MOL410/510 Problem Set 1 - Linear Algebra - Due Friday Sept. 30

Use lab notes to help solve these problems. Problems marked ”MUST DO” are required for full credit. For the remainder of the problems, do as many as you can/need to get above 100 points. As an incentive to do as many problems as you can, we will assign the highest score you can get from all of the non-mandatory problems you attempt; think of each extra problem you do as buying insurance.

1. **Matrix multiplication.**

   • (2 pts) Multiplying it out by hand, find $C = AB$ where

   $$A = \begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix},\ B = \begin{pmatrix} 6 & 5 \\ 3 & 2 \end{pmatrix}$$

   • (2 pts) Again by hand, find $D = BA$. Are C and D the same or different?

2. **Matrix multiplication.**

   • (2 pts) Multiplying it out by hand, find $C = AB$ where

   $$A = \begin{pmatrix} 2 & 1 \\ 4 & 3 \\ 3 & 1 \end{pmatrix},\ B = \begin{pmatrix} 6 & 5 \\ 3 & 2 \end{pmatrix}$$

   • (1 pt) Does BA exist?

3. **Matrix multiplication.**

   • (2 pts) Multiplying it out by hand, find $C = AB$ where

   $$A = \begin{pmatrix} 3 & -3 & 2 \\ 1 & 5 & -1 \\ -4 & -6 & 2 \end{pmatrix},\ B = \begin{pmatrix} 8 \\ 1 \\ 4 \end{pmatrix}$$

   • (2 pts) Now multiply $D = B^{\mathrm{T}} A^{\mathrm{T}}$

   $$A^{\mathrm{T}} = \begin{pmatrix} 3 & 1 & -4 \\ -3 & 5 & -6 \\ 2 & -1 & 2 \end{pmatrix},\ B^{\mathrm{T}} = \begin{pmatrix} 8 & 1 & 4 \end{pmatrix}$$

   what is the difference between C and D?

4. **Dot Product.** Let $\mathbf{x}^{\mathrm{T}} = (1, 3)$ and $\mathbf{y}^{\mathrm{T}} = (0, 2)$.

   • (1 pt) What is their dot product $\mathbf{x}^{\mathrm{T}} \cdot \mathbf{y}$?

   Now consider the rotation matrix

$$R = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

   • (2 pts) For $\theta = 30°$, find $\mathbf{x}' = R\mathbf{x}$ and $\mathbf{y}' = R\mathbf{y}$, compute the dot product between $\mathbf{x}'$ and $\mathbf{y}'$, and compare it to the original before rotation.

   • (3pts) Now do the same but for an arbitrary angle θ.

5. **Dot product and a non-orthonormal base transform.** Once again, as in problem 4, let $\mathbf{x}^{\mathrm{T}} = (1, 3)$ and $\mathbf{y}^{\mathrm{T}} = (0, 2)$, and find their dot product $\mathbf{x}^{\mathrm{T}} \cdot \mathbf{y}$.

   • (2 pts) Now find $\mathbf{x}'$ and $\mathbf{y}'$, which are $\mathbf{x}$ and $\mathbf{y}$ but expressed in the basis given by $\mathbf{b_1}^{\mathrm{T}} = (1/\sqrt{2}, 1/\sqrt{2})$ and $\mathbf{b_2}^{\mathrm{T}} = (0, 1)$.

   • (3 pts) What is the dot product of $\mathbf{x}'$ and $\mathbf{y}'$, and how does it compare to the original? Comment on what you found.

6. **Matrix multiplication.** It has been proposed that integration over time, in the sense of calculus, is an important operation carried out in many systems of the brain (e.g., Major and Tank, 2004). In one well-known example, it is thought that decisions are formed after integrating evidence over time (Gold and Shadlen, 2007). Let the vector $\mathbf{v}$ represent a time series; that is, the i^{th} element of $\mathbf{v}$, v_i, represents the input to a neural system at the i^{th} time step.

   • (5 pts) show that the integral of $\mathbf{v}$ over time, namely $s_i = \Sigma_{j \leq i} v_i$, can be computed by $\mathbf{s} = L\mathbf{v}$ where L is a matrix that has ones on the diagonal and in every element to the lower left of the diagonal, but zeroes everywhere else.

7. **Matrix multiplication. MUST DO.** Take the matrices

$$M = \begin{pmatrix} a1 & b1 & c1 \\ a2 & b2 & c2 \\ a3 & b3 & c3 \end{pmatrix} \qquad \Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

   • (5 pts) Compute $M\Lambda$. How would you describe each column?

• (5 pts) Now suppose that you had N equations of the form $M\mathbf{v}_i = \lambda_i \mathbf{v}_i$, where i runs from 1 to N, and N is the number of rows and columns of M. Write the set of vectors $\mathbf{v}_i$ into a matrix

$$V = \begin{pmatrix} | & | & | \\ \mathbf{v_1} & ... & \mathbf{v_N} \\ | & | & | \end{pmatrix}$$

and then show that the set of equations can be written in a single matrix equation

$$MV = V\Lambda$$

where Λ is a matrix with $\lambda_1, \lambda_2, ...\lambda_N$ as the diagonal entries and zeros everywhere else.

Keep this problem in mind; the result will be useful for the lectures on eigenvalues and eigenvectors.

8. **Inverses and matrix multiplication. MUST DO** Take the matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

• (3 pts) show that

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

• (2 pts) What happens when $ad = bc$?

• (5 pts) Show that when $ad = bc$, the two vectors formed by the rows of A are parallel to each other; show also that the two vectors formed by the columns of A are parallel to each other. This means that neither the rows nor the columns can form the basis to describe a full 2-d space.

9. **Inverses.** (3 pts) Show that $(AB)^{-1} = B^{-1}A^{-1}$.

10. **Changing bases. MUST DO** You have a 3-dimensional vector $\mathbf{x}$, represented in Cartesian basis coordinates as

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Now suppose that you want to change the basis in which this vector is represented to the three axes defined by vectors $\mathbf{b_1}$, $\mathbf{b_2}$, and $\mathbf{b_3}$, such that

$$\mathbf{x} = \mathbf{b_1} y_1 + \mathbf{b_2} y_2 + \mathbf{b_3} y_3$$

where y_1, y_2, and y_3 are scalars.

• (5 pts) Arrange y_1, y_2 and y_3 into a vector $\mathbf{y}$, and write a matrix equation that solves for $\mathbf{y}$.

11. **Solving linear equations. MUST DO**

    • (5 pts) Using MATLAB or by hand using matrix inversion, solve the following set of algebraic equations.

$$\begin{array}{rcrcl} 2x & + & 3y & = & 20 \\ x & - & y & = & 4 \end{array} \tag{1}$$

12. **Solving a set of linear equations. MUST DO** You're studying a simple network of linear neurons. The network has two layers, an input layer e and an output layer o. Neuron o_1 has a firing rate that is twice that of neuron e_1 plus three times that of e_2. Neuron o_2 has a firing rate that is the firing rate e_1 minus the firing rate of e_2. It turns out that the input layer e is very hard to access. So you put electrodes in the output layer, and find that o_1 is firing at 20 spikes/sec, and o_2 is firing at 4 spikes/sec.

    • (3 pts) Use matrix notation and methods to answer. What is the firing rate of the e neurons?

13. **Index notation. MUST DO** In index notation, we use A_{ij} to mean the element in the i^{th} row, j^{th} column (counting from the top left) of the matrix A.

    The "trace" of a matrix is the sum of its diagonal elements: $\mathrm{Tr}(A) = \Sigma_i A_{ii}$.

    • (5 pts) Use index notation to show that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$.

    As you will see in the next lecture, most matrices M can be written as $M = VDV^{-1}$, where D is a matrix that has non-zero entries only on its diagonal, i.e., all its off-diagonal entries are zero. (D is therefore called a "diagonal" matrix.)

    • (3 pts) Show that $\mathrm{Tr}(M) = \mathrm{Tr}(D)$.

    • (2 pts) Show that $\det M = \det D$.

    You just proved a very big deal, since you proved the last two bullet points in general, for *any* matrix M that can be written as $M = VDV^{-1}$ (and that's most of 'em).

14. **Rotation matrices. MUST DO** In two-dimensions a rotation matrix can be written as

$$R = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

• (2 pts) Show that the dot product of the two vectors formed by the rows is zero.

