

Introduction to Matrices and Linear Algebra on Maple

Math 2270-3

September 2, 2009 – updated September 16 – posted on our lecture page

With Maple you can create combined text–math documents. Alternately, you can create math output and export it to other documents. This text field was created using the "T" menu item. To do math in a document, move your cursor to just above the place you want to do it, and use the "[>" menu item – a math field will be created as near to directly below your current cursor position as possible. If you're in a Math field and want text below it, hit the "T" menu item.

```
[> 3 + 4; 5·7;
    6·5 : #commands end with ";" if you want to see output, with ":" if you don't
        #commands are executed as soon as you hit the "enter" or "return" key.
        # to enter multiple command lines, hold down the shift key while you hit return
[>
```

The newer Maple package for matrix and linear algebra is called "LinearAlgebra". Commands in LinearAlgebra tend to be concatenated capitalized words for the complete name of the procedure you wish to do. There are also shortcuts.

Load the LinearAlgebra package:

```
[> with(LinearAlgebra) : # to see commands, end with ";" to hide them, use ":"
```

Define a matrix:

```
[> A := Matrix(3, 4, [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]);
    # a 3 by 4 matrix, with entries moving left to right, and down the rows
    # ":" is how we define things in Maple
```

```
[> A;
```

```
[> B := Matrix([ [1, 2, 3], [1, 2, -1], [0, 2, -1] ]);
    # another format – enter each row
```

Compute the reduced row echelon form:

```
[> ReducedRowEchelonForm(A);
```

Define a (column) vector:

```
[> b := Vector([2, 2, 2]);
```

Augment a matrix and a vector, or two matrices with the same number of rows:

```
[> Aaugb := <A|b>; #the augmented matrix
    AaugB := <A|B>;
    Baugb := <B|b>;
```

Multiply two matrices (or a matrix times a column vector):

```
[> B.A; #we'd get an error if we tried A.B!
    B.b;
```

Matrix addition, scalar multiplication, integer powers, etc:

```
[> C := Matrix(3, 3, [1, 0, 2, 2, 0, 1, 0, 0, 1]);
    2·C; #scalar multiplication
        #(shift-enter to add multiple line commands)
    (B + 2·C)2; #what it looks like!
    B-1; #inverse matrix if it exists
    B-2; #the inverse squared
```

Solve a non-singular square matrix problem (non-singular means inverse transformation (and matrix) exist. We are solving $Bx=b$. Any one of the following commands will work!

```
[ > B-1.b;  
  ReducedRowEchelonForm(⟨B|b⟩);  
  LinearSolve(B, b);
```

Solve any solvable matrix problem. Here we are solving $Ax=b$:

```
[ > LinearSolve(A, b); #Maple free parameters look funny!  
  ReducedRowEchelonForm(⟨A|b⟩);
```