Math 291: Lecture 10

Dr. Fagerstrom

Minnesota State University Moorhead
web.mnstate.edu/fagerstrom
fagerstrom@mnstate.edu

April 5, 2018

Outline

1. The pst-func package
   - Using psplot and psaxes
   - Using psgrid
   - Using psclip
2. Another Example
3. Other function types
   - Parametric Curves
   - Solid of revolution
4. Three Dimensions
5. Resources

Graphing functions

Our goal this week is to learn to use some LaTeX commands to graph without having to do as much work as we did with the postscript commands we learned a few weeks ago.

Graphing functions

Our goal this week is to learn to use some LaTeX commands to graph without having to do as much work as we did with the postscript commands we learned a few weeks ago.
We will need two to use two new packages to do this.

Graphing functions

Our goal this week is to learn to use some LaTeX commands to graph without having to do as much work as we did with the postscript commands we learned a few weeks ago.
We will need two to use two new packages to do this.

- Start a document and load the packages pst-func, pst-3dplot, and graphicx.
- Note: Including pst-func will also load the packages pst-plot, pstricks, pstricks-add, pst-math, and pst-xkey.

The basic commands

As with almost any command that starts with 'ps', these commands are used within a pspicture environment. Remember that the syntax for the pspicture environment is:

```
\begin{pspicture}(xmin,ymin)(xmax,ymax)
...commands...
\end{pspicture}
```

- `\psplot[algebraic]`
  This graphs curves (even if they aren't algebraic, like sin(x) and cos(x)). The command for psplot is as follows:

The basic commands

As with almost any command that starts with 'ps', these commands are used within a pspicture environment. Remember that the syntax for the pspicture environment is:

```
\begin{pspicture}(xmin,ymin)(xmax,ymax)
...commands...
\end{pspicture}
```

- `\psplot[algebraic]`
  This graphs curves (even if they aren't algebraic, like sin(x) and cos(x)). The command for psplot is as follows:
  `\psplot[algebraic,options]{xmin}{xmax}{f(x)}`

The basic commands

As with almost any command that starts with 'ps', these commands are used within a pspicture environment. Remember that the syntax for the pspicture environment is:

```
\begin{pspicture}(xmin,ymin)(xmax,ymax)
...commands...
\end{pspicture}
```

- `\psplot[algebraic]`
  This graphs curves (even if they aren't algebraic, like sin(x) and cos(x)). The command for psplot is as follows:
  `\psplot[algebraic,options]{xmin}{xmax}{f(x)}`
  - Note that there are sometimes issues with certain functions. If that happens, the "plot a lot of points" option still works.

The basic commands

As with almost any command that starts with 'ps', these commands are used within a pspicture environment. Remember that the syntax for the pspicture environment is:

```
\begin{pspicture}(xmin,ymin)(xmax,ymax)
...commands...
\end{pspicture}
```

- `\psplot[algebraic]`
  This graphs curves (even if they aren't algebraic, like sin(x) and cos(x)). The command for psplot is as follows:
  `\psplot[algebraic,options]{xmin}{xmax}{f(x)}`
  - Note that there are sometimes issues with certain functions. If that happens, the "plot a lot of points" option still works.
  - Note that you need to carefully enter your function, and the syntax sometimes isn't what you expect.

The basic commands

- `\psset{options}`
  allows you to set values of certain commands until another psset command is entered. To set the length of 1 unit in the x- and y-directions, we could use:

  `\psset{xunit=1cm,yunit=0.2cm}`

The basic commands

- `\psset{options}`
  allows you to set values of certain commands until another psset command is entered. To set the length of 1 unit in the x- and y-directions, we could use:

  `\psset{xunit=1cm,yunit=0.2cm}`

  - Note that this should happen before you actually tell LaTeX to draw anything or otherwise set aside space for a graphic.

The basic commands

- `\psset{options}`
  allows you to set values of certain commands until another psset command is entered. To set the length of 1 unit in the x- and y-directions, we could use:

  `\psset{xunit=1cm,yunit=0.2cm}`

  - Note that this should happen before you actually tell LaTeX to draw anything or otherwise set aside space for a graphic.
- `\psaxes[options](x0,y0)(xmin,ymin)(xmax,ymax)`
  gives axes centered at (x0,y0) with the minimums and maximums as described.

The basic commands

- `\psset{options}`
  allows you to set values of certain commands until another psset command is entered. To set the length of 1 unit in the x- and y-directions, we could use:

  `\psset{xunit=1cm,yunit=0.2cm}`

  - Note that this should happen before you actually tell LaTeX to draw anything or otherwise set aside space for a graphic.

- `\psaxes[options](x0,y0)(xmin,ymin)(xmax,ymax)`
  gives axes centered at (x0,y0) with the minimums and maximums as described.
  - The option [Dx=2,Dy=5] changes the increment for each axis. The default values are 1.

Let's Try Some Examples

Let's begin by graphing a few functions. Try graphing $\sin(x)$ and $\cos(x)$ on the same coordinate axes (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}
```

Let's Try Some Examples

Let's begin by graphing a few functions. Try graphing $\sin(x)$ and $\cos(x)$ on the same coordinate axes (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}
```

Make the sine curve red and the cosine curve blue by adding a linecolor command in the options.

Let's Try Some Examples

Let's begin by graphing a few functions. Try graphing $\sin(x)$ and $\cos(x)$ on the same coordinate axes (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}
```

Make the sine curve red and the cosine curve blue by adding a linecolor command in the options.
Now add some axes of appropriate length using

```
\psaxes(0,0)(-7,-2)(7,2)
```

Let's Try Some Examples

Let's begin by graphing a few functions. Try graphing $\sin(x)$ and $\cos(x)$ on the same coordinate axes (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}
```

Make the sine curve red and the cosine curve blue by adding a linecolor command in the options.
Now add some axes of appropriate length using

```
\psaxes(0,0)(-7,-2)(7,2)
```

Now add the options to change the increments using Dx=3.14 and Dy=0.5.

Grids

- Create another graph:

```
\psset{xunit=0.5cm,yunit=0.5cm}
\begin{pspicture}(-10,-10)(10,10)
\psplot[algebraic]{-6.325}{6.325}{1/4*x^2}
\psaxes[Dx=2,Dy=2](0,0)(-10,-10)(10,10)
\end{pspicture}
```

and build.

Grids

- Create another graph:

```
\psset{xunit=0.5cm,yunit=0.5cm}
\begin{pspicture}(-10,-10)(10,10)
\psplot[algebraic]{-6.325}{6.325}{1/4*x^2}
\psaxes[Dx=2,Dy=2](0,0)(-10,-10)(10,10)
\end{pspicture}
```

  and build.
- Now change the psaxes to psgrid.

Grids

- Note that the grid ignored the Dx and Dy commands, but

Grids

- Note that the grid ignored the Dx and Dy commands, but
- there are other options that can be used with psgrid.

Grids

- Note that the grid ignored the Dx and Dy commands, but
- there are other options that can be used with psgrid.
- Add the following options to the command
  subgriddiv=1,griddots=10,gridlabels=0

Grids

- Note that the grid ignored the Dx and Dy commands, but
- there are other 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.

Grids

- Note that the grid ignored the Dx and Dy commands, but
- there are other 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 your previous command to change how the grid on your example looks.

Clipping

- Note that in the parabola example, we plotted from -6.325 to 6.325.
- These were chosen so that the graph didn't go off the grid.
- We can also use the psclip command.

```
\begin{psclip}{control object}
object to be clipped
\end{psclip}
```

Clipping

- Note that in the parabola example, we plotted from -6.325 to 6.325.
- These were chosen so that the graph didn't go off the grid.
- We can also use the psclip command.

```
\begin{psclip}{control object}
object to be clipped
\end{psclip}
```

- This command allows you to clip an object so that all that you see of the object is the portion inside of the control object.

Clipping

- Change your parabola commands to:

```
\psset{xunit=0.5cm,yunit=0.5cm}
\begin{pspicture}(-10,-10)(10,10)
\psplot[algebraic]{-10}{10}{1/4*x^2}
\psaxes[Dx=2,Dy=2](0,0)(-10,-10)(10,10)
\psgrid[gridlabels=0,griddots=5,
subgriddiv=1](0,0)(-10,-10)(10,10)
\end{pspicture}
```

Clipping

- Change your parabola commands to:

```
\psset{xunit=0.5cm,yunit=0.5cm}
\begin{pspicture}(-10,-10)(10,10)
\psplot[algebraic]{-10}{10}{1/4*x^2}
\psaxes[Dx=2,Dy=2](0,0)(-10,-10)(10,10)
\psgrid[gridlabels=0,griddots=5,
subgriddiv=1](0,0)(-10,-10)(10,10)
\end{pspicture}
```

- Build it and see what happens.
- Note that LaTeX is perfectly happy to graph beyond the space set aside for the pspicture.

Clipping

- Now clip your graph using a rectangle:
  - Just before the psplot command add `\begin{psclip}{\pspolygon[linestyle=none](-10,-10)(10,-10)(10,10)(-10,10)}`
  - And just after the psplot command add the end command for the psclip environment.
  - Now build again

Clipping

- Now clip your graph using a rectangle:
  - Just before the psplot command add `\begin{psclip}{\pspolygon[linestyle=none](-10,-10)(10,-10)(10,10)(-10,10)}`
  - And just after the psplot command add the end command for the psclip environment.
  - Now build again
- See what happens if you delete the last point of the polygon...

Clipping

- Now clip your graph using a rectangle:
  - Just before the psplot command add
    `\begin{psclip}{\pspolygon[linestyle=none](-10,-10)`
    `(10,-10)(10,10)(-10,10)}`
  - And just after the psplot command add the end command for the psclip environment.
  - Now build again
- See what happens if you delete the last point of the polygon...
- If you remove the linestyle command you will see what is going on.

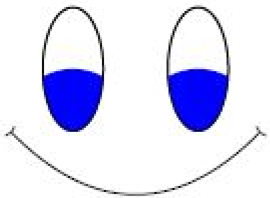
Outline

I always feel like somebody's watching me

Here is a cool example using psclip:

I always feel like somebody's watching me

Here is a cool example using psclip:



I always feel like somebody's watching me

Here is the code for this example:

I always feel like somebody's watching me

Here is the code for this example:

```
\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{)-(}(0,1)(2,0)
\end{pspicture}
```

Outline

Parametric Curves

- To graph a parametrized curve the notation changes slightly.

Parametric Curves

- To graph a parametrized curve the notation changes slightly.
- Suppose we want to use sine and cosine to graph a parametrized circle.

Parametric Curves

- To graph a parametrized curve the notation changes slightly.
- Suppose we want to use sine and cosine to graph a parametrized circle.
- To your code for the sine and cosine graphs add the line:

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

Parametric Curves

- To graph a parametrized curve the notation changes slightly.
- Suppose we want to use sine and cosine to graph a parametrized circle.
- To your code for the sine and cosine graphs 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.

A Graphic to Depict the Area Under a Curve

- Suppose you want to shade the area under a curve for integrating.
- We can use psclip, but note that our curve is what will need to do the clipping.

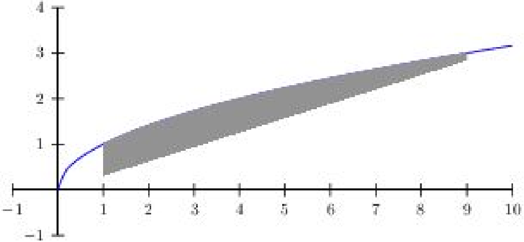
A Graphic to Depict the Area Under a Curve

- Suppose you want to shade the area under a curve for integrating.
- We can use psclip, but note that our curve is what will need to do the clipping.
- 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).
- Use the psclip option to shade it:

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

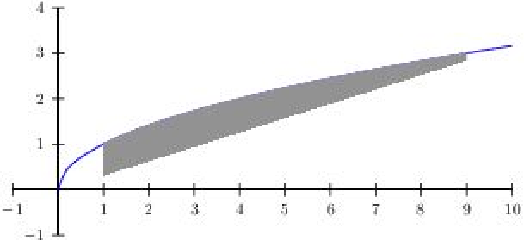
A Graphic to Depict the Area Under a Curve

- You should get



A Graphic to Depict the Area Under a Curve

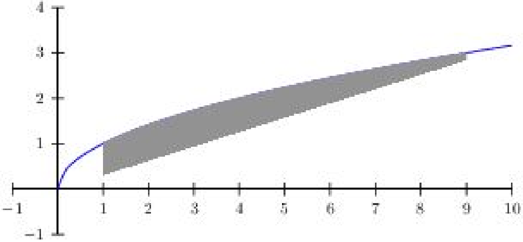
- You should get



- This isn't what we wanted.

A Graphic to Depict the Area Under a Curve

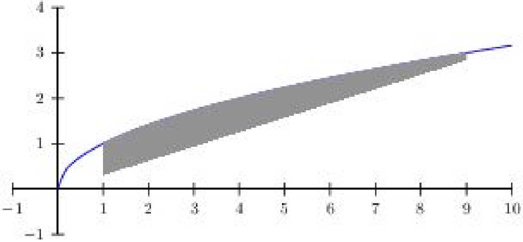
- You should get



- This isn't what we wanted.
- What happened?

A Graphic to Depict the Area Under a Curve

- You should get



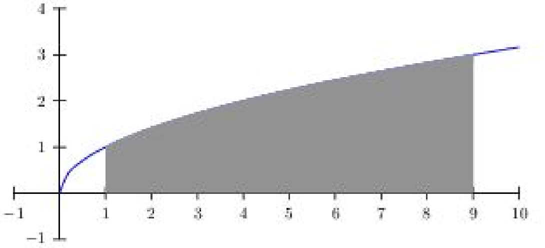
- This isn't what we wanted.
- What happened? LaTeX wants the clip to be a closed shape, so is graphing the curve, but then connecting the first and last points, in this case the points (0,0) and $(10, \sqrt{10})$.

A Graphic to Depict the Area Under a Curve

- So lets fix it by adding the missing bit after the clip ends:
  `\pspolygon*[linecolor=gray](1,0)(9,0)(9,3)(1,1)`

A Graphic to Depict the Area Under a Curve

- So lets fix it by adding the missing bit after the clip ends:
  `\pspolygon*[linecolor=gray](1,0)(9,0)(9,3)(1,1)`
- You should now get

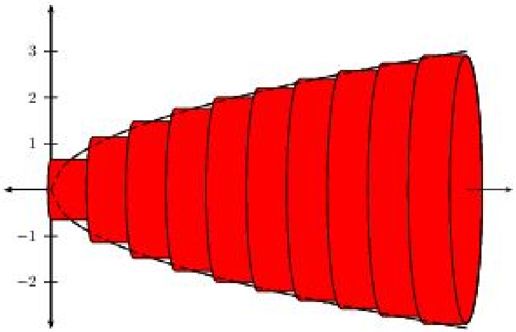


Solid of Revolution

- 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:

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:

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.

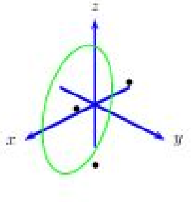
A Sphere

- 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}
```

3D

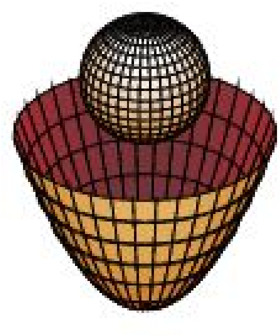
- Here is a fun 3D graphic:



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



The Code:

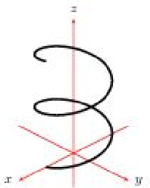
```
\begin{pspicture}(-3,0)(3,6)
\pstParaboloid[showInside=false]{3}{2}
\pstThreeDSphere(0,0,4){1}
\end{pspicture}
```

A Helix

- We can also do parametric plots in 3D.

A Helix

- We can also do parametric plots in 3D.



The Code

```
\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}
```

Outline

Some Resources

Here is a list of links to websites that have useful information about using pst-plot.

Some Resources

Here is a list of links to websites that have useful information about using pst-plot.
(The file names are long, so follow the links embedded in the posted .pdf file for this week)

Some Resources

Here is a list of links to websites that have useful information about using pst-plot.
(The file names are long, so follow the links embedded in the posted .pdf file for this week)

- General PSTricks
- Pst-Plot with the algebraic option
- pst-3dplot