Math 290: LaTeXSeminar Week 10

Justin A. James
Minnesota State University Moorhead
jamesju@mnstate.edu

March 22, 2011

Outline

Getting Func-y!

Our goal this week is to learn to use some powerful LaTeX macros to graph without having to do as much work as we did with standard postscript commands.
We will need two to use two new packages to do this.

- Start a document and put this in the preamble.

  ```
  \usepackage{pst-func,pst-3dplot,graphicx}
  ```

- including pst-func will load the following packages: pst-plot, pstricks, pstricks-add, pst-math, and pst-xkey.

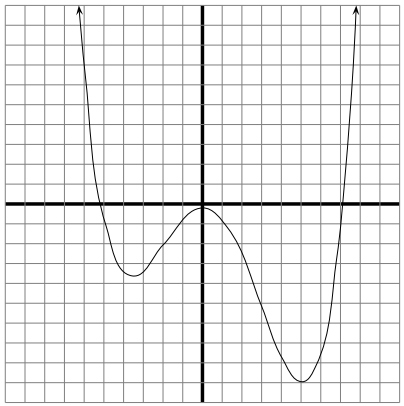
Outline

A Blast From the Past:

Recall that back in week 6, we used the following commands to graph the function $\frac{1}{12}x^4 - \frac{1}{6}x^3 - 3x^2 - 1$:

```
\pscurve[linewidth=1.2pt,arrowsize=10pt]{<->}
(-6.27,10)(-5.95,6.547)(-4.89,-1.10)(-3.56,-3.624)
(-2,-2.06667)(0,-.2)(1.277,-1.203)(3,-5.15)(4.04,-7.76)
(5.06,-8.954)(5.745,-8.17)(6.81,-2.72)(7.234,1.425)
(7.65,6.99)(7.8,10)
```

If we add this command to postscript commands for the large grid (see the files from week 6) , we get the following graph:



A Better Way?

Enter the following text into your document and build:

```
\scalebox{.3}{\begin{pspicture}(-10,-50)(10,50)
\psplot[algebraic,plotstyle=curve]{-10}{10}
{1/(12)*x^4-1/6*x^3-3*x^2-1}
\end{pspicture}}
```

This is not exactly what we were looking for...

Hey Jude...

We will now add some additional commands to make our graph a bit better.

Just before

```
\begin{pspicture}
```

type

```
\psset{xunit=1cm,yunit=.2cm}
```

Now after the plot command type

```
\psaxes(0,0)(-10,-50)(10,50)
```

Now let's adjust the axes a bit. Change the command to read

```
\psaxes[Dx=1,Dy=5](0,0)(-10,-50)(10,50)
```

This is a step in the right direction, but there is still a problem. Our function runs right off of our grid.

We can use the clip command to stop this.

More Commands

Before the line with the psplot command on it, type

```
\begin{psclip}{\pspolygon[linestyle=none](-10,-50)
(10,-50)(10,50)(-10,50)}
```

After the psplot command type

```
\end{psclip}
```

What Do These Commands Do?

Let's look at each of these commands in detail.

- `\psplot[algebraic]`
  allows you to graph nearly any curve your interested in. Even if it isn't an algebraic curve. For example we could plot sin(x) and cos(x) if we wanted to. The commands for psplot are as follows:

  `\psplot[options]{xmin}{xmax}{f(x)}`
- `\psset{options}`
  allows you to set values of certain commands until another psset command is entered. We used it to set the length of 1 unit in the x and y directions.
- `\psaxes[options](x0,y0)(xmin,ymin)(xmax,ymax)`
  gives axes centered at (x0,y0) with the minimums and maximums as described. The commands Dx and Dy allow us to change the increment of each axis. The default for these is 1.

Clipping

The syntax for the "clip" commands is as follows:

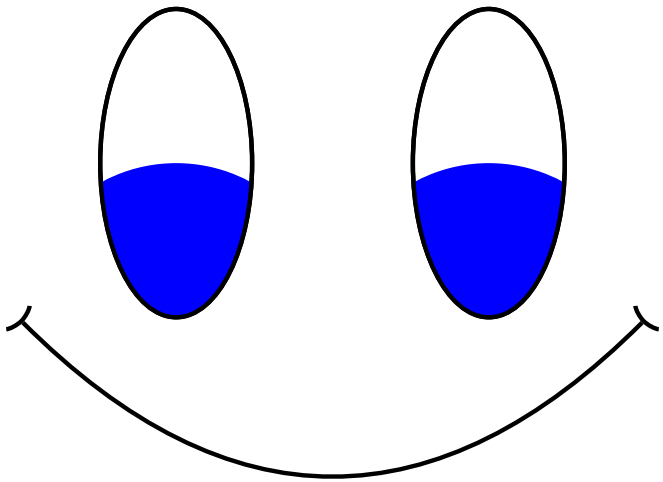
- `\begin{psclip}{control object}`
  `object to be clipped`
  `\end{psclip}`
  This commands allows you to clip an object so that all that you see of the object is inside of the control object. In our example we created a rectangle using pspolygon to clip our object. The rectangle covers the entire coordinate grid. We can use any object to clip. See what happens when you delete the last point of the polygon.

