# Pictures in LaTeX

This file contains sample LaTeX code for creating pictures in LaTeX and including pictures in LaTeX. For example, you an directly create nice pictures of graphs and other diagrams using Tikz. You can also use other programs that allow you to draw pictures, save them in PDF format and then include them in the document.

One of the most versatile is to create a PDF file using your favorite program (for example, Sage) and then to use the LaTeX picture environment to import it. For this to work, the PDF file for your diagram must be in the same directory (or folder) as your .tex file.

The LaTeX command "\begin{picture}(300,120)" creates an imaginary box, with a set of coordinate axes, and then allows you to put things at various locations in the box. The origin is in the lower-left corner of the box, and the axes are what you expect—something like this:


You can now put various things in the box, like circles, lines, and text. For example, the code for producing the exact picture above looks like this:

```
\begin{picture}(300,130)
\put(0,0){\circle*{5}}
\put(0,0){\line(1,0){300}}
\put(0,0){\line(0,1){120}}
\put(305,0){$x$}
\put(0,125){$y$}
\end{picture}
```

We won't have to do anything this complicated, because LaTeX also includes a command for importing a PDF file. If the name of the file is "mygraphic.pdf" then you would use

```
\put(0,0){\includegraphics[scale=0.5]{mygraphic.pdf}}
```

The (0,0) location in the **put** command can be varied in order to place the diagram in a better location. Also, the 0.5 factor in the **scale** option can be varied in order to make the diagram bigger or smaller.

1. For exampole, you can create PDF using Sage. Open up a Sage notebook and input the following commands:

   ```
   P=plot(x^2,(-10,10))
   show(P)
   P.save('parab.pdf')
   ```

   This creates a plot of $y = x^2$, shows it to you, and then saves the graph into a PDF file. Note that you could have chosen any name for the file by changing what is written inside the quotes.

   (a) Now, in your .tex file find the spot where problem 1 begins, and remove the percent sign in front of the \put command, inside the picture environment. Insert the name of your PDF file into the **includegraphics** command, compile the .tex file, and see what it looks like in the PDF previewer. Most likely it will look pretty bad at first!

   The specific code we are looking at has the form

   ```
   \begin{picture}(300,140)
   \put(B,C){\includegraphics[scale=D]{mygraphic.pdf}}
   \end{picture}
   ```

   where $B$, $C$, and $D$ are certain numbers. Play around with changing these numbers/compiling/checking the output until you have a result that looks acceptable to you.

   Now, if your diagram was very big then it might be that the picture box I provided requires you to make $D$ so small that the diagram is tiny. If this is the case, you can also play around with the 140 in the first line of the picture environment—try making it bigger, to make a bigger box. Probably you won't want to go beyond 200, but anything in the 100–200 range is reasonable. Note that if you change the 140 into something too big then there will be too much space either above or below the diagram. There is a certain balance between changing the $C$ value and changing the 140 value.

   (b) Sometimes we want to use LaTeX to add something to our diagrams, like a label. Add a line to the picture-environment code we are working with so that it looks like

   ```
   \begin{picture}(300,A)
   \put(B,C){\includegraphics[scale=D]{mygraphic.pdf}}
   \put(0,0){$y=x^2$}
   \end{picture}
   ```

(where $A$, $B$, $C$, and $D$ are appropriate numbers that you figured out in the last part). When you compile, you will now have the equation $y = x^2$ in the lower-left corner of your picture box. By changing the "(0,0)" values, try to put the equation in a more reasonable place, near upper part of the parabola. This usually takes a few minutes of playing around with compiling/checking/changing/re-compiling/etc.

If the equation is just too big to fit into your diagram, change the added line to instead read

\put(0,0){$\scriptstyle{y=x^2}$}

The **scriptstyle** command tells LaTeX to use a smaller font, the same one it uses for making superscripts and subscripts.

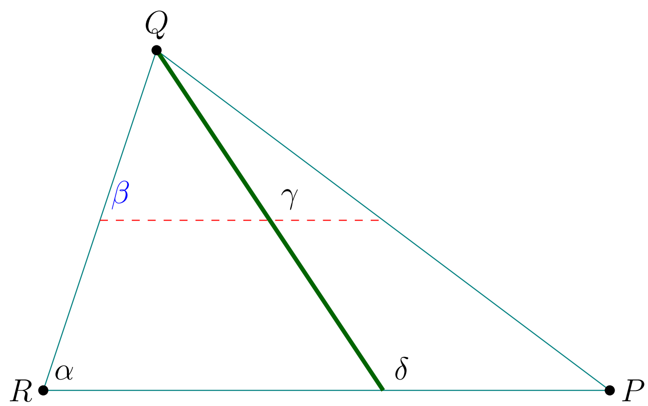
2. The previous problem showed a technique for including pictures into your LaTeX file, and you did an example where you used Sage to produce a PDF file and then imported that PDF file into your LaTeX document. Although Sage can do a lot, for custom pictures it can be pretty unwieldy. The good news is that you can use *any* software that produces a PDF, PNG, or JPG file—you don't have to just use Sage. The best things to look for are "vector graphics" drawing programs. There are lots of free ones out there; the only downside is that these usually involve installing software on your computer. Here are some examples that I know about:

   - **LaTeXDraw**. This is by far the best option that I know. It's available for Macs, Windows, and Linux. It also allows you to put TeX code directly into the pictures (for labels and so forth), which none of the others do. It requires that you have (or are willing to install) Java on your computer.
   - **Drawberry**. This is a free drawing program for Macs. Very basic, but it is okay for simple things.
   - **Inkscape**. This is a free drawing program, available for both Macs and PCs. The one for Macs requires XQuartz and so installation might be too hard for many people; also, I couldn't easily understand how to use the Mac version that I tried, but maybe this is because I'm old. Many people say this program is great, and the images that I see online look pretty impressive.
   - **Adobe Illustrator**. Very expensive, but if you happen to have it then it is good.

Another way to produce diagrams in LaTeX is via the TikZ package (pronounced sort of like "ticks"). The main advantages of TikZ are

- TikZ code is completely internal to the LaTeX document, and therefore does not require an external graphics file;
- TikZ interfaces well with LaTeX, providing a nice scheme for placing all kinds of labels within the diagram.

The following diagram was produced with TikZ:




The main disadvantage of TikZ is that it is somewhat more difficult to use than a simple drawing program like GeoGebra. To use TikZ you basically have to have coordinates for everything in your diagram. In the above diagram $R$ was placed at $(0, 0)$, $P$ at $(5, 0)$, and $Q$ at $(1, 3)$. Once you have coordinates for everything, TikZ is easily instructed to draw lines and circles, and add labels.

In the frontmatter should have three important lines:

```
\PassOptionsToPackage{svgnames}{xcolor}
\usepackage{tikz}
\usetikzlibrary{arrows}
```

These lines must appear in the order shown. The first line is not really necessary for TikZ, but it expands the set of color names that TikZ will recognize. For a complete list of such colors, see Figure 5 on the page

calque.pagesperso-orange.fr/latex/latexps.html

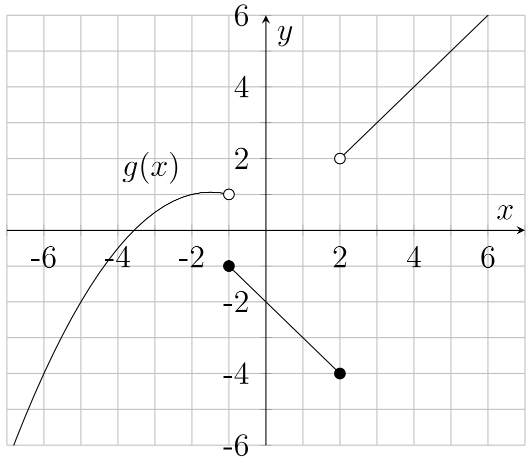
The second line tells LaTeX to load the TikZ package, and the third line enables a special "arrows" option that is important for the interface with GeoGebra talked about below.

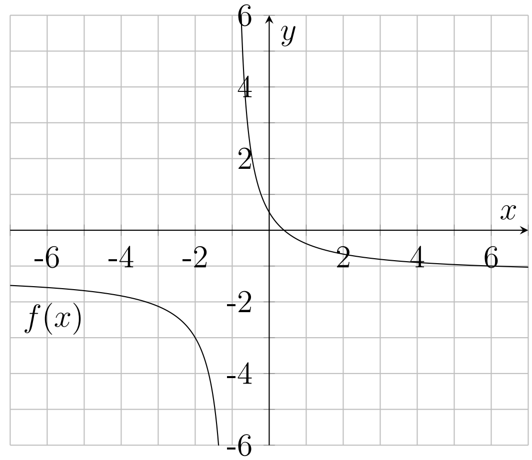
A good introduction to TikZ can be found here:

http://cremeronline/LaTeX/minimaltikz.pdf

Here are some other examples of simpler things produced using TikZ.

In the first one notice the "minipage" environment which places the images next to each other. Also, you will need the package \usepackage{pgfplots}.




Here's one more: