Serious open-source software for streetwise folk

RWD Nickalls,
Department of Anaesthesia,
Nottingham University Hospitals,
City Hospital Campus,
Nottingham, UK.

dick@nickalls.org
www.nickalls.org

Contents

Jan 2008 howtodn6.tex

1 Introduction

WELCOME to the 'Free Universe'! As a general rule for any given commercial software there is a much better one in what I call the 'free universe' of the internet—its all just a question of knowing where to look, and how to install it.

So which software should we be interested in and why? Well, I would suggest starting by installing a recent version of Linux e.g. Mandriva[1] 2008, and then proceed to learn how to use some of the standard computing tools which will help you upshift into the 21st century. Note that all the useful software tools (and much more of course) are included as standard in any Linux system. The three main systems to consider are as follows.

1.1 Typesetting

I would strongly recommend you master the TEX and LATEX system established by Donald Knuth and Leslie Lamport respectively. There are free systems for Linux, Microsoft and Macintosh. On the way you will also need to understand about writing ASCII text files, and how to edit them using so-called 'clean' text editors. You will then also be able to generate pdf files (best and free via PDFLATEX), and use pdf for computer presentations.

To facilitate using TEX and LATEX most people use some sort of software software'front-end', typically either Emacs (experts only really) or the more recent and excellent Kile. These facilitate writing, typesetting, and automatic conversion to PDF, HTML, XML etc. and generally make life easy for you.

1.2 Programming

Learning a programming language is *amazingly* useful, and I seriously recommend it. It allows you to do all your mathematics & statistics (all those things you can't really do on a calculator), solve problems, write data to files, control printers, control electronic devices in the outside world via the serial port (e.g. interface the PC to your kids' Technik-Lego etc.), make graphic shells and interfaces to facilitate using other software etc etc. Note that all the software you use at the moment is of course written using one of the programming languages—usually Perl or C.

The most established and widely used languages are the free Open Source[2] systems Perl, Python, C and C++. These are freely available for all computer platforms and have massive online libraries available on the internet. Furthermore, all these programming languages come as standard on all Linux implementations.

Of the available languages I would suggest you only consider the main 'open source' languages mentioned above. Of these I would strongly suggest Perl (http://www.perl.com/), which is pretty easy, its structure is similar to C and TEX, and it has vast libraries of routines which can be downloaded from the

[1] Mandriva was originally called *Mandrake*.

[2] The Open Source initiative defines nine requirements which software must comply with for it to be regarded as 'Open Source'—see www.opensource.org/docs/definition.html.

internet.

Two important reasons for using an open source programming language are (a) any programs you write can then be run on any computer platform (i.e. you are not restricted to a particular platform, e.g. Micro$oft), and (b) the vast libraries mean you don't have to re-invent the wheel—the chances are that there already exists a suitable library of just the sort of routines you require.

1.3 Graphics

The Linux system for editing and manipulating an image is **The GIMP** (equivalent to the commercial PhotoShop). As regards actually drawing curves, graphs etc then the most useful and flexible software is programmable since this gives fine control and batch capability, for example, PostScript, GNUplot, mathsPIC.

2 Linux

2.1 Introduction

Linux is a public domain (free) Unix-clone operating system for PCs, and has many advantages over the Micro$oft Windows operating system, being extremely stable and essentially bug-free. Remember, Unix was around long before Micro$oft. Linux currently has about 20% of the market share, and this is increasing fast owing to the fact that it is extremely powerful, secure, and essentially free. The extraordinary growth of the open-source culture has of course played a part too.

Since Linux is freely available, you can in fact install it by downloading all the software bits and pieces from the internet. However, this is not easy unless you really know what you are doing. In practice, it is actually much more convenient to install one of the commercial implementations (e.g. I use Mandriva; the recent 2007 PowerPack version costs about £60 from PC World, or internet websites). The great advantage of such a package is that it is easy to install 'out of the box' (DVD), and it comes with all the standard 'tools' and support packages etc. The Mandriva package includes special configuration tools which are extremely useful—essential unless you really know what you are doing. While Mandriva is a good and popular Linux system, there are many other implementations, for example Ubuntu, KUbuntu (the KDE version), Debian (experts only) and SUSE. Only buy the very latest version of a Linux system—so ask around to check what the latest version is.

Updating these systems has been revolutionised by the widespread availability of broadband internet services, since these allow fast download of large files. However, buying the 'boxed' DVD version is very useful since everything is on the DVD. Note that Linux can be generally categorised regarding the system it uses for installing and upgrading software ('packages'), namely by the 'package management system' it uses, either the **rpm**[3] system (e.g. RedHat, Mandriva, SUSE) or **dpm** (e.g. Debian, Ubuntu, Kubuntu).

In practice most people have both Linux and Micro$oft on their hard drive, each installed in separate partitions. The important thing to remember is to always install Micro$oft first, and only then install Linux on another partition

[3] Redhat Package Manager, after RedHat which invented it.

(this is because Micro$oft seems unable to contend with competing partitions during installation). At startup you can then choose from a menu whether to boot into Linux or Micro$oft etc. Note that an extremely useful feature of Linux is that it is able to read and manipulate files in the Micro$oft partition (Micro$oft is unable to look into the Linux partition), and can therefore move files from one partition to the other.

Linux is very flexible, and gives you access to both a 'command line' (where you can type commands and instructions to get Linux to do what you want), and windows (the Linux X-windows system is rather better than the Miscro$oft equivalent, but the look-and-feel is essentially similar).

Starting to use Linux for those who were not weaned on DOS or Unix may be a bit of a culture-shock at first—a bit like going from the UK to the French Riviera—but since the quality of life is so much better folks tend to stay even if it does mean they have to learn the language and drive on the other side of the road :-).

In practice you can still use your old WORD documents in Linux (using OpenOffice), but generally Linux folks use the much better TEX system. Similarly, you can run Powerpoint presentations on OpenOffice, but Linux users generally make their presentation in PDF, which is far more useful and powerful. Photoshop users will find the Linux equivalent **The GIMP** just as excellent. Some of the most commonly used Linux utilities are as follows.

```
k3b                     %% burning CDs & DVDs
mozilla                 %% web browser
Konqueror               %% web browser
TeX and LaTeX           %% typesetting system
OpenOffice              %% an OS version of MS-WORD
kwrite                  %% text editor
kedit                   %% text editor
kile                    %% advanced text editor for TeX & LaTeX
emacs                   %% advanced text editor
emacs-X11               %% Xwindows version of Emacs
GIMP                    %% picture editor (equivalent to PhotoShop)
Perl                    %% programming language
Python                  %% programming language
C, C++                  %% programming languages
Quanta Plus             %% html editor
wine                    %% Microsoft Windows emulator
kdegraphics-kview       %% picture viewer
kdegraphics-kuickshow   %% picturer viewer
kdegraphics-ksnapshot   %% screenshot utility
```

Once you have Linux installed you can then gradually learn how to use it; no more entering 20-character registration codes, no more viruses, no more document corruption, no more crashes. In fact before long you will have completely forgotten how to navigate your old Micro... er ... Micro$oft stuff :-)

2.2 Linux usenet groups

There are loads of these, and all are *very* active—hundreds of postings/day. However the main ones are

```
alt.os.linux.mandrake
comp.os.linux.questions
comp.os.linux.development.apps
uk.comp.os.linux
```

2.3 Linux magazines/journals

There are several monthly paper Linux magazines. Some of the electronic magazines and information sites are as follows.

- Linux Journal

  This is a monthly Linux magazine

  http://www.linuxjournal.com/

- Linux Format

  http://www.linuxformat.com/

- The Linux Chronicles

  This is a series of articles on things-linux by Ron Stephens.

  http://www.awaretek.com/linuxchronicles.html

3 TEX and LATEX

3.1 Introduction

TEX and the associated system of macros known as LATEX, together with the character font system METAFONT, is the ultimate typesetting system—and completely free! It is currently the most powerful and widely used formatting system for producing book-quality text, as well as for generating all the usual publishing formats e.g. PDF, HTML, XML etc. TEX, METAFONT and LATEX are all in the public domain and are therefore completely free. The web home of TEX is http://www.tug.org/.

TEX and METAFONT were created by Donald Knuth, the now famous professor of computing at Stanford University, during the period 1977–1985. In his forward to one of Knuth's books (Knuth, 1979)[4] Gordon Bell wrote that Don Knuth's TEX

> "...is potentially the most significant invention in typesetting this century. It introduces a standard language in computer typography and in terms of importance could rank near the introduction of the Gutenberg press."

So how did TEX come about? It appears that the motivation for TEX arose from Knuth's dissatisfaction with the early galley proofs of his book "*The art of computer programming*." Knuth describes it as follows: (see also http://www.tug.org/whatis.html).

