Math 291: Lecture 10

Dr. Fagerstrom

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

April 18, 2019

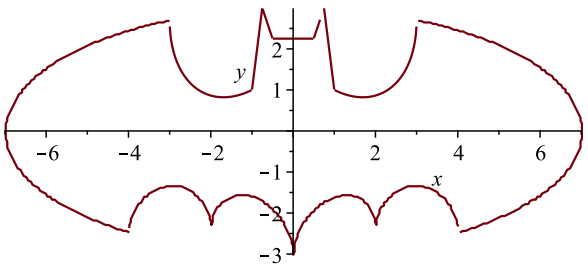
Outline

Including Postscript Graphics Files

- Use the packages: graphicx, pstricks
- Change the Build Option to:
  LaTeX => PS or to LaTeX => PS => PDF
- First, create the image file: "Bat.eps"
  - It will be based on a graph created using Maple
  - Open the file "Bat.mw", then right-click on the graph.
  - In the context menu, choose "Export".
  - Then choose the file type ".eps" ("encapsulated postscript").
  - Save the resulting file as bat.eps in the same folder as your current document (the compiler will only be able to find graphics files in the same folder).

Including Postscript Graphics Files

- Include this new file in your document using the command \includegraphics[width=4in]{bat.eps}
- **Caution**: Trying to mix .jpg and .eps graphics in a single file creates problems — the .eps files won't load unless you go through PS in your build profile, but the .jpg files won't load if you do go through PS in your build profile. So you can't mix the graphic type.
- The result should look something like this:



Outline

Basic Structure and Scaling

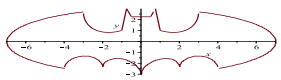
- The environment's name is pspicture (so use \begin{pspicture} and the appropriate end statement).
- You must also give LaTeX the coordinates of the bottom left corner and of the upper right corner as required arguments. These tell LaTeX how much room to set aside for the picture.
- Within that environment are put the ps commands.
- \psset{options} allows you to set values of certain commands until another psset command is entered.
- Note that this command should happen before you actually tell LaTeX to draw anything or otherwise set aside space for a graphic (so before the begin pspicture command).

Basic Structure and Scaling

- To create a pspicture from $(0,0)$ to $(5,5)$ with length of 1 unit to be 0.5 cm in both the x- and y-directions, we could use:

```
\psset{xunit=0.5cm,yunit=0.5cm}
\begin{pspicture}(0,0)(5,5)
...
\end{pspicture}
```

- Another option is to resize the box that is used to create the picture. Doing this to the bat picture:
- `\resizebox{0.7in}{0.4in}{\includegraphics[width=4in,keepaspectratio=true]{bat.eps}}`



- Note that \resizebox{width}{height}{...} is a command in the graphicx package, so can be used for anything in a box (including ps pictures, jpgs, tikz pictures, etc.).

Basic pstricks commands

The following is a list of some ps commands.

- `\psline[...]{arrows}(x1,y1)(x2,y2)`
- `\pscurve[...]{arrows}(x1,y1)...(xn,yn)`
- `\psdots[...](x,y)` [or `\psdot` for just one]
- `\psgrid[...](x0,y0)(x1,y1)(x2,y2)`
- `\psaxes[...](x0,y0)(x1,y1)(x2,y2)`
- `\psframe[...](x0,y0)(x1,y1)`
- `\psdiamond[...](h,k)(x-length,y-length)`
- `\pscircle[...](h,k){r}`
- `\psellipse[...](h,k)(x-length,y-length)`
- `\pswedge[...](h,k){r}{start-angle}{end-angle}`
- `\parabola[...](x0,y0)(h,k)`
- `\psarc[...](h,k){r}{start-angle}{end-angle}`
- `\psbezier[...](x1,y1)(x2,y2)(x3,y3)(x4,y4)`
- `\pspolygon[...](x1,y1)...(xn,yn)`

Postscript Graphics Options

Some options that can be used to modify PS Graphics:

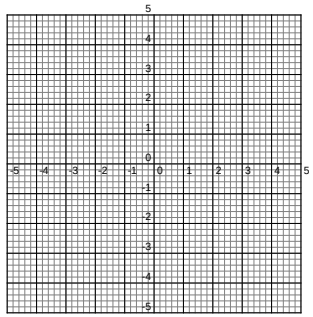
- linewidth=#, linecolor=color, arrowsize=#, dotsize=#, arrowlength=#
- the numbers must include units
- Colors: black, darkgray, gray, lightgray, white, red, green, blue, cyan, magenta, yellow, or
  - `\definecolor{name}{rgb}{# # #}`, where each # $\in [0, 1]$
- Line terminators: $-$, $<$, $>$, $<<$, $>>$, (,), $|$, [,], o, $*$, $|<$, $>|$, c
- doubleline=true (and then doublesep=#)
- linestyle=solid, dashed, dotted, none
- dotstyle=$*$, $+$, $|$, o, x, asterisk, diamond*, diamond, oplus, otimes, pentagon*, pentagon, square*, square, triangle*, triangle
- linearc=# showpoints=true/false
- fillstyle=none, solid, vlines, vlines*, hlines, hlines*, crosshatch, crosshatch*, (and gradient, with the pst-grad package)
- fillcolor=...

Grid and Axes

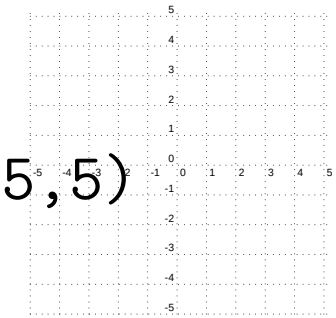
- \psgrid creates a grid with labels on the axes
  subgriddiv=1,griddots=10,gridlabels=0
  subgriddiv=# determines the number of subdivisions of the grid (default is 5)
  griddots=# determines the number of dots to use between ticks (default is 0, which gives a solid line)
  gridlabels=# determines the size of the labels (0 gets rid of them altogether)
- \psaxes creates just the axes with labels
  you need the pst-func package for \psaxes
  option [Dx=2,Dy=5] changes the increment for each axis (default values are 1)

Grid Examples

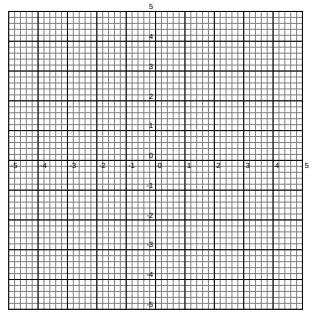
```
\begin{pspicture}(-5,-5)(5,5)
\psgrid(0,0)(-5,-5)(5,5)
\end{pspicture}
```



```
\begin{pspicture}(-5,-5)(5,5)
\psgrid[subgriddiv=1,griddots=5](0,0)(-5,-5)(5,5)
\end{pspicture}
```

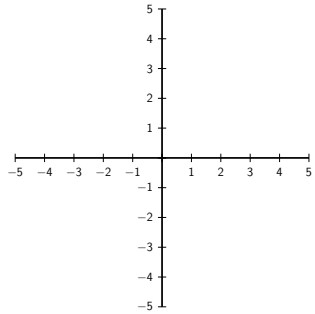


```
\begin{pspicture}(-5,-5)(5,5)
\psgrid[gridlabels=0.25](0,0)(-5,-5)(5,5)
\end{pspicture}
```

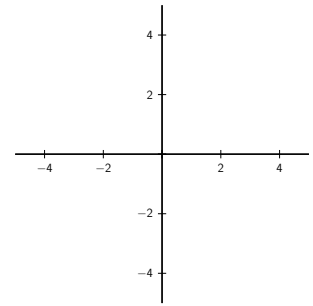


Axes Examples

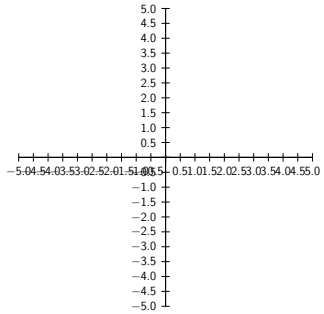
```
\begin{pspicture}(-5,-5)(5,5)
\psaxes(0,0)(-5,-5)(5,5)
\end{pspicture}
```



```
\begin{pspicture}(-5,-5)(5,5)
\psaxes[Dx=2,Dy=2](0,0)(-5,-5)(5,5)
\end{pspicture}
```

