

# Computer tools for secondary school

Michele Tuttafesta

# Contents

<b>1</b>	<b>Gnuplot</b>	<b>5</b>
1.1	Plots of piecewise-defined functions . . . . .	6
1.2	Graphs of Multiple Functions on the Same Plane . . . . .	7
1.3	Surfaces in Space . . . . .	8
1.4	Plotting Data from a File . . . . .	10
1.5	Animations . . . . .	11
	Uniform circular motion . . . . .	12
	Ballistic motion with viscous drag . . . . .	15
	2D Elastic Collision of Two Spheres: One Initially at Rest . . . . .	17
	Elastic collision in 2D of two spheres . . . . .	19
	Electric potential . . . . .	22
	Motion produced by the Lorentz force . . . . .	25
	Gravitational conics: 2D astrodynamics . . . . .	29
	3D Astrodynamics . . . . .	36
	Brachistochrone . . . . .	39
<b>2</b>	<b>Tracker</b>	<b>45</b>
2.1	Basic Usage . . . . .	45
	Fall in water . . . . .	50
	Fall in Air with Drag . . . . .	51
	Uniformly Accelerated Vertical Motion and Measurement of $g$ . . . . .	55
	Projectile motion and measurement of $g$ . . . . .	57
	Ballistic motion and measurement of the viscous drag coefficient . . . . .	61
	Elastic collision in one dimension . . . . .	75
	Motion of the center of mass . . . . .	79
	Harmonic motion of a spring and measurement of the elastic constant . . . . .	84
	Friction model for a string sliding over an edge . . . . .	88
	Mechanical resonance . . . . .	91
<b>3</b>	<b>LTspice</b>	<b>97</b>
3.1	RLC Circuit: Time-Domain Analysis . . . . .	98
3.2	RLC Circuit: Frequency-Domain Analysis . . . . .	103

<b>4</b>	<b>Python</b>	<b>105</b>
4.1	Working Tools . . . . .	105
4.2	Interpreters and IDEs . . . . .	106
4.2.1	Google Colab . . . . .	106
4.2.2	Thonny . . . . .	107
4.2.3	Spyder . . . . .	107
4.3	Interactive Mode and Script Mode . . . . .	107
4.4	Getting Started with Python . . . . .	108
4.5	Important syntactic rules . . . . .	114
4.6	Variables, input/output and data structures . . . . .	116
4.7	Structured Programming in Python . . . . .	122
4.8	Functions . . . . .	123
4.9	Modules and Libraries . . . . .	125
4.10	File . . . . .	126
4.11	Matrices in Python . . . . .	129
4.12	Graphs with Python . . . . .	135
4.13	Random Numbers . . . . .	140
4.14	The Python os Module . . . . .	141
	Brachistochrone . . . . .	142
	Damped Harmonic Motion . . . . .	145
4.15	Resources . . . . .	149
<b>5</b>	<b>L<sup>A</sup>T<sub>E</sub>X</b>	<b>151</b>
5.1	Math mode vs. text mode . . . . .	151
5.2	Images . . . . .	152
5.3	Font styles . . . . .	152
5.4	Spaces, paragraphs, and comments . . . . .	153
5.5	Brackets . . . . .	153
5.6	Lists . . . . .	153
5.7	Symbols (Mathematical Mode) . . . . .	155
5.8	Symbols (in text mode) . . . . .	159
5.9	Templates . . . . .	159
5.10	Resources . . . . .	162

## Attività di laboratorio

1	Gnuplot: Uniform circular motion . . . . .	12
2	Gnuplot: Ballistic motion with viscous drag . . . . .	15
3	Gnuplot: 2D Elastic Collision of Two Spheres: One Initially at Rest . . . . .	17

