

Creating graphics using TikZ and SAGEMATH

Ben Salisbury

January 24, 2017

1 TikZ

TikZ is a language to be used inside \LaTeX in order to create publication quality graphics. For some breathtaking examples of the power of TikZ, go to [2] and look at the gallery!

In order to use the language, be sure to add the following line to the preamble of your \TeX document:

```
\usepackage{tikz}
```

The following examples are those which were shown during the talk.

Example 1. The code

```
\begin{tikzpicture}%[scale=2]
  \path[-] (-1,0) edge (5,0);
  \node[circle,draw,scale=.35,label={south:$a$},fill=black]
    (left) at (0,0) {};
  \node[circle,draw,scale=.35,label={south:$b$},fill=white]
    (right) at (1,0) {};
  \node at (3,0) {$\times$};
\end{tikzpicture}
```

produces



The code

```
\begin{tikzpicture}[scale=2]
  \node (left) at (0,0) {};
  \node (right) at (1,0) {};
  \node at (.25,0) {$($};
  \node at (.75,0) {$]$};
  \path[<->] (left) edge (right);
  \path[-,color=red,very thick] (.25,0) edge (.75,0);
\end{tikzpicture}
```

produces



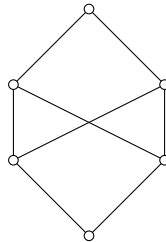
Example 2. One can also create graphs in TikZ. First, so that we don't have to repeatedly add the same (cumbersome) options to each node, we can define a new option which combines them all. Add

```
\tikzset{dot/.style={circle,draw,scale=.35,fill=white}}
```

to the preamble of your document. Now use

```
\begin{tikzpicture}
\node[dot] (0) at (0,0) {};
\node[dot] (-1) at (-1,1) {};
\node[dot] (1) at (1,1) {};
\node[dot] (-2) at (-1,2) {};
\node[dot] (2) at (1,2) {};
\node[dot] (3) at (0,3) {};
\path[-]
(0) edge (-1)
(0) edge (1)
(-1) edge (-2)
(-1) edge (2)
(1) edge (-2)
(1) edge (2)
(-2) edge (3)
(2) edge (3);
\end{tikzpicture}
```

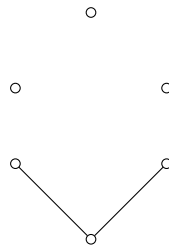
to produce



For-loops can be used in TikZ to streamline code. For example,

```
\begin{tikzpicture}
\foreach \x/\y in {0/0, -1/1, 1/1, -1/2, 1/2, 0/3}
{\node[dot] (\x\y) at (\x,\y) {};}
\path[-]
(00) edge (-11)
(00) edge (11);
\end{tikzpicture}
```

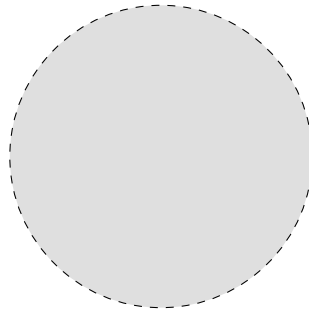
produces



Example 3. One can draw a shaded circle using TikZ. For example,

```
\begin{tikzpicture}
  \draw[dashed,fill=gray!25] (0,0) circle [radius=2cm];
\end{tikzpicture}
```

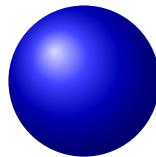
produces



Example 4. TikZ has a shading feature that can make 2-dimensional objects look 3-dimensional. For example,

```
\begin{tikzpicture}
  \shade [ball color=blue] (0,0) circle [radius=1cm];
\end{tikzpicture}
```

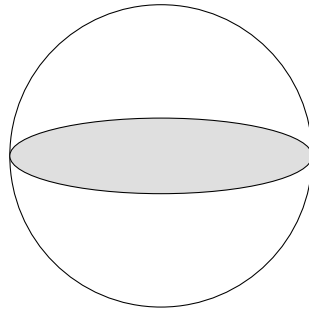
produces



(For some discussion on the “correct” way to use this feature, see [1].) However, maybe you want to draw a sphere like you would on the board? Try

```
\begin{tikzpicture}
  \draw (0,0) circle [radius=2cm];
  \draw[fill=gray!25] (0,0) circle [x radius=2cm, y radius=.5cm];
\end{tikzpicture}
```

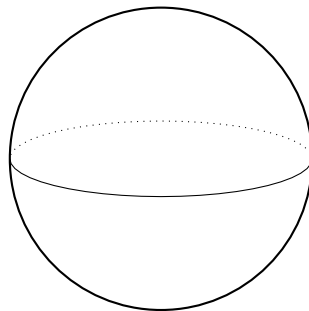
to get



If you insist that the equator of the sphere have a portion of its line dashed in the output, try

```
\begin{tikzpicture}
  \draw[thick] (0,0) circle [radius=2cm];
  \draw[dotted,domain=0:180] plot ({2*cos(\x)}, {.5*sin(\x)});
  \draw[domain=180:360] plot ({2*cos(\x)}, {.5*sin(\x)});
\end{tikzpicture}
```

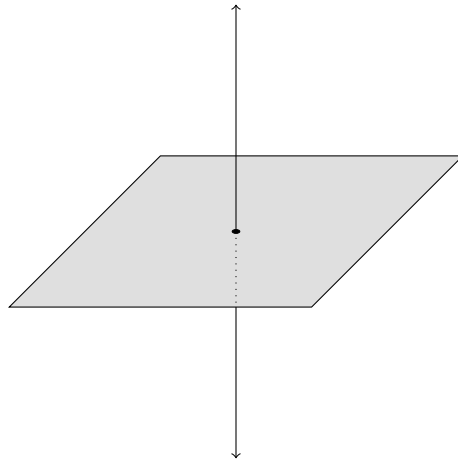
to get



Example 5. Finally,

```
\begin{tikzpicture}
  \draw[-,fill=gray!25]
    (-3,-1) -- (-1,1) -- (3,1) -- (1,-1) -- cycle;
  \draw[fill=black] (0,0) circle [x radius=.05, y radius=.025];
  \path[->] (0,0) edge (0,3);
  \path[->] (0,-1) edge (0,-3);
  \path[dotted] (0,0) edge (0,-1);
\end{tikzpicture}
```

outputs



2 SAGEMATH

There is so much one can do with graphics in SAGEMATH. Please refer to [3] for just some of the two-dimensional possibilities with SAGEMATH.

Before getting started, we need to initialize some variables.

```
var('x,y,z')
```

Example 6. We wish to plot $y = x^2 - 2$ and $y = \frac{1}{2}x$ on the same graph. In SAGEMATH, type

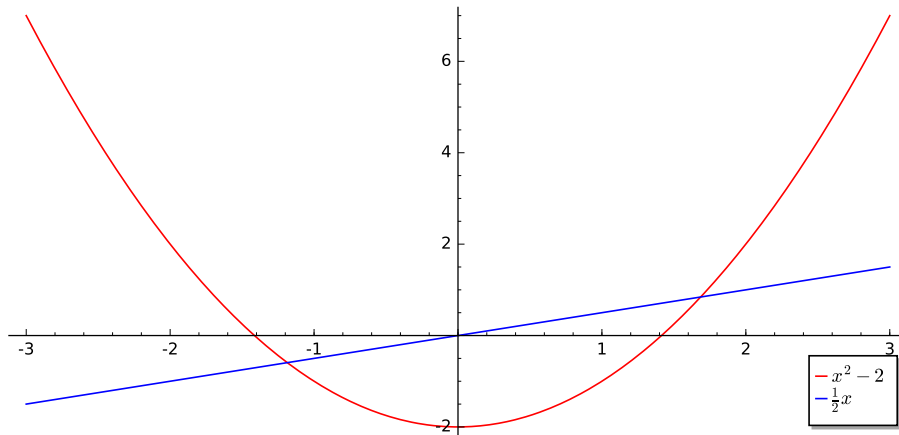
```
u(x) = x^2 - 2
v(x) = (1/2)*x
U = plot(u, (-3,3), color='red', legend_label='$x^2-2$')
V = plot(v, (-3,3), color='blue', legend_label='$\frac{1}{2}x$')
(U+V).show(axes=False)
```

This will produce an image automatically. However, if you want to save the image, say, as a PDF file, then type

```
(U+V).save('<path_to_file>/parabola.pdf')
```

Note that there is nothing special about the name “parabola.pdf”, but it had to be named something! The saved file can then be imported in L^AT_EX using

```
\includegraphics{<path_to_file>/parabola.pdf}
```



It should be added that, specifically, the code

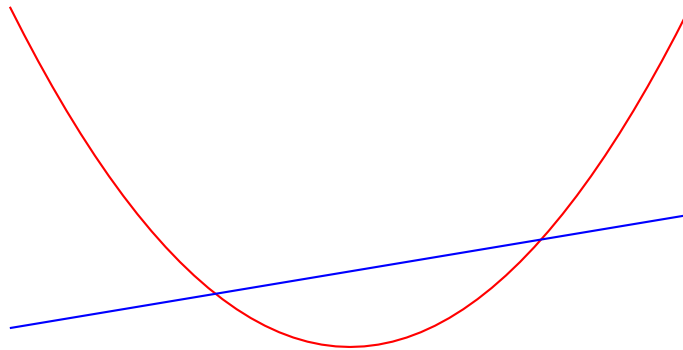
```
\includegraphics[width=.8\textwidth]{<path_to_file>/parabola.pdf}
```

was used so that the size of the image could be controlled.

While this method is best served for more difficult functions (like in the next example), it's nice to compare to TikZ when possible. This same example built using directly in \LaTeX has code

```
\begin{tikzpicture}[xscale=1.5, yscale=.5]
  \draw[-,red,thick,domain=0:3] plot (\x,\x^2-2);
  \draw[-,red,thick,domain=0:3] plot (-\x,\x^2-2);
  \draw[-,blue,thick,domain=-3:3] plot (\x,.5*\x);
\end{tikzpicture}
```

and outputs as



Axes can also be added if desired.

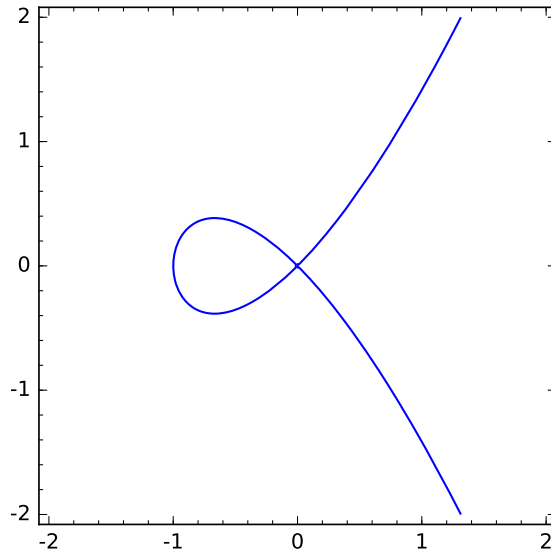
Example 7. To plot the curve whose equation is $y^2 = x^2 + x^3$, we need to use an implicit plot in SAGEMATH. The code

```
f(x,y) = y^2 - x^2 - x^3
F = implicit_plot(f(x,y) == 0, (x,-2,2), (y,-2,2))
F.save('<path_to_file>/implicit.pdf')
```

and

```
\includegraphics[width=.5\textwidth]{<path_to_file>/implicit.pdf}
```

produces



References

- [1] “How to draw a shaded sphere?” <http://tex.stackexchange.com/questions/54193/how-to-draw-a-shaded-sphere>
- [2] T_EXample.net, <http://texample.net/>
- [3] <http://doc.sagemath.org/html/en/reference/plotting/index.html>