Shivani Patel 3033800943 02/28/2020

1. Mechanical Determinants

1a) $\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$ det $= 6-0 \Rightarrow 6 \neq 0$ invertible

1b) $\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$ det $= 6-0 \Rightarrow 6 \neq 0$ invertible

1c) $\begin{bmatrix} 6 & 9 \\ 4 & 6 \end{bmatrix}$ det $= 36-36 = 0$, not invertible

2. The Dynamics of Romeo and Juliet's Love Affair

2a) $A\vec{v} = \lambda$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \lambda_1 \begin{bmatrix} a+b \\ c+d \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$ scalar multiple of $g\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$$\begin{bmatrix} a-\lambda & b \\ c & d-\lambda \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0 \Longrightarrow \begin{bmatrix} a-\lambda+b \\ c+d-\lambda \end{bmatrix} = 0 \quad \boxed{\lambda_1 = a+b}$$

$$A\vec{v}\lambda \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} b \\ -c \end{bmatrix} = \lambda_2 = \begin{bmatrix} ab-bc \\ bc-dc \end{bmatrix} - \begin{bmatrix} b(a-c) \\ c(b-d) \end{bmatrix} - \begin{bmatrix} b \\ -c \end{bmatrix} \quad a+b=c+d$$

$$\begin{bmatrix} a-\lambda & b \\ c & d-\lambda \end{bmatrix}\begin{bmatrix} b \\ -c \end{bmatrix} = 0 \longrightarrow \begin{bmatrix} b(a-\lambda)-bc \\ bc-c(d-\lambda) \end{bmatrix} = 0 \quad \begin{matrix} a+b=c+d \\ (a-c)+(-b+d) \end{matrix} \left[\begin{array}{cc|c} (a-d) & -c & 0 \\ b & -(d-\lambda) & 0 \end{array}\right]$$

2b) $\begin{bmatrix} 0.75-\lambda & 0.25 \\ 0.25 & 0.75-\lambda \end{bmatrix}$

λ_1:
$$\left[\begin{array}{cc|c} -0.25 & 0.25 & 0 \\ 0.25 & -0.25 & 0 \end{array}\right]$$

λ_2:
$$\left[\begin{array}{cc|c} 0.25 & 0.25 & 0 \\ 0.25 & 0.25 & 0 \end{array}\right]$$

det $= (0.75-\lambda)(0.75-\lambda)$

$0.5625 = 1.5 + \lambda^2 - 0.0625$

$\lambda^2 - 1.5\lambda + 0.5 = 0$

$(\lambda-1)(\lambda-0.5)$

$$\boxed{\begin{matrix}\lambda_1 = 1 \\ \lambda_2 = 0.5\end{matrix}}$$

$$\boxed{\vec{v}_1 = \text{span}\left\{\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right\}} \quad \boxed{\vec{v}_2 = \text{span}\left\{\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right\}}$$

2c) Steady state of the system is the span $\left\{\begin{bmatrix}1\\1\end{bmatrix}\right\}$

2d) As the relationship continues, they will slowly fall out of love since the state will progressively reach closer to $\begin{bmatrix}0\\0\end{bmatrix}$

2e) Not sure

2f) $\begin{bmatrix}1-\lambda & 1\\ 1 & 1-\lambda\end{bmatrix}$ det $=(1-\lambda)(1-\lambda)-1$

$=\lambda^2-2\lambda=0$

$\lambda_1=0$

$\left[\begin{array}{cc|c}1 & 1 & 0\\ 1 & 1 & 0\end{array}\right]$

$\lambda(\lambda-2)$

$\lambda=0,2$

$x_1=x_2$

2g) They will love each other, from the calculation of $\vec{s}[n]$ as $n\to\infty=\begin{bmatrix}0\\0\end{bmatrix}$ (zero)

2h) They will love each other indefinitely

2i) $\lambda_1=a+b$ $\quad$ $\lambda_2=a-c$

$=1-2$ $\quad$ $=1+2$

$=-1, \vec{v}_1$ $\quad$ $=3, \vec{v}_2$

$=\begin{bmatrix}1\\1\end{bmatrix}$ $\quad$ $=\begin{bmatrix}1\\-1\end{bmatrix}$

2j)

$\vec{s}[n]=A^n\vec{s}[0]\longrightarrow\alpha\lambda_2^n\vec{v}_2\longrightarrow\alpha 3^n\vec{v}_2$

If $\alpha>0$ then n would appoach ∞ which means romeo has infinite love for juilet but she will have the opposite reaction. If $\alpha<0$ then n would approach $-\infty$ and the outcome is switched

2k) they will remain the same?

