

Part II. Calculations and Calculus with Maple

Section 2. Symbolics: Equations and Assignments

Maple is an excellent calculator. However, calculus and differential equations applications require symbolic capabilities. Although Maple makes symbolic computations in a natural way (see the examples in Section 1), it is essential that users have a clear understanding of how Maple handles variables, equations, and functions.

Variables, equations and assignments

The equation $x = 3$, the expression $y + x$, and the expression y/x all involve the variables x and y . They are entered separately below. Note that $x = 3$ and y/x terminate with semicolons so Maple displays the output. The $y + x$ entry is terminated with a colon, so output is suppressed.

```
> x = 3; y + x: y/x;
```

$x = 3$

$\frac{y}{x}$

- **Key observation:** The equation $x = 3$ does not assign the value 3 to the x variable. This can be seen from the output for the third entry: y/x . Maple echoed the entry back in the output, only in what is referred to as "prettyprint form": $\frac{y}{x}$. This is typical of the output for an entry that Maple does not process. The output is the same as the input, only prettyprinted.

It is possible to assign the number 3 to the variable x , but it must be done using the combination of a colon and an equals sign shown below

$:=$

This is called the **assignment operator**. (No spaces between $:$ and $=$ please.)

The next input contains the same entries, except for the first one where the equation $x = 3$ has been replaced by the assignment $x := 3$. Compare the output to what is displayed above.

```
> x := 3; y + x: y/x;
```

$x := 3$

$\frac{1}{3}y$

Once the number 3 has been assigned to x , it stays that way. Examine carefully the three terms in the output sequence for the next entry.

```
> y + x, x^3 - x*z, (x^2)!;
```

$y + 3, 27 - 3z, 362880$

There are three ways to undo the assignment of 3 to the x variable. One way is to enter

```
unassign('x');
```

as shown below.

```
> unassign('x'); y + x;
```

$y + x$

Note that an "unassign" entry has no output.

Another way to free the x variable is to enter the assignment

```
x := 'x';
```

Think of this as assigning the symbol x to the variable x .

The third way to unassign x is to restart the Maple kernel either by issuing the restart command

```
restart;
```

or by clicking on the restart button on the tool bar. The restart button is the double arrow icon just to the right of the bug, see page 4. Doing this will unassign all of the assigned variables in a worksheet.

Multiple assignments: Full evaluation

Maple will not object if several assignments are made to the same variable. It only remembers the most recent one. This is illustrated in the next input consisting of four entries (two outputs are suppressed).

```
> x := 1; x := 2: x := 3: x := 'x';
```

$x := 1$
 $x := x$

To see that x is currently unassigned, simply enter it.

```
> x;
```

x

Suppose that several assignments are made in a "chain" like this.

```
> x := y^2: y := 2*z: z := 3*ln(w): w := exp(1);
```

$w := e$

Then w is assigned the number e

```
> w, evalf(w);
```

$e, 2.718281828$

and Maple will dig down all the way to e when asked to evaluate x , y , or z . This is referred to as "full evaluation".

```
> x, y, z;
```

$36, 6, 3$

Make sure that you understand the output. What is $\ln(w)$? Why did x evaluate to 36?

If the assignment to w is subsequently changed:

```
> w := sin(theta);
```

$w := \sin(\theta)$

The change will be reflected in the evaluation of x , y , and z .

```
> x,y,z;
```

$$36 \ln(\sin(\theta))^2, 6 \ln(\sin(\theta)), 3 \ln(\sin(\theta))$$

The next entry restarts Maple and then checks that the four variables: x , y , z , and w are unassigned.

```
> restart;
```

```
x, y, z, w;
```

x, y, z, w

As is the case with **unassign**, the **restart** procedure has no output. Note that the two entries in this input appear on two different lines. This was accomplished by first typing **restart**, then pressing [shift-return] to move the cursor to the next line. Pressing [return] won't work, of course, because [return] causes Maple to process the input and jump the cursor to the next input prompt (or create a new one if necessary).

- **Important input rule.** With very few exceptions Maple ignores extra spaces and [shift-return]s in an input. Maple users are encouraged to make judicious use of spacing and carriage returns to make input entries as easy to understand as possible.

Before we can do any calculus applications we must learn more about how Maple solves equations, both exactly and approximately.

Solving equations exactly: The solve procedure

Maple's **solve** procedure is very powerful. Ask Maple for exact solutions for x in "equation" by entering the following:

```
solve( equation, x );
```

If exact solutions are found, they are returned in a sequence.

```
> solve( x^3 - x = x^2, x );
```

$$0, \frac{1}{2} + \frac{1}{2} \sqrt{5}, \frac{1}{2} - \frac{1}{2} \sqrt{5}$$

If the unknown is enclosed in set brackets, then Maple returns the solutions in the form of a sequence equations, each one inside of set brackets. We will see later that this can be very useful.

```
> solve( x^3 - x = x^2, {x} );
```

$$\{x = 0\}, \left\{ x = \frac{1}{2} + \frac{1}{2} \sqrt{5} \right\}, \left\{ x = \frac{1}{2} - \frac{1}{2} \sqrt{5} \right\}$$

If the solution values are to be used in a subsequent input, then assign a name to the solve entry as shown in the next example. Full evaluation implies that the name assigned to the input will also be the name for the output. In the next example a name has also been assigned to the equation. This is recommended or two reasons:

1. The equation can be checked for correctness by examining the output.
2. If the equation has to be used again, simply enter its name.

```
> eqn := x^3 - 5*x^2 = 2 - 6*x;
```

$$eqn := x^3 - 5x^2 = 2 - 6x$$

```
> solns := solve( eqn, x);
```

$$solns := 1, 2 + \sqrt{2}, 2 - \sqrt{2}$$

Having named the solution "solns" the entry

```
solns[k];
```

can be used to obtain the kth solution in the sequence.

```
> solns[1];
```

1

```
> solns[2];
```

$$2 + \sqrt{2}$$

```
> solns[3];
```

$$2 - \sqrt{2}$$

Check by substitution: simplify if needed

Use the substitute procedure (**subs**) to check a solution. The syntax for making a sequence of substitutions for variables v1, v2, ... , vn in a "target" expression is

```
subs( v1=expr1, v2=expr2, ... , vn=exprn, target )
```

The substitutions are made one at a time, in this order. For example, to substitute the third solution, solns[3], into "eqn" make the following entry.

```
> subs( x=solns[3], eqn);
```

$$(2 - \sqrt{2})^3 - 5(2 - \sqrt{2})^2 = -10 + 6\sqrt{2}$$

When it makes a substitution using **subs**, Maple does not automatically **simplify**. To force simplification apply **simplify**:

```
simplify(%);
```

```
> simplify(%);
```

$$-10 + 6\sqrt{2} = -10 + 6\sqrt{2}$$

The output identity confirms that solns[3] satisfies the equation.

If floating point solutions are preferred, apply the evalf procedure to the sequence named solns.

```
> evalf[4](solns);
```

1., 3.414, 0.586

Solving equations approximately: The fsolve procedure

The **solve** procedure often fails to return satisfactory solutions because Maple cannot find an exact solution formula. When this happens, floating point solutions can be obtained by using the **fsolve** procedure. The syntax for **fsolve** is

```
fsolve( equation, x, a..b);
```

This entry asks Maple for a 10 digit solution to "equation" somewhere in the interval from $x = a$ to $x = b$. The next input requests a solution to "eqn" (defined above) in the interval from $x = 3$ to $x = 4$. Compare the output to the middle term in the last output sequence.

```
> fsolve( eqn, x, 3..4);  
3.414213562
```

Read the word "fsolve" as "solve in floating point form". The **fsolve** procedure uses a numerical algorithm and usually returns just one solution. However, if the equation contains only polynomials, replacing the search interval with the keyword "complex" tells Maple to display a sequence of all of the solutions in floating point form.

In the next input we have

1. Entered and named the polynomial equation $x^5 - x^4 + x^3 + x - 1 = 0$,
2. Generated a sequence of floating point solutions named solns (output suppressed), and
3. Applied **evalf** to display the solution sequence using 3 digit representations.

```
> eqn := x^5 - x^4 + x^3 + x - 1 = 0;  
solns := fsolve( eqn, x, complex):  
evalf(solns,3);  
eqn := x5 - x4 + x3 + x - 1 = 0  
-0.580 - 0.739 I, -0.580 + 0.739 I, 0.712, 0.723 - 1.03 I, 0.723 + 1.03 I
```

Observe that the third solution in the solution sequence is purely real.

```
> solns[3];  
0.7124526119
```

