Mth 341 Linear Algebra Spring 2000
MLC Computer Lab Visit 1
Bent Petersen

Login

Press the Ctrl-Alt-Delete keys simultaneously. You should get a login prompt. Enter your ORST user name and press the Tab key. Then enter your ORST password and press the Enter key.

Personal Files

Make sure you save your work on the Z: drive. This drive is actually your personal space on the lab server and will be available if you log in later on a different workstation.

Start Application

To start Maple (or Matlab, or Mathematica, ...) select the Start button (lower left corner of the screen), then Programs from the menu, etc. Alternately, if you use the browser, Internet Explorer or Netscape, to download a Maple Worksheet it will offer to start Maple for you.

Logout

When you are done with your session, you must logout. One way to do it is to press Ctrl-Alt-Delete. A menu will appear. Select Logoff. If you do not logout you leave your account open for the next person to come along. That person will have access to your personal files on the ORST server. Do not forget to logout, even if you are just leaving for a short while!

The Worksheet

In the live version of this worksheet I have removed all of the Maple output. If you press the Enter key over each Maple command below (in order) you will see Maple's responses appear. Fell free to modify anything as you go along!

Sometimes it is desirable to be able to restart more or less cleanly. If we begin the worksheet with a restart command, then we can just re-execute the whole worksheet in order to clean up a mess and do everything in sequential order. This is a useful trick, but it is not a good idea if your worksheet contains some very length calculations (which you do not want to redo).

```
> restart;
```

We will be doing some linear algebra, so we load the linear algebra library, linalg. This library defines some of the commands and data structures that we will use. When we load a library it announces all of the commands that it defines. We can suppress this output by using a colon, in place of the usual command-terminanting semicolon. Here I used a semicolon so you can see the list of commands defined by the linalg library.

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

[*BlockDiagonal*, *GramSchmidt*, *JordanBlock*, *LUdecomp*, *QRdecomp*, *Wronskian*, *addcol*, *addrow*, *adj*, *adjoint*, *angle*, *augment*, *backsub*, *band*, *basis*, *bezout*, *blockmatrix*, *charmat*, *charpoly*, *cholesky*, *col*, *coldim*, *colspace*, *colspan*, *companion*, *concat*, *cond*, *copyinto*, *crossprod*, *curl*, *definite*, *delcols*, *delrows*, *det*, *diag*, *diverge*, *dotprod*, *eigenvals*, *eigenvalues*, *eigenvectors*, *eigenvects*, *entermatrix*, *equal*, *exponential*, *extend*, *ffgausselim*, *fibonacci*, *forwardsub*, *frobenius*, *gausselim*, *gaussjord*, *geneqns*, *genmatrix*, *grad*, *hadamard*, *hermite*, *hessian*, *hilbert*, *htranspose*, *ihermite*, *indexfunc*, *innerprod*, *intbasis*, *inverse*, *ismith*, *issimilar*, *iszero*, *jacobian*, *jordan*, *kernel*, *laplacian*, *leastsqrs*, *linsolve*, *matadd*, *matrix*, *minor*, *minpoly*, *mulcol*, *mulrow*, *multiply*, *norm*, *normalize*, *nullspace*, *orthog*, *permanent*, *pivot*, *potential*, *randmatrix*, *randvector*, *rank*, *ratform*, *row*, *rowdim*, *rowspace*, *rowspan*, *rref*, *scalarmul*, *singularvals*, *smith*, *stackmatrix*, *submatrix*, *subvector*, *sumbasis*, *swapcol*, *swaprow*, *sylvester*, *toeplitz*, *trace*, *transpose*, *vandermonde*, *vecpotent*, *vectdim*, *vector*, *wronskian*]

Matrices can be entered by listing their rows (lists are denoted by [ ] in Maple) or by giving the size and then listing the entries in the matrix row-wise. Here I enter the matrix A in both ways.

```
> A:=matrix([[1,2,3,4],[0,-1,3,4],[2,-1,2,3]]);
```

$$A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & 3 & 4 \\ 2 & -1 & 2 & 3 \end{bmatrix}$$

```
> A:=matrix(3,4,[1,2,3,4,0,-1,3,4,2,-1,2,3]);
```

$$A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & 3 & 4 \\ 2 & -1 & 2 & 3 \end{bmatrix}$$

Personally I prefer the second method for interactive entry. It is also possible to build a matrix by specifying its columns (as vectors or lists). Thus

```
> a[1]:=[1,0,2]; a[2]:=[2,-1,-1];
  a[3]:=[3,3,2]; a[4]:=[4,4,3];
```

$$a_1 := [1, 0, 2]$$

$$a_2 := [2, -1, -1]$$

$$a_3 := [3, 3, 2]$$

$$a_4 := [4, 4, 3]$$

Now we use the augment command to build the matrix:

```
> A:=augment(a[1],a[2],a[3],a[4]);
```

$$A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & 3 & 4 \\ 2 & -1 & 2 & 3 \end{bmatrix}$$

Alternately we can build the matrix by entering its rows and then stacking them on top of each other with the stackmatrix command.

```
> b[1]:=[1,2,3,4]; b[2]:=[0,-1,3,4];
  b[3]:=[2,-1,2,3];
```

$$b_1 := [1, 2, 3, 4]$$

$$b_2 := [0, -1, 3, 4]$$

$$b_3 := [2, -1, 2, 3]$$

```
> A:=stackmatrix(b[1],b[2],b[3]);
```

$$A := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & 3 & 4 \\ 2 & -1 & 2 & 3 \end{bmatrix}$$

There is also a command called entermatrix which may be used to build a matrix. It is a special purpose command though and not very convenient unless you enjoy typing numerous semicolons.

Diagonal matrices can be entered in a natural way.

```
> diag(1,2,3,4);
```

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

The diag command gives you a simple way to define your own n by n identity matrix, IE(n):

```
> IE:=n->diag(seq(1,k=1..n));
```

$$IE := n \to \mathrm{diag}(\mathrm{seq}(1, k = 1 .. n))$$

```
> IE(2);
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

```
> IE(3);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

It would have been more natural to use I for the identity matrix, but Maple uses that for the square root of minus one.

Maple has a built-in command, rref, for computing the row reduced echelon form of a matrix. For example, for the matrix A above we have

```
> rref(A);
```

$$\begin{bmatrix} 1 & 0 & 0 & \frac{3}{19} \\ 0 & 1 & 0 & \frac{-1}{19} \\ 0 & 0 & 1 & \frac{25}{19} \end{bmatrix}$$

Note we still have A available. Ordinarily in Maple an expression may be evaluated (or the value recalled) by entering the name followed by a semicolon.

```
> A;
```

$$A$$

As we see, Maple makes an exception for matrices. The reason is they can be quite large and you may not wish to waste time displaying them, nor to clutter up your worksheet. You can always use the evalm function to force matrix evaluation

```
> evalm(A);
```

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

Now suppose we want to row reduce A ourselves. We'd want to add -2 times the first row to the third row:

```
> A1:=addrow(A,1,3,-2);
```

$$A1 := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & 3 & 4 \\ 0 & -5 & -4 & -5 \end{bmatrix}$$

Note I assigned the result to A1 so I would be able to refer to it in the next step. An alternative approach is to use the ditto operator, %, but it is risky to use except within the same line, because it refers to the previous expression evaluated, which may not be the previous expression on your worksheet if you have done a lot of editing.

Here's an example of using the % operator. I stack up two commands on one line, so I will not have to worry what % points to.

```
> addrow(A1,2,1,2); A2:=addrow(%,2,3,-5);
```

$$\begin{bmatrix} 1 & 0 & 9 & 12 \\ 0 & -1 & 3 & 4 \\ 0 & -5 & -4 & -5 \end{bmatrix}$$

$$A2 := \begin{bmatrix} 1 & 0 & 9 & 12 \\ 0 & -1 & 3 & 4 \\ 0 & 0 & -19 & -25 \end{bmatrix}$$

```
> mulrow(A2,2,-1); A3:=mulrow(%,3,-1/19);
```

$$\begin{bmatrix} 1 & 0 & 9 & 12 \\ 0 & 1 & -3 & -4 \\ 0 & 0 & -19 & -25 \end{bmatrix}$$

$$A3 := \begin{bmatrix} 1 & 0 & 9 & 12 \\ 0 & 1 & -3 & -4 \\ 0 & 0 & 1 & \frac{25}{19} \end{bmatrix}$$

```
> addrow(A3,3,1,-9); A4:=addrow(%,3,2,3);
```

$$\begin{bmatrix} 1 & 0 & 0 & \frac{3}{19} \\ 0 & 1 & -3 & -4 \\ 0 & 0 & 1 & \frac{25}{19} \end{bmatrix}$$

$$A4 := \begin{bmatrix} 1 & 0 & 0 & \frac{3}{19} \\ 0 & 1 & 0 & \frac{-1}{19} \\ 0 & 0 & 1 & \frac{25}{19} \end{bmatrix}$$

We did not have to swap any rows here, but the command for that is swaprow. Here is an example using again our matrix A.

```
> swaprow(A,1,3);
```

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

Consider now a system of the form Ax=b, using again or matrix A.

```
> b:=[2,7,-5];
```

$$b := [2, 7, -5]$$

```
> soln:=linsolve(A,b);
```

$$soln := \left[\_t_1, -\frac{5}{3} - \frac{1}{3}\_t_1, \frac{25}{3}\_t_1 + \frac{128}{3}, -\frac{19}{3}\_t_1 - \frac{92}{3} \right]$$

Maple returns a list of values. The underscore t sub 1 is a parameter. You assign it arbitrary values to obtain all the solutions. If you find the expression difficult to read you can always substitute something more agreeable, for example

```
> soln:=linsolve(A,b,rnk,s);
```

$$soln := \left[s_1, -\frac{5}{3} - \frac{1}{3} s_1, \frac{25}{3} s_1 + \frac{128}{3}, -\frac{19}{3} s_1 - \frac{92}{3} \right]$$

Note the parameter name has to appear as the fourth variable, so we had to insert a third variable, rnk. It turns out that linsolve stuffs the rank of A into the third variable. Thus

```
> rnk;
```

$$3$$

One can pick out the individual components easily

```
> soln[1]; soln[2]; soln[3]; soln[4];
```

$$s_1$$

$$-\frac{5}{3} - \frac{1}{3} s_1$$

$$\frac{25}{3} s_1 + \frac{128}{3}$$

$$-\frac{19}{3} s_1 - \frac{92}{3}$$

We can also solve directly by row reduction of course -

```
> M:=augment(A,b); R:=rref(M);
```

$$M := \begin{bmatrix} 1 & 2 & 3 & 4 & 2 \\ 0 & -1 & 3 & 4 & 7 \\ 2 & -1 & 2 & 3 & -5 \end{bmatrix}$$

$$R := \begin{bmatrix} 1 & 0 & 0 & \frac{3}{19} & \frac{-92}{19} \\ 0 & 1 & 0 & \frac{-1}{19} & \frac{-1}{19} \\ 0 & 0 & 1 & \frac{25}{19} & \frac{44}{19} \end{bmatrix}$$

Here we can read the solution easily. Moreover, we get the canonical form of the solution, with the free variable as the parameter. This is not the form that Maple returned above (though it is equivalent, of course).

```
> for k from 1 to 3 do x[k]:=R[k,5]-R[k,4]*s;
  od; x[4]:=s;
```

$$x_1 := -\frac{92}{19} - \frac{3}{19} s$$

$$x_2 := -\frac{1}{19} + \frac{1}{19} s$$

$$x_3 := \frac{44}{19} - \frac{25}{19} s$$

$$x_4 := s$$

For more complicated situations this could get out of hand and linsolve may be preferable, even if it returns a non-canonical solution.

Note we do not have to use linsolve to find the rank of a matrix. Maple has a command called rank. Let's generate a random matrix to illustrate

```
> B:=randmatrix(5,7);
```

$$B := \begin{bmatrix} -85 & -55 & -37 & -35 & 97 & 50 & 79 \\ 56 & 49 & 63 & 57 & -59 & 45 & -8 \\ -93 & 92 & 43 & -62 & 77 & 66 & 54 \\ -5 & 99 & -61 & -50 & -12 & -18 & 31 \\ -26 & -62 & 1 & -47 & -91 & -47 & -61 \end{bmatrix}$$

```
> rank(B);
```

$$5$$

Actually, it turns out that for a 5 by 7 random matrix the probability of rank 5 is 1. In other words, it is nearly certain!

Consider now the vectors

```
> u:=[2,1,-3,1]; v:=[-1,3,5,0]; w:=[2,-1,1,-3];
  b:=[-16,17,37,3];
```

$$u := [2, 1, -3, 1]$$

$$v := [-1, 3, 5, 0]$$

$$w := [2, -1, 1, -3]$$

$$b := [-16, 17, 37, 3]$$

We wish to write b as a linear combination of u, v, w if possible.

First we form the matrix with columns u, v and w. Then we solve the system of linear equations with this matrix as the coefficient matrix and with b as the inhomogeneous term.

```
> B:=augment(u,v,w); linsolve(B,b);
```

$$B := \begin{bmatrix} 2 & -1 & 2 \\ 1 & 3 & -1 \\ -3 & 5 & 1 \\ 1 & 0 & -3 \end{bmatrix}$$

$$[-3, 6, -2]$$

We see b = -3u + 6v -2w. What happens if there is no solution? Here's an example:

```
> C:=matrix(2,2,[1,1,1,1]); b:=[1,2];
```

$$C := \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$b := [1, 2]$$

```
> linsolve(C,b);
> rref(augment(C,b));
```

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

There is no solution and therefore Maple returns no response.

We have discussed matrix multiplicatio a little bit in class already. In Maple the notation for matrix multiplication is &*. It looks strange, but the ampersand warns Maple's symbolic simplification

routines that this multiplication is non-commutative.
Here's a few examples:

```
> A:=randmatrix(3,3); B:=randmatrix(3,3);
```

$$A := \begin{bmatrix} 41 & -58 & -90 \\ 53 & -1 & 94 \\ 83 & -86 & 23 \end{bmatrix}$$

$$B := \begin{bmatrix} -84 & 19 & -50 \\ 88 & -53 & 85 \\ 49 & 78 & 17 \end{bmatrix}$$

```
> evalm(A &* B); evalm(B &* A);
```

$$\begin{bmatrix} -12958 & -3167 & -8510 \\ 66 & 8392 & -1137 \\ -13413 & 7929 & -11069 \end{bmatrix}$$

$$\begin{bmatrix} -6587 & 9153 & 8196 \\ 7854 & -12361 & -10947 \\ 7554 & -4382 & 3313 \end{bmatrix}$$

Notice I had to force Maple to evaluate these expressions. The default behavior is to return the product unevaluated. This is very useful behavior in some cases where intermediate results are inconveniently large or not actually needed.

Consider now three vectors:

```
> c1:=vector([1,2,3]); c2:=vector([0,1,4]);
  c3:=vector([1,3,7]);
```

$$c1 := [1, 2, 3]$$

$$c2 := [0, 1, 4]$$

$$c3 := [1, 3, 7]$$

Are they linearly independent? We row reduce the matrix with columns c1, c2 and c3.

```
> C:=augment(c1,c2,c3); rref(C);
```

$$C := \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 3 \\ 3 & 4 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Since C has only 2 pivotal columns the vectors c1, c2 and c3 are linearly dependent. Suppose we want to solve the system Cx=b. We know that there will be a compatability condition on b. Let's try it with b symbolic.

```
> b:=vector([b1,b2,b3]);
```

$$b := [b1, b2, b3]$$

```
> linsolve(C,b);
```

No response! Wise, but not useful. Let's try another approach.

```
> rref(augment(C,b));
```

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

That's strange. This result implies no solutions at all, yet there certainly is a solution if b = 0 for example. The problem is that Maple simplifies expressions such as z/z to 1 when z is symbolic. Thus Maple's answer is actually correct in the sense that there is no solution which is a rational expression in b1, b2 and b3.

One way out of the impasse is to note tht we can keep track of the coefficients of b1, b2 and b3 during row reduction by keeping them in separate columns. Thus we augment C by an identity matrix and row reduce.

```
> augment(C,IE(3)); R:=rref(%);
```

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 2 & 1 & 3 & 0 & 1 & 0 \\ 3 & 4 & 7 & 0 & 0 & 1 \end{bmatrix}$$

$$R := \begin{bmatrix} 1 & 0 & 1 & 0 & \frac{4}{5} & \frac{-1}{5} \\ 0 & 1 & 1 & 0 & \frac{-3}{5} & \frac{2}{5} \\ 0 & 0 & 0 & 1 & \frac{-4}{5} & \frac{1}{5} \end{bmatrix}$$

There is no pivot in the first 3 columns in the last row, so the remaining part of the last row yields a compatability condition on b

```
> evalm(submatrix(R,3..3,4..6) &* b) = 0;
```

$$\left[b1 - \frac{4}{5}b2 + \frac{1}{5}b3 \right] = 0$$

Now we can read of the solutions easily. At this point you know enough about Maple to check your answers to all of the homework problems for the material covered so far in class. Try it out!

One comment - Maple commands are polymorphic. Their behavior depends on the number and type of parameters that you feed to them. Thus even though lists and vectors are different data types in Maple, the linear algebra routines for the most part do not care which you use.

```
> a:=[1,2,3]; b:=vector([1,2,3]);
  c:=matrix(1,3,[1,2,3]);
```

$$a := [1, 2, 3]$$

$$b := [1, 2, 3]$$

$$c := \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

```
> type(a,vector); type(a,list); type(a, array);
  type(a,matrix);
```

false

true

false

false

```
> type(b,vector); type(b,list); type(b,array);
```

```
type(b,matrix);
```

true

false

true

false

```
> type(c,vector); type(c,list); type(c,array);
  type(c,matrix);
```

false

false

true

true

If you get really mysterious results try thinking hard about your data types. Sometimes a lack of clarity in thought will bite you!

Another thing to watch out for is 0. When Maple multiplies a symbolic expression by 0 the result is 0. But which zero?

```
> 0*p;
```

$$0$$

This behavior can cause problems. For example

```
> N:=matrix(2,2,[1,2,3,4]);
```

$$N := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

```
> 0*N; evalm(3*N);
```

$$0$$

$$\begin{bmatrix} 3 & 6 \\ 9 & 12 \end{bmatrix}$$

Ouch! Our 2 by 2 matrix became a scalar when we multiplied by 0. Some very strange error messages could result later! Here the solution is to use the scalar multiplication function

```
> scalarmul(N,0); scalarmul(N,3);
```

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 6 \\ 9 & 12 \end{bmatrix}$$

```
>
```

Of course, it is rare for people to be so careful all the time. The key is to think about your results, to understand what is going on and to add the right level of precision to guarantee your results. Even with computer algebra systems there is no substitute for understanding.

Have fun with Maple!