4	Gnuplot: Elastic collision in 2D of two spheres . . . . .	19
5	Gnuplot: Electric potential . . . . .	22
6	Gnuplot: Motion produced by the Lorentz force . . . . .	25
7	Gnuplot: Gravitational conics: 2D astrodynamics . . . . .	29
8	Gnuplot: 3D Astrodynamics . . . . .	36
9	Gnuplot: Brachistochrone . . . . .	39
10	Tracker: Fall in water - uniform motion . . . . .	50
11	Tracker/Gnuplot: Fall in Air with Drag . . . . .	51
12	Tracker: Uniformly Accelerated Vertical Motion and Measure- ment of $g$ . . . . .	55
13	Tracker: Projectile motion and measurement of $g$ . . . . .	57
14	Tracker/Gnuplot/Python: Ballistic motion and measurement of the viscous drag coefficient . . . . .	61
15	Tracker: Elastic collision in one dimension . . . . .	75
16	Tracker: Motion of the center of mass . . . . .	79
17	Tracker: Harmonic motion of a spring and measurement of the elastic constant . . . . .	84
18	Tracker: Friction model for a string sliding over an edge . . . . .	88
19	Tracker/Gnuplot: Mechanical resonance . . . . .	91
20	Python: Brachistochrone . . . . .	142
21	Python: Damped Harmonic Motion . . . . .	145

# Preface

This book was created with the goal of making the world of computer science a powerful and accessible ally in secondary school education.

You won't find magic formulas or digital spells here (though the results may sometimes feel that way), but rather a reliable toolbox, field-tested and refined through years of hands-on experience by the author.

The main audience is curious teachers and students, those already convinced that a programming language can become a lens through which to view the world, and that computer science is not just a school subject, but a bridge to understanding real-world phenomena.

The PhD in Physics who signs these pages didn't stop at theory: every resource has been tested in real classrooms, with real students, and optimized to meet the needs of a school system often resistant to innovation. Whether your goal is to enrich a science lesson, simulate physical phenomena, or simply bring some excitement into a teaching unit, you'll find here ready-to-use ideas or ones easily adaptable to your context.

There's more. This book goes beyond secondary schools: many of the proposed activities are also well-suited to university settings, demonstrating that the boundary between school and research is often thinner than it seems.

I would like to thank all those who contributed to my education in computer science, in particular: Prof. Mario Capitelli and Dr. Gianpiero Colonna, both from the IMIP Institute of the CNR, and Prof. Giuseppe Pascazio from the Polytechnic University of Bari.

And now, dear readers, get comfortable in front of your digital device (or your sheet of paper, for those who enjoy contrasts), and get ready to discover how technology can turn teaching into an adventure.

Enjoy the read and happy coding!

Michele Tuttafesta

# Introduction

The “Computer tools” considered in this book are represented by software of significant educational and technological relevance.

The first chapter is dedicated to “Gnuplot”<sup>1</sup>, a software that allows students to develop programming skills while simultaneously acquiring abilities in graphical data representation.

The second chapter explains the use of the software “Tracker”<sup>2</sup>, designed to acquire and analyze trajectories (or generic space-time evolutions) of moving systems starting from video files.

The main focus of the third chapter is the software “LTSpice”<sup>3</sup>, a system that can be used to design and simulate electrical circuits, from the simplest to the most complex ones. It also provides the ability to analyze the characteristics (voltages, currents, power) of a circuit, both in the time domain and in the frequency domain. It is a particularly useful tool for teaching Physics in the final years of secondary school or at the university level.

The fourth chapter is dedicated to “Python”, a programming language that probably needs no introduction for those who have made it this far. In the following pages, significant examples will be provided demonstrating the application of this language to data analysis and graphical representation.

The fifth chapter contains an introduction to the amazing  $\text{\LaTeX}$ , a typesetting system of unparalleled elegance and power, designed for advanced typographic composition and universally acclaimed for its ability to produce documents of impeccable publishing quality. Created in 1984 by Leslie Lamport as an extension of the  $\text{\TeX}$  typesetting system developed by Donald Knuth,  $\text{\LaTeX}$  has become the de facto standard for writing scientific papers. Its importance in the mathematical and scientific community is unmatched, thanks to its ability to handle complex mathematical formulas with unsurpassed clarity and precision. In an age

---

<sup>1</sup><http://www.gnuplot.info>

<sup>2</sup><https://physlets.org/tracker>

<sup>3</sup><https://ltspice.it.softonic.com>

dominated by simplification and standardized digital aesthetics,  $\text{\LaTeX}$  remains a beacon of intellectual rigor and typographic beauty.

At the end of almost every section, significant examples related to the topic are provided, along with worksheets—often with data that can be freely chosen—followed by explanatory solutions.

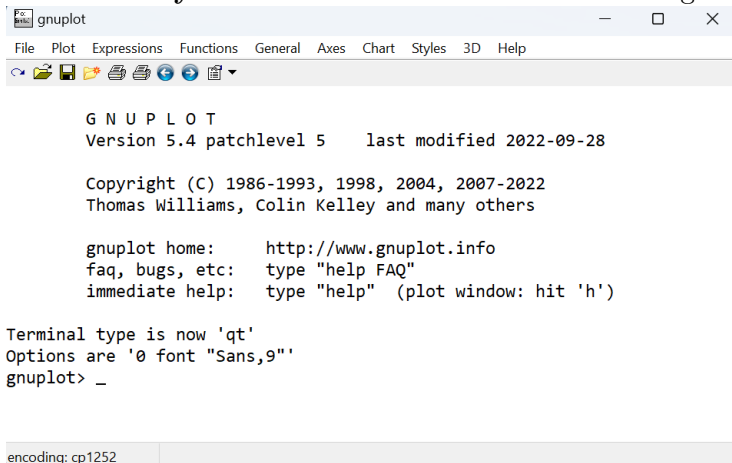
Any clarifications on certain mathematical algorithms proposed in the following pages can be found in the author’s text “Mathematical tools for secondary school” ([1]) or in the excellent books by Domenico D’Ortenzi ([2], [3], [4]).

# Chapter 1

## Gnuplot

We report below what is described on the official software page <http://www.gnuplot.info>: “Gnuplot is a portable command-line driven graphing utility for Linux, OS/2, MS Windows, OSX, VMS, and many other platforms. The source code is copyrighted but freely distributed (that is, you don’t have to pay for it). It was originally created to allow scientists and students to visualize mathematical functions and data interactively, but has grown to support many non-interactive uses such as web scripting. It is also used as a plotting engine by third-party applications like Octave. Gnuplot has been supported and actively developed since 1986.”

**On Windows systems** the software shows the following interface



```
gnuplot
File Plot Expressions Functions General Axes Chart Styles 3D Help
G N U P L O T
Version 5.4 patchlevel 5    last modified 2022-09-28

Copyright (C) 1986-1993, 1998, 2004, 2007-2022
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:   type "help FAQ"
immediate help:   type "help" (plot window: hit 'h')

Terminal type is now 'qt'
Options are '0 font "Sans,9"'
gnuplot> _

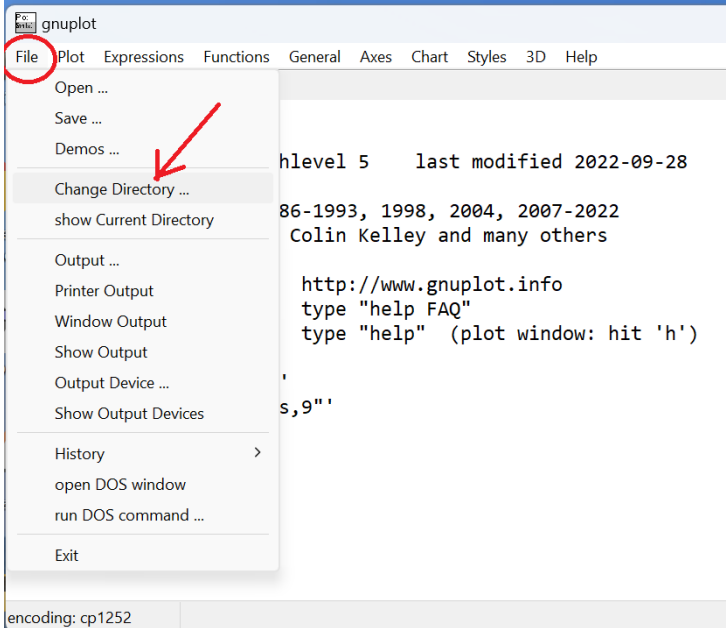
encoding: cp1252
```

One way to use it is by entering commands directly at the command line. For example, by typing:

```
plot sin(x)
```

and pressing [Enter], a simple plot of the function  $y = \sin(x)$  is displayed.

