0 Learning to Use Maple V

In order to start a Maple V session when sitting at a UNIX workstation you must first issue the following command from an xterm.

```
machine%  add maple
```

You will see a response like the following:

```
Maple V Release 2 - Mathematical Manipulation Language
Use the following commands to start Maple:
        xmaple          to start Maple with the X Windows interface
        maple -x        same as 'xmaple'
        maple           to start Maple with the text interface
```

This indicates that Maple V is in the correct path to be invoked. So now issue the command:

```
machine%  xmaple
```

This will cause the application Maple V to be launched. A new window will be opened with the Maple V worksheet. The window includes a menu bar with menus entitled: **File, Edit, Options, Input, Interrupt, Pause, and Help**. The cursor should be located to the right of the prompt which is the symbol $>$. At this point you can start issuing Maple V commands. For example, if you type in the number "247*3756;" at the prompt and press the return key, the workstation will respond with the number "927732." What this means is that you have asked Maple V to multiply the two numbers "247" and "3756" and Maple V responds with the product of the two numbers "927732." The Maple V segment would look like the following.

```
> 247*3756;
                                  927732
```

> **NOTE:**
> Every Maple V command must end with either a semicolon or a colon. If the command ends with a colon then the calculation is made but no output is printed. When a semicolon is used the calculation is made and the result is printed.

What happens if you use the colon?

```
> 247*3756:
```

Notice that no result is printed, on the other hand the computation was made as we will see later.

What happens if you omit the semicolon or colon? The following segment shows that the semicolon is omitted in the first command, and no calculation is made which results in a syntax error when the next command is entered.

```
> 247*3756
> 2+3;
syntax error:
2+3;
^
```

Maple V permits full screen editing in that you can go back to any previous calculation and edit it and perform a new calculation. The following segment illustrates what happens if you go back to the command where the semicolon was omitted and then type in a semicolon and press return.

```
> 247*3756;

                                  927732
> 2+3;
syntax error:
2+3;
^
```

Notice that the next command remains unchanged. You may now press the return key again and you get.

```
> 247*3756;
```

$$927732$$

```
> 2+3;
```

$$5$$

Arithmetic

The basic arithmetical operations such as addition, multiplication, division, and raising to a power are known to Maple V.

$+$ and $-$	add and subtract
$*$ and $/$	multiply and divide
$\hat{}$ or $**$	raise to a power

Maple V is an interactive program that permits the user to enter commands at the prompt, press the return key, and read or use the output in other calculations. You can add two numbers.

```
> 253+7775;
```

$$8028$$

You can add fractions.

```
> 25/27+3/51;
```

$$\frac{452}{459}$$

You can perform operations on a previous result by using the “ditto” symbol, or double quotes ”. Thus the next calculation multiplies the answer obtained in the last calculation by 23.

```
> 23*";
```

$$\frac{10396}{459}$$

Two quotes refer to the next to the last result.

```
> 23*"";
```

$$\frac{10396}{459}$$

One can raise a number to a power.

```
> 3^7;
```

$$2187$$

```
> 3**7;
```

$$2187$$

A feature of most computer algebra sytems is that that they use exact arithmetic. For example, if you divide two integers Maple V returns an exact answer.

```
> 3235/7478;
```

$$\frac{3235}{7478}$$

There is a built-in Maple V function **evalf** - evaluate using floating-point arithmetic, that will give the decimal to *n* digits.

```
> evalf(");
```

```
.4326023001
```

The default number of digits used in floating point output is 10, but if you wish to have any other number of digits then you can specify them when using **evalf**. The following is the 30 digit floating point approximation of the fraction $\frac{3235}{7478}$.

```
> evalf("",30);
```

```
.432602300080235357047338860658
```

If you have a question about a Maple V command you can use the Maple V on line help facillity. If you wish to know what on line help is available for the **evalf** command or any other Maple V command then you may request it by typing a question mark followed by the name of the command and then pressing the return key.

```
> ? evalf
```

If you do this a Maple V help window will be opened that tells all about the command such as the syntax, the options, and also often gives examples. Another, perhaps better, way of obtaining help is to click on the **Help** menu in the upper right-hand corner of the Maple V worksheet and selecting the "Maple Help Browser." The use of this browser should become your preferred method for obtaining answers about how to use Maple V after your own trial and error methods can't answer the question that you might have.

Constants and Built-in Functions

The following table lists some of the constants that are built into Maple V. Note that Maple V is case sensitive, *i.e.*, the symbols "a" and "A" are regarded as distinct symbols.

Constant	Notation in Maple
e	E
π	Pi
$\sqrt{-1}$	I

There are also many functions that are included. An essential part of the study of calculus is the study of elementary functions such as polynomials, rational functions, trigonometric, exponential, and logrithmic. All of these will be discussed in the next chapter. Whenever you have a question see if you can answer it yourself through the use of the help facillity. Consider the following Maple V segment.

```
> Pi;
```

$$\pi$$

```
> evalf(");
```

```
3.141592654
```

```
> evalf("",50);
```

```
3.1415926535897932384626433832795028841971693993751
```

```
> sin(Pi/4);
```

$$\frac{1}{2}\sqrt{2}$$

```
> evalf(");
```

$$.7071067810$$

```
> ln(E);
```

$$1$$

```
> sqrt(3);
```

$$\sqrt{3}$$

```
> evalf(",30);
```

$$1.73205080756887729352744634151$$

```
> I^2;
```

$$-1$$

Defining Variables and Functions

You can assign a value or a function to a variable with the colon-equal symbol ":=".

```
> A := 5;
```

$$A := 5$$

```
> A;
```

$$5$$

This means that the variable "A" has been assigned the value 5 and it will have this value through the remainder of the session unless you assign it another value or **unassign** it.

```
> 4*A +12;
```

$$32$$

The following statement is the way to **unassign** the variable.

```
> A := 'A';
```

$$A := A$$

```
> A;
```

$$A$$

There are essentially two ways of working with functions. One way is to define a function as a variable. Suppose we wish to analyze the function given by $f(x) = x^2$. Then we can enter the following.

```
> f := x^2;
```

$$f := x^2$$

The last command defines the function and you can check that out with the following command.

```
> f;
```

$$x^2$$

There is a Maple V procedure called **subs** which allows you to evaluate this expression. The format is **subs**(variable = something, expression involving variable).

```
> subs(x=5,f);
```

$$25$$

A word of warning here. Many beginners want to use standard functional notation for a Maple V expression, such as $f(5)$. This results in nonsense and is not understood by Maple V at all.

```
> f(x);
```

$$x(x)^2$$

```
> f(5);
```

$$x(5)^2$$

If you want to use standard functional notation then you can do so using the minus-greater than symbol "->," made by typing the "minus sign" followed by the "greater than" sign. For example:

```
> f := x -> x^2;
```

$$f := x \mapsto x^2$$

```
> f(x);
```

$$x^2$$

Now we have

```
> f;
```

```
f
```

```
> f(5);
```

```
25
```

If you already have f defined as an expression and want to convert it to a function, you can use the Mapple V command **unapply**. Suppose that g is already defined and you wish to convert it to a function.

```
> g := x^3;
```

$$g := x^3$$

The expression g does not evaluate as a function.

```
> g(5);
```

$$x(5)^3$$

```
> g := unapply(g,x);
```

$$g := (x- > x)^3$$

```
> g(5);
```

125

Which is the best way to define functions? It depends on the situation. All three methods are used.

Algebra

One of the great benefits of a computer algebra system such as Maple V is that it allows you to to manipulate algebraic expressions much in the same way that a calculator permits you manipulate numbers. Some of the algebra commands used in Maple V are listed below.

