

In this assignment you will be writing a function file that row-reduces a matrix (“performs Gaussian elimination”) until the matrix is in upper triangular form. As before, we’ll take it a step at a time. Initially you will write a script file that row reduces the specific  $4 \times 4$  matrix

$$A = \begin{bmatrix} 2 & -5 & 1 & -3 \\ 3 & 0 & 9 & -4 \\ -2 & 7 & -2 & 6 \\ 1 & -1 & 11 & -1 \end{bmatrix}$$

1. (a) Start a new script file named *yourfirstname\_rr1* whose first line is `pause on;`.
  - (b) Enter the matrix in the command window and copy it into a new script as well. In the script window, define a new matrix *B* by setting it equal to *A*. Work with the matrix *B* for the rest of the time, in order to preserve *A*. You will want to rename it in the command window as well.
  - (c) In the command window, write a line of code that replaces the second row of the matrix (*B*) with the second row minus a fraction multiple of the first row, as described in class yesterday, so that the new (2,1) entry is zero. If/when it works, copy it into your script file.
  - (d) Repeat step (b) to get zeros in the remaining entries in column one. Each time one works in the command window, add that line to your script. When these are all done, add a line with the command `pause`.
  - (e) Now you will need to get zeros in the second column, in all entries below the (2,2) entry. Replace the third row with itself minus the appropriate fractional multiple of the second row in order to get zero in the (3,2) entry. Repeat for the fourth row and add another `pause`.
  - (f) Now you need one more line to get another zero in the fourth row and you’ll be done. Save it and e-mail it to me - **due at 4 PM Monday, May 5th**
2. (a) Rename your script file as *yourfirstname\_rr2*. Replace the numbers in each of your fraction multipliers with the corresponding locations in the matrix. (So each number will become `B(m,n)` for some values of *m* and *n*.
  - (b) Write a for loop to execute each section of code between the pauses as a loop with only one line of code within the loop.
  - (c) Change your file to a function file that reads in a  $4 \times 4$  matrix and row reduces it until it is in upper triangular form. You will need to remove where you have entered the specific matrix within your file. To test it, enter that matrix in the command window and try running your file as a function file now. E-mail the (hopefully working) result to me **by 4 PM Tuesday, May 6th**
3. Save the previous script file as *yourfirstname\_rr3*. Then make the following two changes:
  - (a) Your file has three loops, all with almost the same line in each. If you look carefully at the lines of code and the starting points of the for loops, you will see the opportunity for another counter, from two to four. Figure out how to incorporate another for loop to reduce your three lines of code to one. Test to make sure it works.
  - (b) Note that your code should work for any matrix whose number of columns is at least the same as the number of rows; the rows are the critical part. Modify your code to read in and row-reduce a matrix of any size, as long as it still meets this condition that the number of columns is at least the number of rows. Test to make sure it works and **e-mail it to me by 4 PM on Friday, May 9th.**