Animations

by Bob Bradshaw, Ohlone College

There are three forms of two-dimensional animation in *Maple*. The first form is the command *animatecurve*. This command imitates the gradual drawing of a graph that you would see on a graphing calculator. The second form is the command *animate*. This command allows you to modify an equation. The third form is through the use of sequences. This method provides the most control of the animation.

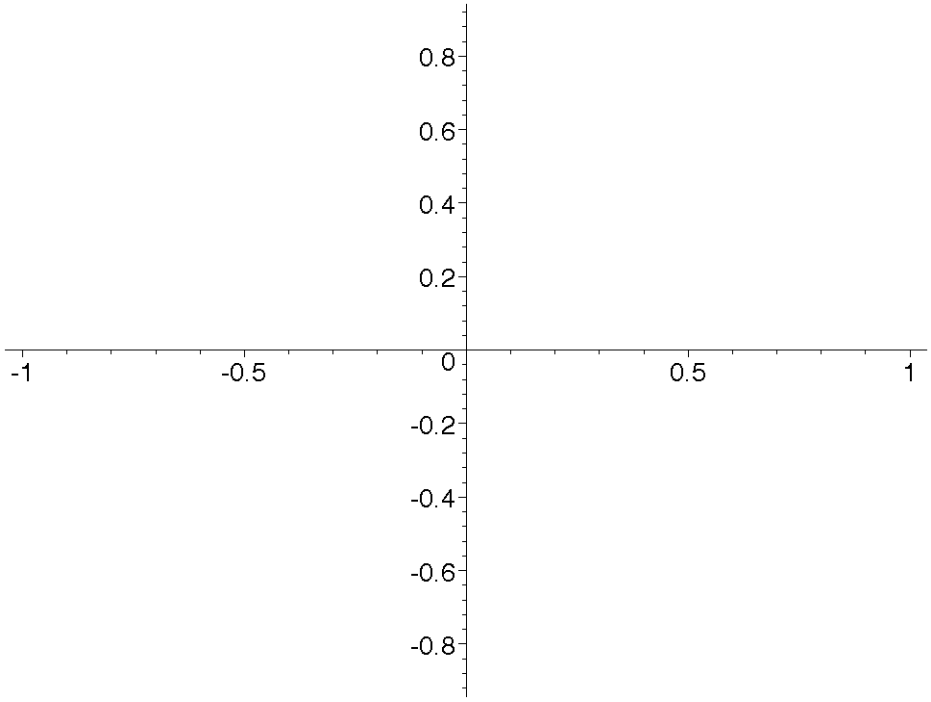
All three of the forms require the use of the plotting subroutines so start by loading this package.

```
> with(plots):
Warning, the name changecoords has been redefined
```

To see any of the following animations, execute the commands and then click once on the picture. You should see a set of VCR-type controls at the top of the screen. Use these to control the animation.

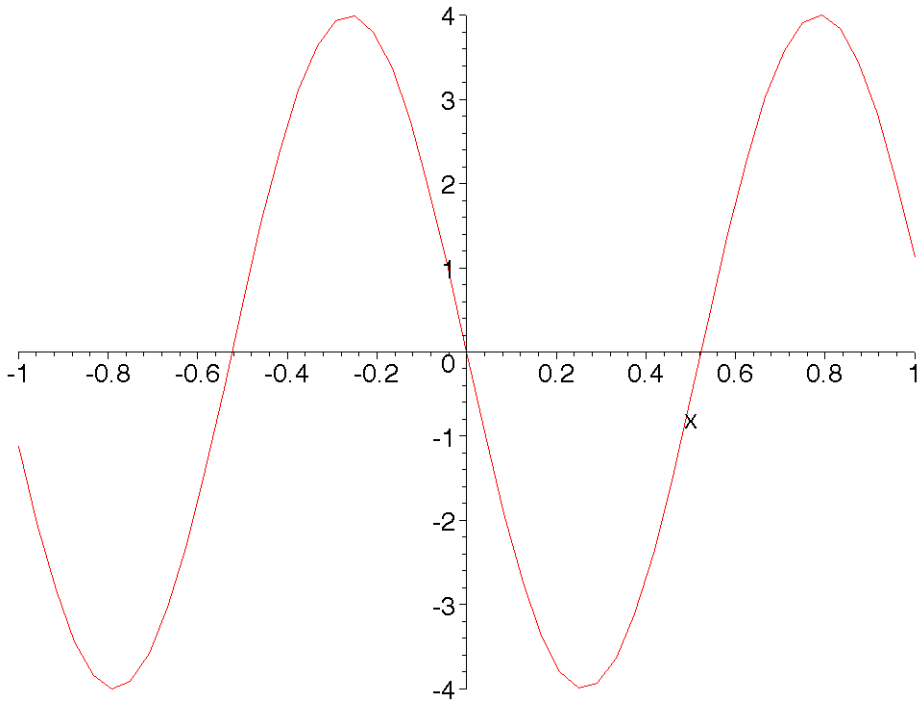
First we look at a simple example of the *animatecurve* command. This command will produce animations of standard functions or parametric equations. below is an animation of a set of parametric equations.

```
> animatecurve([cos(3/2*t)*cos(t),cos(3/2*t)*sin(t),t=0..4*Pi],numpo
  ints=400);
```



