EECS16A: Homework 5

Problem 3: Noisy Images

In [1]:

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

Let's load some data to start off with.

In [2]:

```
H3 = np.loadtxt("cond_10e6.txt", delimiter=',').reshape(100,100)
H2 = np.loadtxt("cond_1e3.txt", delimiter=',').reshape(100,100)
H1 = np.eye(100)
img = np.loadtxt("image.txt", delimiter=',').reshape(10,10)
```

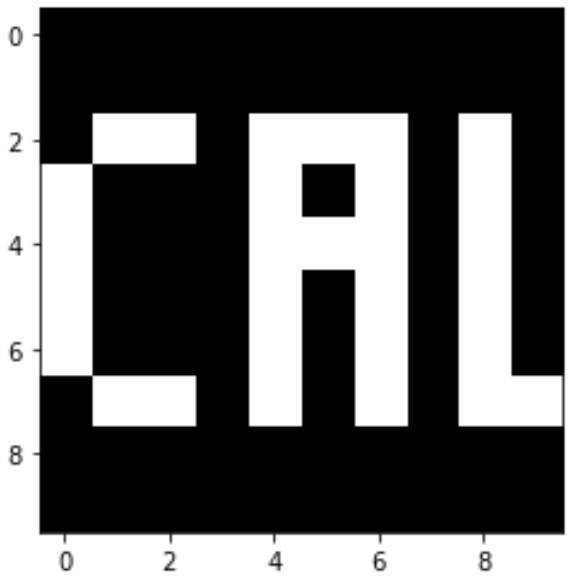
The code below displays the image.

In [3]:

```
plt.figure(0)
plt.imshow(img,cmap='gray')
```

Out[3]:

```
<matplotlib.image.AxesImage at 0x10f41ed50>
```



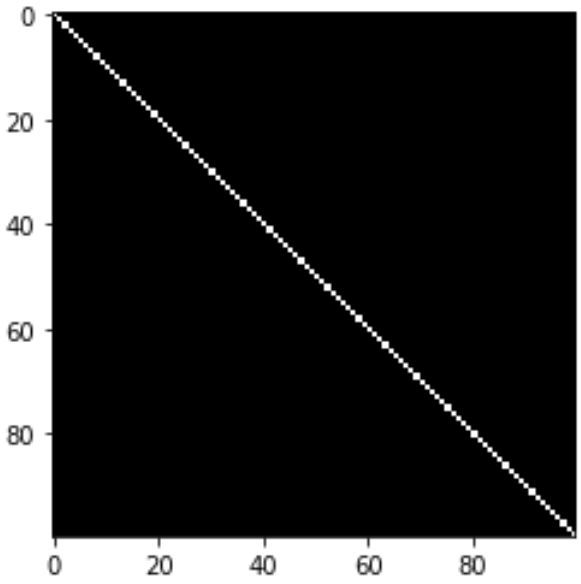
Then, lets display the set of masks

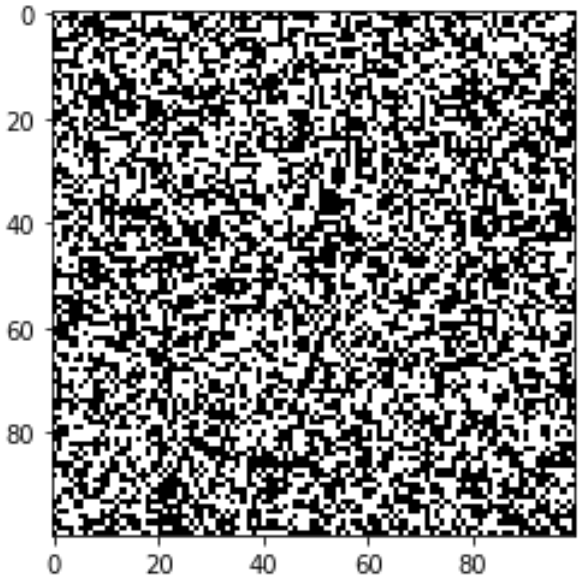
In [4]:

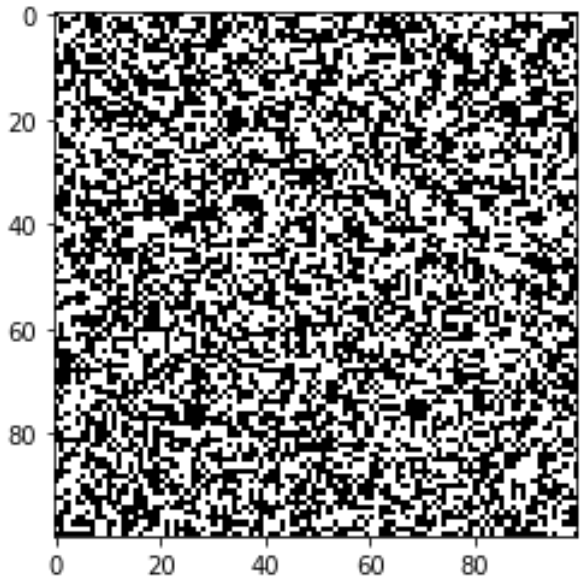
```
plt.figure(1)
plt.imshow(H1,cmap='gray')
plt.figure(2)
plt.imshow(H2,cmap='gray')
plt.figure(3)
plt.imshow(H3,cmap='gray')
```

Out[4]:

```
<matplotlib.image.AxesImage at 0x1115906d0>
```







We'll use numpy.random to make some noise.

In [5]:

```
noise = np.random.normal(0.5,0.1)
```

Lets compute the $\vec{b}$ vector for each matrix and add some noise to the $\vec{b}$ vector.

In [6]:

```
b1 = H1.dot(img.reshape(100)) + noise
b2 = H2.dot(img.reshape(100)) + noise
b3 = H3.dot(img.reshape(100)) + noise
```

First, let's compute $\vec{x}_1$ after adding noise and find the minimum eigenvalue of $\mathbf{H_1}$.

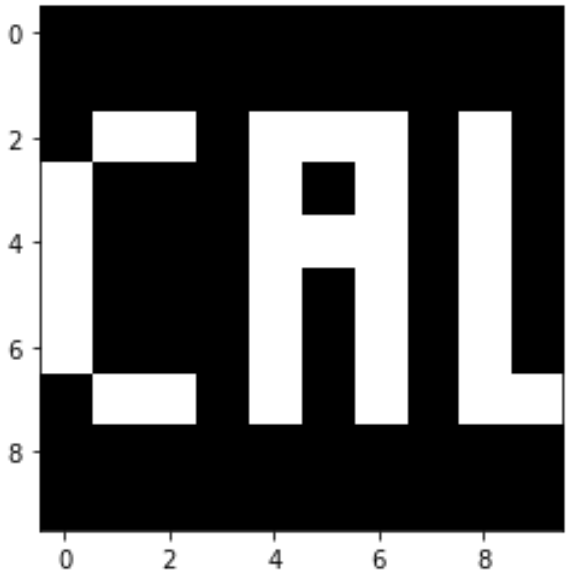
In [7]:

```
x1 = np.linalg.inv(H1).dot(b1)
eigenvalues1 = np.linalg.eig(H1)[0]
print("Is the matrix invertible?", abs(np.linalg.det(H1)) > 0.5)
print("The smallest eigenvalue is:", min(np.absolute(eigenvalues1)))
print("Number of eigenvectors:", len(eigenvalues1))
plt.imshow(x1.reshape(10,10), cmap='gray')
```

```
Is the matrix invertible? True
The smallest eigenvalue is: 1.0
Number of eigenvectors: 100
```

Out[7]:

```
<matplotlib.image.AxesImage at 0x1119f0ed0>
```



Now let's compute $\vec{x}_2$ and find the minimum eigenvalue of $\mathbf{H_2}$.

