A Quick Tour of Maple

Jim Carlson, Jennifer Johnson

September 18, 1993

Contents

This quick tour is intended to give you an idea of how Maple can help you in a mathematics course. For more information see *A Brief Guide to Maple*. The best way to use the material below is to try out the Maple examples at the computer, preferably with a more experienced user to help you get started.

There is no need to read all the sections in order, although it is a good idea to read 1, 2, 3, and 4 first. The section on loops can be read immediately after the section on functions. It can can also be read with profit after reading the material on series. The material on linear algebra is independent of the material on calculus. The beginning calculus student should concentrate on the first part of section 5, reading the part on series and differential equations when ready. The student who needs to use Maple for vector and matrix computations can ignore the parts which he or she is unfamiliar.

A postcript version of this document is available in the directory `/u/cl/lib/maple` as `quicktour.ps`.

1. Basics

As you read through the examples below, try them out at a computer. Do each example as it is given! If you have trouble, compare what you have typed with what is written in the notes. Simple things, like forgetting a parenthesis, not matching parentheses, etc., can prevent Maple from understanding your intentions. If you still can't get Maple to do what you want it to, ask the lab assistant or someone sitting next to you for help.

Commands to Maple are written after the prompt, which on our system looks like this: `>` All Maple commands end with a semicolon, except for questions that you ask it. For example, if you want to know about factoring, you do this:

```
> ?factor
```

You can even ask for general help, which you do like this:

```
> ?
```

The most helpful part of Maple's help files are the examples. These are at the end of each help file.

Arithmetic

You can use Maple like a calculator to do arithmetic:

```
> (12.1 - 7.3)*(31 - 8.1) + 3.0^(2.1);

                                     119.9651086
```

The last number, `119.9651086`, is Maple's answer. Notice that multiplication is represented by `*` and exponentation (powers) by `^`. Sometimes you need to ask Maple to force a result into decimal form. This is what `evalf` does: it evaluates whatever you give it to "floating point" form. The quote sign `"` stands for the result of the last-computed expression. Note that Maple commands end with a semicolon.

Maple prefers to make exact computations when it can:

```
> 654321/123456;
                                   218107/41152
```

However, you can force it to convert the answer to decimal form ("floating point form") by using the `evalf` function:

```
> evalf( " );
                                        5.300034020
```

The quote sign " stands for the most recently computed quantity. You can do this all in one step if you like:

```
> evalf( 654321/123456 );
```

Here is one last example:

```
> sqrt(2);     # compute the square root of 2
> evalf(");    # convert to floating point form
> sqrt(2.0);   # another way of getting decimal answers
> Pi;
> evalf( " );
> evalf( Pi );
```

Try this now. Notice that if we use decimal numbers as inputs we tend to get decimal numbers as outputs.

Variables and algebra

Maple can also handle variables:

```
> F := m*a;              # the formula for force
> m := 2.1; a := 5;     # define mass and acceleration
> F;                     # compute the force
> a := 21.9;             # redefine the acceration
> F;                     # recompute the force
> a := 'a';              # clear a
> F;                     # recompute F
```

Try this now. The text following the # signs are *comments*; Maple ignores them.

With variables you can, of course, do algebra:

```
> factor( a^2 - b^2 );
> expand( (a + b)^5 );
> g := (a + 1)^2/(b-1)^3 + a/(b-1);
> simplify( g );
```

You can also make substitutions:

```
> subs( a = 1, g );
> simplify( " );
> subs( a = 1, b = 2, g );
> g;
```

Note that substitutions have only a temporary effect. See

```
> ?factor
> ?expand
> ?simplify
> ?subs
> ?collect
```

for more information.