Now, we look at the *animate* command. In the following example, we use *animate* to see the effects of changing the amplitude of a sine curve..

```
> animate(a*sin(6*x),x=-1..1,a=-4..4,view=[-1..1,-4..4]);
```



The most powerful method of creating animations is through the use of sequences. A basic *sequence* command has the following form.

```
> seq(i^2,i=1..6);
```

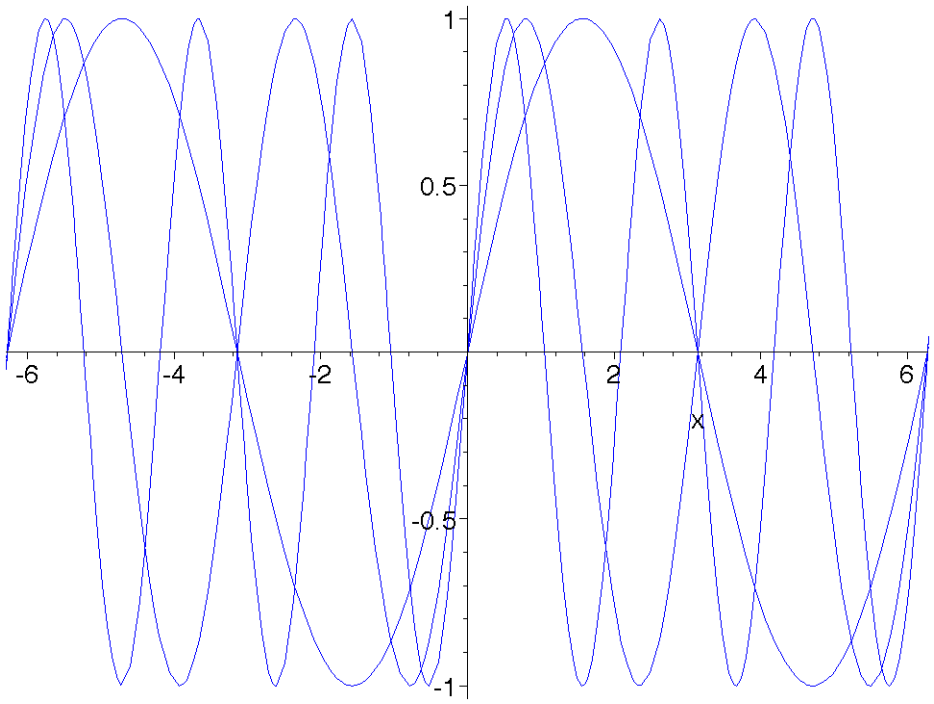
1, 4, 9, 16, 25, 36

We can now use the *sequence* command to generate the plot of sin(*ax*) for three different values of a. Create the sequence first and name it.

```
> bb1:=seq(plot(sin(a*x),x=-6.3..6.3,color=blue),a=1..3):
```

