640:192:01 Part II: playing with algebra on `maple` 9/1/2005

The most attractive feature of `maple` may be its ability to do intricate symbolic computations. Just try this:

*(x+2*y)^5* **RET**

Darn it: lazy `maple`! There are ways of telling `maple` not to be so lazy, but we want to go slowly. Just type:

expand(%) **RET**

and see what happens. Type

x=2 **RET**

and see what happens. Then type

x **RET**

Now try

x:=2 **RET**

followed by

x **RET**

and appreciate the difference. The character string *:=* assigns the **value** on the right to the **name** on the left. Now type

x^3 **RET**

Did you expect that? Try to evaluate $(x+1)^8$ when x has value 2. After doing that, type

restart **RET**

then try

(x+1)^8 **RET**

again and "expand" it. Is the result unexpected? What do you think *restart* does, and (more importantly!) how could you check that restart has that function? Hint: try *help(almost any word!)* when you're curious or confused.

Please assign x the value 17 and then type the following character string **exactly** as written here **but** don't hit **RET** yet!

*x;2x;x2;x*2;x^2;2*x;2^x*

The semicolon **;** is used to separate several `maple` statements on the same input line. Think about what's here and try to predict what will happen. Now hit **RET** and fix up any problems and hit **RET** again. Were you correct?

Warning! In older `maple` implementations and even in `maple`'s current command line version, the character string *2x* will be rejected (with the message **syntax error, missing operator or `;` :**). The graphical interface to `maple 10` will accept *2x* as an implied `2*x`.

People are sometimes reluctant to use long variable names, but I think this can be a very useful `maple` feature. Long names can help you remember what entries represent during complicated computations. For example, try

sumsqrts:=sqrt(x)+sqrt(y)+sqrt(x) **RET**

followed by

expand(sumsqrts^3) **RET**

I agree that more letters take more time to type (and increase the chance for error), but remember this freedom exists: you can call something by a character string close to its real name or with some important attribute recognized. This can reduce confusion.

OVER

Here are a few other algebraic things:

factor(y^4–16) **RET**

We can tell `maple` to use imaginary numbers. See the *help* information about *factor*. Specifically, try the command *factor(y^4–16,I)* **RET**. `maple` is born knowing some constants. I is a number whose square is -1. You may be able to guess what *Pi* is. And the constant *infinity*. In older `maple` systems, E was the number whose decimal approximation begins 2.71828..., but the latest releases of `maple` don't have this. If you do need E, you can define it with the command *E:=exp(1)* **RET**.

Work with a typical expression occurring at the beginning of calculus: $\dfrac{(W+\triangle W)^5 - W^5}{\triangle W}$.

I used the variables W and *deltaW* in my `maple` analysis and (after *expand*ing!) got the usual calculus mess involving both W and $\triangle W$. Notice now (if you haven't already) that `maple` is "case-sensitive". Therefore x and X need not be the same. Be careful!

You can now have fun doing algebraic things which no sane human being would ever think of doing "by hand". For example, what is the coefficient of r^7 in $(r^2+3r+4)^{10}$? Please remember all the necessary parentheses and ^'s. (I think the answer is 175173120.)

`maple` can also substitute in algebraic expressions. Try

*subs(a=t,5a^3+3t*a+2sqrt(a))* **RET**

This command changes a to t. It isn't equality. Try the following command with no $*$ between t and a:

subs(a=t,5a^3+3ta+2sqrt(a)) **RET**

`maple` will compute exactly what you ask! In *3t*a* the initial *3* will multiply the product of the variables t and a. *3ta* is interpreted as a request to multiply the variable *ta* by 3.

Braces or curly brackets, { and }, are used to create a list of variables for the *subs* command. Please try to predict what the result of the following will be **before** hitting **RET**:

*subs({a=t,b=t^2,c=t^3},a*b^2*c^3)* **RET**

`maple` can solve some equations. Try

solve(x^3=7x+1,x) **RET**

and

*solve({a*b+3=2,a+b=0})* **RET**

followed by

*solve({a*b+3=2,a+b=1})* **RET**

and I don't know why there's such a difference in the answer (hey, `maple` will tell you the roots of quadratics using the quadratic formula – just ask it). You could always try *help(solve)* which brings up a huge amount of information, and probably the reason for the difference is explained there somewhere!

`maple` will also find approximate numerical solutions. You could explore the difference in the answers to the command

solve(x^7–x^2+1); **RET**

(`maple` assumes you mean to ask for a root of the equation obtained by setting the expression $x^7 - x + 1$ equal to 0) and the command

fsolve(x^7–x^2+1); **RET**

which gives an approximate numerical solution.

Let's go on to calculus.