# Linear Algebra Project #1: Mathematica and Matrices - Arnav Dani

Directions: I encourage you to use Mathematica (MM) as much as possible in this assignment. There may be times when you will be better served resorting to paper and pen, but in reality, this entire assignment can be done in MM. Of course, if individual instructions say to do something in MM you have to do that. And if you do some work by hand, which you will have to, you will have to transfer that into this notebook, by typing it in manually. Select the cell bracket of each answer and go to Format + Background Color + Light Green. This will be submitted to Schoology.

## Working with matrices in MM

Evaluate each cell below (one at a time) and pay attention to the output. Note that the first cell is just defining the matrices; you won't see any output, yet. The output comes in the cells after this. However, look at how each matrix/vector was defined in that first cell. That is what's important.

In[∘]:=
```
matformA = {{1, 2}, {3, 4}} // MatrixForm;
matA = {{1, 2}, {3, 4}};
vecformB = {5, 6} // MatrixForm;
vecB = {5, 6};
```
insmatA = $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$;

```
(*created using Insert + Matrix. The same is true for insvecB, below*)
```
insvecB = $\begin{pmatrix} 5 \\ 6 \end{pmatrix}$;

In[∘]:=
```
matformA
```
Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

In[∘]:=
```
vecformB
```
Out[∘]//MatrixForm=

$$\begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

In[∘]:=
```
matA
```
Out[∘]= {{1, 2}, {3, 4}}

In[∘]:=
```
vecB
```
Out[∘]= {5, 6}

In[ ]:=
```
matformA.vecformB // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

In[ ]:=
```
matA.vecB // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 17 \\ 39 \end{pmatrix}$$

In[ ]:=
```
insmatA.insvecB // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 17 \\ 39 \end{pmatrix}$$

Lesson to learn: If you want MM to work with, or do anything with a matrix, do NOT define the matrix in MatrixForm. You can make it look like that IF you use the Insert + Matrix, though. It will then look like a matrix that you're used to, but that's not how MM actually sees it.

## How to create a random integer on the interval [*m*, *n*]. The command is pretty simple. It is:  RandomInteger[{*m*, *n*}]. Below I am creating 100 random integers between –5 and 5.

In[ ]:=
```
Table[RandomInteger[{-5, 5}], {k, 1, 100}]
```

### Note: as you do these problems below I should be able to see your use of MM. That is, I need to be able to confirm that you used MM in the way that I am instructing and not making up your own stuff.

1. Create a random 3 *x* 3 upper triangular matrix whose entries on, and below the main diagonal, are integers between –5 and 5, inclusive.  If **any** main diagonal entry turns out to be zero, create a new random matrix. You cannot do this problem with a 0 on the main diagonal. Define this matrix as matA and display it in matrix form. Note: Do NOT **define** matA in matrix form, as MM will not operate with it. Note: An upper triangular matrix is a matrix that has all zeroes below the main diagonal.

```
randint := RandomInteger[{-5, 5}];
```

In[ ]:=
```
matA = {{randint, randint, randint}, {0, randint, randint}, {0, 0, randint}}
```

Out[ ]=
```
{{4, -5, -4}, {0, -3, 5}, {0, 0, 1}}
```

2. Create a random 3 *x* 1 vector whose entries are between –10 and 10. Define this vector as xVec and display it as a column vector.

In[ ]:=
```
randint2 := RandomInteger[{-10, 10}];
```

In[ ]:=
```
xVec = {randint2, randint2, randint2};
xVecForm = xVec // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 5 \\ 2 \\ -7 \end{pmatrix}$$

3. Use MM to compute matA∗xVec. Display your answer as a column vector.

In[ ]:=
```
prodVec = matA.xVec;
prodVecDisplay = prodVec // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 38 \\ -41 \\ -7 \end{pmatrix}$$

4. Create another random 3 $x$ 1 vector whose entries are between −15 and 15. Define this vector as bVec and display it as a column vector. Ib, by some chance (1 in 29791) you get all 0’s, generate a new vector...we don’t want all 0’s.

In[ ]:=
```
randint3 := RandomInteger[{-15, 15}];
bVec = {randint3, randint3, randint3};
bVecForm = bVec // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 6 \\ -2 \\ -4 \end{pmatrix}$$

5. Now consider the equation matA∗$\boldsymbol{x}$ = $\boldsymbol{b}$, where $\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ and $\boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$. Find $\boldsymbol{x}$ in terms of $\boldsymbol{b}$. Your answer should look something like

$\boldsymbol{x} = \begin{pmatrix} \frac{1}{2} b_1 + 5 b_2 + 4 b_3 \\ -3 b_2 - 4 b_3 \\ 2 b_3 \end{pmatrix}$. Note: I just made that up. Finally, write $\boldsymbol{x}$ as the product of a 3 $x$ 3 matrix, which you will define as matB, and the vector $\boldsymbol{b}$. That is, $\boldsymbol{x}$ = matB.$\boldsymbol{b}$

```
b = {b1, b2, b3};
matX = LinearSolve[matA, b] // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} \frac{1}{12} \times (3\,b1 - 5\,b2 + 37\,b3) \\ \frac{1}{3}\,(-b2 + 5\,b3) \\ b3 \end{pmatrix}$$

