Matrices for image blurring

In [7]:

```
import numpy as np
import matplotlib.pyplot as plt
```

In [8]:

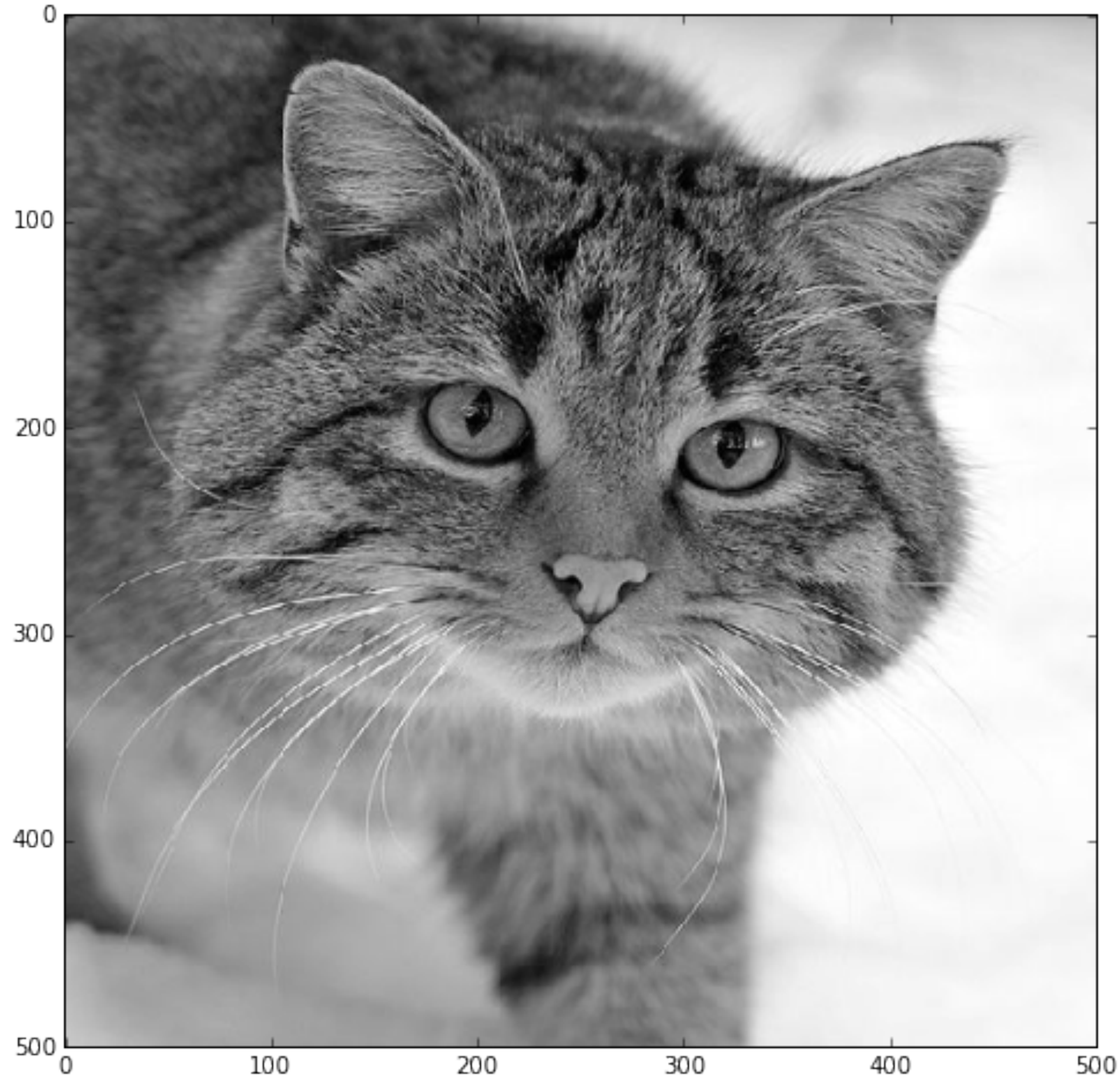
```
from PIL import Image

with Image.open("cat.jpeg").resize((500,500)) as img:
    img = np.array(img).sum(axis=-1)

h, w = img.shape
```

In [17]:

```
plt.figure(figsize=(8,8))
plt.imshow(img, cmap="gray")
```

Out[17]:

```
<matplotlib.image.AxesImage at 0xb08cf0ec>
```



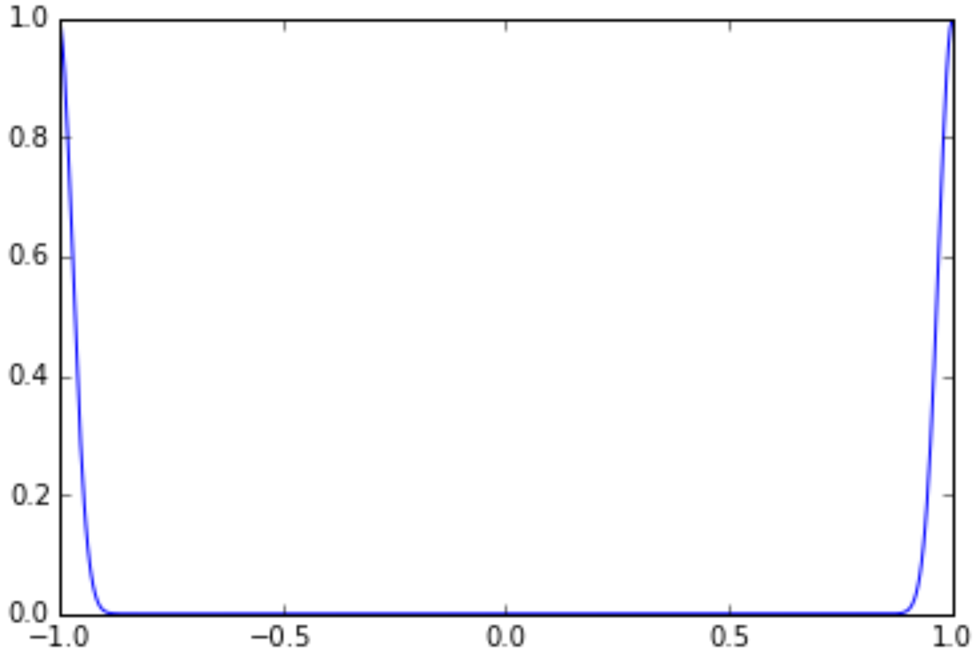
Now make a Gaussian with as many pixels as the image is wide.

In [10]:

```
x = np.linspace(-1, 1, w)
gaussian = np.exp(-500*x**2)
gaussian = np.roll(gaussian, -w//2)
plt.plot(x, gaussian)
```

Out[10]:

```
[<matplotlib.lines.Line2D at 0xb0ba3c2c>]
```



Now, fill a $w \times w$ matrix with shifted versions of this:

In [11]:

```
A = np.zeros((w,w))
for i in range(w):
    A[:, i] = np.roll(gaussian, i)
```

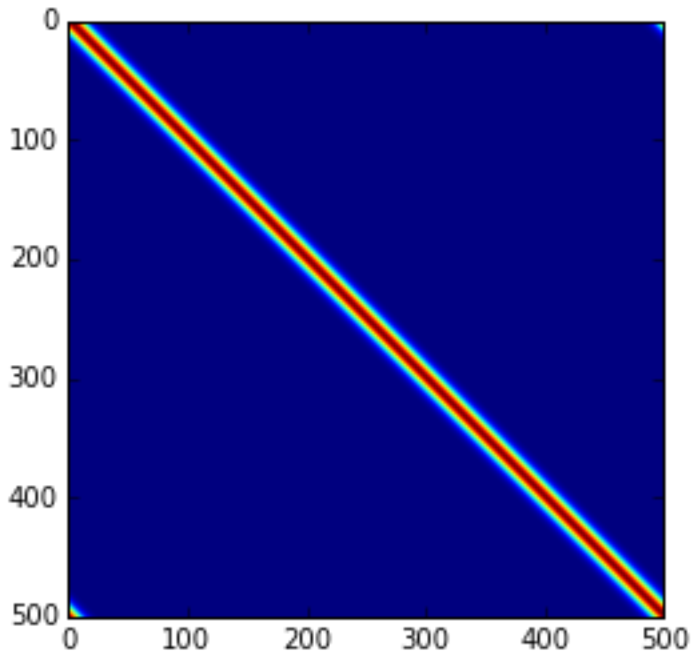
Here's a plot to show what just happened:

In [14]:

```
plt.imshow(A)
```

Out[14]:

```
<matplotlib.image.AxesImage at 0xb09728ec>
```



Multiply the cat by this.

In [15]:

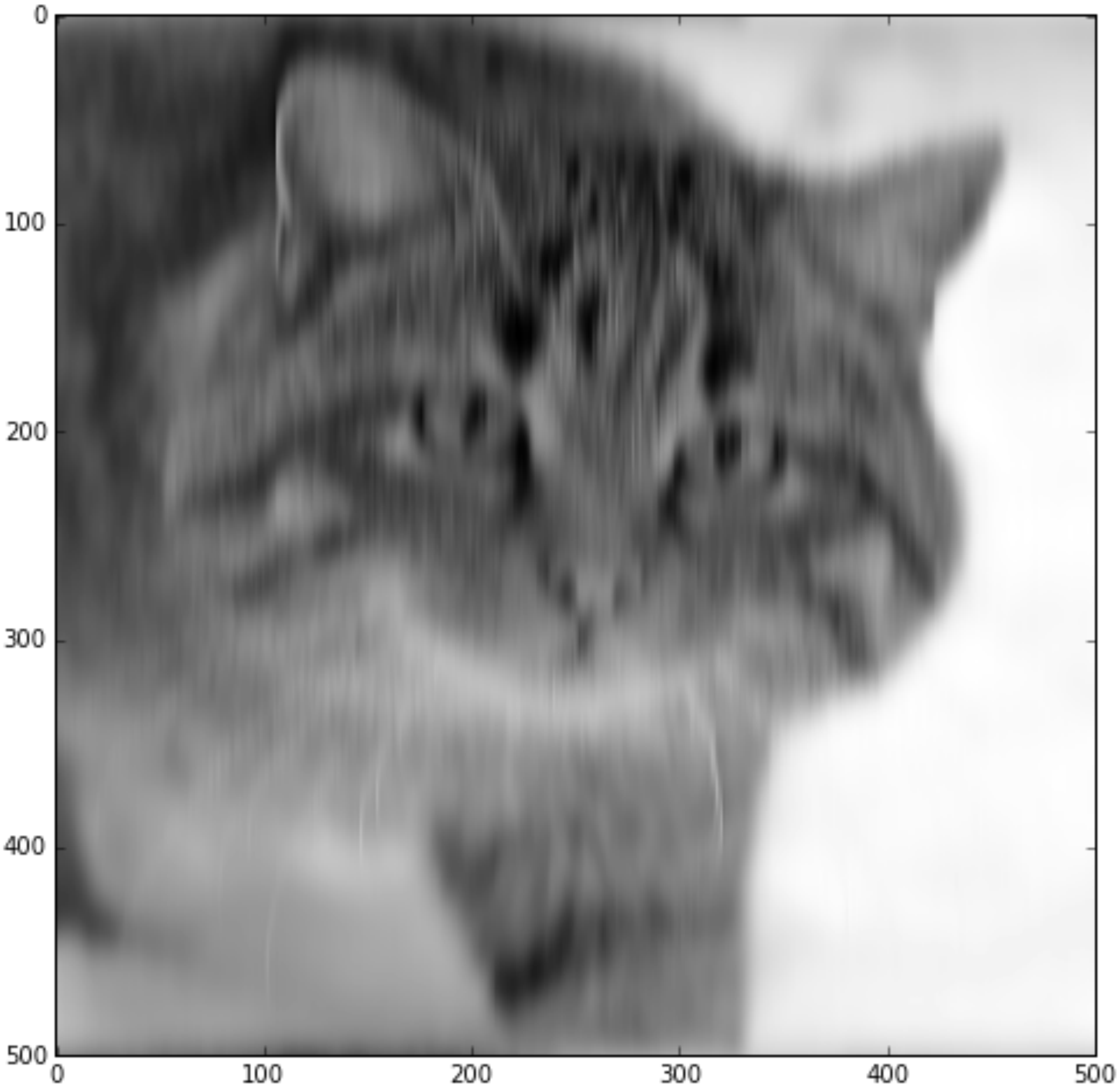
```
blurrycat = A.dot(img)
```

In [16]:

```
plt.figure(figsize=(8,8))
plt.imshow(blurrycat, cmap="gray")
```

```
Out[16]:
```

```
<matplotlib.image.AxesImage at 0xb091e5cc>
```



```
In [ ]:
```