Scientific Computing with Free software on GNU/Linux HOWTO

Manoj Warrier

<m_war (at) users.sourceforge.net>

Shishir Deshpande

<shishir (at) ipr.res.in>

V. S. Ashoka

<ashok (at) rri.res.in>

2003−10−03

Revision History

Revision 1.2	2004−10−19	Revised by: M. W
1 Correction and new additional links		
Revision 1.1	2004−06−21	Revised by: M. W
Updates and evaluated distros		
Revision 1.0	2003−11−18	Revised by: JP
Document Reviewed by LDP.		
Revision 0.0	2003−10−01	Revised by: M. W
first draft proposed		

This document aims to show how a PC running GNU/Linux can be used for scientific computing. It lists the various available free software and also links on the world wide web to tutorials on getting started with the tools.

Table of Contents

1. Preamble....1
- 1.1. Copyright and License....1
- 1.2. Disclaimer....1
- 1.3. Motivation....1
- 1.4. Credits / Contributors....1
- 1.5. Feedback....2
- 1.6. Translations....2

2. Introduction....3

3. Code Development Tools....4
- 3.1. Programming Languages....4
- 3.2. Debugging Tools....5
- 3.3. Version Control Tools....5
- 3.4. Integrated Development Environments....6

4. Mathematics Packages....7

5. Numerical Methods and Libraries....9
- 5.1. Repositories....9
- 5.2. Other topic specific numerical libraries....9

6. Graphics and Visualization....11

7. Programming systems for GNU/Linux....13
- 7.1. The GNU/Linux Workstation....13
- 7.2. Parallel Processing and Symmetric Multiprocessing: Supercomputing....13
  - 7.2.1. Parallel computing document links....13
  - 7.2.2. Parallel processing software for Linux....14

8. Word−Processing and Poster presenting tools on Linux....15
- 8.1. Word Processing Tools....15
- 8.2. Poster presentation tools....15

9. Free Database Management Systems for Linux....16

10. Linux in the laboratory....17

1. Preamble

1.1. Copyright and License

1.2. Disclaimer

No liability for the contents of this document is accepted. Use of the concepts, examples, links and information is entirely at your own risk. There may be errors and inaccuracies, that could damage your system, waste your time, etc... Proceed with caution, and although this is unlikely, the author takes no responsibility whatsoever.

All copyrights are held by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. Naming of particular products, software or brands should not be seen as endorsements

I have not used many of the software applications to which links are provided. There are simply too many applications that do the same thing, that one cannot be expected to have used all of them. In a book on Scientific Computing using GNU/Linux, one would mention ones favorite tool to carry out a task and describe it in detail. However this is a howto providing links to various available free tools for scientific computing and may contain links to some software that promises much but delivers little and vice versa.

1.3. Motivation

This howto mainly consists of the links provided at http://Scilinux.sourceforge.net which has to be disbanded due to a name conflict. The best alternative seems to be to make it a Linux document and host it at the LDP site. Another reason is that there seems to be many free software applications doing the same things. We hope to provide links to the available software thereby making it easy for the scientific community to make a choice without spending much time.

1.4. Credits / Contributors

In this document, I have the pleasure of acknowledging:

- Linus Trovalds, Richard M. Stallman and their merry men for Linux, GNU and also for indirectly broadening various perspectives which were not really obvious.
- A host of colleagues and friends from the *Institute for Plasma Research, India* for discussions at various times.
- Marcel Bose, Ivan Lamouret, K. Scott Hunziker, Livine Christin, W. Herbert, Simon Pinches and many others for suggesting various links mentioned in this document.
- Vasudha my wife for letting me do what I wish and egging me on with comments like "let us hope that you will finish at least this project"

Shishir and Ashoka are co−authors of this document because such a collection of links was Shishir's idea and Ashoka is always contributing by providing links, suggestions and a second point of view. They will be helping me maintain this HOWTO too.

1.5. Feedback

Feedback is most certainly welcome for this document. Send your additions, comments and criticisms to the following email address : <m_war at users.sourceforge.net>.

1.6. Translations

No translations yet.