Problems

1. Use Maple to define the formula for compound interest: $A = P * (1 + r/100)^n$, where P is the principal, r is the annual interest rate expressed as a percentage, n is the number of years the principal is invested, and A is the amount of capital (principal plus accumulated interest) at the end of n years. Now set P to \$100, r to 5. Compute A. Then compute A with $n = 1$, $n = 5$, and $n = 10$.
2. Find the coefficient of $a^2 b^7$ in $(2a + 3b)^9$.
3. Use the function `ifactor` to factor 123456789;

2. Functions

Many functions are predefined in Maple:

```
> sqrt(2);              # exact square root of 2
> evalf(");             # convert previous result to decimal form
> sin(Pi/2);            # pi is defined, angle is in radians
> arcsin(1/2);          # result, an angle, given in radians
> arcsin(1/4);          # !!!
> evalf( " );           # convert previous result to decimal form
> exp(2);               # the exponential of 2
> evalf(");             # convert previous result
> exp(2.0);             # get decimal result directly
> log(2);               # the natural log of 2
> evalf( log(2) );      # the natural log of 2 in decimal form
> exp(log(2));          # check!
> log(exp(2));          # another check
> ?sin                  # ask about the sine function; gives more info
```

Try this now.

You can also define your own functions. Let's begin by making a function which compute the sine of an angle given in degrees. We do it this way:

```
> Sin := x -> sin( evalf(Pi*x/180) );
```

The part after the arrow tells how `Sin` will compute `x`. The definition made, let's test it:

```
> Sin(30);          # sine( 30 degrees )
```

Next let's make a function which computes the arcsine, with the result in degrees:

```
> Arcsin := x -> evalf( 180*arcsin(x)/Pi );
```

Again we test:

```
> Arcsin(1/2);      # result is in degrees
> Sin(Arcsin(1/2);  # check!
> Arcsin(Sin(90);   # another check
```

You can even define functions of several variables:

```
> distance := (u,v,x,y) -> evalf( sqrt( (u - x)^2 + (v - y)^2 ));
```

This function computes the distance from the point (u, v) to the point (x, y).

Problems.

1. The shadow cast by a flagpole is 14 feet long when the angle the sun's rays make with the vertical is 11 degrees. How tall is the flagpole?
2. Use the `distance` function defined above to see how far (11.34, 24.17) is from (−9.61, 12.49).

3. Equations

Maple can solve equations:

```
> solve( x^2 + 3*x = 2.1 );
> solve( x^3 + x = 27 );     # gives exact but complicated answer
> solve( x^3 + x = 27.0 );   # better results
```

Notice that small changes, such as using 27.0 instead of 27, force Maple to give decimal approximations. This is sometimes preferable to exact but complicated answers.

It is helpful to be able to use the results of solving one equation in another problem. Here is one example of how to do this:

```
> s := solve( x^2 + 3*x = 2.1 );  # store solutions in s
> s[1] + s[2];                     # sum of first and second solution
> s[1]*s[2];                       # their product
> cos( s[1] );                     # apply a function to one of them
```

We can also solve systems of equations:

```
> solve( { 2*x + 3*y = 1, 5*x + 7*y = 2 }, {x, y} );
```

Of course, you can have many equations and many unknowns. If you want x and y to take the values given by solving this equation, the following will work:

```
> s := solve( { 2*x + 3*y = 1, 5*x + 7*y = 2 }, {x, y} );
> assign(s);
> x; y;        # check to see if it works
```

Even shorter is

```
> assign( solve( { 2*x + 3*y = 1, 5*x + 7*y = 2 }, {x, y} ) );
```

For more information about solve and related commands try

```
> ?solve      # for exact solutions
> ?assign     # to assign solution values to variables
> ?fsolve     # for numerical solutions
> ?factor
```

These queries will give information about useful options, e.g., the `complex` option for `fsolve`. Rememmber to look at the examples at the end of the help file if you find the general description confusing.