3. Noisy Images

3a) $\vec{s} = H\vec{i} + \vec{w}$

$H^{-1}\vec{s} = H^{-1}(H\vec{i} + \vec{w})$ scalar multip.

$H^{-1}\vec{s} = H^{-1}H\vec{i} + H^{-1}\vec{w}$

$H^{-1}\vec{s} = \vec{i} + H^{-1}\vec{w}$

$\vec{i} = H^{-1}\vec{s} - H^{-1}\vec{w}$

3b) A is invertible which means that 0 cannot be an eigenvalue

$A\vec{v} = \lambda\vec{v}$

$A^{-1}A\vec{v} = \lambda A^{-1}\vec{v}$

$\vec{v} = \lambda A^{-1}\vec{v}$

$A^{-1}\vec{v} = \frac{1}{\lambda}\vec{v}$

3c) iPhython code

3d) $\hat{\vec{w}} = H^{-1}\vec{w} = H^{-1}(\alpha_1\vec{b}_1 + \ldots + \alpha_N\vec{b}_N)$

$= \alpha_1 H^{-1}\vec{b}_1 + \ldots + \alpha_N H^{-1}\vec{b}_N$

$= \alpha \frac{1}{\lambda_1}\vec{b}_1 + \ldots + \alpha_N \frac{1}{\lambda_N}\vec{b}_N$

the small eigenvalues will amplify the sound, shown in lab will make the image less clear

the opposite will happen for large eigenvalues

4. Reservoirs that give & take

4a)

$$A = \begin{bmatrix} a & d & f \\ d & b & e \\ f & e & c \end{bmatrix}$$

4b)

$$\begin{bmatrix} a & d & f \\ d & b & e \\ f & e & c \end{bmatrix}\begin{bmatrix} x \\ x \\ x \end{bmatrix} = \begin{bmatrix} ax+dx+fx \\ dx+bx+ex \\ fx+ex+cx \end{bmatrix} = \begin{bmatrix} x \\ x \\ x \end{bmatrix}$$

the equilibrium state is 0

4c)

$$\begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0.4 & 0.2 & 0.4 \\ 0.4 & 0.4 & 0.2 \end{bmatrix}$$

since all pivots are not $\emptyset$ the matrix A is invertible!

$$\downarrow$$

$$\begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0 & -0.6 & -0.4 \\ 0 & -0.4 & -0.6 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0 & -0.6 & -0.4 \\ 0 & 0 & -1/3 \end{bmatrix}$$

5. power Iteration

5a) since the dominant eigenvalue is greater than one, there will be linearly independent columns, so $b^{-1} = \sum_{n}^{n} \vec{c}, \vec{v}$

5b) $\lim_{k\to\infty} \frac{1}{\lambda_1^k} A^k \vec{b} = \lim_{k\to\infty} \frac{1}{\lambda_1^k} A^k (c_1\vec{v}_1 + \vec{v}_2\vec{v}_2 + \ldots + c_n \vec{v}_n)$

$= \lim_{k\to\infty} \frac{1}{\lambda_1^k} (c_1 \lambda_1^k \vec{v}_1 + \ldots + c_n \lambda_n^k \vec{v}_n)$

$\lim_{k\to\infty} \left(c_1\vec{v}_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k \vec{v}_2 + \ldots + c_n \left(\frac{\lambda_n}{\lambda_1}\right)^k \vec{v}_n \right)$

$\rightarrow \boxed{c_1\vec{v}_1}$

5c) ?

5d) ?

8. Homework Process & Study Group

I worked on this homework for 6 hours

I did most of it on my own but had Sadia Qureshi (3034541667)

I also got an extension due to my medical injury.