Maple Command	What it does
simplify	Simplifies an algbraic expression
expand	Expands an expression
factor	Factors an expression
solve	Solves a system of equations for a set of unknowns

We give a sequence of statements or Maple V segment which illustrates each of these commands.

```
> expand((x^2-4)*(x+1)*(x-2)*(x^2+x+1));
```

$$x^6 - 6x^4 - 3x^3 + 6x^2 + 12x + 8$$

```
> factor(");
```

$$(x+2)(x+1)\left(x^2+x+1\right)(x-2)^2$$

```
> sol := solve({2*x-5*y=12,12*x+4*y=17},{x,y});
```

$$sol := \left\{y = -\frac{55}{34}, x = \frac{133}{68}\right\}$$

Observe that we used braces "{" and "}" in the preceding Maple V input. Maple V understands this to be a **set**. Observe that the Maple V output for the expression "sol" is in the form of a set. One thing to notice about sets is that they do not distinquish as to order. For example, Maple V might have equally well have produced an equivalent output of

$$sol := \left\{x = \frac{133}{68}, y = -\frac{55}{34}\right\}.$$

Maple V regards a set as a kind of array and you can pick out its elements. For example the last Maple V output indicated above is a set consisting of two elements: the first is the equation $y = -\frac{55}{34}$, and the second element is the equation $x = \frac{133}{68}$. The way to select the first element is by entering "sol[1];" and pressing return.

```
> sol[1];
```

$$y = -\frac{55}{34}$$

The second element is obtained by entering "sol[2]" and pressing return.

```
> sol[2];
```

$$x = \frac{133}{68}$$

Notice that these answers are given in the form of an equation. There are two useful Maple V commands for equations: **lhs**-left-hand side, and **rhs**-right-hand side.

```
> rhs(sol[1]);
```

$$-\frac{55}{34}$$

```
> lhs(sol[1]);
```

y

You may use **subs** to check your answer.

```
> subs(sol,{2*x-5*y=12,12*x+4*y=17});
                                {12 = 12, 17 = 17}
```

Observe that we have employed three different kinds of grouping symbols: "()","{}", and "[]". They are used for different purposes and Maple V requires that you use it correctly. The standard parentheses "()" are used with functions as in **factor**(x^2-1), or **sin**(Pi). The braces "{ }" are used to group a set of things together as in {x,y}. The square brackets are used to pick a coordinate out of a group as in sol[2]. There will be other examples of the use of these brackets in the future.

Two-Dimensional Plots

Maple V has a procedure **plot** for graphing a function of one variable. Thus if a function f is defined on an interval $[a, b]$ you can plot it with one of the following commands.

$$\mathbf{plot}(f, a..b, options)$$

or

$$\mathbf{plot}(f(x), x = a..b, options).$$

We will illustrate this with a few examples. First define a function. We will use the function

$$f(x) = e^{-x} \sin 3x.$$

```
> f := x -> E^(-x)*sin(3*x);
```

$$f := x \mapsto E^{(-x)} \sin(3\,x)$$

Note: The command

```
> f := x -> exp(-x)*sin(3*x);
```

defines the same Maple V function since to Maple V the expressions **E^x** and **exp(x)** are the same. We now plot this function over the interval [0, 3]. See Figure 1.

```
> plot(f,0..3);
```

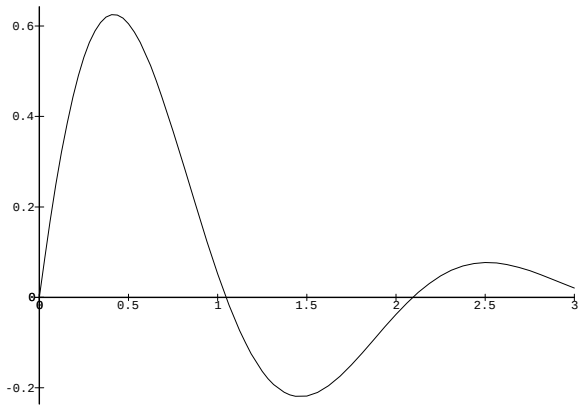

Figure 1: Graph of $\exp(-x)\sin(3x)$ (1)

Now we can plot the same curve with the $x-$ and $y-$axis labelled. See Figure 2.

```
> plot(f(x),x=0..3,y=-0.3..1);
```

