New Maple Commands

This week in Maple:

- Sets versus Lists in Maple.
- More information about plots.
- Sequences.
- The ”Do” Loop.
- Commands with Dates.

Sets and Lists

In Maple, a **Set** is a sequence enclosed by curly braces, and a **List** is a sequence enclosed by square braces. In a **set**, every element occurs only once- In a list, the order of the elements matters, and elements may be repeated (so typically we will use a **list**).

To see the difference, take a look at the following two commands:

```
A:=[2,3,2,3,1,4];
A:={2,3,2,3,1,4};
```

In the first case, Maple keeps the ordering (and double entries) intact. In the second case, Maple re-orders the numbers and removes repeated entries.

More with Plots

When plotting, you can use either a set or a list, but (in order to preserve order), I recommend you always use a list (square brackets). That is, if I want to plot the functions $x, x^2, \sin(x)$ on the same axes, either command will work:

```
plot( {x, x^2, sin(x)}, x=-Pi..Pi, color=[red, blue, green]);
plot( {x^2, x, sin(x)}, x=-Pi..Pi, color=[red, blue, green]);

plot( [x, x^2, sin(x)], x=-Pi..Pi, color=[red, blue, green]);
plot( [x^2, x, sin(x)], x=-Pi..Pi, color=[red, blue, green]);
```

I think we see that the second set of commands produces the plot that is more natural in terms of color assignment. The first two commands, the set is ordered by Maple first, then the colors are assigned.

There is one wrinkle- Plotting parametric curves. In this example, we plot $x(t) = \cos(t)$, $y(t) = \sin(t)$ in the (x, y) plane. In the second example, we show how to plot two parametric plots.

```
plot([cos(t),sin(t),t=0..2*Pi]);
plot([[cos(t),sin(t),t=0..2*Pi],[t^2,t^3,t=-1..1]],color=[red,blue]);
```

Here are some things you can do with the tickmarks of a plot:

- Plot $y = \sin(x)$ for x between -10 and 10. Use 6 tickmarks on the $x-$axis and 10 tickmarks on the $y-$axis:

  ```
  plot(sin(x),x=-10..10,tickmarks=[6,10]);
  ```

- Plot $y = \sin(x)$, for x between -6 and 6. Label the $x-$axis for each maximum as A, B, C, etc. The line break is for readability. In this case, we use the default markings for the $y-$axis.

  ```
  plot(sin(x),x=-6..6,tickmarks=[[-3*Pi/2="A", -Pi/2="B", Pi/2="C",
       3*Pi/2="D"],default]);
  ```

Sequences

Recall that a sequence is a list (using Maple vocabulary) of numbers. Usually that list is generated by some formula that depends on the index of the value (values are indexed by $1, 2, 3$, etc.

As a quick example, the following sequence of numbers:

$$1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \cdots$$

could be generated by the function $1/n$, where $n = 1, 2, 3, \cdots$.

Here is how we generate a sequence in Maple:

```
seq(f(n), n=a..b)
```

The sequence itself is just a set of numbers, which we can assign to a variable:

```
B:=seq(1/n, n=1..10);
```

We can make that a list by using square brackets: `[B]` or a Maple set: `{B}`. You can access elements of the sequence using square brackets. For example, `B[5]` should give you 1/5.

More examples of sequences

Here are a couple more examples of using sequences to generate lists of numbers.

```
#Print the square of the integers from 1 to 5:
seq(i^2,i=1..5);
#List integers from 0 to 100 in multiples of 10:
seq(0..100,10);
```

Series

Maple can also work with series. Here are a couple of examples- You can even get the sum of the alternating harmonic series!

```
sum(1/n^2,n=1..infinity);
sum((-1)^n/n,n=1..infinity);
```

Doing Something Over and Over and Over

If you need to perform some computation over and over, consider using a “do loop”. Here is a sample to try:

```
x:=sqrt(2.0);
for i from 1 to 10 do x:=sqrt(x); od;
```

What happened? *Result:* You should see that you first compute $\sqrt{2}$, then assign that result to x. Then compute $\sqrt{\sqrt{2}}$, then assign that result to x. Do that 9 more times.

In general, a loop to do `xxx` 10 times looks like this:

```
for i from 1 to 10 do xxx; od;
```

where xxx is done over and over again.

Here is another example, where we build the sequence a_k, where $a_1 = 3$, $a_2 = 1$, and:

$$a_{k+1} = 3a_{k-1} - a_k$$

```
a[1]:=3;  a[2]:=1;
for i from 2 to 8 do a[i+1]:=3*a[i-1]-a[i]; od;
```

And here is another example, where we print something to the screen **Typing Tip:** To get a new line (without Maple interpreting your commands), use `Shift + Enter` after each line, then just `Enter` when you’re finished. In the next example, we have Maple print out the k^{th} partial sum of the harmonic series (which should diverge). Does it look like diverges? (Try taking it out to 100, or 200).

```
S:=0;
for k from 1 to 10 do
   S:=S + 1.0/k;
   print(k,S);
od:
```

And one more example: Here, find the sum of all the odd numbers between 11 and 99. In the following example, we show an alternate way of ending the loop: “end do” in place of “od”.

```
N:=0:
for k from 11 by 2 while k<100 do
  N:=N+k;
end do:
```

To find the value of N, type: `N;`

Working with Dates

In this section, we take a look at some Maple commands that will allow us to work with calendar dates.

Be sure to type `with(Finance):` since these commands use the Finance package in Maple.

- `DayCount(Date1,Date2);` Counts the number of days that has elapsed between the two given dates.

  Example: Count the number of days since Elvis Presley was born (The King was born on January 8, 1935, but you probably already knew that). Here are two ways of entering dates- We'll assume that today is Feb 12, 2017.

  ```
  DayCount("January 8, 1935","Feb-12-2017");
  ```

- `AdvanceDate(date, number of days);` You can probably guess what this does.

  Example: What day will it be N days from today? To be more specific, find the date that is 1000 days from Feb 12, 2017:

  ```
  AdvanceDate("Feb-12-2017",1000);
  ```

- What does the following snippet of code do? The line breaks are for readability only.

  ```
  with(Finance):

  H:=[seq(4*i=AdvanceDate("Jan-30-2017",3*i),i=-2..2)];

  plot([100*cos(x/5),100*sin(x/3)],x=-10..10,y=-100..100,
      tickmarks=[ H, default],legend=["Plot 1","Plot2"],
      labels=["x-axis","y-axis"]);
  ```

  Notice that the plotting command may not use all of the values in H- But if we expand the graph using the mouse, more of them will appear.