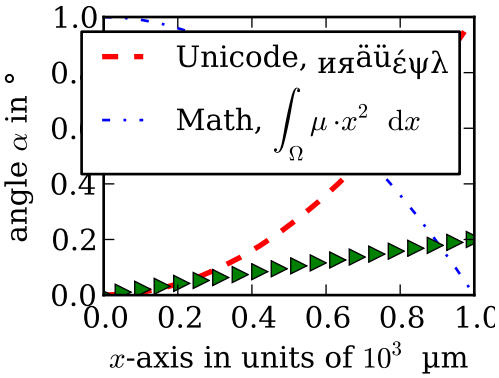
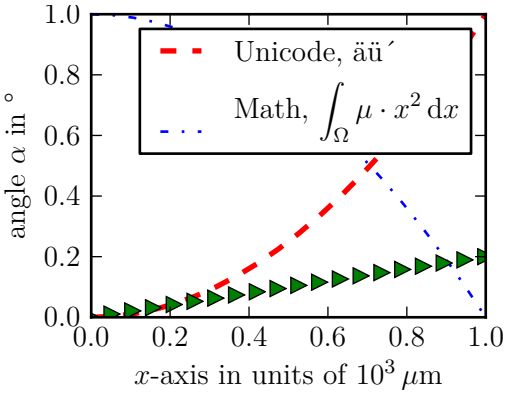
PDF/Latex Output Problems

When creating figures to be included in Latex documents, saving them as PDF is probably the method of choice. The appearance of text elements within the graphs however differs from the Latex typesetting. Naturally, figures created with the *text.usetex* option look more consistent as they are rendered with Latex as well. But Pdflatex, which is used by matplotlib, has no real unicode support and limits the possibilities of using unicode in figures.

The following figure contains some letters from different languages, math text and normal text. It was saved using the default PDF output option:



(a) Matplotlib PDF output. The serif font is not consistent with the default font from Latex. Math-text not consistent with normal text. Some letters are not aligned correctly. Tight-layout is not working correctly (too much space or clipping effect).

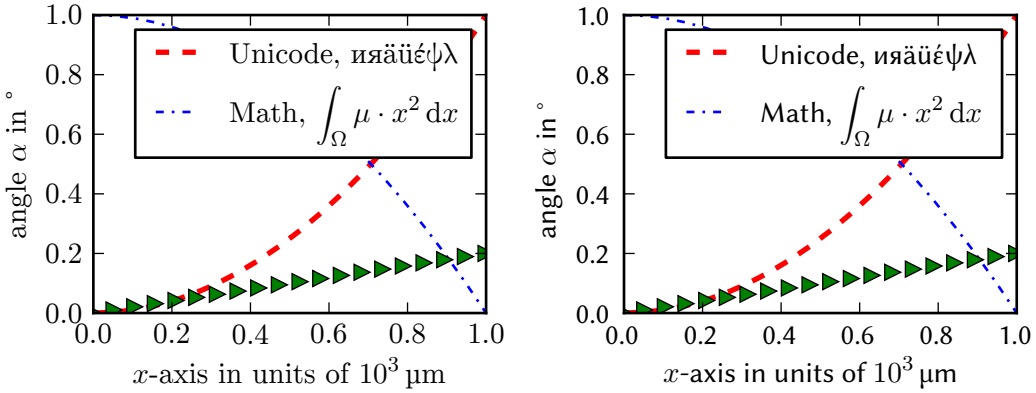


(b) Matplotlib with Latex rendering. Figure texts, math formulas and this document are using the same font and look consistent, but the conversion process to PDF increases the filesize a lot. Some unicode letters are missing completely and the micro sign was replaced by an italic mu-symbol from the math font. Baseline offset of the second label is incorrect.

Figure 1: PDF figures created with matplotlib v1.1.1.

Pgf Backend with Xelatex

For real unicode support the use of Xelatex or Lualatex is mandatory. In order to use these new Latex implementations in Matplotlib, a new backend is required as the *pstricks* methods for drawing the figures are not available anymore. The *pgf* package offers an alternative that works for Pdflatex, Xelatex and Lualatex alike. Figures are drawn using the *pgfpicture* environment and can be included in Latex documents as raw commands (\input) or can be pre-compiled to PDF for inclusion (\includegraphics). Still, Xelatex needs a font with extended glyph coverage for the unicode characters. A unicode variant of the *Computer Modern* fonts used by Latex is available at `http://sourceforge.net/projects/cm-unicode/` and should be used if the figures are supposed to blend in. Of course, any other installed system font will work as well but one has to search for a math font package that harmonizes with the selected text font.



(a) Computer Modern Unicode font

(b) Main font switched to Biolinum

Figure 2: Figure saved as textfile with pgf commands. The file has been included two times. For the second inclusion the document's main font is changed without recreating the figure in matplotlib.