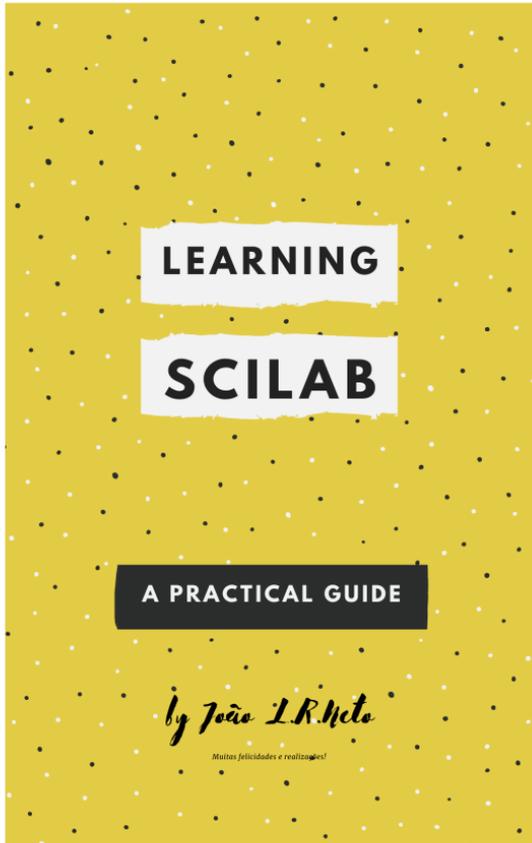


Scilab

Free and Open-Source Software for
Engineers and Scientists

Published by net629



© *net629*
ALL RIGHTS RESERVED.
<https://net629.com/>

For more information about permission to reproduce selections
from this book, write to register@net629.com.

2022, Second Edition

Edited by: net629

Cover design: net629

Subscribe to our newsletter. Simply write to register@net629.com
or visit our website <https://net629.com>.

Dedication



“

To my parents (in memorial), wife, sons and friends
who supported me in this work

Preface



Computer programming is the development of computer applications, a series of instructions written in a computer programming language to do specific work by the computer. We use various terms in the computer programming environment, such as function calls, abstraction, programming languages, data types, function calls, and models. In this book, we will cover the application of computer programming within the scope of Scientific Computing using the environment and the programming language of the Scilab software.

In the Principles of Scientific Computing, formulated by the mathematician John McCarthy in the 1970s, the emphasis is on models and methods, not formulas. The softwares developed for this purpose, such as Scilab, illustrates this principle because it was designed to automatically calculate the solutions of equations representing these models and methods.

Scilab - the acronym for Scientific Laboratory - is open-source software for scientific computing. The platform offers hundreds of predefined math functions and a high-level programming language.

The history of Scilab software goes back to the 1980s, developed in France at IRIA - Institut National de Recherche en Informatique et en Automatic to provide a useful automatic control tool for researchers. Since early 2017, is part of **ESI Group**, a pioneer and world-leading provider in Virtual Prototyping, leveraging materials physics.

Scilab's classification can contemplate several aspects. As a tool for automation, analysis, synthesis, and design of mechanical and electrical systems, such as controllers, servo motors, power supplies, and generators. However, the most popular software applications are software engineering, scientific computing, automotive design and manufacturing, aerospace, chemical engineering, computer graphics, CAD design, simulation, etc.

The development of the contents resulted from my experience during all academic activities in teaching computer programming for engineering courses, in particular. The topics covered do not exhaust the subject, but offer a factual basis capable of assisting the reader in software resources.

It is the first book for anyone who wants to know the possibilities of this software. It serves beginners in programming as well as for those who already work with other platforms. As free and open-source software, Scilab is an excellent alternative for those working in scientific computing with proprietary software. This guide aims to present the fundamentals of the environment and the programming language, showing practical examples of its functionalities.

João L. R. Neto

Belo Horizonte, Brazil. January, 2022



Introduction



“

Good friends, good books, and a sleepy conscience:
this is the ideal life.”

—Mark Twain

Scilab¹ software has been a leader in scientific computing since the mid-1990s. The first product was a spreadsheet preparation program that made mathematical calculations fast and easy enough for every school student to use. Since then, **Scilab!** products have added advanced features, like scientific computing, to grow in popularity among professional scientists and mathematicians.