```
(*take the coefficients from x to find matB *)

matB = {{3 / 12, -5 / 12, 37 / 12}, {0, -1 / 3, 5 / 3}, {0, 0, 1}};
matX2 = matB.b // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} \frac{b1}{4} - \frac{5\,b2}{12} + \frac{37\,b3}{12} \\ -\frac{b2}{3} + \frac{5\,b3}{3} \\ b3 \end{pmatrix}$$

6. Find the vector ***x*** that solves matA * ***x*** = bVec using what you did in #5, and display your answer as a column vector.

In[ ]:=
```
xVec = LinearSolve[matA, bVec];
xVecDisplay = xVec // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} -10 \\ -6 \\ -4 \end{pmatrix}$$

7. Find matA*matB and matB*matA.

In[ ]:=
```
matA.matB // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

In[ ]:=
```
matB.matA // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
(*both are identitify matrices*)
```

8. Consider the system $\begin{pmatrix} 2 & 2 & 0 \\ 1 & 2 & 1 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$.

i. Write the augmented matrix first and then apply Gauss-Jordan elimination and display this in matrix form. You are going to have to do this by hand.

```
(*transfering the hand written GJ elimination*)
step1 = {{2, 2, 0, b1}, {1, 2, 1, b2}, {1, 0, -1, b3}} // MatrixForm;
(* R1/2 → R1*)
step2 = {{1, 1, 0, b1 / 2}, {1, 2, 1, b2 }, {1, 0, -1, b3 }} // MatrixForm;
(* -R1 + R2 → R2, -R1 + R3 → R3*)
step3 =
  {{1, 1, 0, b1 / 2}, {0, 1, 1, b2 - b1 / 2}, {0, -1, -1, b3 - b1 / 2 }} // MatrixForm;
(* R3 + R2 → R3*)
step4 = {{1, 1, 0, b1 / 2}, {0, 1, 1, b2}, {0, 0, 0, b3 + b2 - b1}} // MatrixForm;
(* R3 * 1 / (b3 + b2 - b1) → R3 *)

(*displaying all the steps*)
step1
step2
step3
step4
```

Out[127]//MatrixForm=

$$\begin{pmatrix} 2 & 2 & 0 & b1 \\ 1 & 2 & 1 & b2 \\ 1 & 0 & -1 & b3 \end{pmatrix}$$

Out[128]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 0 & \frac{b1}{2} \\ 1 & 2 & 1 & b2 \\ 1 & 0 & -1 & b3 \end{pmatrix}$$

Out[129]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 0 & \frac{b1}{2} \\ 0 & 1 & 1 & -\frac{b1}{2} + b2 \\ 0 & -1 & -1 & -\frac{b1}{2} + b3 \end{pmatrix}$$

Out[130]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 0 & \frac{b1}{2} \\ 0 & 1 & 1 & b2 \\ 0 & 0 & 0 & -b1 + b2 + b3 \end{pmatrix}$$

ii. Use a result from part (a) to tell which of the following vectors ***b*** will the system be consistent. Explain your answers.

$a. \begin{pmatrix} 5 \\ 9 \\ -4 \end{pmatrix}$ $b. \begin{pmatrix} 1 \\ -7 \\ -8 \end{pmatrix}$ $c. \begin{pmatrix} 14 \\ 5 \\ 9 \end{pmatrix}$ $d. \begin{pmatrix} -2 \\ -1 \\ 3 \end{pmatrix}$ $e. \begin{pmatrix} -3 \\ -3 \\ 0 \end{pmatrix}$

```
a = {{5}, {9}, {-4}};
b = {{1}, {-7}, {-8}};
c = {{14}, {5}, {9}};
d = {{-2}, {-1}, {3}};
e = {{-3}, {-3}, {0}};

Print["\nA:"]
Print[-a[[1]] + a[[2]] + a[[3]] === {0}];
Print["\nB:"]
Print[-b[[1]] + b[[2]] + b[[3]] === {0}];
Print["\nC:"]
Print[-c[[1]] + c[[2]] + c[[3]] === {0}];
Print["\nD:"]
Print[-d[[1]] + d[[2]] + d[[3]] === {0}];
Print["\nE:"]
Print[-e[[1]] + e[[2]] + e[[3]] === {0}]
```

```
A :
True

B:
False

C:
True

D:
False

E:
True
```

I used the last line in the reduced matrix to find which of the following sets of constants would result in consistent systems. To do that, I checked whether b2 + b3 - b1 was zero for all 5 options. It was for options A, C, and E, showing that the matrix was consistent for those sets of constant values. It was not for B and D meaning that with those constants the matrices would be inconsistent.