A Brief Maple Tutorial
Part 2: Functions of a Single Variable and Graphing

Michael Penna, Indiana University – Purdue University, Indianapolis

Objective

In this project we discuss how to define and graph functions in Maple.

Narrative

In Maple, the function $f = f(x)$ is defined by the command:

`f := x ->` <expression in `x`>

and:

`plot(f(x),x=a..b)`	plots f from $x = a$ to $x = b$,
`plot(f(x),x=a..b,y=c..d)`	plots f from $x = a$ to $x = b$, limiting output to points with y-coordinates between c and d,
`plot({f(x),g(x)},x=a..b,y=c..d)`	plots the functions f and g from $x = a$ to $x = b$, restricting output to points with y-coordinates between c and d.

Task

1. Type the command lines in the left-hand column below into Maple in the order in which they are listed. They draw several versions of the graph of $f(x) = \sin x$. The effect of each command is described in the right-hand column for your reference.

`> # Your name, today´s date`	
`> # Functions of a Single Variable and Graphing`	
`> # Task 1`	
`> restart;`	Clear Maple's memory.
`> f := x -> sin(x);`	Let $f(x) = \sin x$. (See Comment 1 below.)
`> f(Pi/2);`	Evaluate f at $x = \pi/2$. (Note that trig function arguments are measured in radians in Maple.)
`> plot(f(x),x=-Pi..Pi);`	Plot f over the interval $[-\pi, \pi]$.
`> plot(f(x),x=-Pi..Pi,y=-0.5..0.5);`	Plot f over the interval $[-\pi, \pi]$, restricting the range to $[-0.5, 0.5]$.
`> plot(f(x),x=-Pi..Pi,color=blue,scaling=constrained);`	Draw the graph of f in blue, and use the same units on the x- and y-axes.
`> plot(f(x),x=-1000..1000);`	Note how choosing the correct interval for graphing makes a big difference!
`> plot(f(x),x=-0.01..0.01);`	Again, choosing the correct interval for graphing makes a big difference! Note also how that the graph of f looks linear when you zoom in on it.

2. Continue by typing the command lines below into Maple. They draw several versions of the graph of $f(x) = \tan x$.

```
> # Task 2
> f := x -> tan(x);
> plot(f(x),x=-Pi..Pi);
> plot(f(x),x=-Pi..Pi,y=-1..1);
> plot(f(x),x=-Pi..Pi,y=-1..1,discont=true);
```

3. Continue by typing the command lines below into Maple. The second line illustrates how we can plot more than one function in one graphic.

```
> # Task 3
> plot({sin(x),cos(x)},x=-Pi..Pi);
```

4. Use the commands you've learned in this project to create the graph of a function of our own choice. (Be creative!)

At this time, make a hard-copy of your typed input and Maple's responses. Then, in each of the graphics you created, label by hand the graph of each function. (In the first graphic you created, for example, label the graph of f by "$f(x) = \sin x$".)

Your lab report will be a hard copy of your typed input and Maple's responses (both text and hand-labeled graphics).

Comments

1. Note that when using a function such as $\sin x$ in Maple you must type "`sin(x)`" (being careful to include parentheses) rather than "`sin x`" (which you might write by hand).

2. Maple allows you to control numerous aspects of the appearance of graphics via "options" such as

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

   While there are many advantages to using different colors when viewing a graphic, a potential disadvantage to using certain colors is that they don't always come out well when you print them. Since this issue is hardware dependent, try the code in the projects to follow as written, but if your graphics don't come out well when you print them, try another option such as "`color=black`" to get a better hard copy.

3. Maple has several "packages" of functions which, while not automatically loaded when you start a Maple session (to conserve on memory), are easy to load and very useful. For example, one of Maple's packages is the `plots` package; you load the `plots` package (or any package, for that matter) with the `uses` command:

   ```
   > uses(plots);
   ```

   And one of the useful functions in the `plots` package is the `setoptions` command: If you want to use the same options throughout a Maple session, you can specify them once at the beginning of the session using `setoptions`, and ignore them thereafter (saving time while reducing both typing and subsequent command length). For example, if we had included the command lines

   ```
   > uses(plots);
   > setoptions(color=blue);
   ```

   immediately after the `restart` command, then all subsequent plots would have (by default) included the option "`color=blue`".

4. Even though the graphs of curves may appear to be smooth, the way programs like Maple graph functions is by plotting a finite number of points and "connecting-the-dots" with short line segments. Thus, rather than seeing a smooth curve when you plot a function such as $\sin x$, you are actually seeing a polygonal approximation to its graph. (To see the dots that Maple uses to graph f, enter the command "`plot(f(x),x=a..b,style=point)`".)

5. To specify a piecewise-defined function such as

$$f(x) = \begin{cases} x^2 & \text{if } x \le 0 \\ 2x+1 & \text{if } 0 < x \le 1 \\ 4 & \text{if } x > 1 \end{cases}$$

in Maple you may use:

```
f := x -> if x <= 0 then x^2 elif x <= 1 then 2*x+1 else 4 end if;
```

In addition to the "`if/then/elif/else/end`" control structure, Maple offers many other structures; to learn more about them, check out Maple's Help. You may also use the `piecewise` command to specify a piecewise-defined function: to use this command to define the function f above, you could write

```
f := x -> piecewise(x<=0, x^2, x>0 and x<=1, 2*x+1, x>1, 4);
```

To learn more about the `piecewise` command, check out Maple's Help.