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)
```

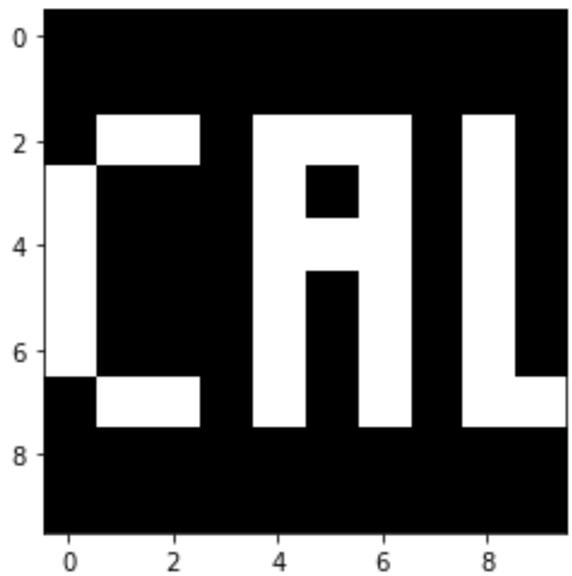
The code below displays the image.

In [3]:

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

Out[3]:

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



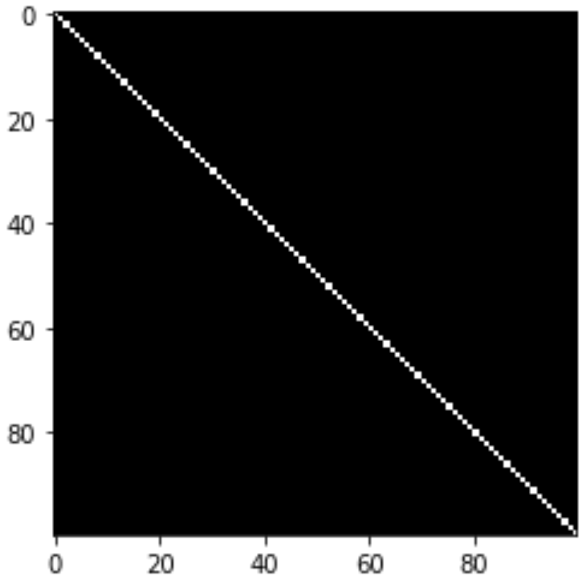
Then, lets display the set of masks

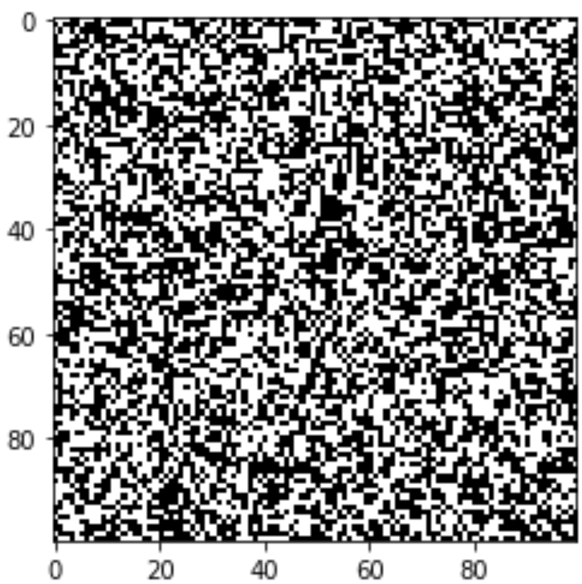
In [4]:

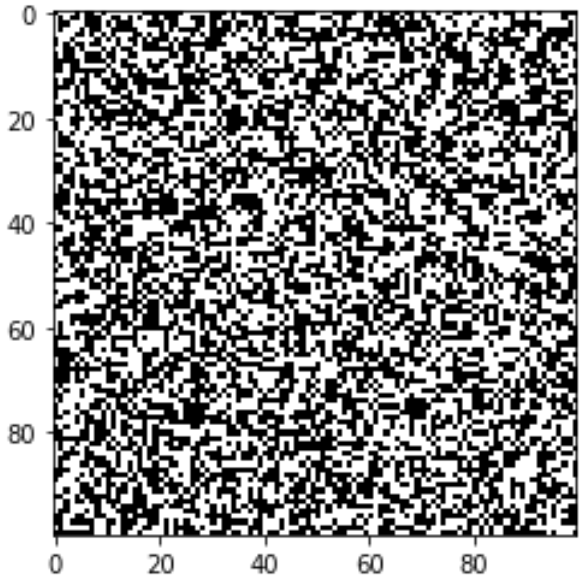
```
plt.figure(1)
plt.imshow(H1,cmap='gray')
plt.figure(2)
plt.imshow(H2,cmap='gray')
plt.figure(3)
plt.imshow(H3,cmap='gray')
```

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 $\mathbf{H_1}$.

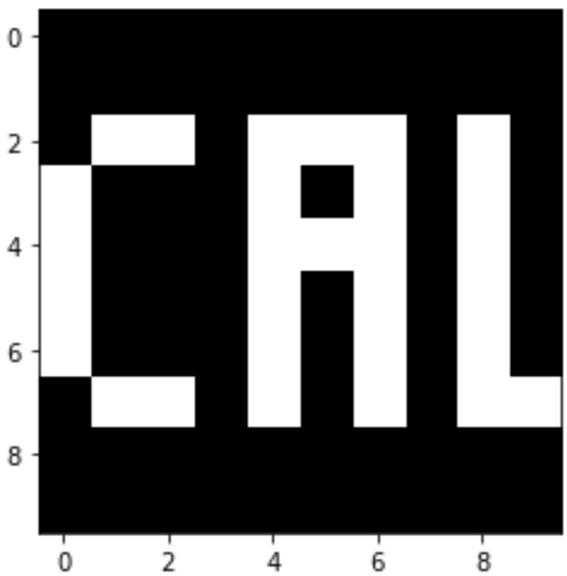
In [7]:

```
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 $\mathbf{H}_2$.

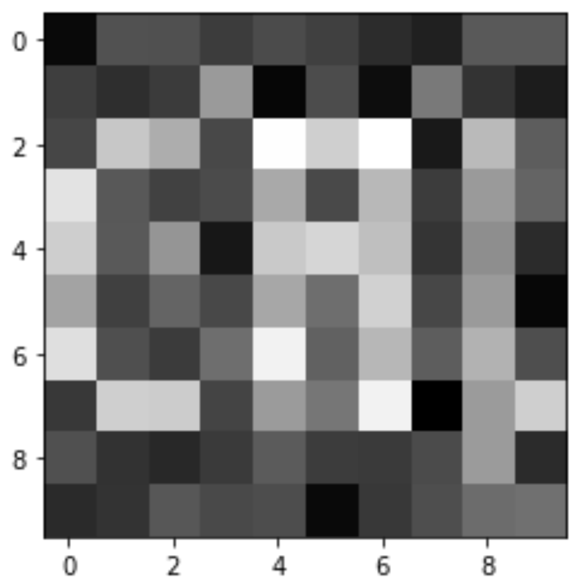
In [8]:

```
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 $\mathbf{H}_3$.

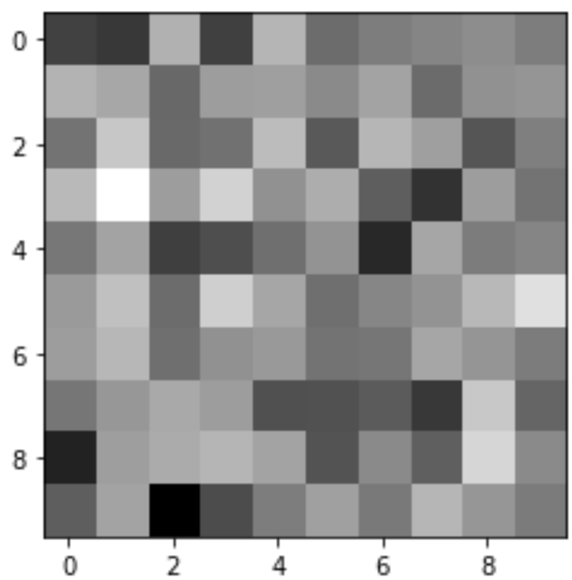
In [9]:

```
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)
```