

Honors Project 7: Visualizing Functions of Several Variables and Animation










One useful way to visualize functions of several variables is as animations. Maple provides two approaches to animation: one uses the `animate` and `animate3d` commands, and the other uses the `display` and `display3d` commands. In this project we discuss and illustrate both of these approaches.

First we consider the `animate` and `animate3d` commands. The key to using the `animate` command is understanding that if $f = f(x, y)$ is a function of two variables, and instead of denoting the second variable by y we denote it by t , then $f = f(x, t)$ can be thought of as a sequence of functions of x — one function for each t . By plotting $f = f(x, t)$ for each of a sequence of values of t , we create a sequence of frames which, when displayed sequentially, produce an animation.

For example, type the following command lines into Maple in the order in which they are listed.

```
> # Bending line
> restart: with(plots):
> plot3d(cos(x*t), x=-10..10, t=0..1, color=blue);
> animate({cos(x*t), cos(x)}, x=-10..10, t=0..1, frames=50, color=blue, scaling=constrained);
```

The second line produces a static graphic of the graph of $f(x, t) = \cos xt$; the third line produces an animation. To see the animation, click on the graphic produced by the `animate` command. A new menu line will appear at the top of the page. Clicking on:

-  starts the animation
-  stops the animation
-  steps through the animation, frame by frame
-  changes the animation from forward to reverse
-  changes the animation from reverse back to forward
-  speeds up the animation
-  slows down the animation
-  runs the animation in a loop
-  runs the animation from first frame to last (undo “looping”)

Note that there is an up-down arrow on the right-hand side of this menu line that will allow you to toggle back and forth between the animation menu and the standard graphic menu.

Some observations: First, the effect of this animation is to show that $f(x, t) = \cos xt$ can be viewed as bending (or “morphing”?) a horizontal line into a cosine wave. Second, to animate more than one function, enclose the functions to be animated in curly brackets `{` and `}`. (In the code above we animated two functions; one of them — namely $\cos x$ — did not depend on t , however: it was static, not dynamic.) Third, a similar construct can be performed using `animate3d`; now, however, instead of using a function of the form $f(x, t)$ we use a function of the form $f(x, y, t)$.

Let us now consider the `display` and `display3d` commands. The usage of these commands is similar to that of the `animate` and `animate3d` commands. The major difference is that with the `display` and `display3d` commands, we create a sequence of images which we eventually display using a command such as:

```
> display(seq(fig.i, i=N..M), insequence=true);
```

(Recall that the sequence fig_i is written `fig.i` in Maple.) To make this concrete, type the following command lines into Maple in the order in which they are listed. They produce an animation which, in step-by-step mode, can be used to illustrate the convergence of successive McLaurin series expansions of $\sin x$ to $\sin x$

```

> # Animated Mclaurin Series
> restart: with(plots):
> f := x -> sin(x);
> N := 8;
> myset := {f(x)}:
> fig.0 := plot(myset,x=-2*Pi..2*Pi,y=-2..2,color=blue):
> for i from 1 to N do
  t := taylor(f(x),x=0,2*i); p :=convert(t,polynomial);
  myset := myset union {p}:
  fig.i := plot(myset,x=-2*Pi..2*Pi,y=-2..2,color=blue):
od:
> display(seq(fig.i,i=0..N), insequence=true);

```

All that we have just said for functions $f : R^n \rightarrow R$ can be extended to transformations $\mathbf{F} : R^n \rightarrow R^m$. If, for example, $\mathbf{F}(x, t) = [f(x, t), g(x, t)]$ is a function whose domain is R^2 and whose range is R^2 :

$$F : R^2 \rightarrow R^2$$

then we can think of $\mathbf{F}(x, t)$ as a sequence of curves in the plane — one function for each t — and by plotting $\mathbf{F}(x, t) = [f(x, t), g(x, t)]$ for each of a sequence of values of t , we can create a sequence of frames which, when displayed sequentially, produce an animation of curves in the plane. (In other words, earlier we talked about curves $\mathbf{x} : R \rightarrow R^2$ in the plane and curves $\mathbf{x} : R \rightarrow R^3$ in space. *Now* we are talking about curves in a space of curves! A curve in a space of curves is known as a *homotopy*. The study of homotopies is important in areas such as mechanical engineering where paths are studied in conjunction with robotics and, more specifically, path planning strategies.)

To illustrate these ideas, type the following command lines into Maple in the order in which they are listed.

```

> # More animations
> restart: with(plots):
> animate([cos(x*t),sin(x*t),x=0..2*Pi],t=0..1,frames=100,
  color=blue,scaling=constrained);
> animate([(1+t*cos(x))*cos(x),(1+t*cos(x))*sin(x),x=0..2*Pi],t=0..2,frames=64,
  color=blue,scaling=constrained);

```

The second line animates an evolving circle, and the second animates a circle evolving into a limaçon. With regard to the second, it should be noted that if $\mathbf{x}_0 = [x_0(t), y_0(t)]$ and $\mathbf{x}_1 = [x_1(t), y_1(t)]$ are any two curves in the plane then

$$(1-t)\mathbf{x}_0 + t\mathbf{x}_1 = (1-t)[x_0(t), y_0(t)] + t[x_1(t), y_1(t)] = [(1-t)x_0(t) + tx_1(t), (1-t)y_0(t) + ty_1(t)]$$

is a homotopy which starts at \mathbf{x}_0 when $t = 0$ and ends at \mathbf{x}_1 when $t = 1$. (The technique illustrated here is very useful. Another application of it involves finding a parametrization of the line segment determined by the points $P_0(x_0, y_0, z_0)$ and $P_1(x_1, y_1, z_1)$: such a parametrization is given by

$$(1-t)(x_0, y_0, z_0) + t(x_1, y_1, z_1) = ((1-t)x_0 + tx_1, (1-t)y_0 + ty_1, (1-t)z_0 + tz_1)$$

where $t \in [0, 1]$.)

Task

Create an animation which illustrates some sort of interesting behavior, and write a paragraph or two about what the animation is supposed to illustrate.