Introduction to Maple, Part 1

Bent E. Petersen

Jan 13, 2000
Revised Feb 1, 2002

These notes[1] are part 1 of an introduction to using Maple in the MLC Computer Lab. The notes are based on notes that I produced for Linear Algebra and Numerical Analysis over the last several years.

Contents

1 Introduction to Maple

When you are using Maple interactively in a window environment such as Windows NT it is possible to move around on the worksheet by left-clicking the mouse to move the insertion point. As a result, you may end up executing commands in an order different from the order in which they appear on the worksheet. This can cause some confusion, since there is no visual clue. One way to fix a mess is to have Maple re-execute the entire worksheet in order from beginning to end (look on the Edit menu). Re-executing the worksheet may not be enough if you have changed some Maple parameters along the way (for example, the precision, Digits). You can reset Maple to its initial state (almost) by executing the command

```
> restart;
```

Note each Maple command must be terminated by a colon or a semicolon. You can spread the command over several lines by postponing the terminating colon or semicolon. You simply move to the next line by pressing Enter. Maple may chatter at you when you move to a new line in this way without terminating a command. Don't worry about it. Everything will be fine if you eventually terminate your command correctly.

[1] This document is available online, http://www.onid.orst.edu/~petersen

You can also stack up several commands on one line by terminating them individually with colons or semi-colons. The effect of the colon is to suppress output from the corresponding command, though the command is still carried out.

Do note that commands are not executed until you press Enter. Just because a command shows up on the worksheet does not mean it has been executed. This fact is particularly important to remember if you edit a command that you have previously entered. The new version will not replace the old one before you press Enter.

Maple has some built in constants

```
> Pi; evalf(Pi); I; I^2;
```

$$\pi$$

$$3.141592654$$

$$I$$

$$-1$$

Note the upper case letters. If you enter pi you will just get the Greek letter π, not the real number π. By the way, the evalf() function takes a second parameter specifying the precision in decimal digits.

```
> evalf(Pi,60);
```

$$3.14159265358979323846264338327950288419716939937510582097494$$

Specifying the precision inside the evalf function in this way does not affect Maple's default precision.

You can also set the precision by assigning a value to Digits (the default is 10). This action changes Maple's default precision for all subsequent calculations. Maple usually does exact calculations, but when floating point numbers are involved the Digits sets the precision. Here's an amusing example

```
> Digits:=4:  convert(evalf(Pi),rational); Digits:=10:
```

$$\frac{22}{7}$$

It is not obvious, but the conversion to a rational number makes use of Digits. Thus we would not get the correct result if we simply pass 4 as the second parameter in evalf().

If by some chance you want a rational number which yields π to 100 decimal places you can proceed as above

```
> Digits:=100:  convert(evalf(Pi),rational); Digits:=10:
```

$$\frac{4731971026350510701148958282416551810987607945 9487}{15062331588226261899547243783212652098313206994173}$$

In the above commands we see the Maple assignment operator is :=. Be careful about that! For example consider:

```
> a = 10; a; a:=10; a;
```

$$a = 10$$

$$a$$

$$a := 10$$

$$10$$

The first 'equals' statement has no effect. Maple just echoes it as a literal string. The second 'equals' statement however assigns the value 10 to a. Note by the way, simply entering a variable, followed by a semicolon, causes Maple to echo its value.

You can also unassign in Maple. Thus to restore a to its pristine unassigned state do

```
> a:='a';
```

$$a := a$$

In general a variable is unassigned by assigning it its name in single quotes. There is a bit of magic here. Putting the single quotes around a prevents Maple from evaluating a and so a is assigned its name. Maple does have an unassign function but it only works on unevaluated names, so you still need the single quotes. The unassign function does have the advantage that it accepts more than one name.

```
> a:=10; b:=20; c:=30;
```

$$a := 10$$

$$b := 20$$

$$c := 30$$

```
> a; b; c;
```

$$10$$

$$20$$

$$30$$

```
> unassign('a','b','c');
> a; b; c;
```

$$a$$

$$b$$

$$c$$

Another way unevaluate a variable is to use the evaln() function to obtain the name of the variable, and then assign the name to the variable Thus

```
> a:=5;
```

$$a := 5$$

```
> a:=evaln(a);
```

$$a := a$$

```
> a;
```

$$a$$

The advantage over other methods is you do not have to remember what quotes to use - single, double, or backticks - since you don't use any.