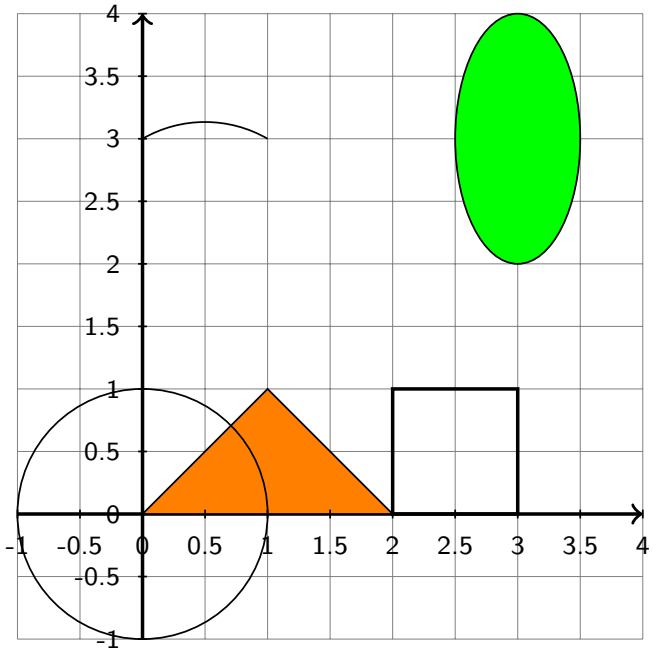


```
\begin{pspicture}(-5,-5)(5,5)
\psaxes[Dx=0.5,Dy=0.5](0,0)(-5,-5)(5,5)
\end{pspicture}
```



Example again

Let's try to recreate the first example from the Ti*k*Z lecture.

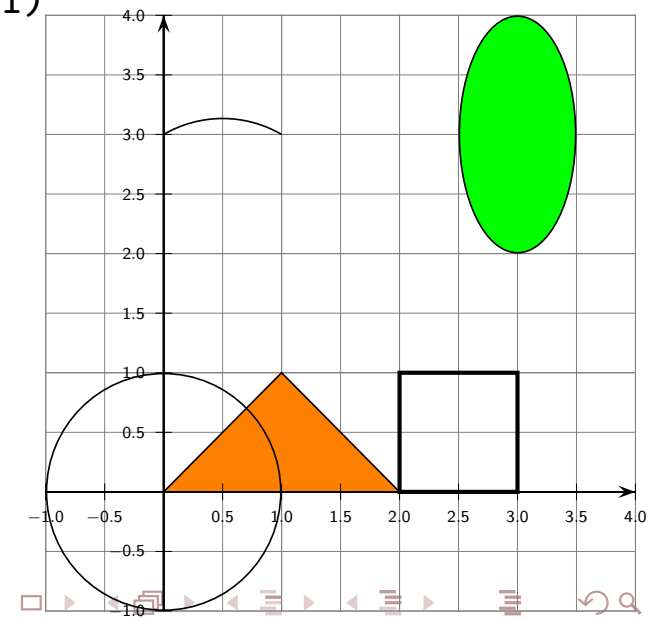


Example again

- We'll need to make the grid a little lighter than the default. We can do this using the gridcolor, gridwidth, subgridcolor, and subgridwidth options.
- Note that the circle is on top of the triangle.
- Note that the tic marks for the axis are on top of the triangle as well.
- For the arc, and looking at the code for the Ti*k*Z picture, we can see that the radius is 1 and the angles go from 60° to 120°. Using a little trigonometry, the center of the circle that it is an arc of is $\left(\frac{1}{2}, 3 - \frac{\sqrt{3}}{2}\right) \approx (0.5, 2.134)$.

Example again

```
\psset{unit=2cm}
\begin{pspicture}(-1,-1)(4,4)
\psgrid[gridlabels=0,subgriddiv=2,gridwidth=0.5pt,gridcolor=gray,
subgridwidth=0.25pt,subgridcolor=gray](0,0)(-1,-1)(4,4)
\psline[arrowsize=6pt]{->}(0,-1)(0,4)
\psline[arrowsize=6pt]{->}(-1,0)(4,0)
\pspolygon[linecolor=black,fillstyle=solid,fillcolor=orange](0,0)(1,1)(2,0)
\pscircle(0,0){1}
\pspolygon[linewidth=2pt](2,0)(2,1)(3,1)(3,0)
\psellipse[fillstyle=solid,fillcolor=green](3,3)(0.5,1)
\psarc(0.5,2.134){1}{60}{120}
\psaxes[Dx=0.5,Dy=0.5](0,0)(-1,-1)(4,4)
\end{pspicture}
```

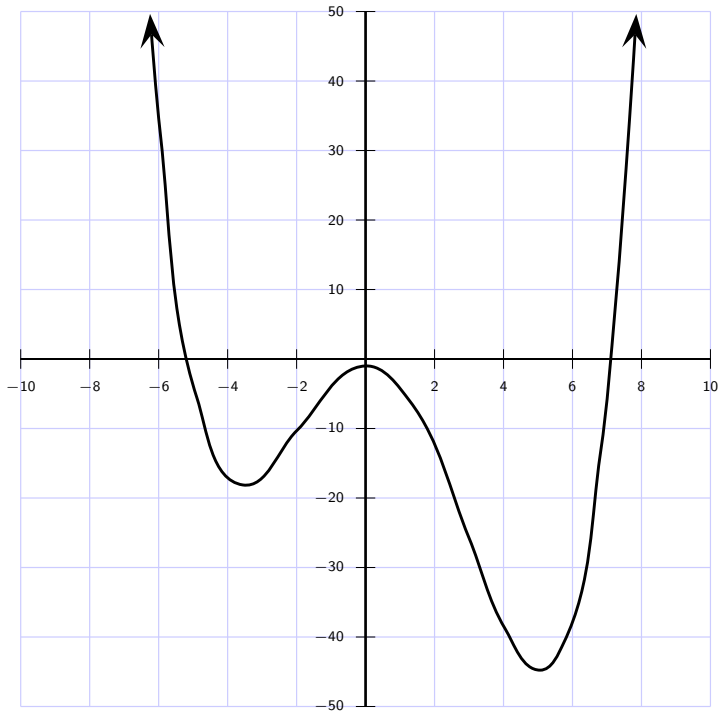


Creating Postscript Graphics: Curves

- To draw a curve, use \pscurve
- In the command above, I generally add points until I have the shape that I want
- Notice that this command draws a curve fitting the points you supplied, in the **exact** order that you supplied them.
- On the following frame I graph the function $f(x) = \frac{1}{12}x^4 - \frac{1}{6}x^3 - 3x^2 - 1$.
- Since the subgriddiv value must be an integer, which creates a background grid that is too 'busy', I draw the grid manually.
- The points used were found by evaluating the function $f(x)$ at various inputs (and rounding the results).

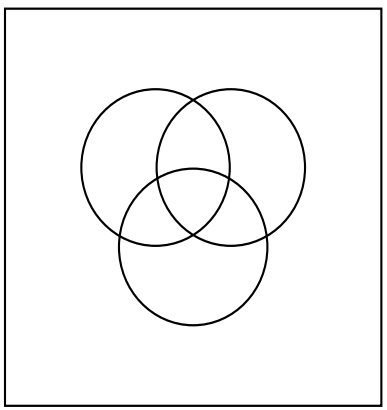
Creating Postscript Graphics: Curves

```
\psset{yunit=0.1cm,xunit=0.5cm}
\begin{pspicture}(-10,-50)(10,50)
\psline[linewidth=0.5pt,linecolor=blue!20](-10,-50)(-10,50)
\psline[linewidth=0.5pt,linecolor=blue!20](-8,-50)(-8,50)
\psline[linewidth=0.5pt,linecolor=blue!20](-6,-50)(-6,50)
\psline[linewidth=0.5pt,linecolor=blue!20](-4,-50)(-4,50)
\psline[linewidth=0.5pt,linecolor=blue!20](-2,-50)(-2,50)
\psline[linewidth=0.5pt,linecolor=blue!20](2,-50)(2,50)
\psline[linewidth=0.5pt,linecolor=blue!20](4,-50)(4,50)
\psline[linewidth=0.5pt,linecolor=blue!20](6,-50)(6,50)
\psline[linewidth=0.5pt,linecolor=blue!20](8,-50)(8,50)
\psline[linewidth=0.5pt,linecolor=blue!20](10,-50)(10,50)
\psline[linewidth=0.5pt,linecolor=blue!20](-10,-50)(10,-50)
\psline[linewidth=0.5pt,linecolor=blue!20](-10,-40)(10,-40)
\psline[linewidth=0.5pt,linecolor=blue!20](-10,-30)(10,-30)
\psline[linewidth=0.5pt,linecolor=blue!20](-10,-20)(10,-20)
\psline[linewidth=0.5pt,linecolor=blue!20](-10,-10)(10,-10)
\psline[linewidth=0.5pt,linecolor=blue!20](-10,10)(10,10)
\psline[linewidth=0.5pt,linecolor=blue!20](-10,20)(10,20)
\psline[linewidth=0.5pt,linecolor=blue!20](-10,30)(10,30)
\psline[linewidth=0.5pt,linecolor=blue!20](-10,40)(10,40)
\psline[linewidth=0.5pt,linecolor=blue!20](-10,50)(10,50)
\psaxes[Dx=2,Dy=10](0,0)(-10,-50)(10,50)
\pscurve[linewidth=1.2pt,arrowsize=10pt]{<->}(-6.25,49.66)(-5.95,32.34)(-4.89,-5.60)(-3.56,-18.12)
(-2,-10.33)(0,-1)(1.277,-6.02)(3,-25.75)(4.04,-38.76)(5.06,-44.77)(5.745,-40.84)(6.81,-13.54)
(7.234,7.12)(7.65,34.22)(7.85,49.65)
\end{pspicture}
```



