How to include an SVG image in LaTeX

Johan B. C. Engelen[*]

MESA+ and IMPACT Research Institutes, TST group, University of Twente, Enschede, the Netherlands

Abstract

How to make a LaTeX document with vector images, where the text in the images has exactly the same font and size as in normal text? This article describes how this is done using the 'PDF/EPS/PS + LaTeX' output feature of Inkscape 0.48. Inkscape can export the graphics to PDF/EPS/PS, and the text to a LaTeX file. When the LaTeX file is input in the LaTeX document, the PDF/EPS/PS image is included with overlaid text. Because typesetting of the text is done by LaTeX, LaTeX commands can be used in images, such as writing equations, references and shorthand macros. *(requires Inkscape version 0.48 or higher)*

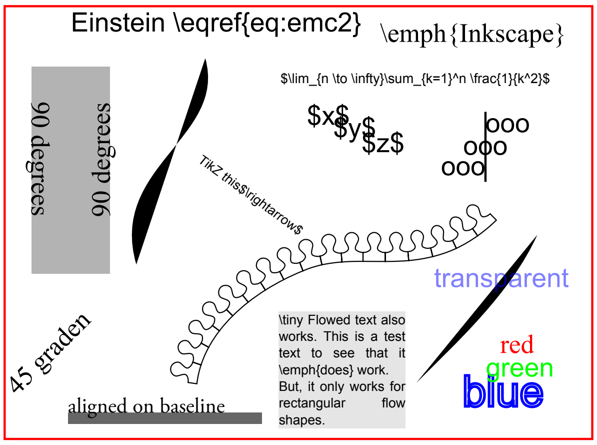

Einstein \eqref{eq:emc2}
\emph{Inkscape}
$\lim_{n \to \infty}\sum_{k=1}^n \frac{1}{k^2}$
90 degrees
90 degrees
x
y
z
ooo
ooo
ooo
TikZ this$\rightarrow$
transparent
\tiny Flowed text also works. This is a test text to see that it \emph{does} work. But, it only works for rectangular flow shapes.
red
green
blue
45 graden
aligned on baseline

Figure 1: The test SVG image, as it is seen in Inkscape (exported to PDF *without* LaTeX option).

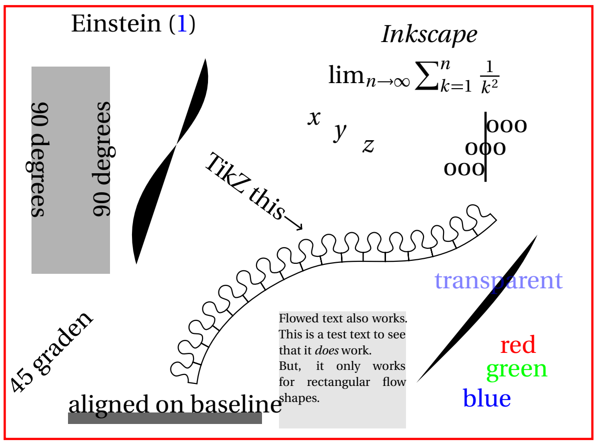

Einstein (1)
Inkscape
$\lim_{n\to\infty}\sum_{k=1}^n \frac{1}{k^2}$
90 degrees
90 degrees
x
y
z
ooo
ooo
ooo
TikZ this→
transparent
Flowed text also works. This is a test text to see that it *does* work. But, it only works for rectangular flow shapes.
red
green
blue
45 graden
aligned on baseline

Figure 2: The test image, exported to PDF *with* LaTeX option.

1. Introduction

An advantage of 'drawing' images in LaTeX code using e.g. TikZ, is the more uniform layout of the document as the image text will be typeset by LaTeX: the image font will be equal to the rest of the document (size, ligatures, etc.). Moreover, TikZ enables easy use of mathematical expressions inside images. But, the programming nature of drawing with TikZ is not liked by many, and most will turn to a vector graphics drawing program, such as Inkscape. However, then, obtaining the document's font and mathematical expressions in images becomes a problem.

Inspired by the 'epslatex' GNUPlot output terminal, a new feature was added to Inkscape, combining the advantages of LaTeX text typesetting and drawing in Inkscape. This article describes how to use Inkscape to create images in which the text font is equal to the document's font, LaTeX mathematical expressions can be written, and LaTeX commands can be used. The automated inclusion of SVG graphics while using the new 'PDF+LaTeX' Inkscape option is described.

(This article assumes the use of PDFLaTeX, but most will probably work just as well for EPS/PS documents.)

[*]Corresponding author: j.b.c.engelen@utwente.nl

2. Creating the image with Inkscape

The image can be drawn as usual in Inkscape, see figure 1. All text in the image will be compiled by LaTeX, and therefore may contain LaTeX commands. For example, it is possible to refer to

$$E = mc^2. \tag{1}$$

Font type and size do not matter and will not be exported to LaTeX.

2.1. *Text alignment*

I have tried to get the positioning of text in the end result as close as possible to the positioning in the SVG file. Text is aligned on the anchor point visible in Inkscape. This means that vertically, the baseline of the resulting text corresponds to the baseline of the text in Inkscape. Horizontally, it depends on the alignment (left, center, right).