> "... I had prepared a second edition of Volume 2, but when I received the

[4] Knuth, D.E. (1979). *TEX and* METAFONT*, new directions in typesetting.* (American Mathematical Society and Digital Press, Stanford).

> galley proofs they looked awful—because printing technology had changed drastically since the first edition had been published. The books were now done with phototypesetting, instead of hot lead Monotype machines; and (alas!) they were being done with the help of computers instead of by hand. The result was poor spacing, especially in the math, and the fonts of type were terrible by comparison with the originals. ... Then a nice thing happened. I was on a committee to revise a reading list ... and one of the things we had to do was to evaluate a book ... these galleys had been generated on a new machine in Southern California ... all based on a discrete high-resolution raster. ... I was astounded, because the resolution was so good I couldn't tell that the type was actually digital. In fact the digital type looked a lot better than what I had been getting in my own galley proofs. ... Therefore it dawned on me for the first time that I, as a computer scientist, would be able to help solve the printing problem that was worrying me so much. I didn't need to know about metallergy or optics or chemistry or anything scary like that; all I had to do was construct the right pattern of 0s and 1s and send it to a high-resolution digital typesetter like that machine in Southern California; then I'd have my books the way I wanted them. ... Therefore it was almost an *obligation* for a computer scientist like myself to study the problem carefully. ... Within a week after seeing the galleys of Winston's book, I decided to drop everything else and to work in digital typography.
>
> Knuth 1986; *Computers and typesetting*. TUGboat 7, 95–98.[5]
> http://www.tug.org/TUGboat/Articles/tb07-2/tb14knut.pdf

In the early 1980s Leslie Lamport developed a system of high-level macros called LaTeX, which used Knuth's TeX typesetting engine. In the early 1990s TeX was declared essentially bug-free and further development of TeX itself ceased. LaTeX continues to be progressively upgraded and developed, and the current version of LaTeX goes by the name LaTeX 2ε.

There is an enormous world-wide TeX community (predominantly in the foreign languages and scientific domains), which communicates via the internet newsgroups, the CTAN archives, the various TeX Users Groups which are established in most countries, and their journals e.g. *TUGboat*.[6]

In what follows I shall mostly use the word TeX as a generic term to embrace both TeX and the LaTeX macro system.

3.2 Books and information

There are plenty of good books for guidance, but you do need to be careful to distinguish between TeX itself (the low-level code known as Plain TeX), and LaTeX (the particularly useful macro libraries). I list the main contenders below.

Although one generally uses LaTeX for most things, most people actually include some (plain) TeX code as well as graphics in their documents. Consequently I would strongly recommend buying at least three books: one book on LaTeX, for example *The LaTeX companion* (Mittelback F and Goossens M 2004), one book on graphics, for example *The LaTeX graphics companion* (Goossens M, Mittelback F, Rahtz S, Roegel D and Voss H 2007), plus the main book on TeX, namely *The TeXbook* by Knuth (1990).

[5] Reprinted in: Knuth D.E. (1999). *Digital typography*. (CSLI Publications, Stanford, California.) Also available from Cambridge University Press, UK.

[6] TUGboat is the main publication of the international TeX Users Group (TUG).

Unfortunately, these books are difficult to find in the usual High Street bookshops, and are generally found only in the bigger shops or university bookshops. Of course they can be bought direct from the publisher or from the TEX Users Groups (see the TUG web site at http://www.tug.org).

LATEX

Of the books on LATEX the first six in the list below are probably the main ones to consider.

- Gratzer G (2007). *More math into LATEX*. pp. 619, (Springer) ISBN 978-0-387-32289-6.
- Mittelback F and Goossens M (2004). *The LATEX companion*. 2nd ed. pp. 1087 (Addison-Wesley) ISBN 0-201-36299-6.
- Goossens M, Mittelback F, Rahtz S, Roegel D and Voss H (2007). *The LATEX graphics companion*. 2nd ed. pp. 925 (Addison-Wesley) ISBN 0-321-50892-0.
- Goossens M and Rahtz S (1999). *The LATEX web companion*. (Addison-Wesley).
- Kopka H and Daly PW (2004). *A guide to LATEX 2ε ; document preparation for beginners and advanced users*. 4th ed. pp. 597. (Addison-Wesley).
- Syropoulos A, Tsolomitis A and Sofroniou N (2003). *Digital typography using LATEX*. pp. 510 (Springer) ISBN 0-387-95217-9.
- Diller A (1999). *LATEX line by line*. (John Wiley & Sons Ltd.).
- Lamport L (1994). *LATEX: a document preparation system—user's guide and reference manual*. (Addison-Wesley).
- Hoenig A (1998). *TEX unbound: LATEX and TEX strategies for fonts, graphics and more*. pp. 580 (Oxford University Press) ISBN 0-19-509685-1
- Walsh N (1994). *Making TEX work* (O'Rielly & Associates, Inc., Sebastopol, CA, USA).

  This book, which is now out of print, is available free on-line in the CTAN archive at
  http://tex.ac.uk/tex-archive/info/makingtexwork/mtw-1.0.1-html.tar.gz
  http://tex.ac.uk/tex-archive/info/makingtexwork/html/index.html

TEX

All these, except the two books by Knuth, are out of print. They are however all well worth acquiring and can often be found via the internet.

- Abrahams PW (1990). *TEX for the impatient*. pp. 357 (Addison Wesley) ISBN 0-201-51375-7.
- Knuth D (1990). *The TEXbook*. (Addison-Wesley).
- Knuth D (1991). *The* METAFONT*book*. (Addison-Wesley).

- Snow W (1992). *TEX for the beginner*. pp. 377 (Addison Wesley) ISBN 0-201-54799-6
- Eijkhout V (1992). *TEX by topic; a TEXnician's reference*. (Addison-Wesley Publishing Company, Wokingham, UK).

  This book, which is now out of print, is available free on-line at http://www.eijkhout.net/tbt/

Electronic information

There are many excellent on-line documents which are worth looking at as well, some of which are as follows. For other electronic sources see the TUG website (http://www.tug.org).

- PracTEX (a free online TEX journal)
- *A not so short introduction to LATEX*
  http://www.tex.ac.uk/tex-archive/info/lshort/english/lshort.pdf
  (about 900KB).
- *A gentle introduction to LATEX*, by Michael Doob.
  http://www.tex.ac.uk/tex-archive/info/gentle/gentle.pdf
- *A simplified introduction to LATEX*, by HJ Greenberg.
  http://www.tex.ac.uk/tex-archive/info/simplified-latex/
- The TEX FAQ (*Frequently Asked Questions*).
  http://www.tex.ac.uk/tex-archive/doc/TeX-FAQ.txt.

3.3 Installation

Free TEX implementations are available for practically all computer operating systems; for some operating systems there are several implementations to choose from. Although there are also many commercial implementations (e.g. PCTEX), the general feeling is that these are (a) horribly expensive and (b) significantly inferior to the free versions.

Two important electronic resources are produced each year by the TEX community, namely (a) TEXLive—produced by Sebastian Rahtz and published by Lehmans[7] (two CD-ROMs plus one DVD in 2004), and (b) a copy of the CTAN archive—produced by Dante[8] and also published by Lehmans (five CD-ROMs in 2004). These include all the open-source TEX implementations, as well as all the available archived TEX software world-wide.

Note that different commputer platforms (e.g. Linux, Micro$oft, Mac) require different TEX implementations (i.e. systems written specially for particular platforms); for example, the TEX implementation for Linux is the system called TEXLive. The implementations for the various platforms are as follows.

- Linux: (TEXLive)
- Microsoft Windows

[7] Lehmans, Berlin
[8] The German TEX User Group

  - MiKTEX: This is available for most Windoz platforms, and certainly works well on Windoz ME, 2000, NT. See also http://www.miktex.org for more information. Easily installed from TEXLive.

- MS-DOS: (emTEX)
- Macintosh: (OzTEX) Easily installed from TEXLive.

All these implementations are available on the TEXLive DVD, which is most easily obtained by joining one of the TEX users groups, e.g. TUG, or UK-TUG (see Section 3.7), since a new version of the DVD is available each year as a membership benefit.

All TEX-related files are archived on the internet at what is known as CTAN (Central TEX Archive Network) which is mirrored in most countries. The main UK CTAN is in Cambridge at http://www.tex.ac.uk/tex-archive/. While downloading via the internet is an option, remember these distributions are very large. One can also purchase disks and DVDs with all the necessary files from the German publisher Lehmans.

Linux TEX implementation

The TEX system for standard GNU-Linux is easily installed from the TEXLive DVD. Once installed you will be able to create a TEX file using a text-editor (e.g., Kwrite). After saving the file, you run LATEX on the TEX file, and view and print it using the following commands typed at the command-line in an Xterm window.

```
latex myfile.tex     %% runs latex
xdvi myfile.dvi      %% views the output
dvips myfile.dvi     %% sends the file to the printer
dvips -o myfile.ps  myfile.dvi  %% creates a PostScript version
gv myfile.ps         %% view the PostScript file using GhostView
pdflatex myfile.tex  %% generates a pdf version of the file
acroread myfile.pdf  %% reads the pdf file using Acrobat Reader
```

Once you have mastered the process of using the text-editor (Kwrite)[9] and processing the files using the above commands, you can then progress to learning how to use the extremely useful "KDE Integrated LATEX Environment" (Kile). For this reason I would suggest you install the KDE system (as opposed to the Gnome system)—at least initially—since Kile is so useful.

Kile—a front end for TeX

Although TEX and LATEX can be run and processed simply by using command-line commands, life is made a lot easier by using one of the two main so-called 'front-ends' designed for processing TeX documents, namely Emacs or Kile. I would recommend only Kile for the beginner (Emacs is really only for experts) as it is pretty simple to learn and set up. The Kile text-editor is truly excellent and has good color encoded highlighting of commands and environments which greatly facilitates writing TEX documents (see Figures 1 and 2).

[9] There are of course plenty of other text-editors you can experiment with, but in my view Kwrite is probably the best stand-alone text-editor.

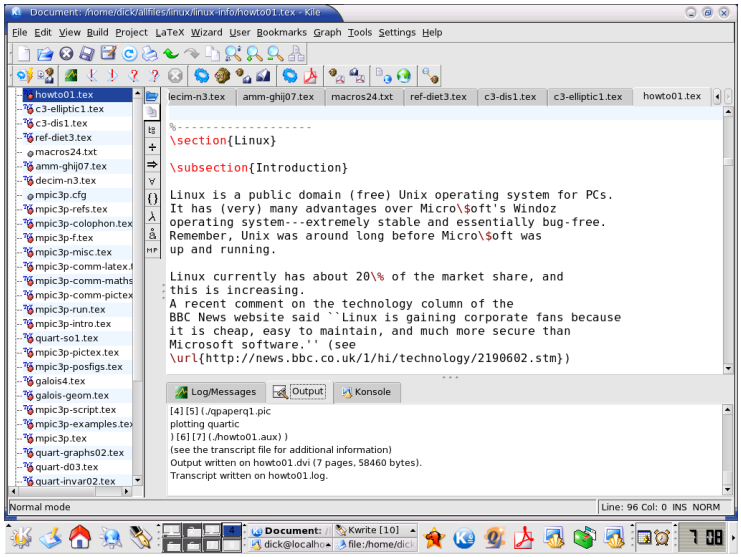

Figure 1: Screen shot showing Kile in use.The main window is the text-editing window; the lower small window gives access to the command-line and the LaTeX output log data; the narrow vertical window on the left lists currently loaded files. The current `.tex` file is run, processed and viewed by clicking on the appropriate icons above. Note the color encoding of commands (red) and their arguments (blue).

3.4 Fonts

Probably the most difficult aspect of TeX is getting to grips with fonts and font creation, especially with respect to Type-I (scalable) PostScript fonts. The following guide considers most scenarios.

- *Font installation guide (v 2.14)(2004)*, by Philipp Lehman
  `CTAN: .../tex-archive/info/Type1fonts/fontinstallationguide`

3.5 Drawing utilities

Programs which allow easy conversion to .eps format are:

```
skencil
inkscape (http://www.inkscape.org/)
dia
cenon (http://www.cenon.info/)
```

3.6 Typesetting mathematics

Once you have got to grips with TeX then you will be able to typeset your articles, books etc with ease. Technical and mathematical details are particularly well accommodated since TeX is also the world's leading maths typesetting software; for example, equations,

$$z_2, z_3 = \frac{-z_1}{2} \pm \frac{\sqrt{3}}{2}\sqrt{(2\delta)^2 - z_1^2}. \tag{1}$$

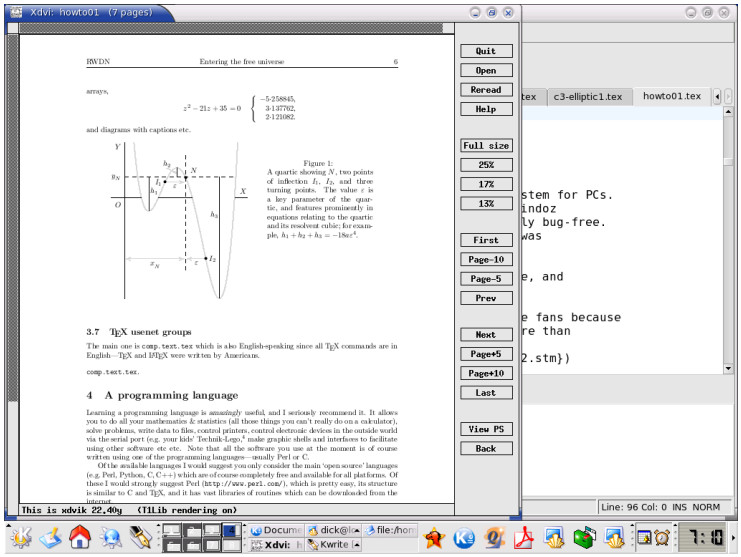

Figure 2: Screen shot showing Kile in use. This view shows the use of the `.dvi` viewer **Xdvi**, showing a page of the typeset output.

arrays,

$$z^2 - 21z + 35 = 0 \quad \begin{cases} -5{\cdot}258845, \\ 3{\cdot}137762, \\ 2{\cdot}121082. \end{cases}$$

and diagrams with captions (see Figure 3).

3.7 TEX users groups

The UK TEX Users Group (UK-TUG) http://uk.tug.org/

Joining the UK TEX Users Group (£12/year) gives you all the latest key TEX CD-ROMS & DVDS (updated each year) which includes all the CTAN files and the TEXLive DVD too.

The TEX Users Group (TUG) http://www.tug.org/

TUG is the main world-wide users group, and not surprisingly, it is USA-based. TUG membership is currently about £45, for which you get the annual set of CDs & DVDs (and TEXLive and CTAN) and also the quarterly journal TUGboat. The TUG website is a good place to start when looking for TEX-related information.

TEX usenet groups `comp.text.tex`

There are several usenet groups covering a variety of languages. This one is the main one and is also English-speaking since all TEX commands are in English—both TEX and LATEX were written by Americans.

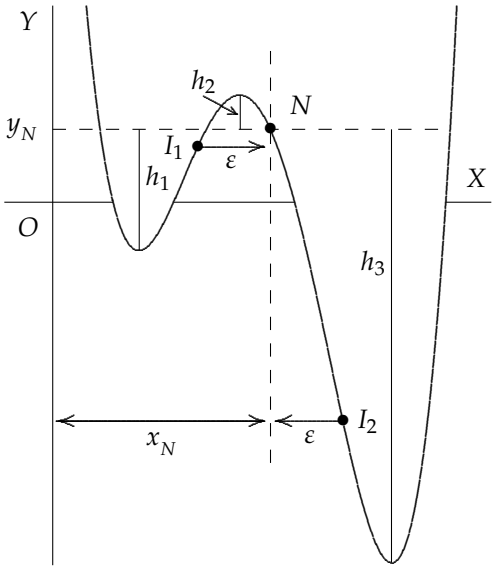

Figure 3: A quartic showing N, two points of inflection I_1, I_2, and three turning points. The value ε is a key parameter of the quartic, and features prominently in equations relating to the quartic and its resolvent cubic.

4 Perl

4.1 Introduction

Perl (Practical Extraction and Report Language) is a general purpose programming language with excellent mathematical and scientific capabilities. It was developed by Larry Wall in the early 1990s. The current stable release version is 5.8.4 (Mandriva-Linux 10 (2004) ships with version 5.8.4). However, work is currently underway with a new rewrite which will form the basis of Perl 6.

Perl is available for all computer platforms, and can be freely downloaded from Perl web-sites. Importantky, it comes as standard with all Linux implementations. Indeed the Perl Institute (http://www.perl.org) works towards *'keeping Perl available, usable, and free for all'*. The Perl language page is at http://www.perl.com/perl/.

A huge library of add-on modules for Perl is available from the internet archive known as CPAN (Comprehensive Perl Archive Network) at http://www.cpan.org. The Perl FAQ is at http://www.perl.com/perl/faq/, and a quarterly (commercial) Perl journal is available from http://orwant.www.media.mit.edu/the_perl_journal/. Useful texts on Perl are as follows (note these books are updated quite frequently).

- Schwartz RL, Foy BD and Phoenix T (2006). *Intermediate Perl*. 2nd ed. (O'Reilly).
- Cozens S (2005). *Advanced Perl programming*. 2nd ed. (O'Reilly).
- Conway D (2005). *Perl best practices*. (O'Reilly).

- Conway D and Poe C (2006). *Perl hacks*. (O'Reilly).
- Christiansen T and Torkington N (1998). *Perl cookbook*. (O'Reilly and Associates Inc., Sebastopol, CA, USA).
- Wall L, Christiansen T and Orwant J (2000). *Programming Perl*. 3rd ed (O'Reilly and Associates Inc., Sebastopol, CA, USA).
- Lidie S and Walsh N (2002). *Mastering Perl/Tk*. (O'Reilly and Associates Inc., Sebastopol, CA, USA).
- Holzner S (1999). *Perl: core language*. (Coriolis Technology Press, Arizona, USA).
- Feiler J (2000). *Perl 5 programmer's notebook*. (Prentice Hall PTR, New Jersey, USA)
- Patwardhan N, Siever E and Spainhour S (2002). *Perl in a nutshell: a desktop quick reference*. 2nd ed (O'Reilly and Associates Inc., Sebastopol, CA, USA). pp 737.

Useful electronic resources are as follows:

- The Perl Foundation: http://www.perlfoundation.org/
- The Perl Institute http://www.perl.org/
- Perl documentation (Plain Old Documentation) http://perlpod.com/5.8.4/
- Online perl library http://learn.perl.org/library/
- Comprehensive Perl Archive Network (CPAN): useful for downloading all Perl software, modules, etc. http://www.cpan.org/ and also http://cpan.perl.org.

4.2 Modules

Standard computer Perl systems come with a large number of standard library modules which usually need to be invoked towards the beginning of a program if their routines are required. The following is a list of the standard maths modules for Perl 5.005 at the time of writing (note that Linux file-names are case sensitive).

```
Math::BigFloat    (Supports large floating numbers)
Math::BigInt      (Supports arbitrary length integers)
Math::BigInt::Calc   (Supports big integer calculations)
Math::BigRat      (Supports big rationals)
Math::Complex     (Supports complex numbers)
Math::Trig        (Supports extra trignometrical functions)
```

Useful information on the maths Perl modules can be found at http://www.mit.edu/afs/sipb/project/seven/info/pm.info-242/ Other special modules can be downloaded from the internet as required.

One particularly useful downloadable tool designed for solving polynomials is the Perl module `Math::Polynomial::Solve` (see below). I shall therefore use this as an example.

Math::Polynomial::Solve module

This is a Perl module for solving univariate polynomials of degree 1–5 written by Gamble (2001). The URLs for this module (version 2.00) and its documentation are as follows.

```
   www.cpan.org/modules/by-module/Math/JGAMBLE
/Math-Polynomial-Solve-2.00.readme
   http://www.cpan.org/modules/by-module/Math/JGAMBLE
/Math-Polynomial-Solve-2.00.tar.gz
```

The documentation can also be found on the useful http://search.cpan.org website as follows.

```
http://search.cpan.org/~jgamble/Math-Polynomial-Solve-2.00/Solve.pm
```

Note that since equations can have both real and complex values this module requires the standard `Math::Complex` module to be loaded as well.

Installing the module

On the Linux platform the `.tar.gz` file is unzipped, untared, and then installed using the following commands.

```
gunzip Math-Polynomial-Solve-1.00.tar.gz
tar -xvf Math-Polynomial-Solve-1.00.tar
perl Makefile.PL
make
make install  #do as Root
make test
```

Usage

Once installed the package is implemented using the Perl command

```
use Math::Complex;
use Math::Polynomial::Solve;
```

Note that Perl translates the `::` in the module package name into `/` and so forces Perl to load the module file called `Solve.pm` located in the directory tree `.../Math/Polynomial/Solve.pm`. The module supplies five functions for solving polynomials as follows:

```
linear_roots()
quadratic_roots()
cubic_roots()
quartic_roots()
poly_roots()
```

If you wish to use only some of the available functions, say, `cubic_roots()` and `quartic_roots()`, then these need to be listed in the `use` command as follows.

```
use Math::Polynomial::Solve qw(cubic_roots quartic_roots);
```

The function `poly_roots()` will actually solve all equations of degree 1–4, and acts accordingly depending on how many coefficients are given as arguments. For example, the command

```
@x=poly_roots(1,2,3,4,5)
```

will solve for a quartic since five coefficients are supplied as arguments.

Example Perl program

In the following example program (`cubicprog.pl`) the function `cubic_roots()` is used to solve the cubic equation $x^3 + 3x^2 + 9x - 13 = 0$. Note that in Perl all active lines must be terminated with a semicolon; the # symbol is the Perl 'comment' character.

```
1  #!/usr/bin/perl
2  # cubicprog.pl
3  use Maths::Complex;
4  use Maths::Polynomial::Solve;
5  # solve the cubic  x^3 + 3x^2 + 9x -13 = 0
6  my ($a,$b,$c,$d) = 1,3,9,-13;
7  my ($x1,$x2,$x3)=cubic_roots($a,$b,$c,$d);
8  print "the coefficients are $a, $b, $c, $d\n";
9  print "x1 = $x1\n";
10 print "x2 = $x2\n";
11 print "x3 = $x3\n";
```

The details of the program are as follows:- first we install the Perl modules for using complex arithmetic (line 3) and for solving polynomials (line 4). Next, we place the equation's coefficients $(1, 3, 9, -13)$ into the variables `$a, $b, $c, $d` (line 6). Finally, we call the `cubic_roots()` function[10] in such a way that the three roots are placed into the array `x[]` (line 7), and, lastly, we print out the coefficients as a check (line 8) and then the three roots `x1, x2, x3` (lines 9–11).

The program `cubicprog.pl` is run using the command

```
perl cubicprog.pl
```

and gives the following result.

```
x1 = 1
x2 = -2 + 3i
x3 = -2 - 3i
```

5 Python

Python (currently at version 2.2) is an Open Source general purpose programming language similar in many regards to Perl, and also has excellent mathematical and scientific capabilities. However, a slightly irksome 'feature' of Python is that the relative indentation of certain key words (e.g. `for`, `else` etc) is an integral part of the syntax, such that error messages are generated unless position is also correct.

[10] from the module `Maths::Polynomial::Solve`

Python is available for most computer platforms, and can be freely downloaded from Python web-sites. It comes as standard with all Linux implementations. An extensive library of add-on modules for Python is available from the internet archive at http://www.python.org/. The Python FAQ is at http://www.deitel.com/. A useful text on Python is

- Deitel HM, Deitel PJ, Liperi JP and Wiedermannn BA (2002). Python: how to program. (Prentice Hall, New Jersey, USA).

Example program

The following simple Python program defines and then uses a subroutine `quadratic()` to return the roots of a quadratic. The program 'imports' the `sqrt()` function from the `math` module (line 4). Note that 2.x versions of Python use a new module called `__future__` (see line 5) to enable floating-point division to be implemented using the usual / symbol (earlier versions of Python required the // symbol for this). For Python 3.0 onwards this and some other changes will be fixed—see http://python.sourceforge.net/peps/pep-0238.html for information on these future changes (Deitel *et al.* 2002, p 46-47).

```
1  #!/usr/bin/python
2  # for python 2.2
3  # testprogram.py
4  from math import sqrt
5  from __future__ import division
6  #---------------------
7  def quadratic (a,b,c):
8      a1=float(a)
9      b1=float(b)
10     c1=float(c)
11     if b==0: xn=0
12     else: xn= -b/(2*a)
13
14     yn1 = a*xn**2 + b*xn + c
15
16     if yn/a == 0 :
17       rootmode = "real";
18       r1 = xn
19       r2 = xn
20     if yn/a > 0 :
21       rootmode = "complex";  r=sqrt(yn/a)
22       r1 = xn - r
23       r2 = xn + r
24     else:
25       rootmode = "real"; r=sqrt(-yn/a)
26       r1 = xn -r
27       r2 = xn + r
28     return (rootmode, r1, r2)
29 #-------------
30 print "the roots are ", quadratic(1,0,-4)
```

To run the Python program `testprogram.py` simply type the following command at the command-line

```
python  testprogram.py
```

6 Gnuplot

http://www.gnuplot.info/
This is one of the main all-purpose graphing tools.

MS-Windows 32-bit version

ftp://ftp.gnuplot.info/pub/gnuplot/gp371w32.zip (this is version 3.7 patch-level 1; runs nicely on Win2K).

Gnuplot usenet group

The main English-speaking one is

- `comp.graphics.apps.gnuplot`.