Another method, in many respects more effective, is to write the commands in a text file, using for example Notepad, one instruction per line. Then, from the 'File' menu, choose the directory where the text file containing the commands is located, as shown in the following figure:



and load the file, for example named 'script.txt', from the command line with the instruction:

```
load 'script.txt'
```

**On Linux systems**, using the terminal, simply move to the directory where the script (e.g., 'script.txt') is located and type:

```
$ gnuplot script.txt
```

We now present some significant examples, partly taken from the author's Moodle website ([5]), available by logging in as a "Guest" at <https://www.a049.it> and then accessing the course "Geometricamente", where additional documentation about the software can also be found.

## 1.1 Plots of piecewise-defined functions

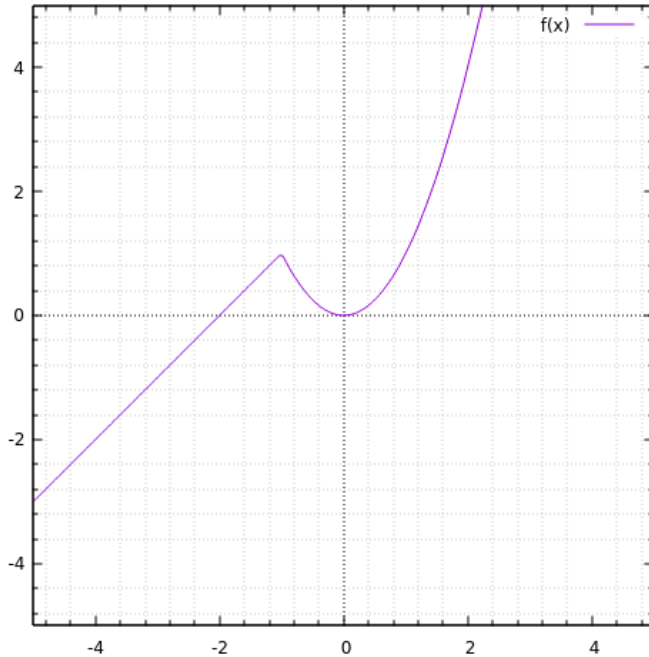
By running the following script

```
set size ratio 1
set zeroaxis
set grid mxtics mytics
```

## 1.2. GRAPHS OF MULTIPLE FUNCTIONS ON THE SAME PLANE<sup>7</sup>

```
set mxtics 5
set mytics 5
set xrange [-5:5]
set yrange [-5:5]
set samples 200
f(x) = x<-1 ? x+2 : x**2
plot f(x)
pause -1
```

we obtain



## 1.2 Graphs of Multiple Functions on the Same Plane

By executing the following script

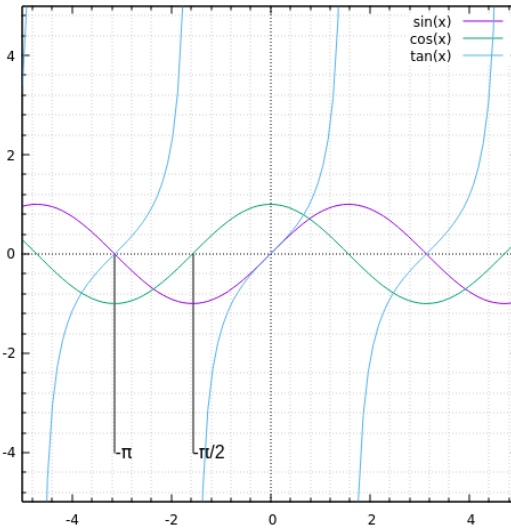
```
set size ratio 1
set zeroaxis
set grid mxtics mytics
set mxtics 5
set mytics 5
set xrange [-5:5]
```

```

set xrange [-5:5]
set arrow 1 nohead from -pi,-4 to -pi,0
set arrow 2 nohead from -pi/2,-4 to -pi/2,0
set label 1 "-{/Symbol p}" at -pi,-4 font "Arial,12"
set label 2 "-{/Symbol p}/2" at -pi/2,-4 font "Arial,12"
plot sin(x),cos(x),tan(x)
pause -1

```

you obtain



### 1.3 Surfaces in Space

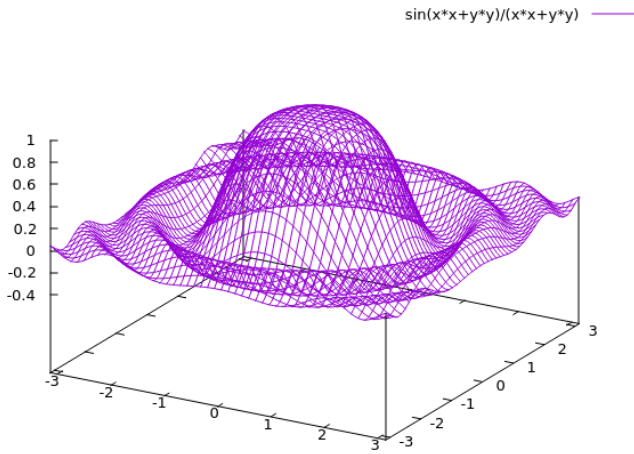
Surfaces in space can be defined by functions of two variables. By executing the following script

```

set xrange [-pi:pi]
set yrange [-pi:pi]
set isosamples 50,50
splot sin(x*x+y*y)/(x*x+y*y)
pause -1

```

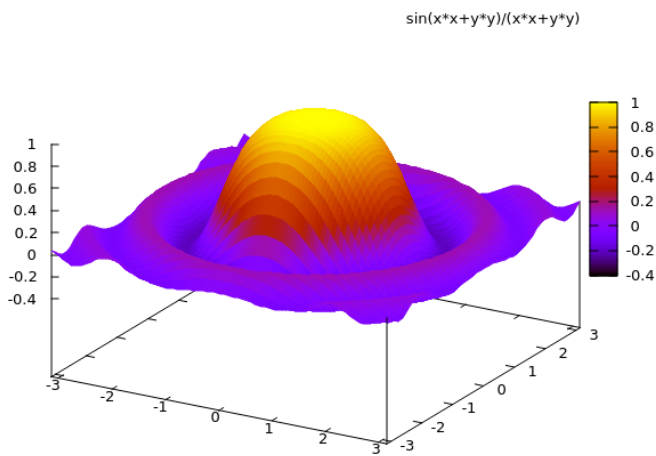
you obtain



or, using the “pm3d” option

```
set xrange [-pi:pi]
set yrange [-pi:pi]
set isosamples 50,50
splot sin(x*x+y*y)/(x*x+y*y) w pm3d
pause -1
```

you obtain





# Chapter 3

## LTspice

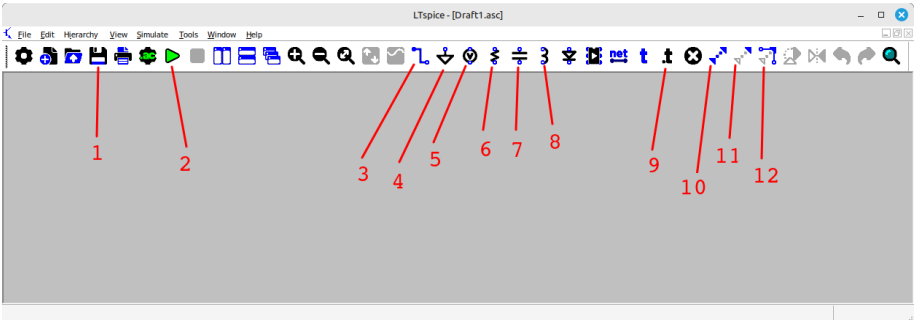
Here is the description taken from the distribution website:

<https://ltspice.it.softonic.com>:

“LTspice is a free graphical design software that allows you to create circuit designs. Developed by Analog Devices <sup>1</sup>, this program is a SPICE-based analog circuit simulator capable of analyzing the performance of circuit designs before prototyping. It offers various features appreciated by both beginners and advanced users. Moreover, it provides extensive online documentation. ...

**SPICE** or “Simulation Program with Integrated Circuit Emphasis” is an open-source analog electronic circuit simulator used in integrated circuit design. Its purpose is to verify the integrity of designs and predict their behavior before physical implementation. The program became popular due to its stability, speed, and ease of use. Among the many SPICE-based programs, **LTspice** includes enhancements over the original tool.”

The program interface, from the menu: File→New Schematic, is shown in the following figure.



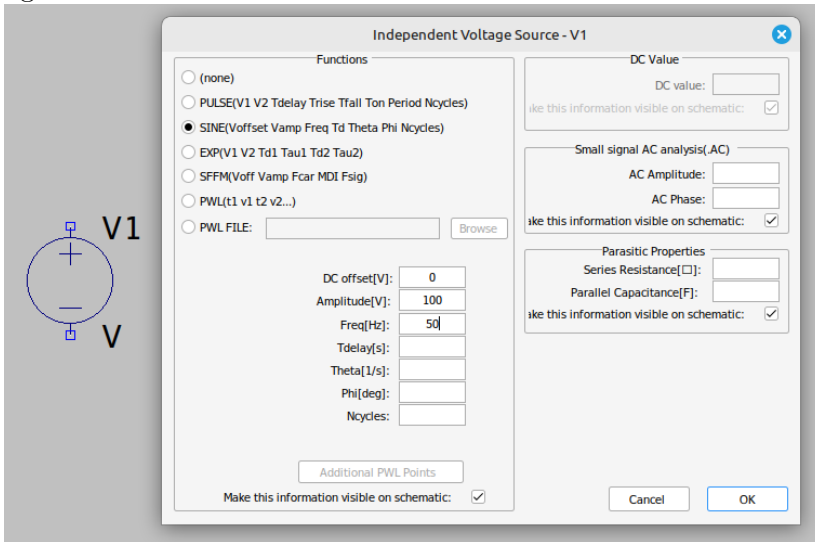
<sup>1</sup><https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html>

We illustrate its basic usage through a few examples.

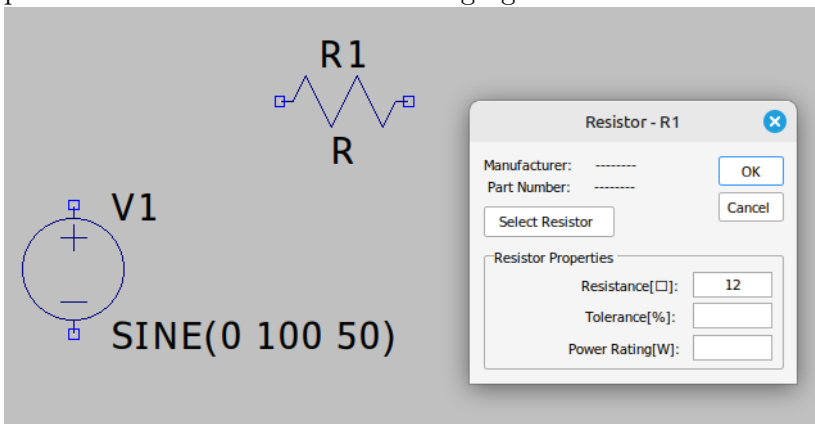
### 3.1 RLC Circuit: Time-Domain Analysis

First, we draw a series RLC circuit with an alternating voltage generator.

From the toolbar, select the voltage source (5) and place it in the drawing area using the left mouse button; then press [Esc] to cancel the automatic duplication of the component. Next, right-click on it, then select [Advanced], and enter its characteristics as shown in the following figure.



Now select a resistor (6) and, before placing it with the left mouse button, rotate it using Ctrl+r. Then, right-click on it and enter its parameters as shown in the following figure.



Repeat the same procedure for an inductor (8) and a capacitor (7), entering the parameters as shown in the following figures.