Problems

1. Find as many roots of $x^4 + x = 16$ as you can. (It may help to write 16 as 16.0). How many roots do you expect on theoretical grounds?
2. Find as many roots of $x^5 + x = 32$ as you can. How many roots do you expect on theoretical grounds? You may also want to see how `fsolve` works on this one. Here are some things to try:

```
> solve( x^5 + x = 32 );
> solve( x^5 + x = 32.0 );
> fsolve( x^5 + x = 32.0, x );           # specify unknown
> fsolve( x^5 + x = 32, x, complex );    # specify complex option
```

Note that Maple may not always return the answers you expect. For example, `solve( x^2 + x = 32.0 )` *returns just one root — on my machine, with version of Maple, anwyay.*

3. Solve the system of linear equations below:

$$7x + 3y + 8z = 1$$
$$2x - 4y + 11z = 2$$
$$x - 7y + 9z = 1$$

4. Use `solve` and `fsolve` to find solutions of the system

$$x^2 + 3y^2 = 1$$
$$5x^2 + y^2 = 1$$

   Do the results agree? Can you draw a picture to clarify the situation? Ask yourself what kind of curves $x^2 + 3y^2 = 1$ and $5x^2 + y^2 = 1$ are. For more information on how to plot two curves like this, see the *Brief Guide*.

   Graphs can tell a lot about where the solutions of one or two equations lie. For example, the solutions of $f(x) = 0$ are the places at which the graph $y = f(x)$ crosses the x axis. The solutions of the two equations $f(x, y) = 0$ and $g(x, y) = 0$ are given by the points wher two curves cross, namely, the curves defined the two equations. The next section will tell you more about graphing. Consult also the *Brief Guide*.

4. Graphs

Maple can make many kinds of graphs. Here we will look at the most basic ones. Use `?plot` or see the *Brief Guide* for more information. To plot $y = x^3 - 3x^2 + x + 1$ on the interval $-4 \leq x \leq 4$ use

```
> plot( x^3 - 3*x^2 + x + 1, x = -4..4 );
```

You can also set the range on the y axis:

```
> plot( x^3 - 3*x^2 + x + 1, x = -4..4, y = -4..4 );
```

With the same range on both axes you get a truer picture. Note that you can find roots by looking for the places where the graph crosses the x-axis. If you like, you can title your graph:

```
> plot( x^3 - 3*x^2 + x + 1, x = -4..4, y = -4..4,
        title = `y = x^3 - 3x^2 + x + 1` );
```

Notice that your commands to Maple don't have to be confied to a single line.

Sometimes it is useful to plot more than one curve at once. The following example illustrates this:

```
> f := sin(x);
> g := x - x^3/6;  # Taylor's approximation to sin(x)
> plot( { f, g }, x = -Pi/2..Pi/2.
        title = `y = sin(x) and its 3rd order Taylor approximation`
```

Problems

1. Graph the the function $y = \cos(x) - (1/3)\cos(3x)$ on the interval $[-\pi, \pi]$. Then print your graph.
2. Graph $\cos(x) - (1/3)\cos(3x)$ and $\cos(x) - (1/3)\cos(3x) + (1/5)\cos(5x)$ simultaneously on the interval $[-\pi, \pi]$. Then print your graph.
3. The previous two problems establish a pattern: a sequence of functions with more cosine terms. The next one is

$$\cos(x) - (1/3)\cos(3x) + (1/5)\cos(5x) - (1/7)\cos(7x)$$

Try graphing several of these, and see if you can guess what the graph of the function with infinitely many terms is.

5. Calculus

Derivatives and Integrals

Maple can differentiate and integrate functions:

```
> n := 'n';                  # make sure n is cleared
> diff( 4*cos(x^3), x );     # differentiate
> int( x^n, x );             # indefinite integral
> int( x^n, x = 0..1 );      # definite integral
```

Sometimes results are correct but not in expected form:

```
> diff( x^n, x );   # ok, but not standard
> simplify( " );    # better, but not perfect
```

You should be aware that not all integrals can be expressed in terms of elementary functions, i.e., ones made up from four arithmetic operations, extracting roots, trig functions and their inverses, the logarithm, and the exponential. Here are two examples:

```
> int( sqrt(1 + x^2), x );
> int( sqrt(1 + x^3), x );
```

Maple gives a useful answer to the first integral, but expresses the second integral in terms of yet another indefinite integral — not too helpful. Nonetheless, we Maple can use a numerical method to find definite integrals even in these "bad" cases:

```
> int( sqrt(1+x^3), x = 0..1);
> evalf( " );
```

Notice that we had to use `evalf` to force numerical evaluation. We could have done this directly, of course:

```
> evalf( int( sqrt(1+x^3), x = 0..1) );
```

You can tell whether the number just computed is "in the ballpark" by thinking about what the integral means: it is the area under a graph. Try

```
> plot( sqrt( 1 + x^3 ), x = 0..1, y = 0..1.5 );
```

to see if this is reasonable.