Now remove the linestyle command.
This will show you what this command was doing.

Outline

I always feel like somebody's watching me

Here is a cool example that Dr. Goyt designed:



I always feel like somebody's watching me

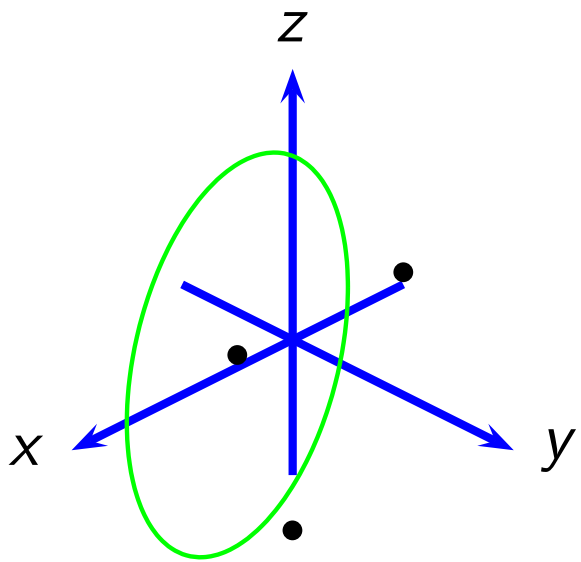
Here is the code for this example:

```
\begin{center} \psset{xunit=1cm,yunit=1cm}
\begin{pspicture}(0,0)(4,4)
\begin{psclip}{\psellipse(1,2)(.5,1)}
\pscircle*[linecolor=blue](1,1){1} \end{psclip}
\psellipse(1,2)(.5,1)
\begin{psclip}{\psellipse(3,2)(.5,1)}
\pscircle*[linecolor=blue](3,1){1} \end{psclip}
\psellipse(3,2)(.5,1)
\parabola[arrows=)-(](0,1)(2,0)
\end{pspicture} \end{center}
```

Outline

3D

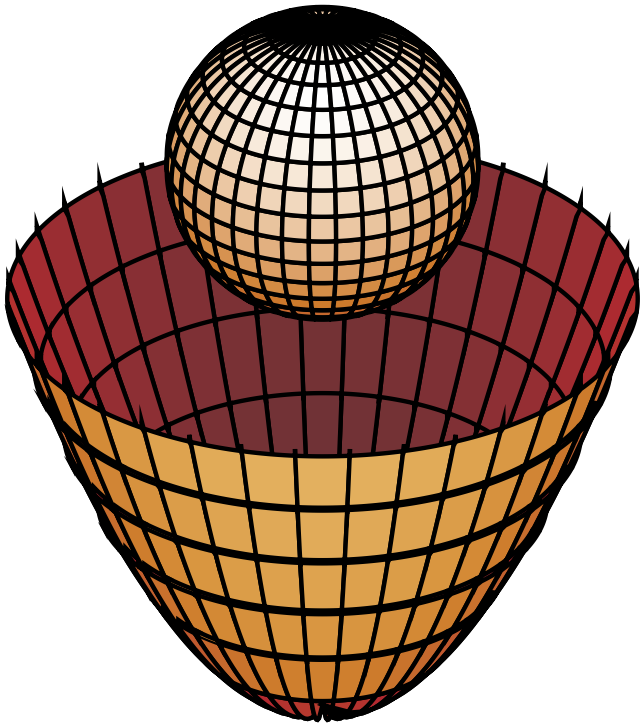
Here is a fun 3D graphic:



The Code:

```
\begin{pspicture}(-2,-1)(1,2)
\psset{unit=1cm}
\pstThreeDCoor[ linewidth=1.5pt,linecolor=blue,
xMin=-1,xMax=2,yMin=-1,yMax=2,
zMin=-1,zMax=2]
\pstThreeDEllipse[linecolor=green]
(1,0.5,0.5)(-0.5,0.5,0.5)(0.5,0.5,-1)
\pstThreeDDot(1,.5,.5)
\pstThreeDDot(-.5,.5,.5)
\pstThreeDDot(.5,.5,-1)
\end{pspicture}
```

More 3D



More 3D

Code:

```
\begin{center}

\begin{pspicture}(-3,0)(3,6)

\pstParaboloid[showInside=false]{3}{2}

\pstThreeDSphere(0,0,4){1}

\end{pspicture}

\end{center} \end{frame}
```

Outline

Let's Try Some Examples

Let's begin by graphing a few functions. Try graphing sin and cos on the same coordinate axes. Graph them from -2π to 2π.

```
\begin{pspicture}(-7,-2)(7,2)
\psplot[algebraic]{-6.283}{6.283}{sin(x)}
\psplot[algebraic]{-6.283}{6.283}{cos(x)}
\end{pspicture}
```

Now add some axes of appropriate length. Make the sin curve red and the cos curve blue.
Now change the psaxes command to psgrid.

Grids

There are many options that can be used with psgrid. Add the following options to the command
[subgriddiv=1,griddots=10,gridlabels=0]
subgriddiv=# determines the number of subdivisions of the grid. The default is 5.
griddots=# determines the number of dots to use between ticks. The default is 0, which gives a solid line.
gridlabels=# determines the size of the labels. 0 gets rid of them altogether.
Try modifying the previous command to change how the grid on your example looks.

Parametrized Curves

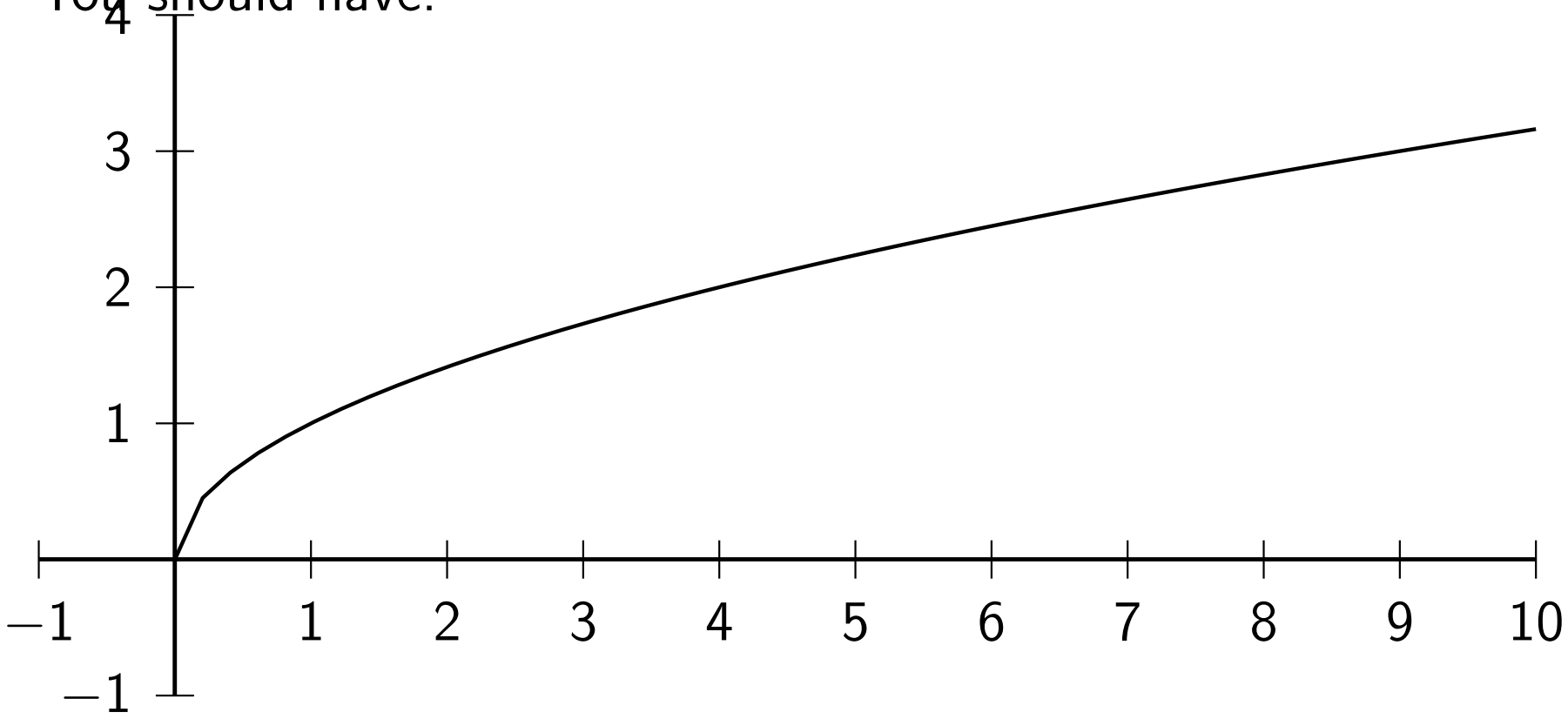
To graph a parametrized curve the notation changes slightly. Suppose we want to use sin and cos to graph a parameterized circle. To your current code add the line:

```
\parametricplot[algebraic,linecolor=green]
{-3.14}{3.14}{2*cos(t)|2*sin(t)}
```