2 Functions, Derivatives and Integrals

Maple distinguishes between functions and expressions. Here's one way to define a function:

```
> f:=x->sin(3*x+x^2);
```

$$f := x \to \sin(3\,x + x^2)$$

We can also define an expression:

```
> g:=sin(3*x+x^2);
```

$$g := \sin(3\,x + x^2)$$

That they are different we can see by trying to evaluate them:

```
> f(1.0); g(1.0);
```

$$-.7568024953$$

$$\sin(3\,x + x^2)(1.0)$$

Note by using a decimal point we forced Maple to do a floating point evaluation. We can also get the precise result of course

```
> f(1); g(1);
```

$$\sin(4)$$

$$\sin(3\,x + x^2)(1)$$

We can see very clearly that g is regarded as an expression involving x and $g(1)$ is just the juxtaposition of the expression g and the expression (1). To evaluate g at say $x = 1$ we use the substitute command subs()

```
> subs(x=1,g);
```

$$\sin(4)$$

We can also use the more natural eval() function in place of substitution.

```
> eval(g,x=1);
```

$$\sin(4)$$

We can convert an expression into a function by using the unapply() command

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

$$h := x \to \sin(3\,x + x^2)$$

Some Maple commands work on expressions, some work on functions, and some on both. For example, here are the derivatives of f and g.

```
> D(f); diff(g,x);
```

$$x \to \cos(3\,x + x^2)\,(3 + 2\,x)$$

$$\cos(3\,x + x^2)\,(3 + 2\,x)$$

In the case of the expression you have to specify the variable you are differentiating with respect to. It should now be clear how to take partial derivatives of expressions.

Now let's bring back another fond memory from calculus - the problem of integration. An example should suffice to see how Maple's integrate command works:

```
> int(x*cos(x),x);
```

$$\cos(x) + x\sin(x)$$

Here's another example where we introduce some more Maple features

```
> Int(1/(1+x^6),x) = int(1/(1+x^6),x);
```

$$\int \frac{1}{1+x^6}\,dx = \frac{1}{3}\arctan(x) - \frac{1}{12}\sqrt{3}\ln(x^2 - \sqrt{3}\,x + 1) + \frac{1}{6}\arctan(2\,x - \sqrt{3}) + \frac{1}{12}\sqrt{3}\ln(x^2 + \sqrt{3}\,x + 1) + \frac{1}{6}\arctan(2\,x + \sqrt{3})$$

Note the int() operator returns the integral, but the Int() operator just returns a symbolic expression for the integral. This behavior of the Int() function is known as postponed evaluation and is sometimes useful. The application here, to obtain our solution as an equation displaying the integral that we wish to evaluate, is trivial but very useful. A more important application of Int(), numeric quadrature, is discussed at the end of this section.

Here is a slicker way to obtain nice equations by using the ditto operator

```
> Int(1/(1+x^6),x):  %=value(%);
```

$$\int \frac{1}{1+x^6}\,dx = \frac{1}{3}\arctan(x) - \frac{1}{12}\sqrt{3}\ln(x^2-\sqrt{3}\,x+1) + \frac{1}{6}\arctan(2\,x-\sqrt{3}) + \frac{1}{12}\sqrt{3}\ln(x^2+\sqrt{3}\,x+1) + \frac{1}{6}\arctan(2\,x+\sqrt{3})$$

What happened here? First Maple accepted the Int() command but didn't return an expression because we terminated the command with a colon. The ditto operator, %, always refers to the immediate previously accepted expression. Here that is the unevaluated integral, which is returned as a symbolic integral. The value function evaluates its argument, and so yields the value of the integral.

The ditto operator is useful, but if your worksheet is executed out of order, it can become difficult to keep track of what the % operator refers to. Thus it is best to confine your use of the % operator to the case where it refers to an expression on the same line (as above).

Definite integrals are easily handled. Moreover we can use the trick above to obtain nice expressions.

```
> Int(1/(1+x^6),x=0..1):  %=value(%);
```

$$\int_0^1 \frac{1}{1+x^6}\,dx = \frac{1}{12}\pi - \frac{1}{12}\sqrt{3}\ln(2-\sqrt{3}) - \frac{1}{6}\arctan(-2+\sqrt{3}) + \frac{1}{12}\sqrt{3}\ln(2+\sqrt{3}) - \frac{1}{6}\arctan(-2-\sqrt{3})$$

An approximate numeric value may be obtained by applying the floating-point evaluation function evalf() to the symbolic value of the integral. Of course this only works if the symbolic value can be found. Maple can also use numeric quadrature to find the approximate value of an integral. This is done by applying evalf() to the unevaluated integral directly.

```
> Int(1/(1+x^6),x=0..1):  %=evalf(value(%),20);
```

$$\int_0^1 \frac{1}{1+x^6}\,dx = .90377177374877204682$$

```
> Int(1/(1+x^6),x=0..1):  %=evalf(%,20);
```

$$\int_0^1 \frac{1}{1+x^6}\,dx = .90377177374877204684$$

Notice the slight difference in the value of two answers. In the first case we integrate symbolically and then evaluate the resulting expression. In the second case we evaluate the integral directly by numeric quadrature. To illustrate the difference more cleary we consider an example where Maple can not evaluate the symbolic integral

```
> int(exp(x^5),x=0..1);
```

$$\int_0^1 e^{(x^5)}\,dx$$

Notice when Maple can not evaluate an intgral it returns it unevaluated. This allows us to continue to manipulate it in various ways. Now consider two examples:

```
> restart: readlib(showtime): on:

> evalf(int(exp(x^5),x=0..1),8);
```

$$1.2248935$$

```
time = 7.52, bytes = 4864654
> off:
> restart: readlib(showtime): on:

> evalf(Int(exp(x^5),x=0..1),8);
```

$$1.2248935$$

```
time = 0.52, bytes = 462410
> off:
```

The readlib(showtime) command loads a timing routine. I included the restart command to make sure we start (fairly) fresh. The first evaluation took 7.52 seconds, the second only 0.52 seconds. The reason is, in the first command Maple tried to evaluate the integral symbolically. After a long time it decided a symbolic answer is not possible and returned the integral unevaluated. At this point the evalf() function applied to the returned unevaluated integral called up a numeric quadrature. In the second command Maple did not waste any time searching for a symbolic solution because we requested that the integral be returned unevaluated. This sort of thing is one of the most important applications of the unevaluated form of the integrate command.

The same example may be used to illustrate a feature of Maple -

```
> restart: readlib(showtime): on:

> evalf(int(exp(x^5),x=0..1),20);
```

$$1.2248935036353106911$$

```
time = 7.94, bytes = 5320890
> evalf(int(exp(x^5),x=0..1),20);
```

$$1.2248935036353106911$$

```
time = 0.02, bytes = 4098
```

```
> off:
```

Maple keeps track of expressions that it has already evaluated. If it is asked to evaluate the same expression again, it simply remembers the value. Thus it is not necessary to assign the result of a calculation to a variable even if you plan to use it several times. I hope it is now clear why I had to use restart in the examples above.

Note by the way that Maple can handle some improper integrals

```
> Int(exp(-x^2),x=0..infinity): %=value(%);
```

$$\int_0^\infty e^{(-x^2)}\,dx = \frac{1}{2}\sqrt{\pi}$$

Bent E. Petersen
Department of Mathematics
Oregon State University
Corvallis, OR 97331-4605

phone numbers
office (541) 737-5163
home (541) 753-1829
fax (541) 737-0517

petersen@math.orst.edu
http://ucs.orst.edu/˜peterseb
http://www.peak.org/˜petersen