• (2 pts) Show that the dot product of the two vectors formed by the columns is zero.

• (2 pts) The transpose of R can be written

$$R^{\mathrm{T}} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Calculate $R^{\mathrm{T}}R$.

• (4 pts) Now consider a general column vector $\mathbf{x} = (x_1 \;\; x_2)^{\mathrm{T}}$. Show that the vector $\mathbf{x}' = R\mathbf{x}$ is the same length as $\mathbf{x}$.

15. **Eigenvalues and eigenvectors. MUST DO.** For $\mathbf{A} = \begin{pmatrix} 4 & -2 \\ 1 & 1 \end{pmatrix}$

• (2 pts) Show that $\vec{x} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ is an eigenvector. What is its associated eigenvalue?

• (3 pts) Find the second eigenvector and eigenvalue.

16. **Eigenvalues and eigenvectors. MUST DO** Find the eigenvalues and eigenvectors for the following three matrices:

• (5 pts) $\begin{pmatrix} 3 & 2 \\ 3 & -2 \end{pmatrix}$

• (5 pts) $\begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix}$

• (5 pts) $\begin{pmatrix} 2 & -3 & 1 \\ 1 & -2 & 1 \\ 1 & -3 & 2 \end{pmatrix}$.

For the the 3x3 matrix, use the Matlab function called eig to obtain the answer.

• (5 pts) For each of the three matrices, calculate the trace and determinant. Show that each matrices' determinant is equal to the product of its eigenvalues and that each matrices' trace is equal to the sum of its eigenvalues. For the 3x3 matrix, you can use the functions $\det$ and trace.

17. **Eigenvectors of symmetric matrices.** Imagine a symmetric matrix $A = A^{\mathrm{T}}$ and that A is diagonalized by the matrix V, that is $A = VDV^{-1}$.

    - (3 pts) Show that V is orthonormal, i.e. $V^{\mathrm{T}}V = I$, where I is the identity matrix.

18. **Eigenvectors of symmetric matrices.** For the matrix

$$A = \begin{pmatrix} 9 & 3 & 6 \\ 3 & 7 & 4 \\ 6 & 4 & 3 \end{pmatrix}$$

    - (2 pts) Use Matlab to calculate the eigenvectors and show that they are orthonormal.

19. **Matlab practice. (OPTIONAL)** This is in tutorial format. There's little to do but follow my instructions. Hopefully, you will begin to intuit some Matlab concepts as you follow along. There are a few follow-ups at the end.

Start with the matrix

$$A = \begin{pmatrix} 1.5 & -0.4 \\ -0.7 & 1 \end{pmatrix}$$

Let's visualize the operation of this matrix on different points of space as you saw in lecture.

• First, let's create a set of points that are equally spaced on a grid. We will use the Matlab function called meshgrid:

```
[X,Y]=meshgrid(-1:0.2:1,-1:0.2:1);
```

This generates a matrix of points with pairs of points from -1 to 1 separated by 0.2 in each direction. To see this, type

```
plot(X,Y,'.r');
```

The '.r' tells Matlab to plot red dots with no lines attaching them.

• Right now, X and Y are aligned in a square matrix. We want to align them as column vectors in a matrix, where the first row contains all the x-values of the vectors and the second row contains all the y-values of the vectors. To do this, type:

```
Z=[X(:)'; Y(:)'];
```

Let's take this apart: The square brackets tell Matlab that everything inside is to be put into a matrix. The X(:) tells Matlab to take all the values inside the matrix X and turn it into a column vector. The apostrophe after that says to take the transpose of that column vector so that it is now a row vector. The semicolon inside the brackets tells Matlab to start a new row in the Matrix, and the second row holds the row vector of values in Y. Since the points were aligned together in the matrices X and Y, they are now aligned in the matrix Z. To see this do:

```
plot(Z(1,:),Z(2,:),'.r')
```

• Let's transform the data by the matrix A. First define the matrix in Matlab:

```
A=[1.5 -0.4; -0.7 1];
```

Where, again, the square brackets tell Matlab to put the items inside a matrix, and the semicolon inside the brackets says start a new row. Now, multiply the grid of points stored in Z by the matrix A:

```
Q=A*Z;
```

and plot the new points stored in Q (you should be able to figure this out on your own now).

