Computer Project:

1. Reduced Echelon Form and \texttt{rref}

\textbf{Purpose:} To calculate the reduced echelon form by hand and with \texttt{rref}, and to see some effects of roundoff error.
\textbf{Prerequisite:} Row reduction and echelon forms
\textbf{MATLAB functions used:} -, \texttt{\textbackslash}; \texttt{\textbackslash{}rref}.

\textbf{Background:} Read about elementary row operations and reduced echelon form in Section 1.2. To learn about the functions from Lay's Toolbox, use \texttt{help} or see his Student Study Guide.

1. Define the two matrices \( A \) and \( B \) below. For each of them, first calculate the reduced echelon form by hand, then use MATLAB's function \texttt{rref} to calculate the reduced echelon form again.

\textbf{NOTE:} This function is in the Lay toolbox, and these files need to be available to Matlab path when you type the \texttt{rref} command. See sections 15 and 16 of the previous extra credit assignment regarding how this should be done.

(a) (hand) The matrix \( A \) below is Example 2 in Section 1.2, where an echelon form is calculated. Repeat here the row operations done in the text, and then finish calculating by hand its reduced echelon form. Show all steps:

\[
A = \begin{bmatrix}
0 & -3 & -6 & 4 & 9 \\
-1 & -2 & -1 & 3 & 1 \\
-2 & -3 & 0 & 3 & -1 \\
1 & 4 & 5 & -9 & -7
\end{bmatrix}
\]
b) (MATLAB) Type `rref(A)`. Is the output identical to what you obtained above?  
(If not, redo hand calculations.)

2. Let \( B = \begin{bmatrix} -0.1 & 0.1 & 2 \\ 0.3 & 0.2 & 0.7 \\ 0 & 0.5 & 6.7 \end{bmatrix} \). (a) (hand) Calculate the reduced echelon form of \( B \).  
Show all steps. Hint: do all scaling at the end.

(b) (MATLAB) Type `rref(B)`. Is the output identical to what you obtained above?  
(If not, redo hand calculations.)

**Remarks.** For the vast majority of matrices \( M \), `rref(M)` will return a very accurate result, as it does in the two examples above. One of the reasons `rref` is so accurate is that it does not pivot on a position where the value is extremely small. This is because roundoff error in computer arithmetic can cause a very small nonzero number to show up in a position where theoretically there should be a true zero, and if you ask MATLAB to search in the usual way for the next pivot, then it might find such a position and end up producing the wrong answer.

Thus it is wise to check the size of numbers and not to use one as a pivot if it is too small. The function `rref` uses the value in a variable called `tol` for this purpose.

When the user does not specify a value for `tol`, then `rref` gives it a default value (and that is what happened above). The precise way `tol` is actually used in `rref` is this: before each pivot step, if the absolute value of any matrix entry \( M(i,j) \) is smaller than `tol`, then `rref` sets that entry to zero.
3. (MATLAB) Use the same matrix $B$ as in question 2. Here you will force `rref(B)` to return the wrong answer by making the value of $\text{tol}$ too small; and you will do experiments to figure out a bound on the default value of $\text{tol}$ (i.e., the value of $\text{tol}$ in Matlab if its value is not specified by the user).

(a) Type each of the following lines and record the result. Note the second entry after `rref` below specifies the value of $\text{tol}$ (and replaces the default value of $\text{tol}$):

```
rref( B, 1e-15)
rref( B, 1e-16)
```

At this point you can be certain that a lower bound for Matlab's default value for $\text{tol}$ is $10^{-16}$. That is, this default value is at least $10^{-16}$.

Explain why:

(b) Experiment to find more precise bounds for the default value of $\text{tol}$. Hint: type `rref(B, 2e-16)`, then `rref(B, 3e-16)`, etc. Record your improved lower bounds for $\text{tol}$. Why are these only lower bounds?

2. Visualizing Linear Transformations of the Plane

Purpose: To understand the standard matrix of a linear transformation. In particular, to see the geometric effect of how a $2 \times 2$ matrix transforms $\mathbb{R}^2$; and conversely, to learn how to write $2 \times 2$ matrices that will transform $\mathbb{R}^2$ in specific ways.

Prerequisites: Linear transformations and their matrices

MATLAB functions and notation used: `drawpoly` from Lay's Toolbox.

Background. When a linear transformation $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is given (for example it might be described in words), it can be identified with a matrix, and this is an easy way to get a formula for the entire function. The method is: let $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a linear transformation and let
\( \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n \) denote the columns of the \( n \times n \) identity matrix; figure out what each \( T(\mathbf{e}_i) \) should be, write \( T(\mathbf{e}_i) \) as a column, and define \( A = [T(\mathbf{e}_1) \ T(\mathbf{e}_2) \ldots T(\mathbf{e}_n)] \). Then it must be true that \( T(\mathbf{x}) = A\mathbf{x} \) for all \( \mathbf{x} \), so you now have a formula for the function. See Theorem 10 in Section 1.7. Note: We often identify \( T \) and \( A \), and call such a function a matrix transformation.

\[
\begin{array}{c}
\text{Example 1. The } 2 \times 2 \text{ matrix transformation that maps } \mathbf{e}_1 \text{ to } \mathbf{e}_1 + \mathbf{e}_2 \text{ and } \mathbf{e}_2 \text{ to } \mathbf{e}_1 - \mathbf{e}_2 \text{ is } \\
\begin{bmatrix}
1 & 1 \\
1 & -1 \\
\end{bmatrix}.
\end{array}
\]

The \( 3 \times 3 \) matrix transformation that maps \( \mathbf{e}_1 \) to \( \begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix} \), \( \mathbf{e}_2 \) to \( \begin{bmatrix} 6 \\ 0 \\ 7 \end{bmatrix} \) and \( \mathbf{e}_3 \) to \( \begin{bmatrix} 5 \\ 4 \\ -1 \end{bmatrix} \) is \( \begin{bmatrix} 3 & 6 & 5 \\ -2 & 0 & 4 \\ 1 & 7 & -1 \end{bmatrix} \).

\[
\begin{array}{c}
\text{Example 2. The function that reflects } R^2 \text{ across the line } y = -x \text{ is a linear transformation. To find its matrix, notice it must map } \mathbf{e}_1 \text{ to } -\mathbf{e}_2 \text{ and } \mathbf{e}_2 \text{ to } -\mathbf{e}_1, \text{ so the matrix is } \\
\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}. \text{ See the sketch at the top of page 79 in Lay's text.}
\end{array}
\]

1. (hand) (a) Write a \( 2 \times 2 \) matrix that maps \( \mathbf{e}_1 \) to \( 4\mathbf{e}_2 \) and \( \mathbf{e}_2 \) to \( -\mathbf{e}_1 \):

\[
\begin{array}{c}
\text{(b) Write a } 2 \times 2 \text{ matrix that reflects } R^2 \text{ across the line } y = x:}
\end{array}
\]

More background. A matrix transformation always maps a line onto a line or a point, and maps parallel lines onto parallel lines or onto points. (See exercises 25-28 in Section 1.7.) In the following question, you will verify these things for a particular matrix.

2. (hand) Let \( \mathbf{M} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \). (a) Explain why the function \( T(\mathbf{x}) = \mathbf{M}\mathbf{x} \) maps the \( x \)-axis onto the line \( y = x \), and why it maps the line \( y = 2 \) onto the line \( y = x + 2 \).

(Hints: A general point on the \( x \)-axis is of the form \( \begin{bmatrix} t \\ 0 \end{bmatrix} \); calculate \( \mathbf{M}\begin{bmatrix} t \\ 0 \end{bmatrix} \) and interpret where those image points lie. Similarly, calculate \( \mathbf{M}\begin{bmatrix} t \\ 2 \end{bmatrix} \) and interpret.)

(b) Continue to use the function \( T(\mathbf{x}) = \mathbf{M}\mathbf{x} \). Sketch and label the \( x \)-axis and the line \( y = 2 \) on the left axes below, and sketch and label their images on the right. Verify that \( \mathbf{M} \) does map the first two parallel lines to a pair of parallel lines:
Still more background. Because a matrix transformation maps parallel lines to parallel lines, it will map any parallelogram to another parallelogram (which could be degenerate -- a line segment or a point). When a parallelogram is given, the easy way to draw its image is to plot the images of its four vertices and connect those points to make a parallelogram.

Define the standard unit square to be the square in $R^2$ whose vertices are $(0, 0), (1, 0), (1, 1)$ and $(0, 1)$. When you want to visualize what a $2 \times 2$ matrix transformation does geometrically, it is particularly useful to sketch the image of this standard square -- seeing how this square gets moved or distorted usually gives you a good idea of what the transformation is doing geometrically to the whole plane. Recall that any linear transformation maps the origin to itself (why?), so you only need to figure out where the transformation maps the other three vertices.

Example 3. Continue to use $T(x) = Mx$ where $M = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Sketch the image of the standard unit square:

![Diagram of Example 3]

Example 4. Find a matrix $M$ which maps the standard unit square to the parallelogram with vertices $(0, 0), (3, 1), (2, 2), (1, 1)$. To do this, sketch the parallelogram and recognize that $Me_1$ and $Me_2$ must be $(3, 1)$ and $(-1, 1)$, or vice
versa. (Why?) So either of the matrices \[
\begin{pmatrix}
3 & -1 \\
1 & 1
\end{pmatrix}
\] or \[
\begin{pmatrix}
-1 & 3 \\
1 & 1
\end{pmatrix}
\] will work. Calculate the image of \((1, 1)\) under each of these, to verify that it is \((2, 2)\). Why are these the only two matrices that will work?

3. (hand) Below are seven \(2 \times 2\) matrices, each one of the special, simple types described in Sections 1.7 and 1.8. Each determines a linear transformation of \(\mathbb{R}^2\). We will just write the name of the matrix instead of \(T\). For each matrix, sketch the image of the standard unit square, label the vertices of the image, and describe in words how the matrix is transforming the plane. To get you started, answers are given for the first matrix.
\[ A = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \] Description: This is a vertical shear. It leaves the y-axis fixed and changes the slopes of other lines through the origin.

\[ B = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \] Description: 

\[ C = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \] Description: 

\[ D = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \] Description:
\[ E = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \] Description:__________________________________

\[ F = \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) \\ \sin(\pi/4) & \cos(\pi/4) \end{bmatrix} \] Description:__________________________________

\[ G = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \] Description:__________________________________
4. (MATLAB) In computer graphics, transformations are usually accomplished by applying a succession of simple matrix transformations -- dilations, shears, reflections, rotations and projections. Here you will do a variety of such.

To begin, define the matrices \( A, B, \ldots, G \) used above. You should also define a matrix called \( box \) whose columns contain the coordinates of the vertices of the standard unit square. Your matrix should be:

\[
\begin{bmatrix}
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0
\end{bmatrix}
\]

Note that the first column is repeated at the end - this will be needed for the correct picture of the square to come up using \texttt{drawpoly} below.

Type \texttt{A, box} to see \( A \) and \( box \). Notice the order in which the vertices of the square are stored in \( box \), and notice that the origin is repeated as the last column. Type \texttt{A*box} to see that this command causes MATLAB to multiply \( A \) times each column of \( box \) - i.e., it produces a matrix whose columns are the images of the vertices of the standard unit square.

**Important:** Note that to use \texttt{drawpoly}, you will need to have downloaded the files from the lay tool box (downloading and adjusting the path were discussed in the previous assignment).

More importantly, you might have another problem with \texttt{drawpoly}, especially if you are using Matlab version 6. Specifically, the drawings may appear as white on a white background (sort of like trying to spot a polar bear). If you see the picture frame but not the polygon, you will need to change the color specified in the file \texttt{drawpoly.m}. To do this, go into the file with your editor, and change the line

\[
\text{plot(u(1,:),u(2,:),'w'), if nargin < 5, pause, end}
\]

by replacing the 'w' (standing for 'white') to another color, such as 'b' (be sure to save of course).

When \( X \) is a matrix whose columns are points in \( R^2 \), the command \texttt{drawpoly(X)} (once it has been fixed; see above) will plot those points and draw a line segment from each one to the next one. Try this: type \texttt{drawpoly(box)} to see the standard unit square. Do you see why we needed to repeat the first column as the last in \( box \)? Then type \texttt{drawpoly(A*box)} to see the image of that square under \( A \).

To see both figures on the same axes, type \texttt{drawpoly(box,A*box)} -- first you will see the square, then press [Enter] and you will also see its image under \( A \).
(a) The lines below perform several successive transformations of the standard unit square. The first line does a shear using \( A \); the second does the shear followed by rotation of the plane through \( \pi/4 \) using \( F \); the third does the shear followed by the rotation and then reflects the plane across the line \( y = -x \) using \( E \). Type these lines, and watch carefully to see the result of each successive transformation. Sketch the result of each step:

\[
\text{drawpoly}(A*\text{box}) \\
\text{drawpoly}(F*(A*\text{box})) \\
\text{drawpoly}(E*(F*(A*\text{box})))
\]

(b) First shear using \( A \), then reflect across the \( x \)-axis, then rotate through \( \pi/4 \). Use \text{drawpoly} to sketch the result of each successive transformation, and sketch:

\[
\text{drawpoly}
\]

5. (hand) Sketch the parallelogram with vertices \((0, 0), (4, 2), (0, -4), (4, -2)\) and write two different \( 2 \times 2 \) matrices \( X \) and \( Y \) which would transform the standard unit square into this parallelogram. Use \text{drawpoly} to verify that your matrices work:
6. (hand) Sketch the parallelogram with vertices \((1, 1), (1, 2), (3, 1), (3, 2)\). Explain why no \(2 \times 2\) matrix transformation could map the standard unit square onto this figure.

Example 5. Consider the figure called flag1 shown below. Its flagpole is from \((1, 0)\) to \((1, 3)\), and the vertices of the flag are \((1, 2), (1, 3), (2, 3), \) and \((2, 2)\). Find matrices which transform this figure into the other flag type figures shown, flagA and flagB.

First consider flagA. One way to figure out the matrix would be to simply recognize that rotating the plane through \(-\pi/2\) clearly takes flag1 into flagA, so the matrix must be
\[
\begin{bmatrix}
0 & 1 \\
-1 & 0
\end{bmatrix}
\]. Alternatively, inspect the figures to see where \(e_1 = (1, 0)\) and \(e_2 = (0, 1)\)
must be mapped, so as to find the columns of the matrix. Clearly \((1, 0)\) maps to \((0, -1)\) -- so the matrix we seek must look like \[
\begin{bmatrix}
0 & a \\
-1 & b
\end{bmatrix}
\]. It is also pretty easy to see by inspection that \((1, 1)\) maps to \((1, -1)\), so solve the equation
\[
\begin{bmatrix}
0 & a \\
-1 & b
\end{bmatrix}
\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}
\]
for \(a\) and \(b\), and you will obtain the same matrix \[
\begin{bmatrix}
0 & 1 \\
-1 & 0
\end{bmatrix}
\].

Now consider flagB. Clearly \((1, 0)\) must be mapped to \((3, 0)\). Since there appears to be no skewing or dilation in the vertical direction, you could guess that \((0, 1)\) maps to itself, hence the matrix is \[
\begin{bmatrix}
3 & 0 \\
0 & 1
\end{bmatrix}
\]. Or you could proceed as above, noticing that \((1, 1)\) maps to \((3, 1)\) and using that to solve for the entries of the second column of the matrix. This would also yield \[
\begin{bmatrix}
3 & 0 \\
0 & 1
\end{bmatrix}
\].

Whatever method you use to calculate the matrix in such a problem, you should check that your matrix really does work. Create a data matrix flag1 to store the vertices for flag1 defined above. Be sure that the last column of your flag1 matrix is the same as the first column; this way the drawpoly command will draw a closed polygon:

\[
\text{flag1} = \begin{bmatrix}
1 & 1 & 1 & 2 & 2 & 1 \\
0 & 2 & 3 & 3 & 2 & 2
\end{bmatrix}
\]

The easy way to do the sketching here is to store the matrices and use drawpoly. Type these lines to do that:

\[
\begin{align*}
\text{MA} &= [0 \ 1; -1 \ 0], \ \text{drawpoly(flag1, MA*flag1)} \\
\text{MB} &= [3 \ 0; 0 \ 1], \ \text{drawpoly(flag1, MB*flag1)}
\end{align*}
\]

7. Continue to consider the figure called flag1 discussed above. One at a time, consider each of the other three flags sketched below and find a \(2 \times 2\) matrix which maps flag1 onto it. Use drawpoly to verify that each of your matrices does what you want, and record each matrix below the appropriate figure.
Matrices: