MATHEMATICS 201-105-RE
Linear Algebra
Martin Huard
Winter 2006

Systems of Linear Equations with Maple

When working with matrices on Maple, the first thing to do is to load the *linalg* package, which contains a lot of the commands that we use.

\> **with(linalg);**

Let us consider the following system of linear equations $AX = b$.

$$\begin{aligned} 3x + 3y + 2z &= 3 \\ 4x - 2y + 5z &= -15 \\ 2x - y - z &= 3 \end{aligned}$$

We will see five different ways to solve this system of linear equations with Maple, namely *directly*, using *Gaussian elimination*, *Gauss-Jordan method*, the *inverse* and with *Cramer's Rule*.

Solving Directly

We can solve our system of linear equations directly using the `solve()` command.

\> **solve({ 3*x+3*y+2*z=3, 4*x-2*y+5*z=-15, 2*x-y-z=3 }, {x,y,z});**

Note that this method works for <u>any</u> kind of equations, not just systems of linear equations.

There is an other way to solve this using the `linsolve()` command from the *linalg* package. We begin by defining the matrix of coefficients *A* along with the constant matrix *b*. When using the linsolve command, *b* must be entered as a vector or using the command matrix.

\> **A:=matrix([[3,3,2],[4,-2,5],[2,-1,-1]]);**

\> **b:=matrix([[-3],[-15],[3]]);**

Then, the solution to our system of linear equations is:

\> **linsolve(A,b);**

Gaussian Elimination

The first thing we need to do is find the augmented matrix. We can either rewrite the matrix, or, since we already defined *A* and *b*, use the `augment( , )` command from the *linalg* package.

\> **C:=augment(A,b);**

Now we can reduce our matrix using the `gausselim( )` command from the *linalg* package.

\> **G:=gausselim(C);**

Note that Maple does not give us the matrix in row-echelon form.
To obtain the solution, we use the `backsub( )` command from the *linalg* package.

\> **backsub(G);**

Reducing the matrix row by row

Instead of using the `gausselim()` command, we can obtain the reduced matrix by transforming the augmented matrix through the use of elementary row operations.
The commands for elementary row operations are as follows:

1. `addrow(A,rj,ri,m)` $R_i \to R_i + mR_j$ (Note the position of i and j !)
2. `mulrow(A,ri,k)` $R_i \to kR_i$

3. `swaprow(A,ri,rj)` $R_i \leftrightarrow R_j$

Let us reduce our augmented matrix C with these elementary row operations.
Let us start with $R_1 \to \frac{1}{3}R_1$ to make a leading 1.

> **C1:=mulrow(C,1,1/3);**

and make a zeros under the leading 1, with $R_2 \to R_2 - 4R_1$ and $R_3 \to R_3 - 2R_1$

> **C2:=addrow(C1,1,2,-4);**
> **C3:=addrow(C2,1,3,-2);**

Making a leading one where the -6 is with $R_2 \to -\frac{1}{6}R_2$

> **C4:=mulrow(C3,2,-1/6);**

and a zero under that new leading one $R_3 \to R_3 + 3R_2$

> **C5:=addrow(C4,2,3,3);**

Lastly, making a leading one where the $\frac{-7}{2}$ with $R_3 \to -\frac{2}{7}R_3$

> **C6:=mulrow(C5,3,-2/7);**

Which gives us the row-echelon form for our augmented matrix.

Gauss-Jordan method --- Preferred Method

The Gauss-Jordan method works the same way as Gaussian elimination, except that we use the command `gaussjord( )` from the *linalg* package.

> **J:=gaussjord(C);**

Which gives us the augmented matrix in reduced row-echelon form. The solution can be obtained with the `backsub( )` command from the *linalg* package.

> **backsub(J);**

Using the inverse

In solving the system $AX = b$, we can solve for *X* by finding the inverse of *A*, if *A* is invertible, then multiplying it by b . That is, $X = A^{-1}b$.

> **multiply(inverse(A),b);**

Note that this method will only work if our system has a unique solution.

Cramer's Rule

To use Cramer's rule, we need to find the matrices *A*(1), *A*(2) and *A*(3), where the matrix *A*(j) is the matrix *A* with the jth column replaced by the *b*. This can be done with the `concat( )` command, which enables us to assemble our matrices A(j) from the columns of *A* and the matrix *b*.

> **A1 := concat(b,col(A,2),col(A,3));**
> **A2 := concat(col(A,1), b, col(A,3));**
> **A3 := concat(col(A,1),col(A,2),b);**

With these matrices, the solution is obtained by dividing det(*A*(*j*)) with det(*A*).

> **det(A1)/det(A);**
> **det(A2)/det(A);**
> **det(A3)/det(A);**

Maple Tutor

Maple has a number of built-in tutors, such as Gaussian elimination, that you can explore. Go to TOOLS – TUTORS – LINEAR ALGEBRA and pick the tutor you wish to try.