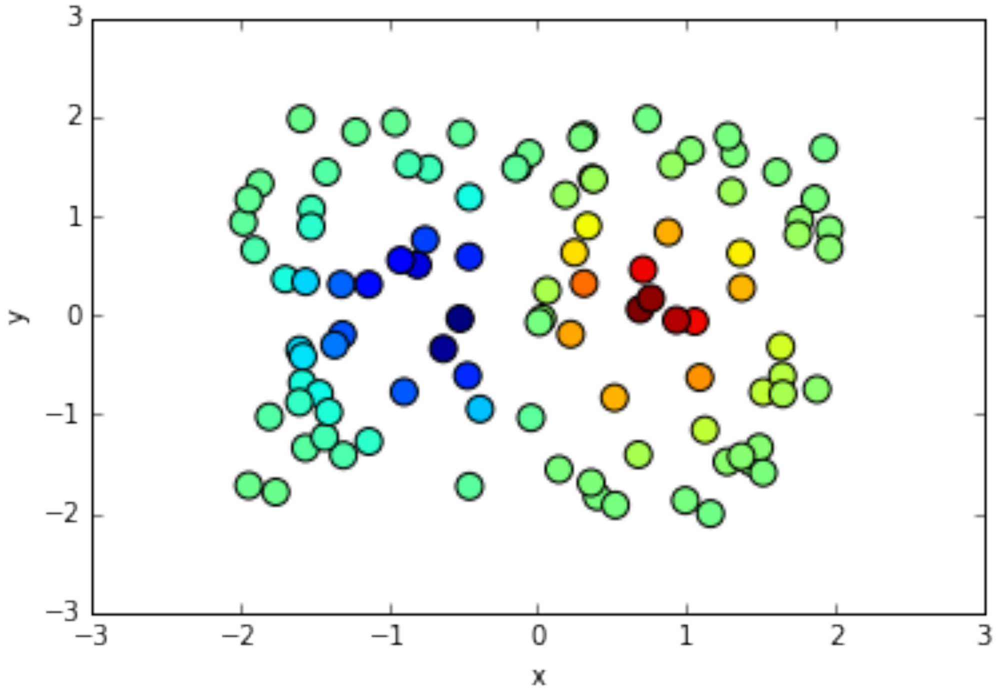
In [2]:

```
import numpy as np
from scipy.interpolate import Rbf
import matplotlib.pyplot as plt
from matplotlib import cm
%matplotlib inline
```

Let's make some data

In [30]:

```
x = np.random.rand(100)*4.0-2.0
y = np.random.rand(100)*4.0-2.0
z = x*np.exp(-x**2-y**2)
ti = np.linspace(-2.0, 2.0, 100)
plt.scatter(x, y, 100, z, cmap=cm.jet)
plt.xlabel('x')
plt.ylabel('y')
xx, yy = np.meshgrid(ti, ti)
```



How doe we "interpolate" this data in 2D?

