Some Linear Algebra Basics in Maple

A note about input: I prefer to use "worksheet mode". Worksheets are built (more or less) from command prompts "[>"
and text boxes (like what we're in right now). At a command prompt, you can type in "Text" or "Math" mode. I prefer
Text mode because then raw code is shown. Math mode might look prettier, but it hides details and can be quite
unhelpful when a mistake is made. To switch between input modes you can either click buttons above or use the key "F5".
If you would like to enter a command, place the cursor somewhere in that line of code and hit "enter". If you would like
to enter more than one line of code at a single command prompt, you can use "shift+enter". Also, don't forget that while
many of the commands used are "long" you don't necessarily have to fully type them out. If Maple offers a suggestion
of what you're trying to type, use "tab" completion to skip typing the rest of the command. Alternatively, you can use
Maple's "pallettes" off to the left to get pre-fabricated templates to fill out.

First, we clear memory and load the package "LinearAlgebra" which contains commands that we might find helpful.
Notice that the "with(...)" command ends in a ":" this suppresses output. If you change it to a ";" Maple will display
a list of all of the commands that the LinearAlgebra package loads.

```
> restart;
  with(LinearAlgebra):
```

In general, to access help you can type "? something" at a command prompt and Maple will look up help on "something".

To find out more about the "LinearAlgebra" package and its
many functions try...

```
> ? LinearAlgebra
```

An older linear algebra package is called "linalg". It is no longer supported. Please don't load or use it.

Remember that Maple uses ":=" for assignment and "<" and ">" to begin and end matrices/vectors/rows/columns.
To move down we use "," and to move over we use "|"

Let enter a matrix M into memory. I'll do it two different ways. Once going accross rows and then again going
down columns.

```
> M := <<1|2|3>,<4|5|6>>;
```

$$M := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad (1)$$

```
> M := <<1,4>|<2,5>|<3,6>>;
```

$$M := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad (2)$$

After code is performed, Maple usually produces some blue output (unless we tell it "shut up!" by using a ":").
If you right click on blue output, you'll get a context sensitive menu. Usually this menu is loaded with all kinds
of helpful commands you can perform.

After entering "A" (below) into memory, try "right clicking" on the displayed matrix. You will get a menu of
common operations to perform on the matrix. I will try...

Solvers and Forms --> Row-Echelon Form --> Reduced
Standard Operations --> Inverse
Standard Operations --> Transpose
Solvers and Form --> LU Decomposition --> Gaussian Elimination --> Gaussian Elimination (P,L,U)

(We are defining "A" row-by-row.)

```
> A := <<1|2|3>,<4|5|6>,<7|8|9>>;
```

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad (3)$$

```
> LinearAlgebra:-ReducedRowEchelonForm( (3) );
```

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix} \quad (4)$$

```
> LinearAlgebra:-MatrixInverse( (3) );
Error, (in LinearAlgebra:-MatrixInverse) singular matrix
```

A is a singular matrix -- it didn't row reduce to the identity -- thus is does not have an inverse!!

```
> LinearAlgebra:-Transpose( (3) );
```

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \quad (5)$$

```
> LinearAlgebra:-LUDecomposition( (3), 'method =
```

```
  GaussianElimination' );
```

$$\left[\begin{matrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{matrix}\right], \left[\begin{matrix}1 & 0 & 0\\ 4 & 1 & 0\\ 7 & 2 & 1\end{matrix}\right], \left[\begin{matrix}1 & 2 & 3\\ 0 & -3 & -6\\ 0 & 0 & 0\end{matrix}\right] \tag{6}$$

The above command spits out P, L, and U where A = PLU. Notice that P is the identity matrix (no row swaps
were needed in the forward pass) so actually A = LU.

```
> P,L,U := LUDecomposition(A, 'method = GaussianElimination' );

  'A' = L.U;
```

$$P, L, U := \left[\begin{matrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{matrix}\right], \left[\begin{matrix}1 & 0 & 0\\ 4 & 1 & 0\\ 7 & 2 & 1\end{matrix}\right], \left[\begin{matrix}1 & 2 & 3\\ 0 & -3 & -6\\ 0 & 0 & 0\end{matrix}\right]$$

$$A = \left[\begin{matrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{matrix}\right] \tag{7}$$

Let's define "B" column-by-column.

```
> B := <<1,2,3>|<4,5,6>|<7,8,9>>;
```

$$B := \left[\begin{matrix}1 & 4 & 7\\ 2 & 5 & 8\\ 3 & 6 & 9\end{matrix}\right] \tag{8}$$

The Student[LinearAlgebra] subpackage has even more
useful tools.

```
> with(Student[LinearAlgebra]):
> ? Student[LinearAlgebra]
```

LinearSolveTutor is an interactive tutor which will allow you
to row reduce A one step at a time.

```
> LinearSolveTutor(A);
```

Other commands of interest...

```
> v := <5,0,2>;
```

```
C := <<3,4,1>|<-1,2,5>>;
```

$$v := \begin{bmatrix} 5 \\ 0 \\ 2 \end{bmatrix}$$

$$C := \begin{bmatrix} 3 & -1 \\ 4 & 2 \\ 1 & 5 \end{bmatrix} \quad \textbf{(9)}$$

Matrix-vector and matrix multiplications are done with a "."

```
> A.v;
  A.C;
```

$$\begin{bmatrix} 11 \\ 32 \\ 53 \end{bmatrix}$$

$$\begin{bmatrix} 14 & 18 \\ 38 & 36 \\ 62 & 54 \end{bmatrix} \quad \textbf{(10)}$$

You can build bigger matrices from smaller ones (and vectors).

Here is [C|v] and [A|v|A|C] and a stack of two copies of A.
Keep in mind that sizes must make sense and that "|" is over
while "," is down...

```
> <C|v>;
  <A|v|A|C>;
  <A,A>;
```

$$\begin{bmatrix} 3 & -1 & 5 \\ 4 & 2 & 0 \\ 1 & 5 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 & 5 & 1 & 2 & 3 & 3 & -1 \\ 4 & 5 & 6 & 0 & 4 & 5 & 6 & 4 & 2 \\ 7 & 8 & 9 & 2 & 7 & 8 & 9 & 1 & 5 \end{bmatrix}$$

(11)

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad (11)$$

Here we have Maple compute 4A-3I where I is the 3x3 identitiy matrix...

```
> 4*A - 3*IdentityMatrix(3);
```

$$\begin{bmatrix} 1 & 8 & 12 \\ 16 & 17 & 24 \\ 28 & 32 & 33 \end{bmatrix} \quad (12)$$

Determinants are easy as well...

```
> Determinant(A);
```

$$0 \quad (13)$$