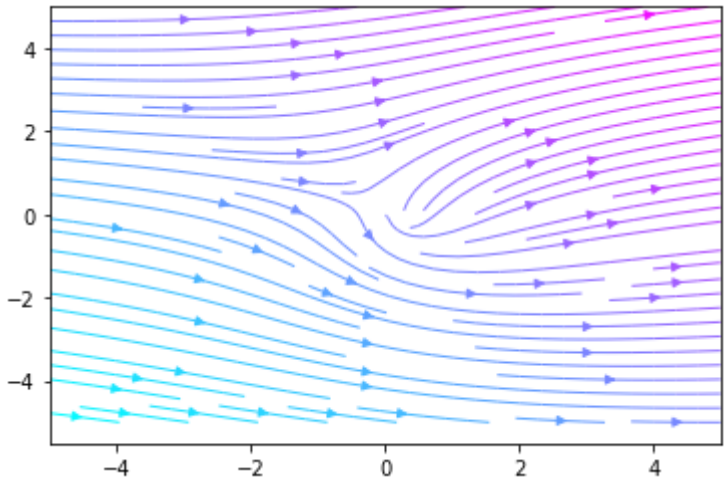
In [6]:

```
# You can also have stream plots of variable densities as follows:

import numpy as np
import matplotlib.pyplot as plt
Y, X = np.mgrid[-5:5:200j, -5:5:300j]
U = X**2 + Y**2
V = X + Y
plt.streamplot(X, Y, U, V, density=[0.5, 0.75])
plt.show()
```



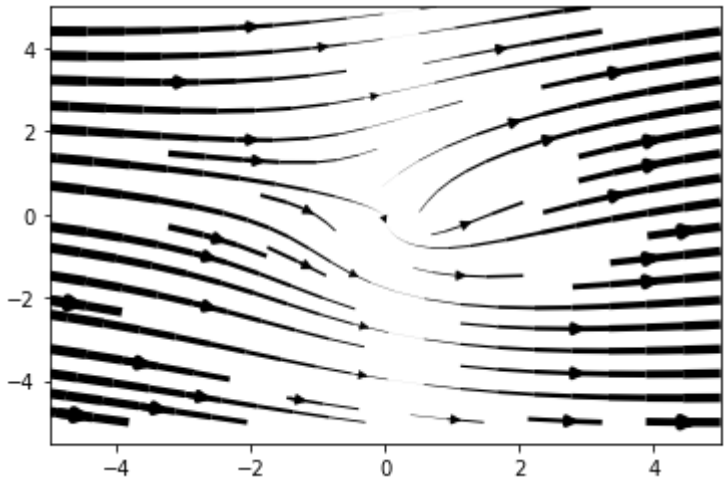
In [7]:

```
# You can also assign colors to the stream plot as follows:

import numpy as np
import matplotlib.pyplot as plt
Y, X = np.mgrid[-5:5:200j, -5:5:300j]
U = X**2 + Y**2
V = X + Y
plt.streamplot(X, Y, U, V, color=V,
               linewidth=1, cmap='cool')
plt.show()
```



In [8]:

```
# You can also create a stream plot with variable line widths as follows:

import numpy as np
import matplotlib.pyplot as plt
Y, X = np.mgrid[-5:5:200j, -5:5:300j]
U = X**2 + Y**2
V = X + Y
plt.streamplot(X, Y, U, V,density=0.6,
               color='k',linewidth=X)
plt.show()
```



In [9]:

```
# You can also use quiver plots for the vector visualizations as follows:

import numpy as np
import matplotlib.pyplot as plt
X = np.arange(-5, 5, 0.5)
Y = np.arange(-10, 10, 1)
U, V = np.meshgrid(X, Y)
plt.quiver(X, Y, U, V)
plt.show()
```