The other four solutions are complex, appearing in conjugate pairs (the symbol I is used by Maple to denote $\sqrt{-1}$):

```
> solns[1]; solns[2];  
-0.5797227182 - 0.7393202821 I  
-0.5797227182 + 0.7393202821 I
```

The next entry checks the third solution. Because a float has been substituted, the output is in floating point form and automatically simplified.

```
> subs(x=solns[3],eqn);  
-1 10-10 = 0
```

The output should be $0 = 0$. It is not because the "solution" is not in exact form. Note, however, that the left hand side of the output equation is as close to zero as it could possibly be for a 10 digit float. The symbol $-1 \cdot 10^{(-10)}$ means $-1 \times 10^{(-10)}$ ($= -0.0000000001$).

More about solve and fsolve

When the solve procedure cannot find exact solutions to an equation, it often returns expressions telling the user that the solution is a root of a certain function, i.e. a zero of the function.

```
> eqn := tan(x) = x;
   solns := solve( eqn, x);
```

$eqn := \tan(x) = x$
 $solns := \text{RootOf}(-\tan(_Z) + _Z)$

We will take this as a signal to use **fsolve** with a search interval in the form **a..b** as shown below.

```
> fsolve( eqn, x, 3..6);
   fsolve( eqn, x, 6..9);
```

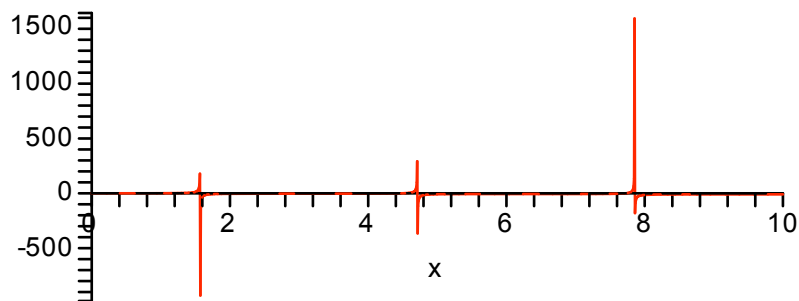
4.493409458
7.725251837

In order to search successfully it would be very helpful to have the graph of the function $\tan(x) - x$. This is easy to obtain using Maple's **plot** procedure. The **plot** procedure has the following basic form.

`plot(f(x), x=a..b)`

The next entry asks for the graph of $\tan(x) - x$ from $x = 0$ to $x = 10$.

```
> plot( tan(x) - x, x=0..10);
```

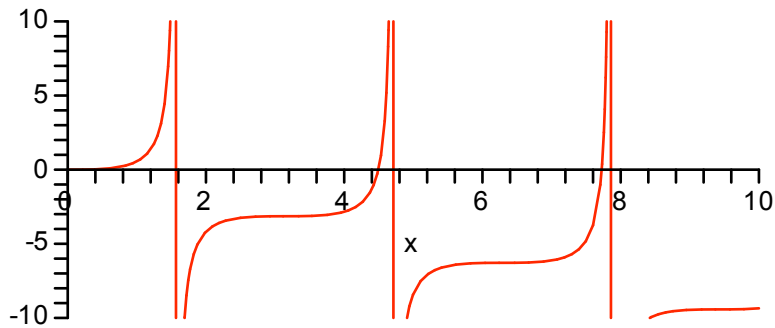


Because the tangent function has vertical asymptotes Maple plots points over a very wide vertical range (see the numbers on the vertical axis). This can be controlled by adding an optional vertical range specification in the form

`c..d`

as shown below.

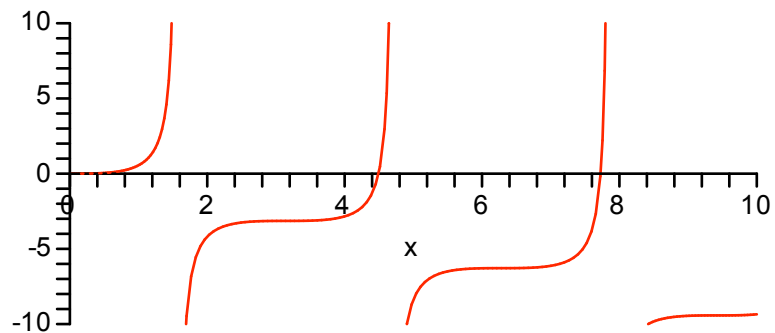
```
> plot( tan(x)-x, x=0..10, -10..10);
```



This is a much better picture. Observe, however, that the four branches of the graph are connected with vertical lines that appear to be the asymptotes. This is because Maple made the graph by plotting lots of points and connecting the dots. This is the default plot behavior. The next **plot** entry removes the vertical lines by adding the optional equation

```
discont=true
```

```
> plot( tan(x)-x, x=0..10, -10..10, discont=true);
```



It now appears that the only solutions to $\tan(x) = x$ in the interval from 0 to 10, other than $x = 0$, are the two that were just found using **fsolve**.