Series

Maple can find Taylor series of functions:

```
> series( sin(x), x = 0);       # expansion around x = 0
> series( sin(x), x = 0, 10);   # more terms
> evalf( " );                   # convert coefficients to decimals
> ?series                       # more info
```

You can also compute partial sums of any series. For example, to find

$$1 + 1/2 + 1/3 + \cdots + 1/10,$$

we do this:

```
> sum( 1/n, n = 1..10);
> evalf( " );                           # convert to decimal form
> s := n -> sum(1/i, i = 1..n);         # useful function
> s(5);                                 # sum of first five terms
> s(10);                                # sum of first ten terms
> evalf( " );                           # convert to decimal
> ?sum                                  # more info
```

Alternating series require a bit more thought. First we define a function which returns 1 if n is odd, -1 if n is even:

```
> sgn := n -> (-1)^n;  # easy, but inefficient
> sgn(1);
> sgn(2);
```

The rest is easy:

```
> t := n -> sum( -sgn(i)/i, i = 1..n );
> t(1);
> t(2);
> t(3);
```

Maple can compute some limits:

```
> limit(sin(x)/x, x=0);
> ?limit                       # more info
```

This occasionally works with series:

```
> w := n -> sum( 1/i^2, i=1..n);
> limit( w(n), n=infinity );
```

Differential equations

Maple can handle some differential equations. For example, to solve

$$\frac{dy}{dx} = ky,$$

the equation governing radiactive decay, population growth, etc., we do this:

```
> dsolve( diff(y(x),x) = k*y(x), y(x) );
> sol := dsolve( { diff(y(x),x) = k*y(x), y(0) =100}, y(x) );
> k := 1; sol;
> k := 2; sol;
```

Note that we can specify initial conditions and can experiment with parameters. If you want to graph the solution, you need only the right-hand side, for which the `rhs` function is designed:

```
> plot( rhs(sol), x = 0..1 );
```

If you want to define a function from what `dsolve` gives you, do it this way:

```
> f := x -> rhs(sol);
> f(1); f(2);
```

Let us now consider a somewhat more realistic populution growth model:

$$\frac{dy}{dx} = ky(1-y)$$

If we try

```
> de := diff( y(x), x ) = k*y(x)*( 1 - y(x) );
> dsolve( de, y(x) );
```

we don't get what we expect. The following works better:

```
> dsolve(  de, y(x), explicit );
```

We can also impose initial conditions:

```
> dsolve( { de, y(0) = 0.1 }, y(x), explicit );
```

Finally, let's try a second-order equation,

$$m\frac{d^2y}{dx^2} + ky = 0,$$

the equation governing the motion of a frictionless mass-spring system. To solve it we say

```
> de := m*diff(y(x),x$2) + k*y(x) = 0;
> dsolve( de , y(x) );
> sol := dsolve ( {de, y(0) = 1, D(y)(0) = 1 }, y(x) );
> sol := simplify( sol );
```

You will need to know about imaginary exponentials to interpret the answer to the first application of `dsolve` The second application gives a correct but complicated answer, so we simplify it. There is a lot more to learn about `dsolve`, so consult `?dsolve`.

Problems

1. Find the derivative of $x^5 + 7x^2 + x + 1$. Then use `solve` to locate the critical points. Determine the nature of the critical points (maximum, minimum, inflection). Finally, verify your answer by graphing. Hint: use constants in floating point form, e.g., 7.0 instead of 7 to force floating point results.
2. Find $\int \sqrt{1 + 7x^2}\, dx$.
3. Find $\int_0^1 \sqrt{1 + 7x^2}\, dx$ in both exact and floating point form.
4. Find the Taylor series of $\sqrt{\cos(x)}$.
5. Compare the graph of the fourth order Taylor polynomial of $\sqrt{\cos(x)}$ with the graph of $\sqrt{\cos(x)}$ itself. Do this on the interval $[-\pi/2, \pi/2]$. Can you draw any conclusions from what you see?
6. Compute the sum of the infinite series
$$1 + 1/2^3 + 1/3^3 + 1/4^3 + \cdots$$
which you think is accurate to two decimal places. Do this first by finding partial sums (the sum of the first n terms).

   Can you get something accurate to four decimal places? Can you do still better? Can you estimate the error committed by using a partial instead of infinite sum? Finally, try the combination of `sum` and `limit` as described above to see what you get.
7. Repeat the previous problem for the infinite series
$$1 - 1/2^5 + 1/3^5 - 1/4^5 \pm \cdots$$
Do you get better or worse answers given the same amount of computational work?

6. Linear algebra: vectors and matrices

Maple is also equipped to do linear algebra, i.e., computations with vectors and matrices. We give only a few examples here and refer to the *Brief Guide* for more information. If there is a matrix notion, e.g., eigenvalue, which you are not familiar with, you may ignore it for now.

To begin you must (always) load the linear algebra package:

```
> with(linalg);
```

Now we define two vectors:

```
> v := vector( [1, 2] ); w := vector( [3,1] );
```

To display a vector or matrix use `print`:

```
> print(v);
```

Here are some vector operations:

```
> add(v, w);        # vector sum
> scalarmul(v, 3)   # multiply v by 3
> dotprod(v, w);    # dot product v.w
```

Next, we define two matrices:

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

Some basic operations with them:

```
> add(A,B);
> multiply(A,B);
> multiply(B,A);
> scalarmul(A,7);
> inverse(A);

> det(A);
> trace(A);

> transpose(A);
> multiply(A,v);
> multiply(transpose(v),A);
```

There are also some commands for building big matrices from small ones, for extracting parts of matrices, and for dealing with individual elements:

```
> stack(A,B);
> augment(A,B);
> ?submatrix

> A[1,2];              # get a component of A
> v[1];                # get a component of v
> A[1,2] := 9;         # change a component of A
> print(A);            # view the whole matrix
> A[1,2] := 1;         # change A back to original value
```

Maple can do Gaussian elimination as well as elementary row and column operations:

```
> C := randmatrix(4,6, entries=rand(0..4));
> C2 := gausselim(C);
> C3 := gaussjord(C);
> ?addrow
> ?mulrow
> ?swaprow
```

It can also find eigenvalues and eigenvectors:

```
> eigenvals(A);
> ? eigenvals
> ? eigenvects
> ? Eigenvals
```

Finally, let's work a little problem: we solve the equation

$$Ax = v$$

where we still have

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \qquad v = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Recall that even though Maple writes v as a row vector (to save space), it thinks of it as a column vector). To solve our equation we multiply on the left by A^{-1} and simplify to get

$$x = A^{-1}v$$

Thus, our solution is given by

```
> AA := inverse(A); x := multiply(AA,v);
```

Remarks: It cumbersome to type out long words like multiply. Here is a shortcut:

```
> alias(m=multiply);
> m(A,B);            # same as multiply(A,B);
```

Another way to do things is like this:

```
> evalm( A + B );    # same as add(A,B);
> evalm( 7*B );      # same as scalarmul(B,7);
> evalm( A &* B );   # same as multiply(A,B);
> evalm( (3*A + 5*B) &* (A - B) );
> evalm( (A + B) &* (A - 2*B)^(-2) );
```

The last expression computes

$$(A + B)(A - 2B)^{-2}$$

Maple needs the ugly symbol `&*` to distinguish matrix and scalar multiplication.

Problems (from the *Brief Guide*)

Vectors.

In the problems below, let $v = (1, 2, 3)$, $w = (-1, 4, 1)$.

1. Find the sum and dot product of v and w.
2. Compute the length of v using the formula

$$\text{Length}(v) = \sqrt{v \cdot v}.$$

3. Define a length function for vectors by

```
Length := v -> sqrt( dotprod(v,v) );
```

Then compute `Length(v)` and make sure that it agrees with what you obtained in the previous problem. Finally, convert to floating point form with `evalf`, e.g., `evalf(Length(v))`.

4. The angle between two vectors is given by the formula

$$v \cdot w = |v||w| \cos \theta$$

where $|v|$ stands for the length of v. Define a function for computing angles by

```
Angle := (v,w) -> arccos( dotprod( v, w ) / (Length(v)*Length(w)) )
```

Then use this function to compute the angle between v and w. You may want to use `evalf` to put the result in decimal form.

5. Is the result of the previous problem in radians or degrees? You may want to modify the definition of `A` to suit your preferences as regards unit of measure.

6. Maple also contains a `crossprod` function:

$$v \times w = crossprod(v, w)$$

Use it to compute $v \times w$ and comment on the geometric significance of the result.

Matrices.

In the problems below, let

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \qquad B = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

7. Compute AB and BA. Are they the same?
8. Compute $A + B$ and $B + A$. Are they the same?
9. Compute $7A$.
10. Define `C := add(A,B)`. Then compute C^2.
11. Compute the determinants of A and B. Is there a geometric interpretation of your result?
12. Where possible, compute the inverses of A and B. Check your result by multiplying: If $U := A^{-1}$, then $UA = I$ and $AU = I$, where I is the appopriate identity matrix.

13. Compute Av. Can you also compute vA? (Think about row versus column vectors).
14. Let x be an unknown vector. Solve the equation $Bx = v$. You can do this by noting that $x = B^{-1}v$. Check your result.
15. Can you solve the equation $Ax = v$? In class you have learned (or will learn) that you can do this by applying Gaussian elimination to the matrix $(A|v)$, gotten by putting v as a "new" column to the right of A. You can do all of this with `gausselim( augment(A,v) )`. The `gaussjord` function works even better.
16. Can you solve the equation $Ax = u$, where u is the column vector with components $(6, 15, 24)$? Hint: apply Gaussian elimination to $(A|u)$.

7. Loops

Sometimes you need to do a computation over and over. Here, for example is how to make a table of squares:

```
> for i from 1 to 10 do i^2 od;
```

Whatever ever is between `do` and `od` is done first for $i = 1$, then for $i = 2$, and so on, up to $i = 10$. Such a gizmo is called a *loop* . We can use it to make a more elaborate table of squares like this;

```
> for i from 1 to 10 do print(i, i^2) od;
```

Here is a table of primes:

```
> for i from 1 to 10 do ithprime(i) od;
```

Loops can be used for more sophisticaed things than tables. We give two examples. First, we compute the quantity

$$1 + 1/2 + 1/3 + \cdots + 1/n$$

Here is how it is done:

```
> S := 0;
> for i from 1 to 10 do S := S + 1.0/i  od;
```

We wrote $1.0/i$ instead of $1/i$ to force floating point compuaation. In the next example we start with the number 1, then apply `sin` to it, then apply `sin` to the result, and so on, for a total of 10 applicatons of the sine fuction. This is called *iteration of mappings*.

```
> x := 1;
> for i from 1 to 10 do x := evalf( sin(x) ) od;
```

Problems

1. Make a table of square roots of the integers from 1 to 20.
2. Compute the quantity

$$1 + 1/2^2 + 1/3^2 + \cdots + 1/10^2$$

3. Let $f(x) = (x^2 + a)/2x$, where $a = 2$. Begin with $x = 1$. Apply f to it. Apply f to the result. Repeat this process ten times. Here is a suggested way to define f:

```
> f := x -> evalf( (x^2 + a)/(2*x) );
> a := 2;
```

The `evalf`, as usual, forces floating point evaluation.

Does the sequence of numbers you generate this way seem to get closer and closer to a fixed number (the *limit*). If so, what is it? Repeat the process with $a = 3$, $a = 4$. Can you guess what the pattern is? Is it part of a more general pattern? Is this process a useful one?