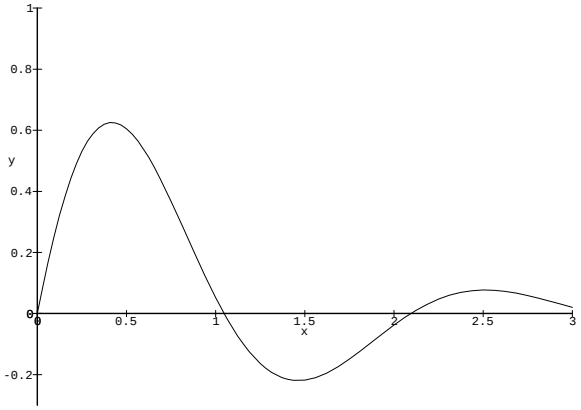

Figure 2: Graph of $\exp(-x)\sin(3x)$ (2)

Don't confuse the two formats, because Maple V won't create the plot. For example, if you express the function f without the argument x and specify $x = 0..3$, then you will only get a coordinate axis for a plot. We do not print out this plot here.

```
> plot(f,x=0..3);
```

On the other hand if you use the following statement then you get an empty plot.

```
> plot(f(x),0..3,-0.3..1);

Warning in iris-plot: empty plot
```

You should carefully read the help information for the **plot** command which will be used a lot in the course of this text. There is a special Maple V Library called **plots** that has many useful procedures defined. In particular, there is the command **display**. There are two ways to invoke a procedure from a Library Package like **plots**, for example you can access all of the routines with the command **with(plots);**. You can also invoke **display** using the command **plots**[**display**]({set of plots }). Sometimes we wish to printout text information on our plots. The **plots** procedure **textplot** works for this. The following will serve as an example that uses the **plots** routines **textplot** and **display**. See Figure 3. make sure that you use colons and semicolons where they are used below.

```
> plot1 := plot(f,0..3):
> plot2 := plots[textplot]([0.7,0.8,‘x=0.4336,y=0.615‘],align = RIGHT):
> plots[display]({plot1,plot2});
```

In the preceding it is important to use the colons to punctuate the statements, otherwise you will get a page full of Maple V text defining the **Plot Structure** rather than a graph. The **textplot** command is used to print text at the point with coordinates (0.7,0.8). The text is aligned to the right in this case. Note the use of backward quotes, ‘, that surround the text statment: ‘x=0.4336,y=0.615‘.

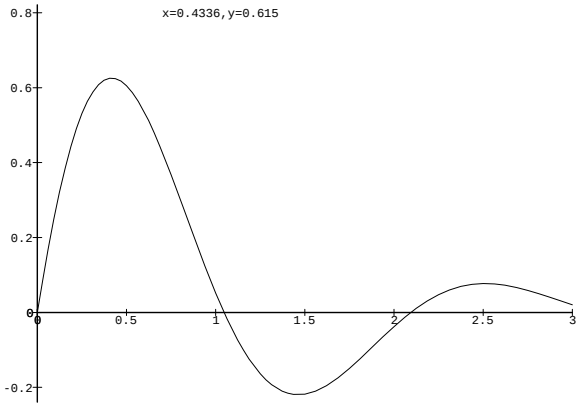

Figure 3: Graph of $\exp(-x)\sin(3x)$ (3)

The *seq*, *map*, and *zip* Commands

If you are already familiar with a programming language like Fortran, C, or Pascal then you probably have learned how to write programs with "loops". You can also write programs in Maple V using loops, but Maple V has some built-in procedures that makes this practice almost never necessary. Suppose that we are given n data points in the form of two lists. X and Y. For example the data might be stored as follows:

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

> Y := [3,2,5,7,1];
                        Y := [3, 2, 5, 7, 1]
```

Note: another use for the square brackets "[]" is to enclose a Maple V array known as a **list**. A list is similar to a set except the order is necessary in a list. For example the *sets*

$$\{a, b\} \quad \text{and} \quad \{b, a\}$$

are the same, but the *lists*

$$[a, b] \quad \text{and} \quad [b, a]$$

are not. If we want to plot the curve containing the points $(X[i], Y[i])$, *i.e.* (0,3),(1,2),(2,5),(3,7), and (4,1), then we need to provide the **plot** command with a list of pairs in the form:

$$[[0, 3], [1, 2], [2, 5], [3, 7], [4, 1]].$$

If you have experience with one of the traditional programming languages mentioned above you would probably form this list using a "loop" as follows (For the resulting plot see Figure 4):