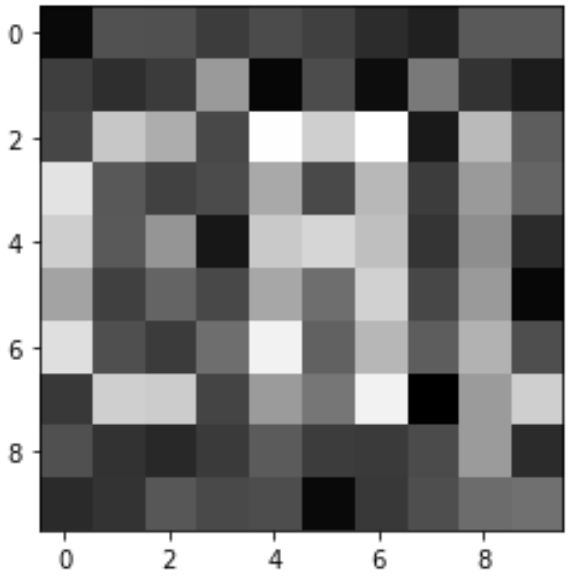
In [8]:

```
x2 = np.linalg.inv(H2).dot(b2)
eigenvalues2 = np.linalg.eig(H2)[0]
print("Is the matrix invertible?", abs(np.linalg.det(H2)) > 0.5)
print("The smallest eigenvalue is:", min(np.absolute(eigenvalues2)))
print("Number of eigenvectors:", len(eigenvalues2))
plt.imshow(x2.reshape(10,10), cmap='gray')
```

```
Is the matrix invertible? True
The smallest eigenvalue is: 0.29516363308629756
Number of eigenvectors: 100
```

Out[8]:

```
<matplotlib.image.AxesImage at 0x111b5c210>
```



Now let's compute $\vec{x}_3$ and find the minimum eigenvalue of $\mathbf{H_3}$.

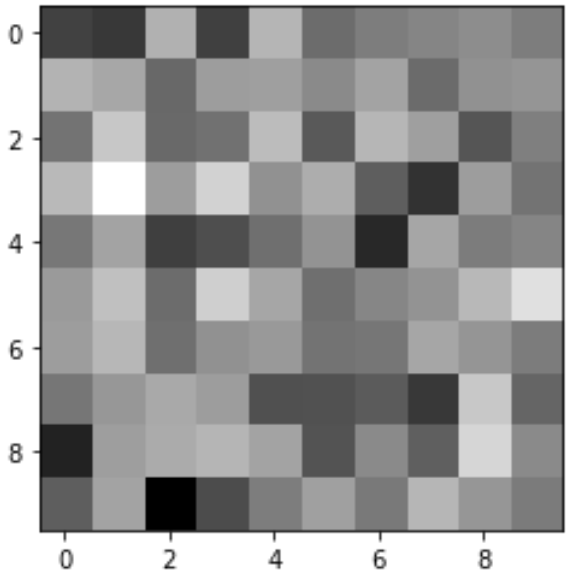
In [9]:

```
x3 = np.linalg.inv(H3).dot(b3)
eigenvalues3 = np.linalg.eig(H3)[0]
print("Is the matrix invertible?", abs(np.linalg.det(H3)) > 0.5)
print("The smallest eigenvalue is:", min(np.absolute(eigenvalues3)))
print("Number of eigenvectors:", len(eigenvalues3))
plt.imshow(x3.reshape(10,10), cmap='gray')
```

```
Is the matrix invertible? True
The smallest eigenvalue is: 1.2184217512913978e-05
Number of eigenvectors: 100
```

Out[9]:

```
<matplotlib.image.AxesImage at 0x111c92550>
```



Problem 6: Page Rank

In [ ]:

```
# Though it is not required you may use iPython for your calculation
s in parts (c) and (g)
```