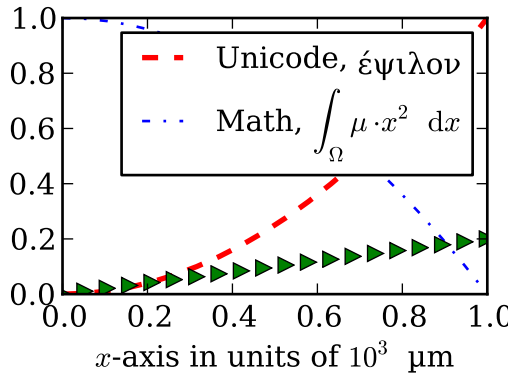


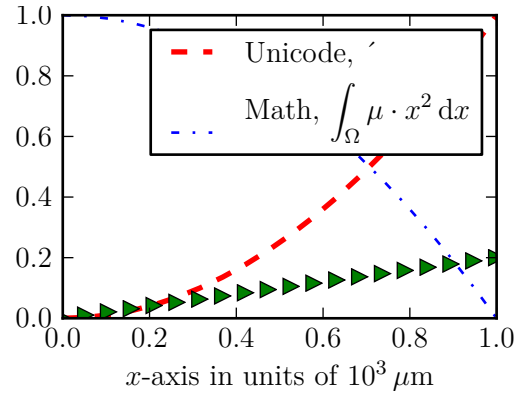
PDF/Latex Backend Problems

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

The following figure contains greek unicode letters, math text and normal text. It is produced using the PDF backend:



(a) Matplotlib PDF backend. Serif font not consistent with the Latex default font. Math-text does not fit to normal text. Greek letters are not aligned correctly.



(b) Matplotlib with Latex rendering. Consistent look in Latex documents, better looking math-text, but no real unicode support. The conversion process to PDF increases the filesize a lot. Greek letters are missing and the micro sign is replaced by an italic mu-symbol from the math font. Baseline offset of the second label is incorrect.

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 *pgfpicture* package offers an alternative that works for Pdflatex, Xelatex and Lualatex alike. These commands can be included in Latex documents directly (`\input`) or be pre-compiled to PDF for inclusion (`\includegraphics`). For a most consistent appearance, the font *Computer Modern Unicode* available at <http://sourceforge.net/projects/cm-unicode/> must be installed, since the default Latex font is most of the unicode letters. Any other installed system font will work as well though.

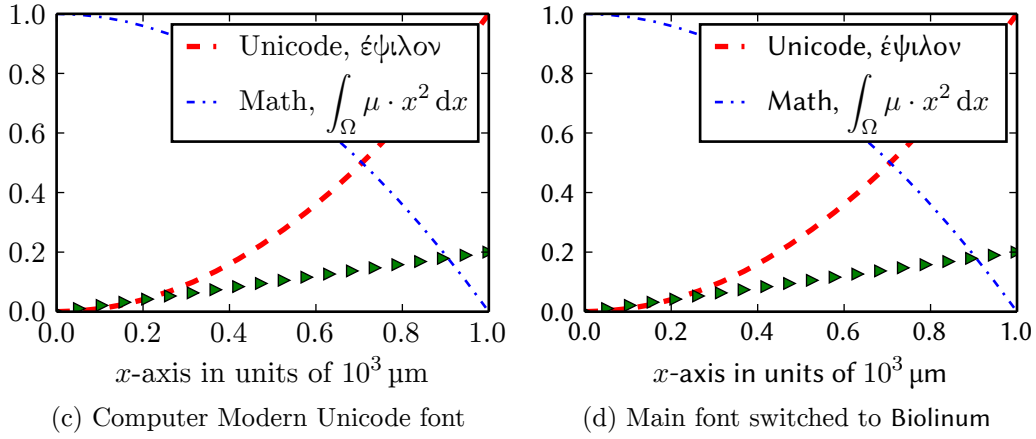


Figure 1: Pgf figure commands included in a Xelatex compiled document. The fonts were changed within the document without recreating the figure in matplotlib.