2. Introduction

GNU/Linux is probably the platform of choice for scientific computing. There exists a wide variety of high level languages, debugging tools and other code development tools for programming, numerical subroutines for solving various types of equations, plotting and visualization packages, word processing software which can display equations and figures and in fact parallel programming software to construct a supercomputer with off the shelf PC parts and some hardware. This document aims to provide a list of *free software* for carrying out the above tasks and links to tutorials and other documents on how to setup and use these software applications.

This document does not aim to provide links to subject specific free software available for GNU/Linux systems. It aims to show how GNU/Linux can be used best to handle scientific computing tasks. It is hoped that people or institutions with interest in a specific subject list, compile a list of the free software available for that subject ... for example see *Linux for Astronomy, Linux for Biotechnology* and *Linux for Chemistry* at The Random Factory . Another site with a lot of links (to commercial and free) scientific software is Scientific Applications on Linux. The GNU Software Directory also has links to many of the links provided in this howto plus many more topic specific software. You may also want to check out The Science and Engineering section at Freshmeat.net.

The software links provided are classified into

- Code development tools
- Mathematics packages
- Numerical subroutines and libraries
- Graphics and visualization
- GNU/Linux Systems
- Publishing tools
- Databases
- Linux in the Laboratory

Just installing GNU/Linux on your PC makes it a powerful workstation. The various popular distributions however do not have all the tools needed to make it the ideal scientific computing machine. This HOWTO aims to fill in this gap by creating a list of free software useful for scientific computing. It is assumed that people reading this document already have a PC with Linux and the GNU utilities installed. For those who do not have such a setup and want to install GNU–Linux can check out GNU/Linux Systems for links to documents on installing GNU/Linux, and also on how to get started using GNU/Linux. Recently there has been an effort by Dirk Eddelbuettel to create a scientific computing environment Quantian which probably is the first GNU–Linux distribution tailored for Scientists. I checked out the latest release and it has almost all the packages mentioned in this document and many packages not mentioned. It is fair to say that if you have any linux distribution in which the packages are managed by rpms or any debian based system, you will find pre–compiled binaries of these packages and will not have to waste much time installing them.

---

3. Code Development Tools

Code development consists of mainly Programming languages, Debugging tools, Version Management tools, Compiling tools, and Integrated Development Environments (IDEs) where all the above are coupled as a single software application.

3.1. Programming Languages

Links are provided to various compilers used in Scientific Computing like FORTRAN, C, C++, Java and more recently Python.

- GNU Compiler Collection : GNU's project to produce a world class optimizing compiler. It works on multiple architectures and diverse environments. Currently GCC contains front ends for C, C++, Objective C, GNU Fortran−95, Java, and Ada, as well as libraries for these languages (libstdc++, libgcj,..).

  For manuals on using the various GCC compilers check out The GCC online documentation
- g77 : The GCC front end for FORTRAN 77. It is a very good FORTRAN77 compiler. It however does not have the −r8 option which compiles a program as double precision. This could be a good compiler design philosophy but in many cases gives problems when porting a code from SUN / DEC / HP workstations onto Linux systems. The g77 manual is available at The Gcc Online documentation site.
- gfortran. I was happy to receive this link by mail. It was 3 years since I had migrated to using the GNU C compiler for scientific computing because there was no "truly free" FORTRAN−95 compiler available then. I thank Paul Thomas for this link.
- g95. gfortran above and g95 are reportedly offshoots from the same CVS tree. Has an impressive list of programs that compiles and runs using this compiler.
- fort77 and f2c: fort77 is a perl program which invokes the f2c command (a Fortran to C translator) transparently, so it can be used just like a real Fortran compiler. Fort77 can be used to compile Fortran, C and assembler code and can link the code with f2c libraries. If you install fort77, you'll also need to install the f2c package. This does not have the "−r8" problem. You can download fort77 and f2c from the above link.
- lush: An object−oriented programming language, which combines the flexibility of an interpretive language, with the efficiency of a compiled language. It has full interfaces to numerical libraries (GSL, LAPACK, BLAS), graphics libraries (OpenGL), which allows creation of graphics and 3D animations and many other features that sound too good to be true. I have not yet tried this out, but it sounds very promising.
- Scientific Python: You may want to explore Python for your scientific computing needs. Python is an interpreted, interactive, object−oriented programming language. It has a number of extensions for numerics, plotting, data storage and combined with Tk lets you develop very good GUIs for your codes. The most exciting aspect is that it simplifies programming because it has modules for almost anything (vectors, tensors, transformations, derivatives, linear algebra, Fourier transforms, statistics, etc ...) are available. You can also wrap C and Fortran libraries from Python. Finally if you want to write a numerical scheme of your own you may find that it is simpler in Python. There are also interfaces to netCDF (portable binary files), MPI and BSPlib (parallel programming).

  You can further explore Python for Scientific computing here:

  - Scientific−Python: A collection of modules for scientific computing on Python. All the

necessary modules can be downloaded as either a tar file or an RPM file from here. The maintainer Konrad HINSEN also has a nice tutorial on Scientific Computing in Python.
  - SciPy An open source library of scientific tools for Python. It includes modules for graphics and plotting, optimization, integration, special functions, signal and image processing, genetic algorithms, ODE solvers, etc.

3.2. Debugging Tools

In this section links are given to mainly debugging tools for GCC and FORTRAN. I understand that python has a debugging module built in though I have not used it. The purpose of a debugger is to allow you to see what is going on inside a program while it executes or what the program was doing when/if it crashed.

- Ftnchek: A FORTRAN checker designed to detect errors in a Fortran program that a compiler usually does not. Therefore it is best to run ftnchek on your FORTRAN programs after it has compiled without errors. Its purpose is to assist the user in finding semantic errors. Semantic errors are legal in the Fortran language but are wasteful or may cause incorrect operation. An on−line manual is available. This project is looking for volunteers to bringing ftnchek up to the Fortran 90 standard.
- gdb : All programs written in the languages supported by GCC can be debugged using *gdb*, an excellent interactive, command line debugger. You can compile your programs using a −g option which then compiles your code with debugging information inserted into the executable. It can start your programs, stop your programs on specified conditions and at specified locations, examine what happened when your program stops. In a large code with multiple cascading calls to various functions it can back trace the function calls. You can also Download the document *Debugging with GDB* and a quick reference card.
- xxgdb: It is a front end to the gdb debugger. Useful for beginners to gdb as it lists out the whole gdb commands as buttons with a area for viewing source on which one can include break points, etc by a click of the mouse, and another area for viewing the debugging results.
- DDD: The GNU Data Display Debugger, GNU DDD, is a graphical front−end for command−line debuggers such as GDB, DBX, WDB, Ladebug, JDB, XDB, the Perl debugger, or the Python debugger. Besides ``usual'' front−end features such as viewing source texts it also has a good interactive graphical data display, where data structures are displayed as graphs. Follow this link for a DDD manual in postscript / HTML / PDF format.

3.3. Version Control Tools

It will be worth your while investing some time in learning to use one of the version control tools below (cvs is what I use ..) if you are into any serious code development.

- Concurrent Versions System : CVS is one of the most popular version control systems running on the Linux operating system. Popular Linux projects like Apache, EGCS, GIMP, and others are using CVS to coordinate their efforts ... This is how the URL linked above describes their effort.

  A tutorial on CVS is available at Gentoo Linux Documentation and a free CVS book is available here
- Project Revision Control System : PRCS, the Project Revision Control System, is the front end to a set of tools that (like CVS) provide a way to deal with sets of files and directories as an entity, preserving coherent versions of the entire set. PRCS was designed primarily by Paul N. Hilfinger, with input and modifications by Luigi Semenzato and Josh MacDonald. PRCS is written and maintained by Josh MacDonald. Its purpose is similar to that of SCCS, RCS, and CVS, but (according to its authors, at least), it is much simpler than any of those systems. This page is where information

on the latest developments in PRCS can be found.
- Gbuild : gbuild is a script written in the Bourne shell language to simplify package maintenance by allowing you to automate code update from CVS, compilation, building tar/rpms/srpms of your package. some external scripts which certain functions of gbuild depend on are written in Perl. gbuild is released under the GPL.

3.4. Integrated Development Environments

Integrated development environments (IDEs) can be very useful for building code and ideally come with all the above tools (i.e a compiler, a debugger and a version control tool). In addition to that IDEs also usually provide a makefile generator, documenting help, online help manuals, etc.