```
> List := NULL;

                              List :=

>  for i from 1 to n do List := List,[X[i],Y[i]] od;
```

```
                                List := [0, 3]

                             List := [0, 3], [1, 2]

                          List := [0, 3], [1, 2], [2, 5]

                       List := [0, 3], [1, 2], [2, 5], [3, 7]

                    List := [0, 3], [1, 2], [2, 5], [3, 7], [4, 1]

> List := [List];
                  List := [[0, 3], [1, 2], [2, 5], [3, 7], [4, 1]]

> i := 'i';
                                       i := i
> plot(List);
```

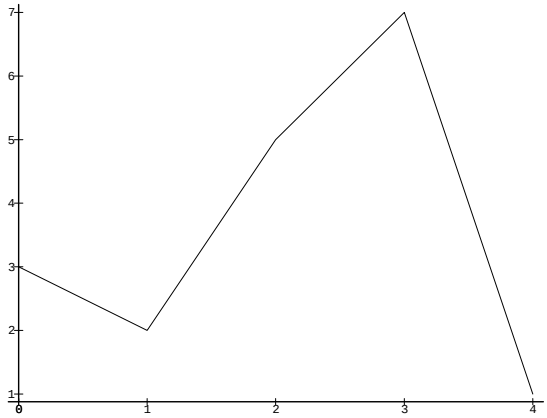

Figure 4:

Note that the last statement "> i := 'i'; " that **unassigns** "i" is necessary if you are going to use the variable "i" again, since it comes out of the above sequence assigned the value 6. If the method used above to construct the list that is denoted here by the variable "List" seemed unnatural don't worry about it. Using the Maple V commands **seq**,**map**, and **zip** will usually make it unnecessary to make these complicated loops. For example, the above list, denoted by *List*, can be constructed in one step.

```
> List := [seq([X[i],Y[i]],i=1..5)];
                  List := [[0, 3], [1, 2], [2, 5], [3, 7], [4, 1]]
> i := 'i';
                                       i := i
```

One can say that the preceding use of the **seq** command "zipped up" the two lists X and Y to create the final list *List*. Maple V actually has a built-in command called **zip** that will also do this. The format for this command is **zip**(f,X,Y) where f is a function of two variables and X and Y are lists. In our example, we want f to be a function that assigns the value $[x, y]$ to the pair (x, y). Thus as with **seq** the variable *List* can be assigned using only one line.

```
> List := zip((x,y)->[x,y],X,Y);
              List := [[0, 3], [1, 2], [2, 5], [3, 7], [4, 1]]
```

Suppose that you have a list of numbers X and you wish to evaluate a function f at each number in the list. The Maple V command **map**(f,X) will perform this action without the necessity of creating a loop. Study the next example.

```
> X := [0,Pi/4,Pi/3,Pi/2];
```

$$X = [0, \frac{\pi}{4}, \frac{\pi}{3}, \frac{\pi}{2}]$$

```
> Y := map(sin,X);
```

$$Y := [0, \frac{1}{2}\sqrt{2}, \frac{1}{3}\sqrt{3}, 1]$$

```
> Z := evalf(Y);
                    [0, .7071067810, .8660254040, 1.]
```

The first command assigns to X the list

$$[0, \frac{\pi}{4}, \frac{\pi}{3}, \frac{\pi}{2}].$$

The next command assigns to Y the list

$$[\sin(0), \sin(\pi/4), \sin(\pi/3), \sin(\pi/2)].$$

The final command assigns to Z the floating point evaluation of each number in Y.

You will find that these three commands will be very useful when dealing with large lists. You will seldom find it necessary to use a loop in this course.