It is still considered one of the most popular programs in use today. The reason for this is that it is exceptionally well designed, intuitive and versatile and is easy to use, even for a computer starter. The **Scilab!** software has many packages that include historical analysis tools, statistical analysis tools, Cartesian or probability calculations, graphical manipulation, time-series data analysis, and mathematical modeling packages.

For a layperson, it might be hard to understand how scientific computation, and especially the language used for scientific studies, differs from the standard language of computer programs. Analysis refers to the procedure done by a computer to solve a problem by searching through an input data set to find an output value. In scientific computation, this results in solving a particular problem using scientific principles. An individual scientist proves a scientific theory or finds a relationship between observed events by making statistical or other tests that allow direct observation of the data.

If you have been researching various science aspects, you may have come across the term scientific computation. If so, then you already know that scientific computing has been around for decades. However, it wasn't until sometime in the last decade that scientific computation was introduced into the world of common computer languages.

Scilab! is free, open-source, and cross-platform software with a

¹Scilab! (Scilab!)

high-performance programming language. It allows the implementation of highly accurate numerical methods with the support of several libraries.

Scilab! offers developers several benefits, such as open-source; he uses a common development infrastructure; supports multiple languages; can be used for simulation and scientific applications; it is friendly and extensible. Researchers have been used in several fields of science, including engineering, computer science, mathematics, physics, chemistry, biology, ecology, climatology, etc.

Scilab! has many advantages over other numerical simulation and analysis languages. First, **Scilab!** does not require any external or proprietary hardware. **Scilab!** runs on almost all operating systems. Also, **Scilab!**'s language compatibility makes it more practical and accessible for broad applications. The environment is easy to install, modify, and use for beginners and professionals.

Simulation and numerical analysis programs using **Scilab!** include simulation packages, scientific and applied applications, and graphical data displays. The users themselves developed most of the numerical models and simulation tools used in **Scilab!**.

1 Organization and Features

One of the best way to learn computer programming is to do computer programming. This is the approach we take in this book.

There are different types of software paradigms, each more or less suitable for solving specific types of problems. The vectors-matrix

paradigm is introduced with **Scilab!**, Matlab®²³ and others. These software is useful for dealing with models or problems involving intensive use of vector and matrix operations, cell arrays, and data structures, as well as problems of recursive structure requiring intensive use of ‘loops’. In this book, we present **Scilab!**, a high-level, free and open source software for scientific computing.

The book consists of six chapters covering everything from installing the software, going through the basics, the programming language, advanced features, and practical applications. Based on the author’s experience in teaching computer programming for decades, this script offers the beginner or advanced reader a comprehensive overview of **Scilab!**’s environment and programming language.

To master the **Scilab!** environment and programming language, we consider it essential to have a comprehensive knowledge of the software with its basics resources. These questions are related to chapters 1 and 2. The programming language starting in chapter 3 presents the basics resources of inputs, outputs and the control structures (branchings, and **loops!** (**loops!**)) and chapters 5 and 6 present advanced features.



²1994-2022 The MathWorks, Inc

³**Matlab!** (**Matlab!**)

Contents

Dedication	iii
Preface	v
Introduction	xi
1 Organization and Features	xiv
1 Getting Started	1
1.1 Installation	3
1.1.1 Environment	3
1.1.2 Installing and uninstalling toolboxes	3
2 Scilab basics	5
2.1 Overview	7
2.2 Objects, Data Types and Variables	7
2.3 Scilab Objects	8
2.3.1 Scalars	8
2.3.2 Matrices	10
2.3.3 Lists	13
2.4 Variables	15
2.4.1 Creating valid identifiers	15
2.4.2 Managing variables	15
2.4.3 Special constants	16
3 Scilab Programming	19
4 Data Visualization	23
5 Advanced Topics	27
The Author	31
An Important Final Note	35

Getting Started



“ Words are all we have.

—Samuel Beckett

1.1 Installation

1.1.1 Environment

1.1.2 Installing and uninstalling toolboxes

Scilab basics



“

Writing comes from reading, and reading is the finest teacher of how to write.

—Annie Proulx

2.1 Overview

2.2 Objects, Data Types and Variables

type() function

size() function

Notes

2.3 Scilab Objects

2.3.1 Scalars

Boolean

The Boolean type stores either a T or F value.

Listing 2.1: Boolean

```
--> 5 == 5
ans =

T

--> typeof(ans)
ans =

"boolean"

--> size(ans)
ans =

1. 1.
```

String

The string type is used to store a string.

Listing 2.2: String

```
--> "Scilab"
ans =

    "Scilab"

--> typeof(ans)
ans =

    "string"

--> size(ans)
ans =

    1.    1.
```

Polynomial

Rational

2.3.2 Matrices

Boolean matrix

A Boolean matrix is a matrix whose entries are either True - %t, %T or False - %f, %F. See Listing 2.3.

Listing 2.3: Boolean matrix

```
--> B = [%t,%f,%T,%F]
B =

   T F T F

--> typeof(B)
ans =

   "boolean"

--> size(B)
ans =

   1.   4.
```

String array

Polynomial matrix

A polynomial matrix has entries expressions that are composed of variables, constants, and exponents.

Listing 2.4: Polynomial matrix

```
--> p = poly(0,"z")
p =

   z

--> Mp = [p,1-p;1,$p^2$]
Mp =

   z   1 -z
```

```
1 z^2
--> typeof(Mp)
ans =

"polynomial"
--> size(Mp)
ans =

2. 2.
```

Rational matrix

Sparse matrix

A sparse matrix or sparse array is a matrix where only non-zero entries are stored.

Listing 2.5: Rational matrix

```
--> Sp = sparse([1,2;4,5;3,10],[1,2,3])
Sp =

( 4, 10) sparse matrix

( 1, 2) 1.
( 3, 10) 3.
( 4, 5) 2.

--> typeof(Sp)
ans =

"sparse"
--> size(Sp)
ans =

4. 10.
```

Boolean sparse matrix

2.3.3 Lists

The list is a more versatile data type available in **Scilab!**, making it possible to relate a list of values of different types separated by commas between braces.

List

Extracting list element

In the Listing 2.6, we extract the third element from the second item in the list created in the Listing ??.

Listing 2.6: Extracting list element

```
--> L(2)(3)
ans  =

-3.
```

Typed-list

Extracting list element

In the Listing 2.7, we extracted the labeled by the name color element from the list created in the Listing ??.

Listing 2.7: Extracting list element

```
--> Lt("color")
ans  =
```

```
"blue"  
--> Lt("weight")  
ans =  
  
10.
```

Standard state-space linear system

Extracting list element

We extracted elements A and C from the list. See Listings ?? and 2.8.

Listing 2.8: Extracting list element

```
--> S1("A"), S1("C")
ans =

    2.    0.    0.
    0.    3.    0.
    0.    0.    4.
ans =

    1.   -1.    0.
```

Transfer matrix

2.4 Variables

2.4.1 Creating valid identifiers

2.4.2 Managing variables

clear

isdef

exists

2.4.3 Special constants

The **Scilab!** has some special constants that cannot be deleted and/or changed.

SCI,WSCI

Variable containing the value of the root path of **Scilab!**.

Listing 2.9: SCI/WSCI

```
ls SCI
ls(SCI)
cd(SCI)
```

SCIHOME

TMPDIR

home

%e

Euler number.

%f or %F

Scilab Programming



“

Books are the quietest and most constant of friends; they are the most accessible and wisest of counselors, and the most patient of teachers.

—Charles W. Eliot

Data Visualization



“

Books have a unique way of stopping time in a particular moment and saying: Let's not forget this.

—Dave Eggers

Advanced Topics

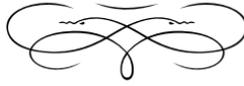


“

If one cannot enjoy reading a book over and over again, there is no use in reading it at all.

—Oscar Wilde

The Author



Ph.D. in Mechanical Engineering and Master in Electrical Engineering.

An Important Final Note

Writers are not performance artists. While there are book signings and public readings, most writers (and readers) follow their passion alone in their homes.

*What applause is for the musician, **reviews** are for the writer.*

Books create a community among readers; you can share your thoughts among all those who will or have read the book.

Leave a thoughtful honest review and help me to create such a community on the platform on which you have acquired this book. *What did you like, what can be improved? To whom would you recommend it?*

Thank you, also in the name of all the other readers who will be able to better decide whether this book is right for them or not! A positive review will increase the reach of the book, a negative review will improve the quality of the next book. I welcome both!