

# FREE/OPEN SOURCE SOFTWARE: AN ALTERNATIVE FOR ENGINEERING STUDENTS

Paulo S. Motta Pires<sup>1</sup>, David A. Rogers<sup>2</sup>

**Abstract** <sup>3/4</sup> Networked or stand-alone computational environments that are available as free or open-source software are very useful for teaching engineering courses. The cost of implementation of these environments is essentially that of the underlying personal computer (PC) hardware. The necessary software, of very high quality, is available free of charge through the Internet. Our purpose is to show that free or open source software can be used as a good alternative for engineering students. The main focus is on Scilab, a numerical software package. Often installed on machines running the Linux operating system, it is a powerful tool that is useful in teaching numerical methods or computationally intensive disciplines. Also presented are other free software packages, GSL (GNU Scientific Library) and LaTeX, that can be useful to the student. A package like Scilab can serve the needs of students when standard commercial mathematical software is not available.

**Index Terms**—Free/open source software, numerical methods, programming language, Scilab.

## INTRODUCTION

There is no doubt that the Internet has changed engineering education, especially in providing rapid and, often, free access to information. Teachers, researchers, and students can use the Internet to improve their professional knowledge and skills.

The Internet is also a vast and rich repository of free/open source computational software environments. Very high quality packages are available for mathematics, numerical analysis, parallel computing, data processing, image visualization, graphics, text preparation and presentation, and so on [1]. In this paper the central focus will be Scilab [2]-[3], often used in combination with Linux [4]. These alternatives give the student or professional with limited resources a long-term capability for performing significant computational studies.

The approach used in this work is to introduce free/open source software in numerically intensive graduate and undergraduate courses as described below:

- First, the free/open source software philosophy is presented. Emphasis is placed on World Wide Web search mechanisms and the opportunity that the students have to access information that will permit them to do their jobs better.

- Next, the Linux operating system is introduced. Options for installation on different computing systems are outlined along with some of its distinctive characteristics.
- Finally, the Scilab software package is presented.

The main body of this paper contains the following elements: (1) a brief description of the main software packages used is given, (2) a survey of some Scilab characteristics and capabilities (includes an overview of commands and functions) that are appropriate for engineering students is presented, and (3) examples of Scilab utilization in numerical methods courses for undergraduates and suggestions for utilization in graduate courses are given. For economy of space, the second and third elements are combined. Finally, some suggestions for including other useful material in these courses will be presented. The procedures used to install the necessary software are outlined in the Appendix.

## DESCRIPTION OF THE SOFTWARE PACKAGES

The Linux operating system is a free, open-source multi-task and multi-user operating system that is well known and widely used. Linux was first developed for 32-bit Intel x86-based personal computers (386 or higher) but has been ported to SUN SPARC and UltraSPARC machines, Motorola 68000-based machines, Power PCs, the IBM S/390, the Intel IA-64, and others.

There are several Linux flavors or distributions. For PCs, the following are examples of such distributions: RedHat [5], Suse [6], Mandrake [7], Slackware [8], Debian [9] and Conectiva [10]. The latter is a distribution that is available to Portuguese- and Spanish-speaking users. Linux can be downloaded through the Internet. The choice would be the ISO (ISO9660 type) files that are about 600 MB in size. With the ISO file the user burns an installation CD for himself or herself. If the Internet connection is slow, one can purchase Linux on CDs from several vendors including local bookstores. Installation instructions come with the distribution files or from the Linux HOWTO guides [11]. These guides are translated into several languages and are available in several file formats.

Scilab, a numerical software package developed at INRIA (Institut National de Recherche en Informatique et en Automatique), France, is similar in some respects to MATLAB® [12]. It uses an interactive graphical

<sup>1</sup> Paulo S. Motta Pires, Depto. de Engenharia de Computação e Automação, Univ. Federal do Rio Grande do Norte (UFRN), Brazil, pmotta@dca.ufrn.br

<sup>2</sup> David A. Rogers, Dept. of Electrical and Computer Engineering, North Dakota State University, Fargo, ND, 58105, David.Rogers@ndsu.edu

environment combined with a higher level programming language. The Scilab language can be interfaced easily with C or FORTRAN programs using dynamic links, through the **link** primitive, or using interfacing or gateway programs. The latter programs can be written by the user following the examples distributed with the Scilab package or generated through the Scilab **Intersi** companion toolbox. Also, Scilab can be used to do symbolic computation through an interface with Maple software [13]. Several high quality toolboxes are available to Scilab users, such as: Scicos, a package for modeling and simulation of dynamical systems; ANN, the Artificial Neural Network toolbox; the Signal Processing toolbox; and FRACLAB (Fractal, Multifractal and Wavelet Analysis toolbox). Scilab also has an extension called Parallel Scilab, Scilab // [14]-[15], that supports the execution of parallel jobs within Scilab. This extension can run on parallel machines or on a network of workstations (NOWs) and is based on the Parallel Virtual Machine, PVM, software paradigm [16]. Scilab has introductory documentation in several languages (English, French, Spanish, and Portuguese).

### SCILAB FOR ENGINEERING STUDENTS

Engineering students at both the undergraduate and graduate levels will normally need an introduction to the graphics environment of the software. Scilab can also run in Linux text mode if it is started as “**scilab -nw**”, but that will not be considered here. Scilab’s menu structure and its built-in libraries are typical initial elements to be presented in a course. Some representative built-in Scilab libraries that would be of interest in most classes are listed below:

- Scilab programming. Consists of several commands and functions that can be used to program with Scilab.
- Graphics functions, the functions to draw 2D and 3D graphics.
- Elementary mathematical functions.
- Input/output functions.
- Linear algebra functions.

In the sections that follow, the use of Scilab in undergraduate and graduate courses will be considered separately.

### SCILAB FOR UNDERGRADUATES

Scilab can be introduced to students initially as a very powerful calculator. At this level, students are exposed to prompt-line operations. They learn to edit command lines, how to use comments, and how to manipulate files and directories in the Scilab environment. Several basic operations are performed with real and complex numbers and with several elementary mathematical functions. The objective at this point is to acquaint the students with the environment. Next, students become familiar with polynomials, vectors, matrices, and lists data types. Several

examples of these data types are given. To illustrate, the polynomial

$$1 + 2s$$

with coefficients 1 and 2 and variable **s**, can be defined using the command **poly**,

```
-->poly([1 2], 's', 'c').
```

The Scilab prompt is the “-->” sign. A row vector can be defined using

```
-->w = [1 2 3].
```

To define a row vector, the elements should be separated by blanks or by commas. To define a column vector, the elements should be separated by a semicolon as in [1; 2; 3]. The handling of input and output of vector and matrix files follows. For example, the elements of the matrix **A**,

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 2 & 3 & 4 \end{bmatrix}$$

can be stored in an ASCII file called **matrixA** and read to the Scilab environment using the command **read**,

```
-->A = read('matrixA', 3, 4).
```

In this case, the numbers contained in the archive **matrixA** will be assigned to the variable **A**. Finally, loops (**for** and **while**), conditional commands (**if-then-else**, **select-case**), the concept of local and global variables, and the structure of functions are presented. At each point the student does some hands-on exercises

The process of introducing students to Scilab culminates with Scilab seen as a programming development environment. The students develop programs as separate files, load these files into the Scilab environment, and execute their code. Here Scilab is used as a programming environment to operate in some of the following areas:

- Non-linear equation solving using the bisection and Newton-Raphson methods
- Direct methods for solving linear equation systems (Gauss triangularization, LU decomposition)
- Iterative methods for solving linear equation systems (Gauss-Jacobi and Gauss-Siedel methods)
- Numerical solutions of non-linear systems of equations (Newton methods)
- Initial-value problems for ordinary differential equations (Runge-Kutta methods)
- Numerical integration (Simpson and Romberg methods)
- Interpolation and polynomial approximation

Figure 1 shows a typical program example written in the Scilab programming language that is appropriate for an undergraduate numerical methods course. This program is used to solve the ordinary differential equation,

$$\frac{dy}{dx} = (x - y) / 2 \quad (1)$$

in the interval [a,b] with step **h** using the traditional fourth order Runge-Kutta numerical integration method defined by:

$$y_{k+1} = y_k + \frac{h}{6}(f_1 + 2f_2 + 2f_3 + f_4). \quad (2)$$

The coefficients  $f_i$  are given by the equations:

$$\begin{aligned} f_1 &= f(x_k, y_k) \\ f_2 &= f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}f_1\right) \\ f_3 &= f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}f_2\right) \\ f_4 &= f(x_k + h, y_k + hf_3). \end{aligned} \quad (3)$$

The Runge-Kutta algorithm is implemented in the file **runge4.sci** while the equation to be solved is implemented in the file **fx.sci**. The extension **sci** is used to name files with Scilab programs or functions. These files also should be pure ASCII files and can be generated using text editors like **emacs**, **joe** or **vi**. It should be observed that comments are preceded by the “//” sign and that the Scilab syntax is close to traditional programming languages such as C or FORTRAN.