Postscript Graphics Options - Example

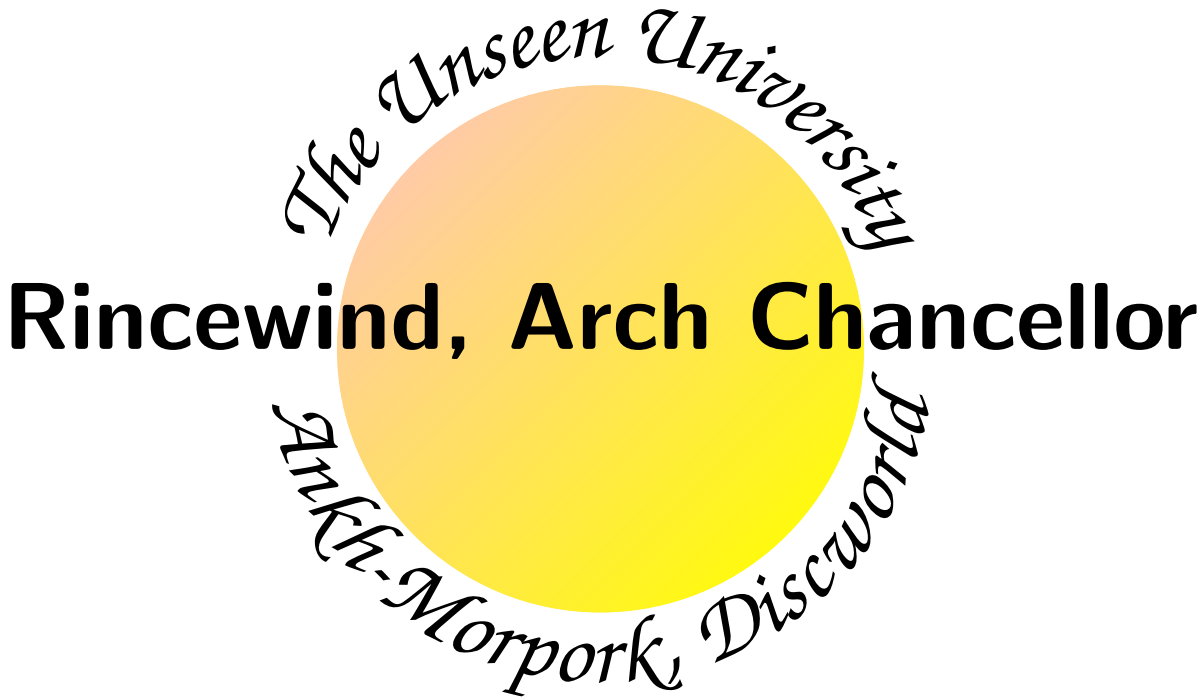
- To see an example of how to use some of these commands, we will build a Venn diagram:
- The following commands create an empty 3-circle Venn diagram:



```
\begin{pspicture}(0,0)(5,5)
\pspolygon(0,0)(5,0)(5,5)(0,5)(0,0)
\pscircle(2,3){1}
\pscircle(3,3){1}
\pscircle(2.5,2){1}
\end{pspicture}
```

Text with graphics

We can also mix text with graphics, as in the following "University Seal". (Note: Example is taken from "The LaTeX Graphics Companion" by Goossens, Rahtz, and Mittelbach, 1997)



Text with graphics

Some notes before I give the code:

- You can use the psset command for things other than setting the default size.
  \psset{linestyle=none}
  this allows us to not have to say this repeatedly for our guideline curves in our textpath
- You can define new colors as well, other than by blending
  `\definecolor{pink}{rgb}{1, .75, .8}`
  tells LaTeX how much red, green, and blue to include in our new color "pink".
- You can define a new font, which is used here for the "University Seal":
  `\newcommand{\curly}[1]{{\fontfamily{pzc}\fontsize{17}{17}\itshape#1}}`
- To get text to go along a path use the \pstextpath command
  `\pstextpath[c]{\psarcn(0,0){2}{180}{0}}{\curly{The Unseen University}}`
  c=centered, which is an alignment command
  the first required argument is the path to follow
  the second required argument is the test to put on that path
- If you want to traverse the path in the opposite direction, use psarcn instead of psarc
- The command rput puts the given whatever at the starting point at the reference point in the [..].
  B stands for baseline (also l,r,t,b for left, right, top, bottom)

Text with graphics

Packages needed: pst-text, pst-grad, pstricks

```
\definecolor{pink}{rgb}{1, .75, .8}
\begin{pspicture}(-3,-2.2)(3,2.2)
\psset{linestyle=none}
\newcommand{\curly}[1]{{\fontfamily{pzc}\fontsize{17}{17}\itshape#1}}
\pstextpath[c]{\psarcn(0,0){2}{180}{0}}{\curly{The Unseen University}}
\pstextpath[c]{\psarc(0,0){2}{180}{0}}{\curly{Ankh-Morpork, Discworld}}
\pscircle[fillstyle=gradient,gradangle=45,gradbegin=pink,gradend=yellow](0,0){1.7}
\rput[B](0,0){{\Large \itshape \bfseries Rincewind, Arch Chancellor}}
\end{pspicture}
```

Additional Resources

- The PSTricks website: PSTricks
- The TeX User's Group website for pstricks: TUG PSTricks Manual
- A user's manual for pstricks-add package: pstricks-add
- A user's manual for pst-plot package: pst-plot