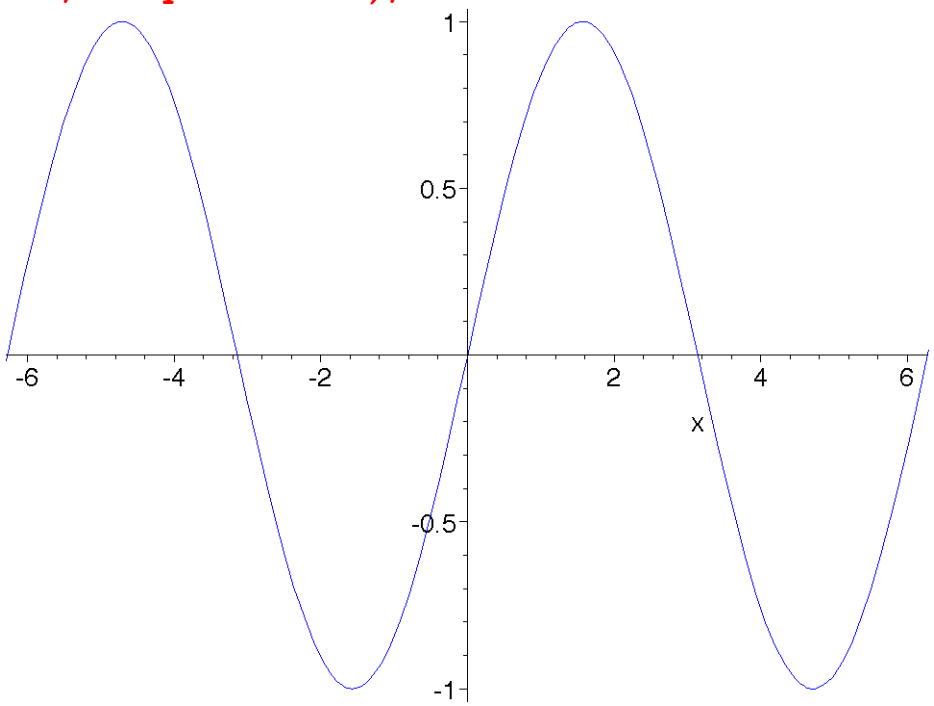
The animation is plotted by using the *display* command. If you use the option *insequence=false*, all frames are shown together and there is no animation.

```
> display(bb1,insequence=false);
```



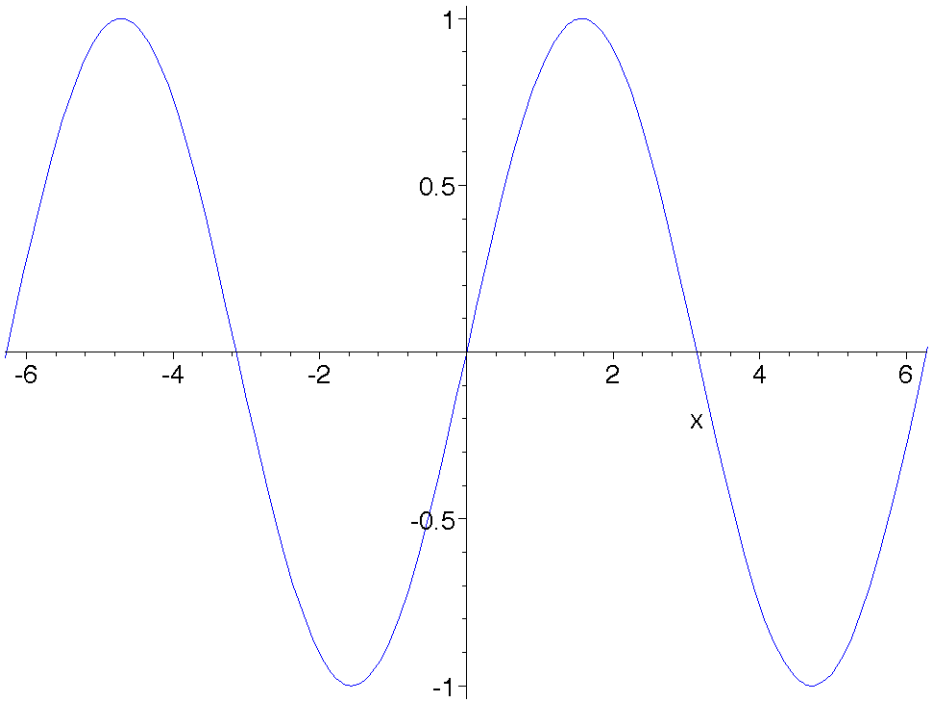
Using the option *insequence=true* shows only one frame at a time.

```
> display(bb1,insequence=true);
```



