

Example of Using Latex to Create PDF Document

Donald D. Daniel

May 8, 2003

Abstract

A simple document to demonstrate and explain how to create pdf documents.

Publication of technical papers on the internet is hampered by the difficulty of presenting fancy mathematical equations and crisp detailed line graphics in an ordinary HTML web page. Pdf documents can solve this problem. They are not read with a browser, but your browser can download them to your disk, then you can read them with Adobe Acrobat on Windows, or with “gv” on Linux.

A simple graph can be generated directly by a simple program written by the user. More elaborate graphs of computer generated data can be produced using the linux program gnuplot. Mechanical and electrical drawings can be generated by the linux program xfig. All of these kinds of graphics and more can be included in the final pdf document using Latex.

This is intended to be a simple practical example of the creation of a pdf document containing a computer generated graph using a linux system. The “tetex” package on linux contains the “Latex” word processing system, and various supporting pieces of software. The “tetex-doc” package contains the documentation, which appears at /usr/doc/texmf after installation. This example was done using Debian Linux 3.0 on a computer using the Alpha chip, with no modification or configuration of the stock tetex package.

This document was first created as the file “pdfex.tex” in plain text using the vi editor. A link to pdfex.tex is contained in www.waltzballs.org/other/pdf.html. In pdfex.tex the first six commands and the last command would

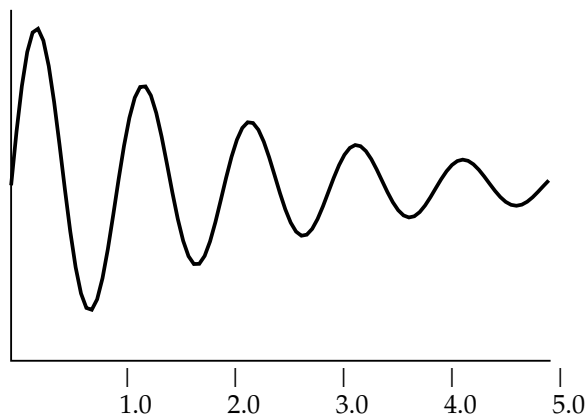


Figure 1: This is temp1.eps

probably be in most simple latex documents that include graphics. The group of six commands to include the figure in the middle of pdfex.tex would be repeated for each figure. Latex “floats” the figures to convenient locations in the text. The figure in this document was created with the program “graf1” contained in the article “How to Program a Computer” found at <http://www.waltzballs.org/other/prog.html>. The computer program creates the file “temp1”. The file “temp1” must be converted from ordinary Postscript to Encapsulated Postscript (eps). This is done by turning on the X11 system and looking at it with the linux program “gv”. With the cursor in gv the lower left coordinates are found to be 154, 366. The upper right are 465, 587. Using the vi editor, the line

```
%%BoundingBox: 154 366 465 587
```

is added in temp1 after the first line. Temp1 is now an eps file. Its name must be changed to temp1.eps.

The graph in Figure 1 shows the result.

The original latex version of this document, pdfex.tex, was converted to pdfex.dvi with the two commands

```
latex pdfex
```

```
latex pdfex
```

where the same command must be issued twice to resolve references to citations and figures. If the command is not issued twice, the references will

appear as question marks. At this point Latex has created for us pdfex.dvi, which is the new form of our document, and pdfex.log, which contains a log of the transformation process, including error messages, if any. Pdfex.aux is for the use of latex, and we can ignore it. Then, the file pdfex.dvi was converted to the postscript file pdfex.ps with the command:

```
dvips -Pcmz pdfex > pdfex.ps
```

Finally, the file pdfex.ps was converted to pdf with the command

```
ps2pdf pdfex.ps
```

The final result, pdfex.pdf can be viewed and printed from gv with the command:

```
gv pdfex.pdf
```

The -Pcmz option for dvips produces type 1 fonts; without it bitmapped fonts would be produced. They both print well on a printer, but on most computers, type 1 fonts look better on the screen. The particular font chosen here is computer modern. Both the printer and the printing software have to be configured for 600 dots per inch or better for type 1 computer modern to print perfectly. It is easy to change 300 to 600 in the filter referred to in /etc/printcap. The filter is referred to in the line :if=/etc/magicfilter/myfiter, where myfilter is whatever filter matches your printer. In myfilter replace all occurrences of 300 with 600.

The documentation in /usr/doc/texmf is scattered in several different subdirectories. It will probably be sufficient for the needs of most users, but references [1] [2] will help for more exacting requirements.

References

- [1] Helmut Kopka and Patrick W. Daly, “A Guide to Latex”, third edition, Addison-Wesley, 1999.
- [2] Henry McGilton and Mary Campione, “Postscript by Example”, Addison-Wesley, 1992.