• Let's plot lines that go from the old points to the new points and cap each with a dot on the new point. Fortunately, in Matlab, this is readily done in a single line.

```
plot([Z(1,:); Q(1,:)],[Z(2,:); Q(2,:)],'b',Q(1,:),Q(2,:),'.r')
```

To understand this code, let's first focus on the first set of brackets. Z(1,:) tells Matlab to give me all of the elements in the first row of the matrix Z, just as Q(1,:) tells Matlab to give me all of the elements in the first row of the matrix Q. By putting them inside the square brackets and separating them with the semicolon, I have created a new matrix that has all of the x-values of the old points, Z(1,:), and the x-values of the new points, Q(1,:). These are the x-values of the lines we created. Similarly, the second square brackets holds the y-values of the old and new points, notice that we used the second row of the matrices Z and Q. Why did Matlab plot individual lines for every column in the square brackets? That's just part of Matlab's syntax; every column in a a matrix is treated as a separate curve.

What about the second part of the plot command? There we are plotting the grid of points, just as we did earlier. Matlab's plot command can produce several curves on the same graph by producing triplets of inputs in the format plot(x1,y1,'k',x2,y2,'.r',x3,y3,'–b'). There are other ways of getting Matlab to plot multiple curves on the same graph. We will meet them next.

• Now, let's do an animation! We want to animate the points starting from their old points and moving to their new points in little increments. In order to move the points smoothly between the starting points to the ending points, we need to take the nth root of the matrix A. How do we do that? Remember in class that we learned that we can do it using eigenvectors and eigenvalues. So let's find that now for the matrix A.

```
[V,D]=eig(A);
```

Since $A = VDV^{-1}$, we also need the inverse of V. To do that, we can divide the identity matrix by V. In Matlab

```
IV=eye(2)/V;
```

where the function eye(2) creates a 2x2 identity matrix. Keep these values for later.

Let's generate a new figure, clear it, and place some points that will be in our animation. First the figure and clearing:

```
figure(10); clf;
```

This line makes a figure with 10 as its "handle" and clears all the contents of it.

Now, let's preallocate a set of handles for each dot in the plot.

```
h=zeros(1,size(Z,2));
```

Handles are a way to adjust plots after they have been created. They are, in fact, more general than that, but that's beyond the scope of this exercise. For now, think of them as pointers to the graphs we will make. Right now, they are pointing to nothing. We have to actually set their values so that they point to some curves. So, let's do some plotting. We could use the plot command as before, but we're going to use a for-loop instead. It will make our code clearer.

```
for k=1:size(Z,2)
    h(k)=line(Z(1,k),Z(2,k),'color','r','marker','.','linestyle','none');
end
```

This code snippet runs a for-loop over the index k for each column that is in Z. The size is a Matlab function that returns the number of elements in each dimension of a matrix. The second input says give me the number of columns. The function line is similar to plot, but it does not erase over what was previously in the plot. It also requires more specific output, no shortcuts like '.r' anymore; you have to specify 'color','r', etc.

Now, let's define the number of steps in our animation, and define our axis limits and make the graph square.

```
N=100; axis(2*[-1 1 -1 1],'square');
```

Now animate:

```
for n=1:N
    B=V*(D.^(n/N))*IV;
    z=B*Z;
    for k=1:size(Z,2)
        set(h(k),'XData',z(1,k),'YData',z(2,k));
    end
    drawnow
end
```

Explanation: the for-loop runs from values of n going from 1 to 100. The first line inside the for-loop defines a new matrix B which is the (n/100)th power of A (where we are using the eigenvalues and eigenvectors to do the calculation). The vector z are the new points of Z when transformed by B. The inner for-loop replaces the data in each dot with the new transformed data. The handles give us access to all sorts of properties of each dot, including its x- and y-data! Finally, the drawnow tells Matlab to refresh the graph right now; otherwise, Matlab will only do it once at the end.

Great, now try these.

- Instead, of using grid points that run from -1 to 1, do it so that they run from -2 to 2. Produce your animation again using your new set of grid points.

- Now, when the animation concludes, draw blue lines on the figure that go from the starting point to the end point.