2.2. *Text colour*

Text colouring works. If the text has a fill colour, this colour will be exported to LaTeX as an RGB colour. Although stroking text will not work in LaTeX, when the text as a stroke colour *and no fill colour*, the stroke colour will be exported to LaTeX as an RGB colour as if the text had that colour as fill. Text transparency is exported too.[1]

2.3. *Exporting to PDF*

When the image is saved to PDF, one has to set the 'PDF+LaTeX' option in the dialog that pops up after specifying which file name to save to. The bounding box of the image in Inkscape *including text* will be used as the bounding box for the image if the exported area is the drawing (`--export-area-drawing`). It is strongly recommended to use this 'exported area is drawing' option, because text will not be clipped to the page size.

2.3.1. *Using Inkscape's command line*

Exporting to PDF and LaTeX can also be performed through the command line interface: `inkscape -D -z --file=image.svg --export-pdf=image.pdf --export-latex`. Note the added `--export-latex` option.

[1]Transparency is not exported when exporting to EPS or PS, because it is not supported by those formats.

3. Including the image in LaTeX

The image should be included in the document by inputting the `.pdf_tex` file created by Inkscape. It is possible to set the width of the image by defining `\svgwidth`:

```
\begin{figure}
  \centering
  \def\svgwidth{\columnwidth}
  \input{image.pdf_tex}
\end{figure}
```

The `.pdf_tex` file created by Inkscape contains a `picture` environment, that includes the PDF exported by Inkscape and places text on top of it. The result is figure 2, note that the font exactly matches the document's font, and the link to the equation works. If no width is specified, the image will have its original width. The `\svgwidth` is forgotten after including a figure, so one must redefine the width for each figure (`\svgwidth` is set empty by `image.pdf_tex`).

3.1. *When images are not in the document's directory*

When images are not located in the document's directory, but in a sub-directory, one has to add that directory to the graphics search path (unfortunately). For example, if one's images are in sub-directory `images`, add the following to the preamble:

```
\graphicspath{{images/}}
```

Alternatively, when the images are not in a subdirectory of the document, they can be accessed with the `import` package, adding

```
\usepackage{import}
```

to the preamble, and including the image with

```
\import{<path>/<to>/<file>}{<filename>.tex}
```

3.2. *Automatic export*

('write18' must be enabled, see the `epstopdf` package documentation. Add `-shell-escape` to the command line when calling `pdflatex`.)

3.2.1. *Workflow*

Whenever the SVG file is updated, it is possible to have LaTeX automatically call Inkscape to export the image to PDF and LaTeX again. This simplifies the workflow to

- Modify the SVG image in Inkscape;
- Save the SVG (Ctrl+S, no need to export to PDF);

- Recompile LaTeX document. PDFLaTeX will notice the SVG file has changed, and will automatically do the export for you.

3.2.2. LaTeX code

Add the following code to the preamble of your document:

```
\newcommand{\executeiffilenewer}[3]{%
 \ifnum\pdfstrcmp{\pdffilemoddate{#1}}%
 {\pdffilemoddate{#2}}>0%
 {\immediate\write18{#3}}\fi%
}
\newcommand{\includesvg}[1]{%
 \executeiffilenewer{#1.svg}{#1.pdf}%
 {inkscape -z -D --file=#1.svg %
 --export-pdf=#1.pdf --export-latex}%
 \input{#1.pdf_tex}%
}
```

When an image is included via `\includesvg`, it is automatically exported to PDF+LaTeX again by Inkscape when the SVG file has changed (note that the image must be specified *without* file extension):

```
\begin{figure}
  \centering
  \def\svgwidth{\columnwidth}
  \includesvg{image}
\end{figure}
```

The preamble code will only work if Inkscape is in the search path of your operating system (Windows: http://www.computerhope.com/issues/ch000549.htm).

4. Example of using another font

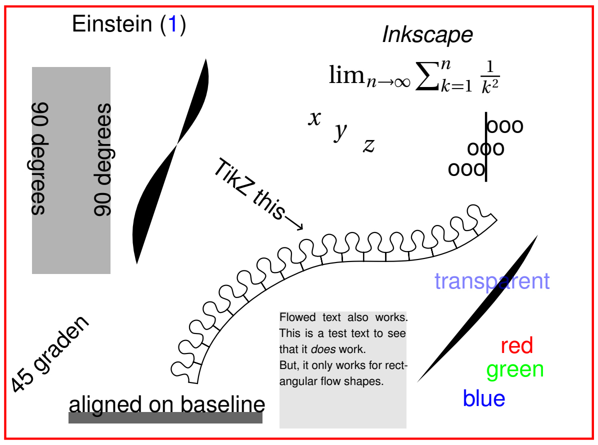
When another font is chosen, the image font will of course also change to this font. This is shown in figure 3, where the same image file is included as in figure 2.

5. Known bugs and limitations

Please report any bug you find on https://bugs.launchpad.net/inkscape.

There are some known limitations/bugs.

- Exporting to EPS+LaTeX, the bounding box is always set snugly around the drawing *without* text. There is debate about whether this is a bug in Inkscape or not. This becomes a problem when text is desired 'outside' the other parts of the drawing. A workaround is drawing a rectangle with zero stroke width as bounding box around the drawing. Alternatively, one can export to PS+LaTeX instead and renaming the resulting `.ps` file to `.eps`. See https://bugs.launchpad.net/inkscape/+bug/595821.
- Flowed text is only exported for rectangular flow shapes.
- There are no other scaling options beside specifying the width.



Flowed text also works. This is a test text to see that it *does* work. But, it only works for rectangular flow shapes.

Figure 3: The test image, exported to PDF *with* LaTeX option. Note that the image's font matches the newly chosen font. The included LaTeX file is the same as in figure 2.