Using variables as functions

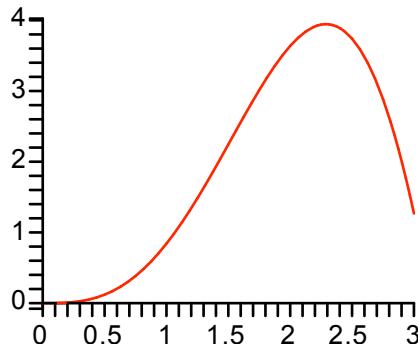
A typical calculus book is full of problems like the following.

Graph the function $y = x^2 \sin(x)$ over the interval from $x = 0$ to $x = 3$. Find its maximum and minimum values on this interval.

To handle this problem in Maple we can define the variable y as a function of x using the **assignment operator**. This is done in the first entry. The second entry uses the plot procedure to display the graph of the function over the given interval.

```
> y := x^2*sin(x);
plot( y, x=0..3);
```

$$y := x^2 \sin(x)$$



Clearly the minimum y value on this interval is $y = 0$, attained at the left endpoint where $x = 0$. To find the maximum y value we will compute the derivative function and then find its zero near $x = 2.5$.

The **diff** procedure can be used to obtain the derivative formula. The syntax for differentiating variable y with respect to variable x is simply

```
diff( y, x )
```

- We would like to name the derivative function y' but Maple does not permit the single forward quote (i.e. the prime symbol) in a variable name. We name it yp instead. (Think of it as " y prime".)

```
> yp := diff(y,x);
```

```
yp := 2 x sin(x) + x2 cos(x)
```

Now use **fsolve** to find the zero of yp between 2 and 3. The **solve** procedure will not work here; try it. We assign the name $xmax$ to the zero of yp so it can be used later to compute the maximum y value.

```
> xmax := fsolve( yp=0, x, 2..3);
```

```
xmax := 2.288929728
```

The maximum y value can now be found using substitution:

```
subs( x=xmax, y )
```

However, because **subs** does not simplify, the evaluate procedure: **eval**, is preferred for this sort of calculation. The general syntax for eval is

```
eval( expression, variable = value )
```

The maximum y value is named y_{max} .

```
> ymax := eval(y,x=xmax);
```

```
ymax := 3.945301625
```

Now that the coordinates of the high point have been found, it can be displayed on the graph by adding the point (x_{max}, y_{max}) to the plot procedure.

The syntax for plotting two expressions over the interval from $x = a$ to $x = b$ is

```
plot( [expression1, expression2], x=a..b);
```

Note that the two expressions to be plotted are entered in a **list**.

Maple also uses **list** notation

[xmax,ymax]

to denote a point and the plot procedure requires that points be placed inside of list brackets. Therefore, the plot entry that will show both the graph of y and the high point has the following basic form

```
plot( [ y, [ [xmax,ymax] ] ] , x = 0..3 );
```

The optional equation

```
style=[line,point]
```

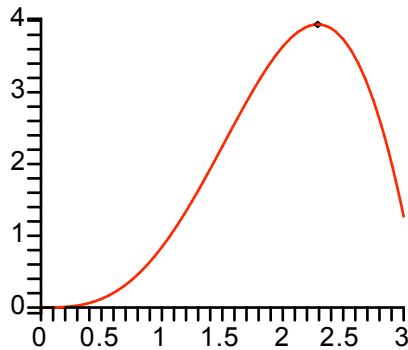
is needed to tell Maple to plot the y curve using the "line style" (i.e. connect the dots) and plot the **list** of points [[xmax,ymax]] in the "point style".

The color equation

```
color=[red,black]
```

tells Maple to make the y curve red and the high point black.

```
> plot( [ y, [ [xmax,ymax] ] ], x=0..3, style=[line,point],  
color=[red,black]);
```



Plots are discussed in more detail in the next section.