- Kdeveloper : A easy to use C/C++ IDE (Integrated Development Environment) for Linux. It supports KDE/Qt, GNOME, plain C and C++ projects. This site has a lot of documentation ..... a highly browsable site for software developers. Specifically, KDevelop manages or provides:

  All development tools needed for C++ programming like Compiler, Linker, automake and autoconf; KAppWizard, which generates complete, ready−to−go sample applications; Class generator, for creating new classes and integrating them into the current project; File management for sources, headers, documentation etc. to be included in the project; The creation of User−Handbooks written with SGML and the automatic generation of HTML−output with the KDE look and feel; Automatic HTML−based API−documentation for your project's classes with cross−references to the used libraries; Internationalization support for your application, allowing translators to easily add their target language to a project; KDevelop also includes WYSIWYG (What you see is what you get)−creation of user interfaces with a built−in dialog editor; Debugging your application by integrating KDbg; Editing of project−specific pixmaps with KIconEdit; The inclusion of any other program you need for development by adding it to the "Tools"−menu according to your individual needs.
- VDKbuilder: VDKbuilder is a tool that helps programmers in constructing GUI interfaces, editing, compiling, linking, and debugging within an integrated environment. Using VDKBuilder dramatically reduces developing time since all code related to GUI construction and signal processing is automatically generated, maintained and updated. It is distributed under the GNU Public License. Visit the site for downloading the software.

4. Mathematics Packages

All the links below are free high level languages and Mathematics Packages for Scientific Computation on Linux. These packages are usually like a Mathematical Laboratory in which numerical computations can be done and usually have their own interpreted language. They either link to a popular (free) plotting package or have their own graphics and plotting capability. They also provide capability to I/O files and interface with other programming languages like C, C++, Fortran, etc ... Now a days some of them have parallel programming capabilities. I have not included MuPAD, a good symbolic math package, since is not really free. Check out if their most free license suits you.

- Octave: An excellent package for numerical computations. It uses gnuplot for plotting and has a online help. It is also easily extensible (i.e. new functions, procedures can be written) either using its own language or by using dynamically loadable modules written in C, C++, Fortran or other languages. An extensive manual is available here. You can get a GNOME based front end for it here. It is distributed under the GNU Public License.
- Scilab: Another superb package numerical computations having a good user interface and a very good online click−able help. Its plotting and graphic capabilities are also very impressive. It also provides for easy interfacing with Fortran and C. It has its own free license.
- Yorick: Yorick is a fast, interpreted language, designed for scientific computing and numerical analysis. The syntax is similar to C, but the variables need not be declared. It offers an interactive graphics package based on X windows. X−Y plots, quadrilateral meshes, filled meshes, cell arrays, and contours are supported. You can embed compiled routines in Yorick to solve problems for which the interpreter is too slow. It is also useful as a pre and post processor for large physical simulation programs. A tutorial like manual is available here. Yorick is open source software, copyright of the Regents of the University of California.
- Algae: As the above link describes it, Algae is a interpreted language for numerical analysis. It was developed as a fast and versatile tool, capable of handling large problems. Algae consists of the programming language Algae, and algae, the interpreter. Its features include speed (generally much faster than octave, RLaB and Scilab), storage of sparse arrays and a code profiling capability (to check where your code spends its time). A user manual is available here. It is distributed under the GNU General Public License.
- YACAS: As the above link describes it, "YACAS is an easy to use, general purpose Computer Algebra System, a program for symbolic manipulation of mathematical expressions. It uses its own programming language designed for symbolic as well as arbitrary−precision numerical computations". Links to documentation (user manual, tutorial, etc ..) is available here. It is distributed under the GNU General Public License.
- RLAB: The above link describes it thus, "Rlab is an interactive, interpreted scientific programming environment. Rlab is a very high level language intended to provide fast prototyping and program development, as well as easy data−visualization, and processing". It is distributed under the GNU General Public License. The author Ian Searle has written an article in The Linux Journal titled An Introduction to Rlab which as he reminds us, is a bit dated, and a Reference Manual is also available.
- Maxima: Maxima is a symbolic computation program. The above link describes it as follows, "Maxima is a descendant of DOE Macsyma, which had its origins in the late 1960s at MIT. It is the only system based on that effort still publicly available and with an active user community, thanks to its open source nature. Macsyma was the first of a new breed of computer algebra systems, leading the way for programs such as Maple and Mathematica. This particular variant of Macsyma was maintained by William Schelter from 1982 until he passed away in 2001. In 1998 he obtained permission to release the source code under GPL".
- The R−Project for Statistical Computing: R is a language and environment for statistical computing and graphics. It provides a large collection of tools for statistical analysis of large arrays of data and

also graphical facilities. R is also a complete effective programming language. For computationally intensive tasks, C, C++ and Fortran code can be linked and called at run time. A comprehensive set of manuals dealing with installation, introduction, writing extensions, etc ... is available here. It is distributed under the GNU General Public License.

- gTybalt: gTybalt is a step towards a free computer algebra system. It is object oriented, allowing symbolic calculations within C++. It is efficient, in the sense that solutions developed with gTybalt can be compiled with a C++ compiler and executed independently of gTybalt. The mathematical formulae are visualized using TeX fonts and can easily be converted to LaTeX. I did not realize that it has good graphic capabilities till I checked out the gTybalt manual. It is distributed under the GNU General Public License.
- JACAL: As the link above describes it, " JACAL is an interactive symbolic mathematics program. JACAL can manipulate and simplify equations, scalars, vectors, and matrices of single and multiple valued algebraic expressions containing numbers, variables, radicals, and algebraic differential, and holonomic functions".
- bc: bc is an arbitrary precision numeric processing language. It supports interactive execution of statements. Click here for a Manual in a variety of formats. It is GNU software and is distributed under the GNU General Public License.

---

5. Numerical Methods and Libraries

The best thing that could happen for scientific computing with free software on GNU/Linux is the GNU Scientific Library GSL. It however has source code only in C and people who use FORTRAN will find that a let down. Pouncing on this opportunity it is recommended that GSL is another reason (in addition to the GCC C compiler, coupled with the advantages of C programming) for starting to learn to use C. In addition to this, the two best source code repositories for Numerical Methods and libraries are Netlib and GAMS. There are new numerical packages being developed outside the usual "write a FORTRAN program, get a numerical subroutine from INTERNET for solving the numerics" concepts. The merits and demerits of this approach are debatable, but there exist more options like Object Oriented Numerics GSL and GiNaC which are exciting developments.

5.1. Repositories

- Netlib: An amazing amount of free source code for Numerical Methods. Netlib is THE source code repository which contains an innumerable amount of source code for Numerical Methods. It also has an active discussion forum wherein you can submit your queries and stay posted for help. Netlib also has a Parallel Tools Library and a search by subject.
- GAMS: Guide to Available Mathematical Software GAMS has a very useful search using which one can search for keywords (example: ``diffusion'' to search for a diffusion equation solver). However the browse by package at GAMS reveals that a lot of the software they provide is a link to the netlib repository.
- Object Oriented Numerics A site devoted to object oriented numerics. It has a Mailing list, Extensive Links to freely available libraries (OO of course) and freely available tools for object oriented scientific computing.
- GNU Scientific Library The GNU Scientific Library (GSL) is a collection of numerical routines written from scratch in C. It provides an Applications Programming Interface (API) for C programmers and also allows wrappers to be written for very high level languages. It covers a wide range of numerical computing topics, has a good manual, is widely portable and is distributed under the GNU General Public License.
- GiNaC GiNaC is designed to allow the creation of software which need symbolic manipulations embedded in them. It extends C++ by a set of algebraic capabilities and is recursively named for GiNaC is not a Computer Algebra system. It is distributed under the terms and conditions of the GNU general public license (GPL).

5.2. Other topic specific numerical libraries

- FFTW FFTW is a collection of fast C routines for computing the Discrete Fourier Transform in one or more dimensions. It includes complex, real, and parallel transforms, and can handle arbitrary array sizes efficiently. This package includes both the double− and single−precision FFTW uniprocessors and the threads libraries.
- LAPACK LAPACK (Linear Algebra PACKage) is a standard library for numerical linear algebra. LAPACK provides routines for solving systems of simultaneous linear equations, least−squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. LAPACK is coded in Fortran77 and is built with egcs. It is well documented and widely used (and therefore widely tested).
- SuperLU SuperLU is a general purpose library which performs an LU decomposition for the direct solution of large, sparse, non−symmetric systems of linear equations on high performance machines.

Its written in C and is callable from either C or Fortran.

- ARPACK ARPACK is a set of Fortran77 subroutines designed to solve large scale eigenvalue problems. A Users Guide is available. The above link also gives information about a parallel version of ARPACK − PARPACK and a object oriented version ARPACK++.
- Computational Fluid Dynamics codes This link contains a comprehensive listing of public domain, shareware and freeware Computational Fluid Dynamics codes links with a description of each CFD code.

---

6. Graphics and Visualization

- Gnuplot Gnuplot is a command−line driven interactive function plotting utility. It handles both curves (2 dimensions) and surfaces (3 dimensions). Surfaces can be floating in the 3−d coordinate space, or as a contour plot. For 2−d plots, there are also many plot styles, including lines, points, lines with points, error bars, and impulses. Graphs may be labeled with arbitrary labels and arrows, axes labels, a title, date and time, and a key. It has multiple plotting capabilities too. It allows saving the graphs in various formats which can be included in word processors. It can be used to generate publication quality plots.
- NCAR Graphics A very popular graphics package which is very well documented and widely used. It provides basic ingredients for creating complex plots as functions / routines that can be called from Fortran and C. There is a contributed programming interface to the NCAR Graphics package: NCL (NCAR Command Language). The programming interfaces provide access to complex graphics utilities like contouring, world map projections, and velocity vectors. For the most part, the C interface is built on top of the Fortran interface... It is distributed under the GNU public license. Click here for going to the documentation of all its various components.
- OpenDX A very good Open Source Data eXplorer. It can handle large amounts of data and creates great visualizations. It was the tool I stumbled upon when I wanted a free graphics routine to make 3−D plots and zoom−in, rotate, and really eXplore the output Data from my codes. The downside is that compiling from source is really challenging and getting started is a difficult. However it has excellent documentation distributed with it and once I started off it was the best tool I have ever used.
- Gri: It is a language for scientific graphics programming. The claim that Gri is similar to LaTeX in the sense that both provide extensive power as a reward for tolerating a learning curve seems exciting and I for one want to check this out!! Check out the following article in The Linux Journal. Go to the gri home page if you are now impressed by it and check out download info and manuals.
- MayaVi: A scientific data visualizer written in Python. It is distributed under the BSD license. The screenshots look promising. Check out the above link for more details.
- PGPLOT: PGPLOT is a Fortran 77 or C callable subroutine package for drawing scientific 2D and Simple 3D plots. One can call these routines during runtime and redirect the output to a variety of devices at run time. It is well documented and the full documentation is available at the above site. It is Free for Non−Commercial Use. A user manual is available online at PGPLOT Users Manual
- PLplot: This is a library of scientific plotting functions that can be called from C, C++, FORTRAN, TCL, PYTHON. PLplot features as described in the above link are, "It can be used to create standard x−y plots, semilog plots, log−log plots, contour plots, 3D plots, mesh plots, bar charts and pie charts. Multiple graphs (of the same or different sizes) may be placed on a single page with multiple lines in each graph. There are almost 2000 characters in the extended character set. This includes four different fonts, the Greek alphabet and a host of mathematical, musical, and other symbols. A variety of output devices are supported and new devices can be easily added by writing a small number of device dependent routines". To download click here .
- Grace Grace is a WYSIWYG 2D plotting tool for the X Window System and Motif. Grace runs on practically any version of Unix. Grace is a descendant of ACE/gr, also known as Xmgr. It is lisenced under the GNU public license. This link also has a tutorial and download information.
- SciGraphica SciGraphica is a application for data analysis and technical graphics. It fully supplies plotting features for 2D, 3D and polar charts. The aim is to obtain a fully−featured, cross−platform, user−friendly, self−growing scientific application. It is free and open−source, released under the GPL license.
- Plotutils: The GNU plotutils package contains software for both programmers and technical users. Its centerpiece is libplot.a powerful C/C++ function library for exporting 2−D vector graphics in many file formats, both vector and raster. It can also do vector graphics animations. Besides libplot, the package contains command−line programs for plotting scientific data. Many of them use libplot to

export graphics.

- DISLIN DISLIN is a high−level plotting library for displaying data as curves, polar plots, bar graphs, pie charts, 3D−color plots, surfaces, contours and maps.
- ImLib3D ImLib3D is an open source C++ library for 3D (volumetric) image processing. It contains most basic image processing algorithms, and some more sophisticated ones. It comes with an optional viewer that features multi−planar views, animations, vector field views and 3D (OpenGL) multi−planar.
- Ptplot: Ptplot is a 2D data plotter and histogram tool implemented in Java. Ptplot can be used as a standalone applet or application, or it can be embedded in your own applet or application.

7. Programming systems for GNU/Linux

This section deals with links to tutorials and documents for installing Linux on a PC, getting started with Linux, and then going a step further –– to optimize your PC for processing power, using multiple processors (Symmetric Muliti Processing – SMP); making a cheap, upgradeable Supercomputing Linux cluster and finally links to software to do parallel programming on Linux.

7.1. The GNU/Linux Workstation

As with most documentation related to GNU/Linux, the Linux Documentation project's home page is a priceless source. You might first want to read The Linux Installation HOWTO. For those who want to install Linux along with Windows might want to browse through The Linux + Windows HOWTO. When installing Linux make sure that you choose to install all documentation. After installing Linux, a good, comprehensive document to getting started with using Linux is The Rute Users Tutorial and Exposition which is a beginners guide to Linux and UNIX like systems. I'd like to give a less intimidating (size–wise) link to a small beginners guide, but U will find this useful after taking the plunge. You might also want to go through The Linux System Administrator's Guide and to check out The Linux Administration Made Easy (LAME) guide It attempts to describe day–to–day administration and maintenance issues commonly faced by Linux system administrators.

7.2. Parallel Processing and Symmetric Multiprocessing: Supercomputing

It is possible to get large volume number crunching without spending millions of rupees on a supercomputer. You only need to link together (by some high speed network) the requisite number of CPUs, with GNU/LINUX as the underlying OS. Add some freely available message passing software and a effective parallel processing number crunching machine is made. Such clusters are called "Beowulf clusters". The other advantages of such a cluster other than building costs is, up–gradation costs are minimal. The two best resources for Linux cluster builders are

- The Beowulf Project home page and
- The Extreme Linux Project

These sites are upgraded frequently with useful information for cluster builders.

7.2.1. Parallel computing document links

You will also want to read this excellent article on Linux Clustering Software (and the large variety of links it provides) by Joe Greenseid. I hope to go through the links and include them subsequently in this HOWTO.

Other free document links for parallel processing are:

- The Beowulf Howto : This document introduces the Beowulf Supercomputer architecture and provides background information on parallel programming, including links to other more specific documents, and web pages. But, before that for an understanding of parallel processing and Symmetric multiprocessing on Linux, check out the following:
- The Parallel Processing on Linux HOWTO
- The Symmetric Multiprocessing HOWTO

7.2.2. Parallel processing software for Linux

Now after reading the above documents, you have an idea of parallel processing. Parallel program libraries are the core of parallel processing on a Linux cluster. There are various free implementations of parallel processing libraries. Since parallel processing is all about performance, these libraries have some very nice functional tools to analyze your parallel program performance. Given below is a set of links to these parallel program libraries and tools.

- Message Passing Interface: MPI is a standard specification of message passing libraries. The above document gives a lot of links to documents on the standard, etc.. A MPI implementation for Linux mpich is also available at that site. There are a lot of documents for Learning to use MPI .
- Local Area Multicomputer − LAM: LAM (Local Area Multicomputer) is an MPI programming environment and development system for heterogeneous computers on a network. With LAM, a dedicated cluster or an existing network computing infrastructure can act as one parallel computer solving one problem. LAM features extensive debugging support in the application development cycle and peak performance for production applications. LAM features a full implementation of the MPI communication standard. You can download the sources (tar−zipped, rpm) or binaries from here A host of MPI tutorial links and also a `getting started with LAM' tutorial is available here
- Parallel Virtual Machine : As the PVM home page describes, it is a software package that permits a heterogeneous collection of Unix and/or NT computers hooked together by a network to be used as a single large parallel computer. Thus large computational problems can be solved more cost effectively by using the aggregate power and memory of many computers. The software is very portable. The source, which is available free thru netlib, has been compiled on everything from laptops to CRAYs.
- Ganglia: Ganglia is an open source cluster monitoring and execution environment developed at the University of California, Berkeley Computer Science Division. As the above link describes it, "Ganglia is as simple to install and use on a 16−node cluster as it is to use on a 512−node cluster as has been proven by its use on multiple 500+ node clusters". It not only can link nodes in a cluster, but also link clusters to other clusters.

8. Word−Processing and Poster presenting tools on Linux

Those of you who do not use LaTeX and find it challenging and want a WYSIWYG word processor, keep your ears tuned to OpenOffice which has released version 1.2 of its openoffice software. Its tools may compare well with the best in the market.

8.1. Word Processing Tools

- Latex: LaTeX is a high−quality typesetting system, with features designed for the production of technical and scientific documentation. LaTeX is the de facto standard for the communication and publication of scientific documents. David R.Wilkin's primer "Getting Started with LaTeX" is a good tutorial to getting started with LaTeX. For those who have to live with a WYSIWYG documenting tool, check LyX. This is a front−end for latex. It isn't as powerful as latex proper, but helps with a good WYSIWIG.
- Lout: A document formatting system similar to latex. Good features, documentation and history. Light weight and outputs postscript. Thanks to Emiliano Gavilan for this link.
- Abiword: As the AbiWord home page says, "AbiWord is suitable for typing papers, letters, reports, memos, and so forth". It has won many awards and seems to be the best open source WYSIWYG word processor. Check out the above link to know more about it and download it.
- kword: As the kword home page says, "KWord is a FrameMaker−like word−processing and desktop publishing application. KWord is capable of creating demanding and professional looking documents. Whether you are a corporate or home user, production artist or student, KWord will prove a valuable and easy to use tool for all your word processing and layout needs". Check out the above link to know more about it and download it. (you might want to know more about the whole koffice suite).

8.2. Poster presentation tools

- KPresenter: KPresenter is the presentation tool of the KOffice suite of office utilities. It allows screen presentations with all the trappings one is used to seeing in costly presentation tools. It also allows honest, real scientific presentations where one does not have to impress the audience with non subject specific stuff :−). The best thing about it is the possibility of saving the presentation as a html file. It makes portable network graphics files with each presentation slide. With a smattering of knowledge of html files one could put in a animated image as a image link thereby allowing one to show movies too when necessary.
- Xfig : Though the man page claims that it is a facility for the interactive generation of figures ...., It in fact much more than that. Other than generating figures for elucidating what you want to say in a poster, it helps you import and export figures in a variety of formats, write text in various fonts and sizes, generate Greek symbols and color text, Save as latex picture file or any other format supported by your word processor for inclusion in your publications, generate GIFs of each page of the poster to put on your web site, and finally it generates *.fig files which are small in size. The only thing on my wish list for xfig is the capability to edit the imported pictures which are not in *.fig format. Therefore for a computer screen projected poster presentation you need a frames capable browser with contents in one frame and the xfig generated posters (exported as *.png or *.jpg from Xfig) on the other.

9. Free Database Management Systems for Linux

Scientific computing has two parallel data needs, one the physical values of the data itself, and the other is Database systems to manage the data. In this document links are provided only to database resources on the net and free Database systems. I personally do not use databases to manipulate the data generated by my codes. gawk, sed, and other basic Unix commands like grep, head and tail seem sufficient to manipulate any data I generate. I thought I should include this section for the large data generators. Hopefully a person with experience in databases will make this section better.

- Free database list
- ACM SIGMOD: Index of publicly available database software.
- MySQL: A relational Database management system.
- PostgreSQL As the link above describes it ...PostgreSQL is a sophisticated Object−Relational DBMS, supporting almost all SQL constructs, including subselects, transactions, and user−defined types

10. Linux in the laboratory

Again this is a section where I have zero experience and hope someone will with experience will contribute towards making this document better. However, I provide below links suggested by Sambaran Pahari and Deepak Gupta. These links seem to be very good from my inexperienced viewpoint.

- The Linux Lab Project A site for "Linux Lab Project."..everything to do with laboratory process, process control, automation and data acquisition on Linux. As the above link says, "The Linux lab project is intended to help people with development of data collection and process control software for LINUX. It is planned to provide a standardized development environment for a wide variety of applications from hardware support to application development".
- Linux Parallel port drivers: The above link says, "If you have a parallel port device and would like to know if there is a Linux driver available for it −−then this is the place to look". Sounds like a confident claim.