There should now be a green circle of radius 2 on your grid.

A Graphic to Depict the Area Under a Curve

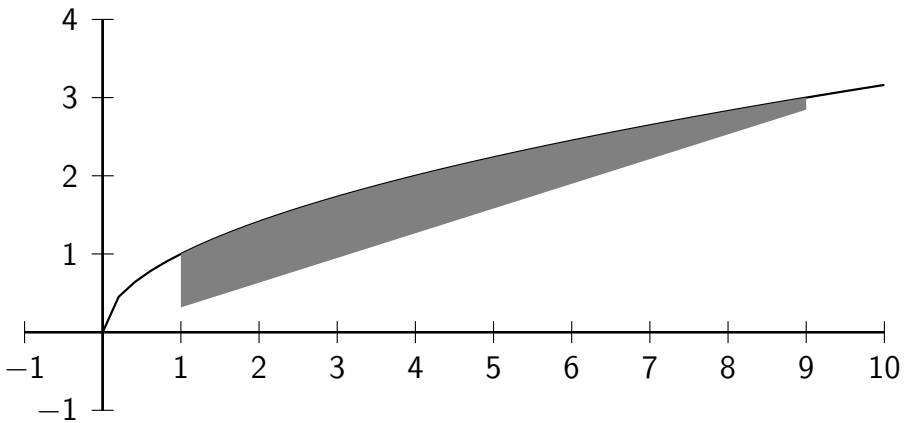
Suppose you want to shade the area under a curve for integrating. Let's use the curve $\sqrt{x}$ from 1 to 9. Graph this with a set of coordinate axes. Make your picture from (-1,-1) to (10,4). You should have:



Shading

Now, let's use the psclip option to shade it. Replace your code by:

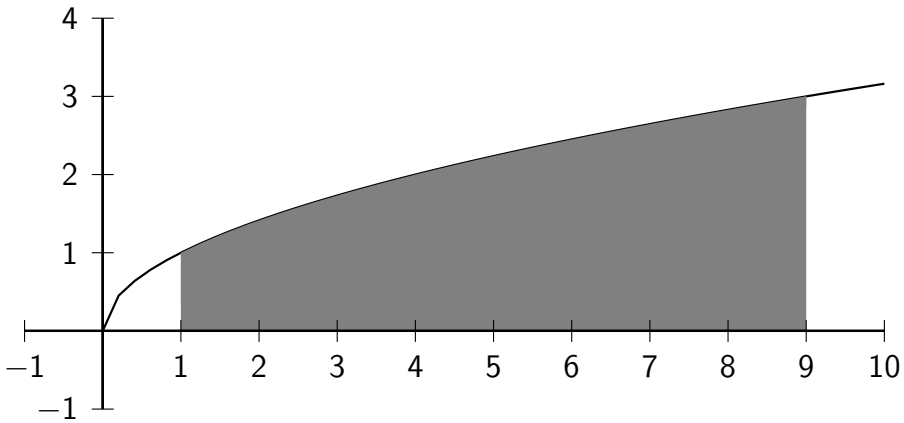
```
\begin{pspicture}(-1,-1)(10,4)
\begin{psclip}{
\psplot[algebraic]{0}{10}{sqrt(x)}}
\pspolygon*[linecolor=gray](1,0)(9,0)(9,4)(1,4)
\end{psclip}
\psaxes(0,0)(-1,-1)(10,4)
\end{pspicture}
```



