

Linear algebra

Maple has a fairly extensive set of linear algebra commands. To use them, you must begin your work with the command:

```
> with(linalg):  
Warning, new definition for norm  
Warning, new definition for trace
```

The basic linear algebra operations you will need to use are :

- a) Defining matrices (the "matrix" command) and displaying them ("evalm")
- b) Matrix multiplication (the "multiply" command)
- c) Row reduction commands ("pivot" and "gaussjord"), making augmented matrices with "augment"
- d) Inverse of a matrix

Matrices are defined with the "matrix" command -- the easiest way to make a matrix is to list its entries, each row being enclosed in brackets as follows:

```
> M:=matrix([[3,-1,2],[2,8,8],[3,1,1]]);
```

$$M := \begin{bmatrix} 3 & -1 & 2 \\ 2 & 8 & 8 \\ 3 & 1 & 1 \end{bmatrix}$$

If you make a typing error defining a matrix, you can change one entry at a time as follows:

```
> M[2,1]:=5;
```

$$M_{2,1} := 5$$

This changes the entry in the second row, first column:

```
> M;
```

M

```
> evalm(M);
```

$$\begin{bmatrix} 3 & -1 & 2 \\ 5 & 8 & 8 \\ 3 & 1 & 1 \end{bmatrix}$$

Note that, to see M , you must use `evalm` (for "evaluate as a matrix").

If we want to solve the system of equations:

$$3x - y + 2z = 4$$

$$5x + 8y + 8z = 6$$

$$3x + y + z = -2,$$

we need to augment the matrix M with the vector $[4,6,-2]$ and do Gaussian elimination. The Maple commands are:

```
> S:=augment(M,vector([4,6,-2]));
```

$$S := \begin{bmatrix} 3 & -1 & 2 & 4 \\ 5 & 8 & 8 & 6 \\ 3 & 1 & 1 & -2 \end{bmatrix}$$

```
> gaussjord(S);
```

$$\begin{bmatrix} 1 & 0 & 0 & \frac{-22}{19} \\ 0 & 1 & 0 & \frac{-86}{57} \\ 0 & 0 & 1 & \frac{170}{57} \end{bmatrix}$$

You can see that the solution of the system is: $x = -22/19$, $y = -86/57$ and $z = 170/57$. Alternatively, we could have found the inverse of M and multiplied it by the vector:

```
> multiply(M^(-1),vector([4,6,-2]));
```

$$\begin{bmatrix} \frac{-22}{19} & \frac{-86}{57} & \frac{170}{57} \end{bmatrix}$$

Thus, the important commands are:

1. `matrix` -- this is used to specify matrices -- note the positions of square brackets in the syntax used above -- square brackets enclose each row, and an extra pair

enclose all the rows.

2. `evalm` -- to see a matrix, you must "evalm" it -- occasionally, it will be necessary to `evalm` partial results of a matrix computation in order to use them again. The general rule is that you must use `evalm` anytime you use the name of a matrix, as in `evalm(A)` , `evalm(M)` etc..

3. `augment` -- `vector` -- use `augment` to create an augmented matrix, i.e., to put the extra column on the right side of a square matrix so that the result represents a system of equations. Note that you must use "vector" to make the right side into a vector, as shown above.

4. `gaussjord` -- performs complete row reduction on a matrix.

5. `multiply` -- `multiply(A,B)` multiplies the two matrices A and B --since matrix multiplication is not commutative, be careful of the order!

6. `inverse` -- For the inverse of a square matrix (if it exists), you can use either `inverse(A)` or A^{-1} . In fact, you may raise a matrix to any power using standard $^$ notation -- this is useful when studying Markov chains.

General hints:

The matrix functions in Maple are generally pretty well-behaved. The thing that goes wrong most often is that people forget to use `evalm` sometimes -- when you get unexpected results, this is usually the reason.

Maple uses exact rational arithmetic in matrix computations unless you put some "floating point" numbers (i.e., numbers with decimal points) in the problem. Floating point computations are usually faster, so even if your problem involves only whole numbers, it is sometimes a good idea to express 3 as 3.0 etc.. if the

computations are going too slowly.