640:291:01 **Part IV: playing with graphs on `Maple`** **9/15/2002**

The graphing capabilities of `Maple` explored here may seem rather familiar to you after your experience with graphing calculators and possibly other programs or devices. I won't discuss polar plots or parametric plots here. `Maple` does really lovely pictures with three-dimensional plots. Begin with the simple command

`plot(x^3-6*x+1);` **RET**

After the program responds, move back and alter your command to read

`plot(x^3-6*x+1,x);` **RET**

Now a graph should be created.`Maple` needs to know the variable name!
For this kind of plot, `Maple`'s default interval for x (what it assumes unless advised otherwise) is $[-10, 10]$. This may be what you want (try graphing $\sqrt{x^2 - 100}$ and see what happens) but you may want more control. The `plot` command has a very large number of options. Read about them during the first free week you have by typing `help(plot)` and following all the references!
Look at the graph shown. When I am graphing, I tend to foul things up a great deal so I frequently end up with 5 or 10 graphs sitting around at the same time. Try to be neat and have everything labelled correctly. Of course, if the graphs hang around until you `quit Maple` entirely, they'll disappear also. But, meanwhile confusion will be increased. You can label things fairly easily. For example, the plot suggested above can be labelled with the command:

`plot(x^3- 6*x +1,x,title=‘cubic‘);` **RET**

The quotes around "cubic" are single opening quotes, on the upper left of a keyboard together with tilde: ˜.
Please look at the graph again. The vertical and horizontal axes are very differently scaled. `Maple` is trained to "autoscale" so it will **distort** the picture to fill the rectangular screen. I wrote that word very large because this feature has repeatedly surprised me, sometimes not pleasantly. Put your cursor on the graph, and right click: go to Projection and select Constrained which shows the way the graph really looks: very, very, very thin. Let's draw another graph.

`plot(sin(1/x),x=.001..1);` **RET**

This graph is attempting to show some misbehavior since $\lim_{x\to 0^+} \sin(\frac{1}{x})$ doesn't exist. The graph should bounce around a lot when x is close to 0. It certainly seems to, but notice (by pure thought) that between each root or x-intercept of the function, there must be a place where the function is $+1$ and there must be a place where the function is -1. If you look closely, the picture doesn't show this. `Maple` doesn't think about functions, it plots points and connects the dots to produce the graph. Initial performance of `plot` is fairly simple-minded (more sophisticated alternatives can be specified). The plotted points can be seen by right clicking on the plot, going to Style and then selecting Point. You'll see the computed points gotten from the function with no interpolated pixels darkened. If you increase the number of points to be plotted, you may get a better picture, but computation time can increase, sometimes a lot.
Here's another bad picture. You can get a nice curve if you type

`plot(1/x,x=.01..1);` **RET**

but things get bizarre and fairly useless if you change the domain to $[-1, 1]$ or to $[-2, 2]$. Remember to be careful. You can help `Maple` avoid nasty errors with these kinds of discontinuities. The `help` screen for `plot` is long with many entries in the SEE ALSO line. Another view of this curve, initially strange, is gotten with

`plot(1/x,x=-.1..(-.01));` **RET**

Again consider the location of the vertical axis. Does the line segment displayed go through the origin?. Warning: if you want to plot from .01 to .03 note that more parentheses are needed – as in `x=.01..(.03)` so that `Maple` will distinguish between the periods needed for a range of values (the ..'s) and the decimal point.
Try the command (but you may want to guess the result before completing the input):

`plot({x^2,x^3},x=4..5);` **RET**

Therefore you can plot collections of functions, also sometimes useful.
Now I'd like you to define a function corresponding to $P(w) = \dfrac{e^w}{1+w^2}$ (you create the appropriate `Maple` command). Remember that `Maple` finds the derivative function of P using `D(P)`. Then type this:

`plot({P(t),D(P)(t)},t=-2..5);` **RET**

OVER

As long as the variable (here, `t`) is used consistently, `Maple` will have no difficulty. Compare the two graphs. Where `D(P)` is 0 then `P` must have a horizontal tangent. Does this seem to occur? What happens to the graph of `P` when the graph of `D(P)` has a maximum?
Let's try to find a root graphically.

```
plot(x*ln(x)-sin(x),x=0..3); RET
```

This function seems to have a root. There are options which will allow you to "zoom" in on the crossing, but can you just alter the domain several times and try to get the crossing located to the nearest ten-thousandth? You can check your answer with a numerical computation:

```
fsolve(x*ln(x)-sin(x),x); RET
```

reports 1.752677281, close my guess at the root graphically. You may read about `fsolve` using `help`.
In a several variable calculus course, we should look at 2-dimensional representations of 3-dimensional graphs. The points (x, y, z) satisfying $z = x^2 + y^2$ can be graphed with the command

```
plot3d(x^2+y^2,x=-10..10,y=-10..10): RET
```

which should show you a square piece of a nice "cup" (hey, the function is non-negative, and as $|x|$ and $|y|$ get large the function's value shown in z direction increases. Put your mouse on the graph and try some of the options of the mouse buttons. See how the appearance changes! Rotate the graph. Look down and up at it, and look at it sideways.
Now try something more difficult. Where do the graphs of the functions $z = x^2 + y^2$ and $z = x^3 - xy^2$ intersect? You can plot more than one graph at a time. Try the command

```
plot3d({x^2+y^2,x^3-xy^2},x=-10..10,y=-10..10): RET
```

The two surfaces shown intersect. How many pieces of curves are there in the intersection?
The picture generated above is difficult for me to see carefully. I tried this instead.

```
RED:=plot3d(x^2+y^2,x=-10..10,y=-10..10,color=red,grid=[100,100]): RET
GREEN:=plot3d(x^3-x*y^2,x=-10..10,y=-10..10,color=green,grid=[100,100]): RET
```

The use of the colon (`:`) and not the semicolon (`;`) to end the commands helps a great deal. The "graphs" (a massive data structure!) are stored in the variables `RED` and `GREEN`. The command

```
display(RED,GREEN); RET
```

shows both of the graphs in strikingly different colors. I think it is easier to see the intersection curves with this picture. You can also rotate this, of course, and try to understand it. The following command sketches a "projection" of the intersection curves into the $(x, y)-$plane.

```
implicitplot(x^2+y^2=x^3-x*y^2,x=-5..5,y=-5..5,grid=[100,100]); RET
```

If you have done what is written here, this command will **produce no picture and no useful output**. Many `Maple` commands must be loaded before they can be used. Now type

```
with(plots); RET
```

which produces a list of procedures that have been loaded, including `implicitplot`. Redo the `implicitplot` command, and see how the curves drawn fit into the red/green graph already drawn. Now answer easily, "How many pieces of curves are there in the intersection?"
You may have forgotten one piece. Even after finishing this exercise, please realize that you have not proved your answer. The graphs drawn by `Maple` merely take approximate function evaluations and "connect the dots". They do not really verify statements, merely make suggestions. I can give examples from my own experience which show how misleading such computations can be. Thought is needed! A complete solution is available on a course web page: see the **Technology** page for a link to `intersect.html`.

That's about all we can do now.

Disclaimer! Non-advertisement!! Important information!!!

Symbolic manipulation programs such as `Maple` are increasingly available. `Mathematica` and `Derive` are other programs with the same capability. There are many special purpose programs in science, engineering, and mathematics which have extensive "intelligence" to analyze models. We're considering `Maple` here because Rutgers has a site license for this program. It is generally available on Rutgers systems. The specific instructions won't be the same from program to program, but many of the same ideas will be present. Students should expect to have a machine do tiresome or elaborate symbolic computations as well as numerical computations.