The Rest

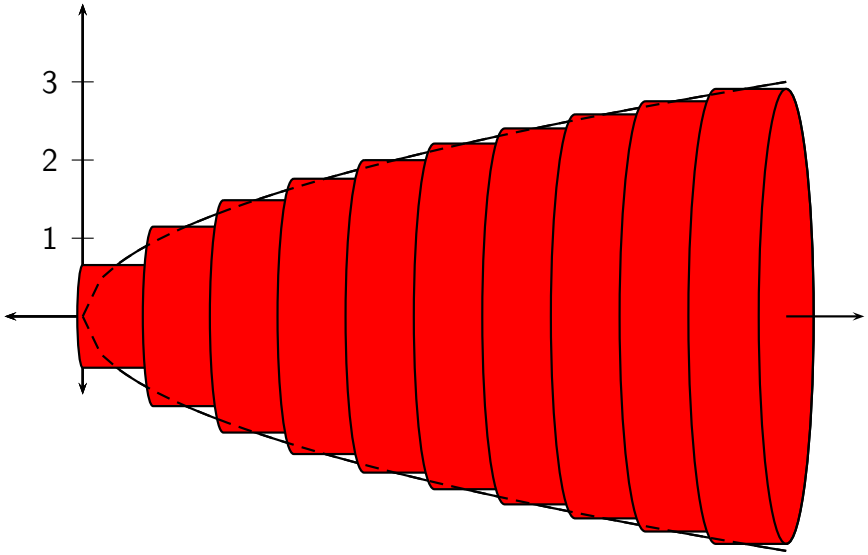
We have not yet shaded in the entire area. To fill in the rest we'll use another pspolygon.
Add the following to your code:

```
\pspolygon*[linecolor=gray](1,0)(9,0)(9,3)(1,1)
```



Solid of Revolution

What if we want to draw a picture of the washer method for finding the volume of the solid formed by revolving the function $f(x) = \sqrt{x}$ around the x-axis.
It would look as follows:



The Code

The code for the object on the previous slide is:

```
\begin{pspicture}(-1,-1)(10,4)
\psaxes[arrows=<->](0,0)(-1,-1)(9,4)
\psVolume[fillstyle=solid,fillcolor=red]
  (0,9){10}{x sqrt}
\psline[arrows=->](9,0)(10,0)
\end{pspicture}}
```

Outline

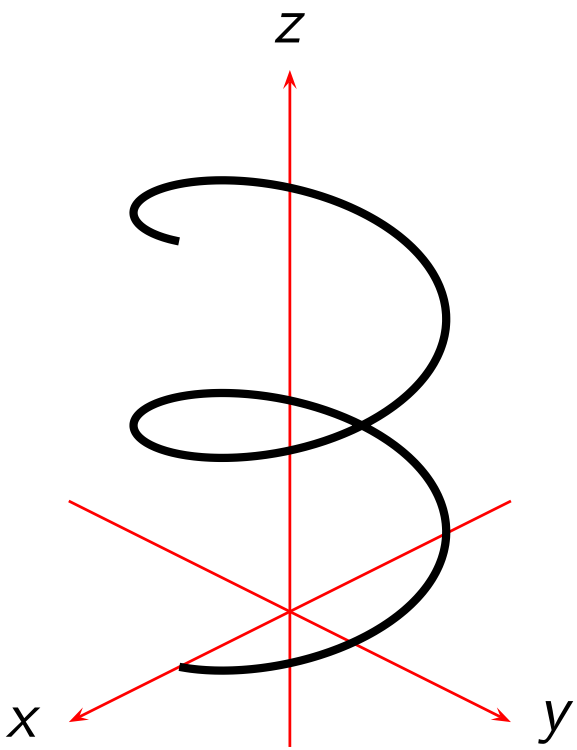
A Sphere

The pst-3dplot allows us to graph in 3 dimensions.
Try the following code.

```
\begin{pspicture}(-4,-4)(4,4)
\pstThreeDCoor
\pstThreeDSphere(0,0,0){3}
\end{pspicture}
```

A Helix

We can also do parametric plots in 3D.



The Code

The Code for the Helix in the Previous Page.

```
\begin{pspicture}(-4,-4)(4,4)
\pstThreeDCoor[xMin=-2,xMax=2,yMin=-2,
yMax=2,zMin=-1,zMax=4]
\parametricplotThreeD[xPlotpoints=200,
plotstyle=curve,algebraic,linewidth=1.5pt]
(0,12.564){cos(t)|sin(t)|t/4}
\end{pspicture}
```

Some Resources

Here is a list of links to websites that have useful information about using pst-plot.
(Follow the links embedded in the posted .pdf file for this week)

- General PSTricks
- Some Pst-Plot without the algebraic option
- Pst-Plot with the algebraic option
- pst-3dplot