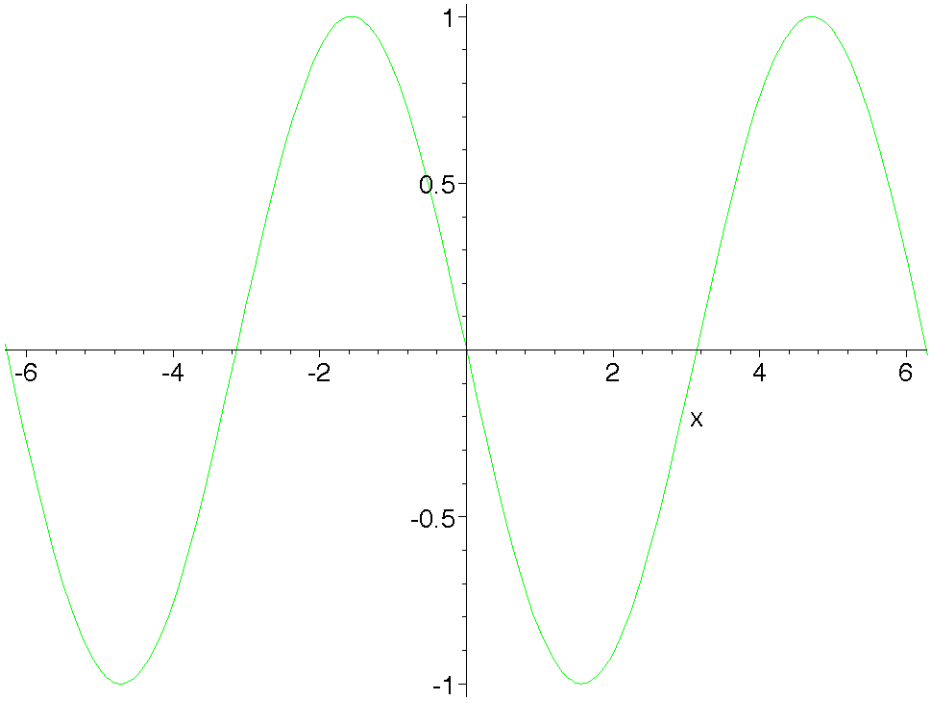
If we create a second sequence, you can compare the results of making *insequence* either true or false.

```
> bb2:=seq(plot(-sin(a*x),x=-6.3..6.3,color=green),a=1..3):
> display(bb1,bb2,insequence=true);
```



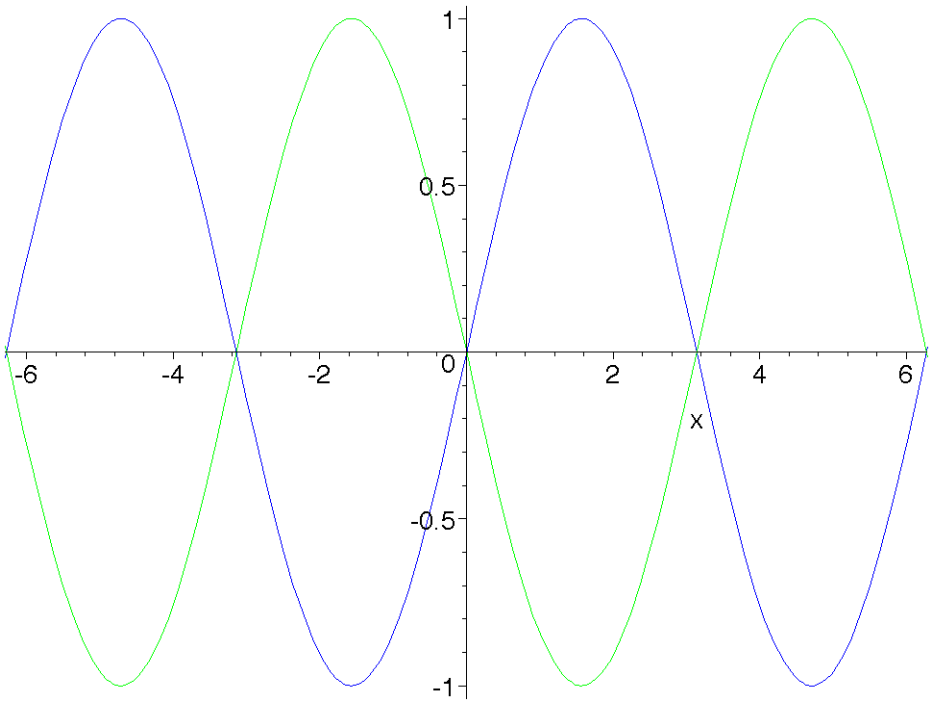
You can also look at the effect of chanign the order of what you are plotting.

```
> display(bb2,bb1,insequence=true);
```