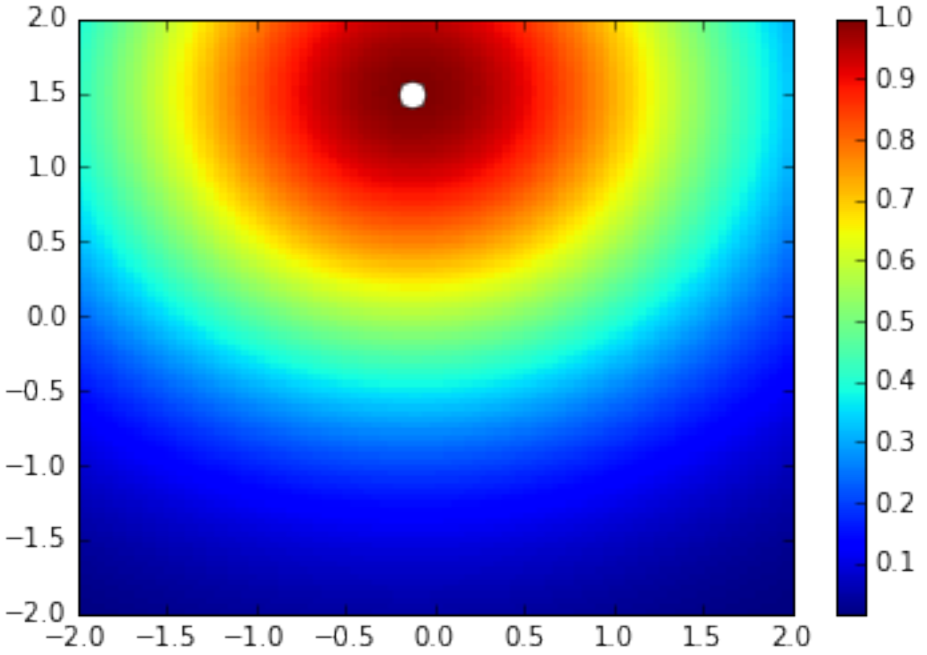
$$\phi(r) = e^{-(\varepsilon r)^2}$$

In [32]:

```
rbf = Rbf(x[0], y[0], 1, epsilon=2, function='gaussian')
zz = rbf(xx, yy)
plt.pcolor(xx, yy, zz)
plt.plot(x[0], y[0], 'o', ms=10, color='1.0')
plt.colorbar()
```

Out[32]:

```
<matplotlib.colorbar.Colorbar at 0x11717fba8>
```



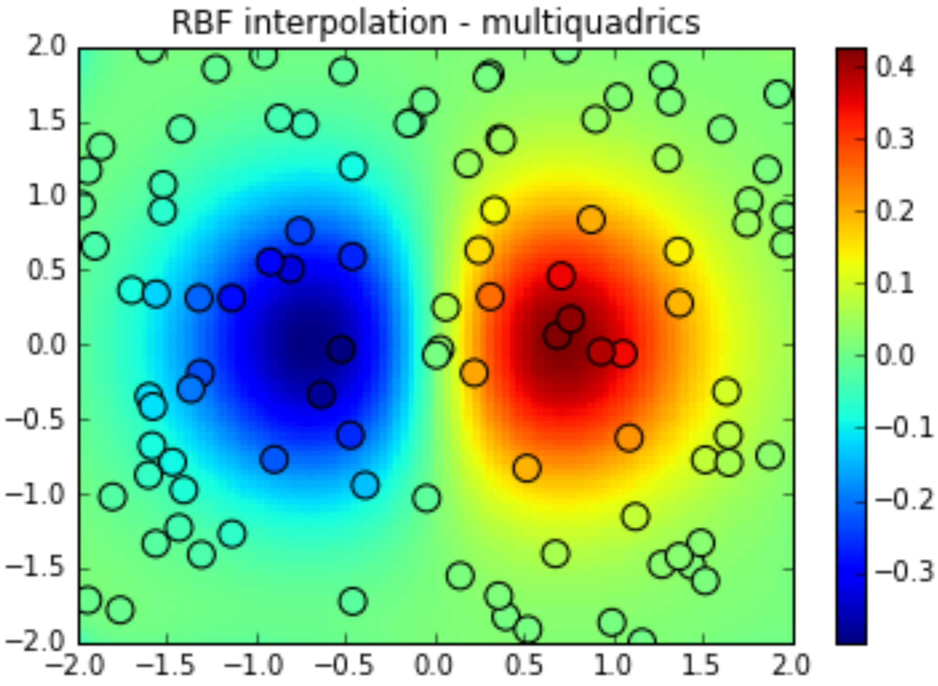
now do this for all of the data

In [33]:

```
rbf = Rbf(x, y, z, epsilon=2, function='gaussian')
zz = rbf(xx, yy)
plt.pcolor(xx, yy, zz)
plt.scatter(x, y, 100, z)
plt.title('RBF interpolation - multiquadrics')
plt.xlim(-2, 2)
plt.ylim(-2, 2)
plt.colorbar()
```

Out[33]:

```
<matplotlib.colorbar.Colorbar at 0x11759cc88>
```



In [ ]: