```
> restart;
> with(Student[Calculus1]):
> with(LinearAlgebra):
```

The goal of this project is to help you learn some of the basics of Maple. The first problem asks you to find the roots of a quadratic polynomial.

We begin by defining the polynomial.

The symbol := is used to define something. For example, if we wanted to define the letter u as the number 7, we write

```
> u:=7;
```

$$u := 7$$

Also, if you end a line with a semicolon (;), your output will display on the screen. If you end with a colon, the output will not display.

If we want to define a function $p(x)$, the following notation is used:

```
> p:= unapply(a*x^2 + b*x + c,x);
```

The computer can now evaluate this polynomial at any x. For example,

```
> p(2);
```

$$4\,a + 2\,b + c$$

In order to compute the roots of this polynomial, we need to define a, b, and c as real numbers. This can be done as follows:

```
> a:=-1; b:= 3; c:= 4;
```

$$a := -1$$
$$b := 3$$
$$c := 4$$

The polynomial $p(x)$ is now

```
> p(x);
```

$$-x^2 + 3\,x + 4$$

We can find the roots of this polynomial using the 'Roots' command: the first argument, $p(x)$, is the polynomial whose roots we wish to find. The second argument tells Maple which variable is the independent variable–we can use any letter we like.

```
> Roots(p(x),x);
```

$$[-1, 4]$$

```
> Roots(p(z),z);
```

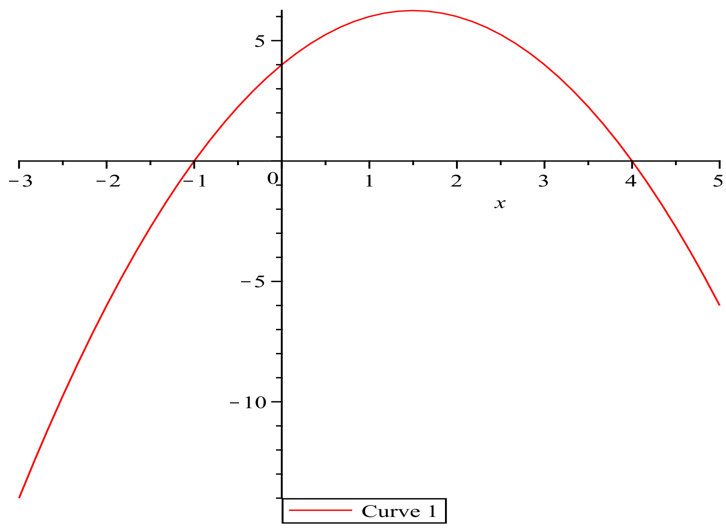
$$[-1, 4]$$

```
> Roots(p(alpha),alpha);
```

$$[-1, 4]$$

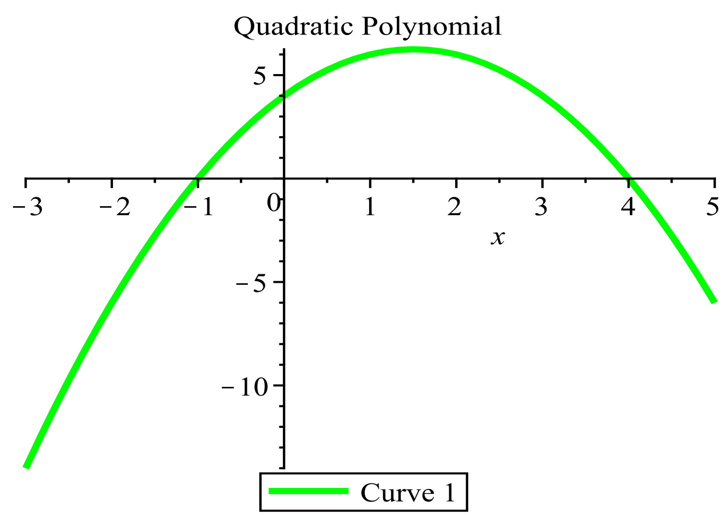
We can also plot the polynomial.

```
> plot(p(x),x = -3..5);
```



The second argument of the 'plot' function is the domain over which the function will be plotted. The 'plot' command contains other options as well; using these options, you can change the color and add a title, among other things. These can be explored by typing the command '? plot'. For example,

```
> plot(p(x), x = -3..5, color = green, thickness = 3, title = "Quadratic
Polynomial");
```



%%%

We will now discuss vectors and how to work with them. We begin by defining 2 vectors in**R**ˆ7.

```
> restart;
```

```
> with(LinearAlgebra):
> v := Vector([1,-2,3,-4,5,-6,7]); w := Vector([0,1,2,3,4,5,6]);
```

$$v := \begin{bmatrix} 1 \\ -2 \\ 3 \\ -4 \\ 5 \\ -6 \\ 7 \end{bmatrix}$$

$$w := \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

We can find the dot product and the norm of the vectors. We can then use this information to find the angle between these two vectors. We need to use the option 'Euclidean' for reasons that are explained in upper level math classes.

```
> a := DotProduct(v,w);
```

$$a := 24$$

```
> nv := VectorNorm(v,Euclidean); nw := VectorNorm(w,Euclidean);
```

$$nv := 2\sqrt{35}$$

$$nw := \sqrt{91}$$

```
> theta := arccos(a/(nv*nw));
```

$$\theta := \arccos\left(\frac{12}{3185}\sqrt{35}\sqrt{91}\right)$$

We can convert this to a decimal number (in radians) using the command 'evalf'.

```
> theta := evalf(theta);
```

$$\theta := 1.356529607$$

To convert this to degrees we simply multiply by 180/Pi. Note that we need to use a capital P if we want to use the decimal approximation of pi.

```
> thetadegrees := evalf(180*theta/Pi);
```

$$thetadegrees := 77.72342126$$

We can also check the Cauchy-Schwarz inequality $|v \cdot w| \leq \|v\| * \|w\|$ and the triangle inequality $\|v + w\| \leq \|v\| + \|w\|$.

```
> abs(a);
```

$$24$$

```
> evalf(nv*nw);
```

$$112.8716085$$

```
> u := v+w;
```

$$u := \begin{bmatrix} 1 \\ -1 \\ 5 \\ -1 \\ 9 \\ -1 \\ 13 \end{bmatrix}$$

```
> un := VectorNorm(u,Euclidean);
```

$$un := 3\sqrt{31}$$

```
> un := evalf(un);
```

$$un := 16.70329309$$

```
> evalf(nv + nw);
```

$$21.37155158$$

%%%

This next section will provide an introduction to working with matrices. We begin by creating our 3 by 3 matrix A.

```
> restart;
> with(LinearAlgebra):
> A := Matrix([[-2, 1, -1],[1, -2, -4],[-1, 2, 3]]);
```

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

One of the nice things about Maple is that it can deal with large matrices. We will stick to the 3 by 3 case for the time being, just so we can plot some of the results.

Remember, one of the most important problems in linear algebra is solving the system$A\boldsymbol{x}=\boldsymbol{b}$. Let's see what happens when we try to do this with our matrix A. Suppose the vector$\boldsymbol{b}$that we are given is

```
> b := Vector([1,0,0]);
```

$$b := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Is there an $\boldsymbol{x}$ so that $A\boldsymbol{x} = \boldsymbol{b}$? There are multiple ways to answer this question. One way is to look at an arbitrary vector $\boldsymbol{d} = A\boldsymbol{w}$ and see if we can find any patterns in this vector that will give us a clue as to what sort of vectors $\boldsymbol{x}$ will solve the system.

```
> w := Vector([x,y,z]);
```

$$w := \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

```
> d := MatrixVectorMultiply(A,w);
```

$$d := \begin{bmatrix} -2x + y - z \\ x - 2y - 4z \\ -x + 2y + 3z \end{bmatrix}$$

In this case, it is hard to see much of a pattern. We can, however, do Gaussian elimination to determine what is going on. We begin by creating an *augmented* matrix that is of the form [A,$\boldsymbol{b}$]. We can then perform Gaussian elimination on this matrix, and it will give us a system of the form $U\boldsymbol{x} = \boldsymbol{c}$. Then back substitution will be straightforward. We first create our augmented matrix:

```
> AA := Matrix([A,b]);
```

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

Notice the matrix A in the first 3 by 3 portion of this matrix and $\boldsymbol{b}$ is in the final column. We can perform Gaussian elimination as follows:

```
> ENA := GaussianElimination(AA);
```

$$ENA := \begin{bmatrix} -2 & 1 & -1 & 1 \\ 0 & -3/2 & -9/2 & 1/2 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

We can find the solution using the following command:

```
> xx := BackwardSubstitute(ENA);
```

$$xx := \begin{bmatrix} -2/3 \\ -1/3 \\ 0 \end{bmatrix}$$

This means that the columns of the matrix A are independent, so linear combinations of these columns fill all of three dimensional space.

What if the columns are not linearly independent? Then there is either no solution or there are infinitely many solutions to $A\boldsymbol{x} = \boldsymbol{b}$, depending on the vector $\boldsymbol{b}$. In the next example, we'll look at the row picture of a matrix to see what happens.

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

Consider the vectors $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$, given below.

```
> u := Vector([-2,1,-1]); v:= Vector([1,-2,2]); w:= Vector([-1,-4,4]);
```

$$u := \begin{bmatrix} -2 \\ 1 \\ -1 \end{bmatrix}$$

$$v := \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix}$$

$$w := \begin{bmatrix} -1 \\ -4 \\ 4 \end{bmatrix}$$

Notice that $\boldsymbol{w} = 2\boldsymbol{u} + 3\boldsymbol{v}$, so these vectors are dependent. Consider the matrix $A = [\boldsymbol{u}\ \boldsymbol{v}\ \boldsymbol{w}]$. We want to see if there are any vectors $\boldsymbol{x}$ so that $A\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{b}$ is given.

```
> A := Matrix([u,v,w]);
```

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

We will consider the following vector $\boldsymbol{b}$:

```
> b := Vector([1,0,0]);
```

$$b := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

We will do the same procedure as before, where we create an augmented matrix and then use Maple to perform Gaussian elimination followed by back substitution.

```
> AA := Matrix([A,b]);
```

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

```
> ENA:= GaussianElimination(AA);
```

$$ENA := \begin{bmatrix} -2 & 1 & -1 & 1 \\ 0 & -3/2 & -9/2 & 1/2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Notice that the last line gives us the equation $0 = 0$. Thus***z***is a*free variable*, so there are infinitely many solutions.

```
> xx := BackwardSubstitute(ENA);
```

$$xx := \begin{bmatrix} -2/3 - 2\,\_t_1 \\ -1/3 - 3\,\_t_1 \\ \_t_1 \end{bmatrix}$$

We pick the third component of this vector to be any number, sayt_1. Then the first component is -2/3 - 2*t_1, and the second component is -1/3-3*t_1.

Let's first determine the row picture of this system. The three equations are:

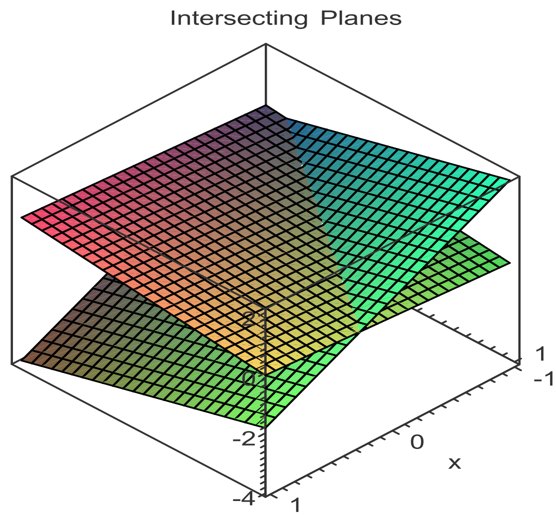
-2x + y - z =1
x -2y -4z =0
-x +2y +4z =0

We can solve each of these equations for$z$to obtain 3 planes. They are

z =-2x + y -1
z = x/4 - y/2
z = x/4 -y/2

We can plot these planes and see what is happening.

```
> plot3d([-2*x+y-1,x/4-y/2],x=-1..1,y=-1..1,axes = boxed, title
= "Intersecting Planes",glossiness = 0.9);
```



%%%

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

Finally, we can also perform row operations using the elementary matrices we discussed in class on Wednesday. Suppose we want to solve the system$A\boldsymbol{x}=\boldsymbol{b}$, where$Aand\boldsymbol{b}$are (the matrix$A$is called a*tridiagonal*matrix):

```
> A := Matrix([[-2, 1, 0],[1, -2, 1],[0, 1, -2]]);
```

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

```
> b := Vector([-1, 1, 2]);
```

$$b := \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix}$$

```
> AA := Matrix([A,b]);
```

$$AA := \begin{bmatrix} -2 & 1 & 0 & -1 \\ 1 & -2 & 1 & 1 \\ 0 & 1 & -2 & 2 \end{bmatrix}$$

Our first pivot is -2, and we would like to eliminate the 1 in the second row ofAA, so we will use the elimination matrix E_21, with the multiplier L_21 = 1/(-2) = -1/2. Remember, this means we start with the identity matrix and then insert the number -L_21 into the (2,1) entry of E_21.

```
> E_21 := Matrix([[1,0,0],[1/2,1,0],[0,0,1]]);
```

$$E\_21 := \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
> AA1 := MatrixMatrixMultiply(E_21,AA);
```

$$AA1 := \begin{bmatrix} -2 & 1 & 0 & -1 \\ 0 & -3/2 & 1 & 1/2 \\ 0 & 1 & -2 & 2 \end{bmatrix}$$

The second pivot is -3/2, and the mulitplier we would like to use is L_32 = 1/(-3/2) = -2/3. The elimination matrix E_32 is, in this case, given by:

```
> E_32 := Matrix([[1,0,0],[0,1,0],[0,2/3,1]]);
```

$$E\_32 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2/3 & 1 \end{bmatrix}$$

```
> AA2 := MatrixMatrixMultiply(E_32,AA1);
```

$$AA2 := \begin{bmatrix} -2 & 1 & 0 & -1 \\ 0 & -3/2 & 1 & 1/2 \\ 0 & 0 & -4/3 & 7/3 \end{bmatrix}$$

We now have an upper triangular system $U\boldsymbol{x}=\boldsymbol{c}$, where U and $\boldsymbol{c}$ are:

```
> U := SubMatrix(AA2,[1..3],[1..3]);
```

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

```
> c := SubMatrix(AA2,[1..3],[4]);
```

$$c := \begin{bmatrix} -1 \\ 1/2 \\ 7/3 \end{bmatrix}$$

In the 'SubMatrix' command, the syntax [1..3],[1..3] tells Maple to build a submatrix from rows 1 through 3 and columns 1 through 3 of the matrix AA2. The command [1..3],[4] tells Maple to build a vector from rows 1 through 3 and column 4 of the matrix AA2.

We can find the solution to our system by using the 'BackwardSubstitute' command.

```
> x := BackwardSubstitute(AA2);
```

$$x := \begin{bmatrix} -1/4 \\ -3/2 \\ -7/4 \end{bmatrix}$$

Let's check our answer–we need to make sure that $A\boldsymbol{x} = \boldsymbol{b}$.

```
>  MatrixVectorMultiply(A,x);
```

$$\begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix}$$