If you create and name a *display* command, these can again be placed in a *display* command. This may be a useful time to have *insequence=false*.

```
> displaygreen:=display(bb1,insequence=true):
> displayblue:=display(bb2,insequence=true):
> display(displayblue,displaygreen,insequence=false);
```



In the following, I first declare a set of parametric functions, the number of frames in the animation, an initial and final *t* value and a delta *t* value.

```
> xt:=cos(3/2*t)*cos(t);
  yt:=cos(3/2*t)*sin(t);
  numframes:=60;
  initialt:=0;
  finalt:=12.56;
  delta:=(finalt-initialt)/numframes;
```

$$xt := \cos\left(\frac{3}{2}t\right)\cos(t)$$

$$yt := \cos\left(\frac{3}{2}t\right)\sin(t)$$

$$numframes := 60$$

$$initialt := 0$$

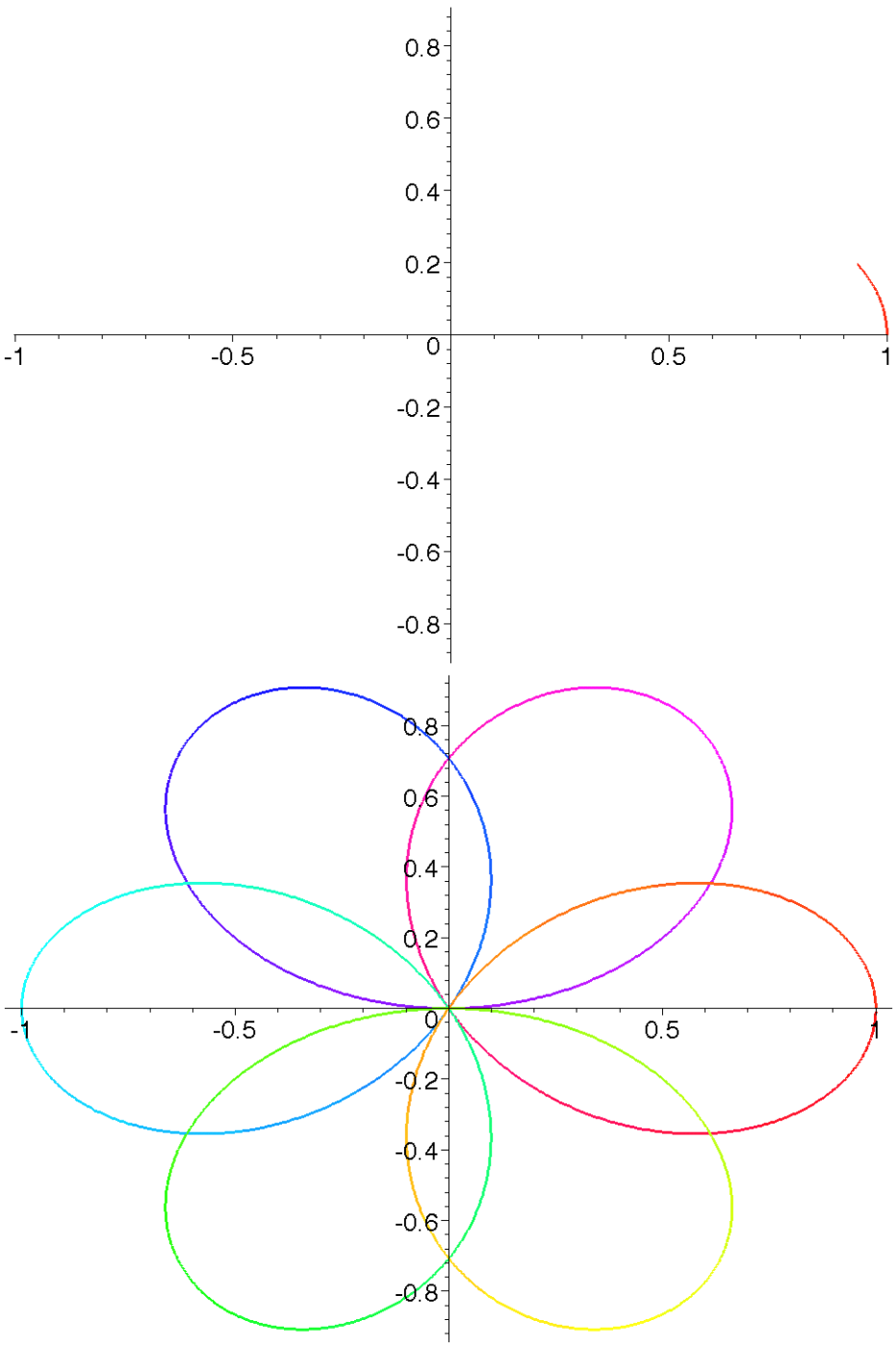
$$finalt := 12.56$$

$$\delta := .2093333333$$

We can create a sequence in which the color is defined using the HUE command, where the HUE is a function of the frame number.

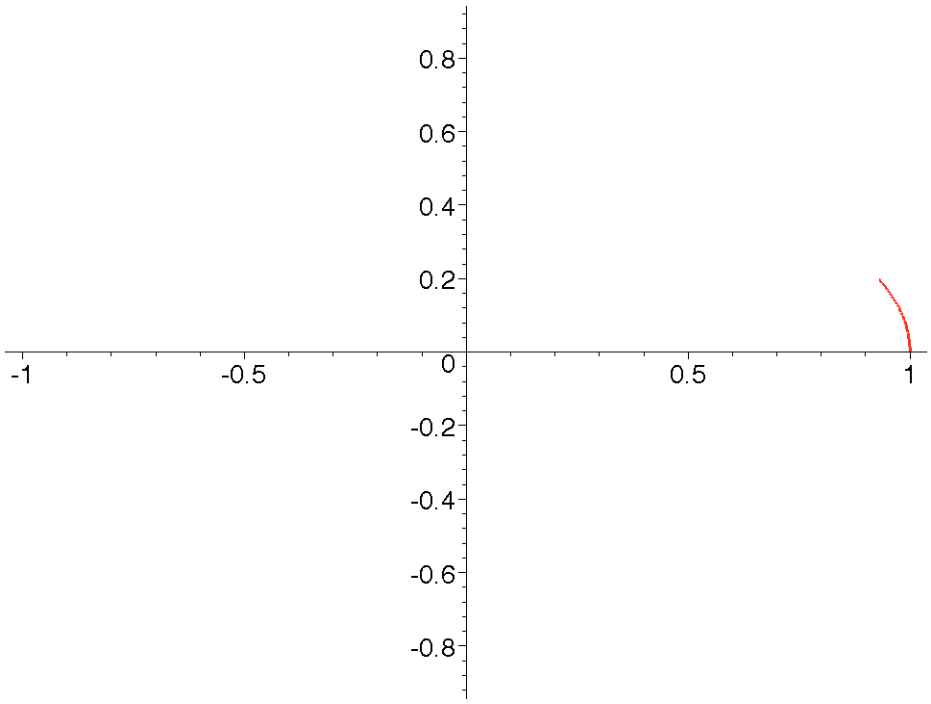
```
> bb5:=seq(plot([xt,yt,t=initialt+(i-1)*delta..initialt+i*delta],col
  or=COLOR(HUE,i/numframes),thickness=3),i=1..numframes):
> display(bb5,insequence=true);

> display(bb5,insequence=false);
```



This is a repeat of the above animation with a slight change in the initial value of *t*.

```
> bb6:=seq(plot([xt,yt,t=initialt..initialt+i*delta],color=COLOR(HUE
  ,i/numframes),thickness=3),i=1..numframes):
> display(bb6,insequence=true);
```


0.8
0.6
0.4
0.2
-1
-0.5
0
0.5
1
-0.2
-0.4
-0.6
-0.8

>

>