**FILE: runge4.sci**

```
function [XY] = rk4(a, b, h, y0)
// Ordinary differential equations
// Runge-Kutta method
// Inputs : [a,b] - integration interval
//          h - integration step
//          y0 - initial condition at x0
// Output : Matrix XY, first column x values
//          second column y (solution) values
X(1) = a
Y(1) = y0
n = (b-a)/h
for k=1:n
    xk = X(k)
    yk = Y(k)
    hf1 = h * f(xk, yk)
    hf2 = h * f(xk + h/2, yk + hf1/2)
    hf3 = h * f(xk + h/2, yk + hf2/2)
    hf4 = h * f(xk + h, yk + hf3)
    Y(k+1) = Y(k)+(hf1 + 2*hf2 + 2*hf3 + hf4)/6
    X(k+1) = X(k) + h
end
XY = [X Y]
```

**FILE: fx.sci**

```
function [fxy] = f(x,y)
// Inputs: x and y values to compute f(x,y)
fxy = (x - y)/2
```

FIGURE 1

PROGRAM TO SOLVE ORDINARY DIFFERENTIAL EQUATIONS USING SCILAB.

With respect to the file **runge4.sci**, **rk4** is the name of the function, the parameters **a**, **b**, **h** and **y0** are the input

arguments and the parameter **XY** is the output argument. The elements of the file **fx.sci** have analogous purposes. The sequence of steps to load the files and run these programs is shown in Figure 2. The **getf** (get function) command is used to load each function shown in Figure 1 into the Scilab environment. The syntax is:

```
-->getf('rkutta4.sci')
-->getf('fx.sci')
```

to load the files **rkutta4.sci** and **fx.sci** into the Scilab environment, respectively. We assume in this example that these files were generated in the same directory used to execute Scilab. Otherwise, the user should give the complete path to find the files. Unlike some other software, the name of the file that has the function can be different from the name of the function. Next, the program is executed. The syntax used is:

```
-->rk4(0, 3, 1/8, 1).
```

In this example the interval of integration is [0, 3], the step size is given by  $h = 1/8$ , and the initial condition is  $y_0 = 1$ . The bottom half of Figure 2 shows a portion of the output, the first column listing elements of  $x$  and the second column listing the corresponding computed values of  $y$ . This output is produced by the last line of the file **rkutta4.sci**, namely  $XY = [X \ Y]$ .

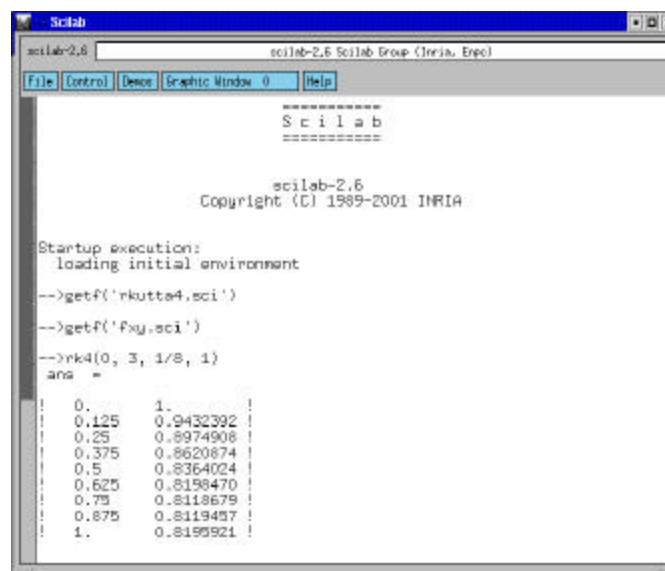


FIGURE 2

SCILAB ENVIRONMENT: LOADING AND RUNNING THE PROGRAM EXAMPLES.

**GRADUATE COURSES**

At the graduate level, a similar sequence is used to present free/open source software in general and Scilab in particular. Numerical linear algebra is an important topic in the authors’ graduate courses. Here students use Scilab to:

- solve linear equation systems by direct and iterative methods,

- compute norms of vectors and matrices,
- plot graphics, and
- do image compression using the SVD (singular value decomposition) method.

The Scilab programming language and built-in functions are introduced. Table I lists those most commonly used. Scilab is also a useful topic in a graduate-level research methods course since it provides a substantial research tool of interest to a broad spectrum of engineering disciplines that is compatible with other software environments and available at a minimal cost.

TABLE I  
EXAMPLES OF SCILAB MATRIX FUNCTIONS

Operation	Built-in function	Description
Matrix Factorization	chol	Cholesky decomposition
	lu	LU decomposition
	qr	Orthogonal decomposition
	schur	Schur decomposition
	svd	Singular value decomposition
Norms	norm	Norms of vectors and matrices (norms 1, 2, $\infty$ , and Frobenius are available)
Matrix Manipulation	A'	Transpose of matrix A
	cond	Condition number
	det	Determinant
	diag	Diagonal elements of a matrix
	inv	Matrix inverse
	rank	Rank of a matrix
	size	Size of a matrix
	spec	Matrix eigenvalues
	tril	Extract lower triangular part of a matrix
triu	Extract upper triangular part of a matrix	

As a companion to graduate courses using Scilab, the instructor can introduce GSL (GNU Scientific Library [17]). This is another free/open source program library written in C/C++. The results obtained from GSL can be compared with those obtained using Scilab. The GSL library includes functions to do manipulations with vectors and matrices, numerical differentiation, linear algebra, eigensystems, etc.

In both undergraduate and graduate courses, students are encouraged to present some project results as technical reports (each one about five to six pages in length). The structure of this short report is:

- Title page
- Abstract
- Introduction
- Description of the theoretical background
- Results (with tables and graphics)
- Conclusions
- References
- Appendix (with the source code developed)

Depending on available resources, the technical reports can be opportunities to introduce LaTeX [18]. A LaTeX

template is made available to download at the home page of the course. Students submit their reports as PS (PostScript) or PDF (Portable Document Format) files. The students learn a report format, develop their writing skills, and become proficient in a word processing (or typesetting) routine that can be used in other courses and even throughout their professional careers.

## STUDENT SURVEYS

In an effort to consider student reaction to an increased software emphasis, some specific questions were composed and included in the general student surveys used in two of the graduate courses offered by the authors. Students were asked their opinions as to whether or not there should be increased coverage of advanced mathematically oriented topics (such as statistics and systems theory). A total of 31 students responded to the survey. Fifty-nine percent of the students favored such a change, 19% opposed it, and 22% were undecided. Students were also asked if they favored increased coverage of engineering software packages and numerical methods in these courses. Seventy-two percent favored this change, 28% were undecided, and none opposed the change. It appears fair to conclude that use of software tools such as those considered in this paper is favored by most of the students who participated in the survey.

## CONCLUSIONS

We have presented some alternatives based on free/open source software for engineering students to use in their classes and even in their future professional work. The Scilab programming environment running under the Linux operating system along with libraries like GSL provides the necessary tools for a computationally intensive engineering activity with a minimal financial commitment. This same environment can serve the engineer in a professional work environment and provides a firm promise of long-term functionality.

## APPENDIX

The simplest and best procedure to install Linux is to boot the system from an installation CD and then follow the instructions that appear on the monitor. The Linux operating system also can be installed from an ftp site.

Scilab can be installed on a wide variety of platforms including Dec Alpha, Sun SparcStations, SGI, PPCs, Macintosh, HP, and PCs. There are pre-compiled (binary) versions and rpm files. Also, the source code is available for compilation as a GUNZIP tar shell script. Scilab, of course, can also be installed on Microsoft Windows machines (through a self-extracting .exe binary file for Windows 9X/NT/2000) or on machines with the \*BSD (FreeBSD, OpenBSD, NetBSD) operating systems. The procedure for installation depends on the file type used and

the target platform, but it is very simple to use and well documented.

### REFERENCES

- [1] Scientific Applications on Linux, available at <http://sal.kachinatech.com/index.shtml>, 2002
- [2] Scilab Home Page, available at <http://www-rocq.inria.fr/scilab/>, 2002
- [3] Gomez , Claude (Ed.), *Engineering and Scientific Computing with Scilab*. Boston: Birkhäuser, 1999.
- [4] The Linux Kernel Archives, available at <http://www.kernel.org>, 2002
- [5] RedHat Linux Distribution, available at <http://www.redhat.com>, 2002
- [6] Suse Linux Distribution, available at <http://www.suse.com>, 2002
- [7] Mandrake Linux Distribution, available at <http://www.mandrake.com>, 2002
- [8] Slackware Linux Distribution, available at <http://www.slackware.com>, 2002
- [9] Debian Linux Distribution, available at <http://www.debian.org>, 2002
- [10] Conectiva Linux Distribution, available at <http://www.conectiva.com.br>, 2002
- [11] Linux Documentation Project, available at <http://tldp.org>, 2002
- [12] MATLAB®, available at <http://www.mathworks.com>, 2002
- [13] Maple, available at <http://www.maplesoft.com>, 2002
- [14] The Scilab // Project, available at <http://www.enslyon.fr/~desprez/FILES/RESEARCH/SOFT/SCILAB/>, 2002
- [15] Deprez, F., Fleury, E., Grigori L., “Scilab//: User Interactive Application and High Performance,” available at <http://www.enslyon.fr/~deprez/OURAGAN/>, 2002
- [16] PVM – Parallel Virtual Machine, available at [http://www.csm.ornl.gov/pvm/pvm\\_home.html](http://www.csm.ornl.gov/pvm/pvm_home.html), 2002
- [17] GSL, the GNU Scientific Library, available at <http://www.gnu.org/software/gsl/gsl.html>, 2002
- [18] Comprehensive TeX Archive Network, available at <http://www.ctan.org/>, 2002