

FREE SOFTWARE

— experiences from using and teaching —

Predrag Pejović

05.05.2016.

a personal note: how did I get here?

because my longtime friend **Prof. Dr. Goce Arsov** invited me.

and he promoted me to the IEEE Senior Member rank ...

the committee was 2/3 from Macedonia and 1/3 from Spain

and why did I say this?

because if we don't state the values we appreciate,
we would forget them

and if we forget our values, we would lose them

and I learned this from Richard M. Stallman, at my old age

and Richard M. Stallman started the free software movement ...

“and you may ask yourself,
where did I come from?”

Professor at the Faculty of Electrical Engineering,
University of Belgrade

Department of Electronics, **power electronics specialist**

but for both more or less ...

diverse passions, diverse topics, mathematics and physics are always
the same ...

and everywhere the same ...

I use to say to my students that Kirchhoff's laws are not brought by
the Bundestag ...

“and you may ask yourself, how did I get here?”

B.S. University of Belgrade, 1990.

Thesis title: “**Software** for the System of Vector Control of Induction Machines”

Advisor: Slavoljub Marjanović, “**Caja**”

M.S. University of Belgrade, 1992.

Thesis title: „Bidirectional AC to DC Voltage Converter with Approximately Unity Power Factor“

Advisor: Slavoljub Marjanović, “**Caja**”

Ph.D. University of Colorado, Boulder, 1995.

Thesis title: “A Method for **Simulation** of Power Electronic Systems Using Piecewise-Linear Device Models”

Advisor: Dragan Maksimović, “**Maksa**”

... and I am “**Peja**”

the dark side of the moon?

Mathematics High School

("Srednje usmereno obrazovanje"):

OVRONTS „Veljko Vlahović“

nowadays translated to: **Matematička gimnazija**,
which was the initial name ...

profession: **programer**

motivated computer **user** ...

... actually, extremely motivated ...

... let's face the fact: **addicted!**

... but every love story has an ending ...

in my case, in the summer of 1996 ...

Introduction

some personal problems . . .

- ▶ Chief of the Department, . . .
- ▶ licensing, \$
- ▶ accumulation of data stored in proprietary formats . . .
- ▶ aware since I wanted to read my Ph.D. Thesis once again . . .
- ▶ $\frac{1}{2}$ self-compatible software
- ▶ really strange idea that I should control the machine, instead of the opposite
- ▶ a need for frequent installations, modifications, and upgrades of computers . . .
- ▶ and a visit to Zürich . . .

Worried Man Blues

“ . . .

it takes a worried man to sing a worried song
I'm worried now, but I won't be worried long

. . . ”

possible solution?

- ▶ free software?
- ▶ “there is no such thing as a free lunch!”
- ▶ or is there?
- ▶ tried free software every time when I bought a new computer — on the old one
- ▶ thus, not really often ...
- ▶ until 2008 ...
- ▶ and the appearance of ...

Ubuntu 8.04, Hardy Heron

Ubuntu 8.04 ...

- ▶ very good release
- ▶ mature technology
- ▶ visually attractive ... to me
- ▶ importance of visual impression!
- ▶ really comfortable working environment, pleasing ...
- ▶ **switched the working environment to Ubuntu!**
- ▶ and that was 2008 ... eight years ago
- ▶ experience: use **two** computers at first ...
- ▶ and I became enthusiastic about computers again!

Free Software

Free Software

- ▶ cost 0\$?
- ▶ **NOT THAT!** Not a single word about the cost!
- ▶ **Four Freedoms:**
 0. **to run the code as you wish**
 1. **to study the program**
(and modify, if you need/want to)
 2. **to share the program**
 3. **to share the modified version**
- ▶ and these four freedoms define free software
- ▶ as defined by Richard Matthew Stallman
- ▶ and the fourth of the freedoms is going to make a difference between permissive and restrictive free software licenses

people: Richard Matthew Stallman, RMS



source: Wikipedia, Wikimedia Commons

numerous excellent speeches, like

https://www.youtube.com/watch?v=Ag1AKI1_2GM

Richard M. Stallman: Free software, Free Society ← this is a link!

a pro-proprietary colleague of mine: "who is this Karl Marx?"

people: Eben Moglen



source: Wikipedia, Wikimedia Commons

very specific lawyer!

very, very nice speeches!!!

how it all started ← this is a link!

where are we now ← this is a link!

Friedrich Engels?

worth to know!

different things:

1. free software
2. open source software
3. freeware
4. shareware

many people do not understand the difference,
especially between free software and freeware,
but there is a **huge** difference!!!

permissive software licenses

- ▶ four freedoms provided
- ▶ no restrictions
- ▶ especially on the fourth freedom ...
- ▶ you can close the code ...
- ▶ examples: BSD license, MIT license ...
- ▶ and this is how it all started ...
- ▶ with X Window System ...
- ▶ SPICE is released under permissive BSD license ...
- ▶ and the rest is history

BSD license

http://en.wikipedia.org/wiki/BSD_licenses

7/17/2007

Spice is covered now covered by the BSD Copyright:

Copyright (c) 1985-1991 The Regents of the University of California.
All rights reserved.

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, and distribute this software and its documentation for any purpose, provided that the above copyright notice and the following two paragraphs appear in all copies of this software.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

MIT license

http://en.wikipedia.org/wiki/MIT_License

Copyright (C) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyleft

- ▶ ©
- ▶ Copyright: “all rights reserved”
- ▶ Copyleft: “all rights reversed” or “all wrongs reversed” or “some wrongs reversed”
- ▶ goal to protect free software as free
- ▶ **restrictive license, the fourth freedom is affected!**
- ▶ “viral license”
- ▶ The GNU General Public License
- ▶ The GNU Lesser General Public License
- ▶ The GNU Affero General Public License
- ▶ The GNU Free Documentation License

a word on “open source”

- ▶ not a hot issue any more
- ▶ but it used to be!
- ▶ essentially, a marketing issue ...
- ▶ that created a big fight ...
- ▶ open source initiative
- ▶ essentially, the same thing ...
- ▶ FOSS
- ▶ ... and I am not going to bother with this any more!

is there free food?

- ▶ software is a form of mathematics?
- ▶ zero marginal cost
- ▶ not a material object
- ▶ flamed debates ...
- ▶ among people that do not have any personal interest ...
- ▶ nowadays mainly resolved:
 1. Wikipedia
 2. embedded in embedded systems
 3. dominates the server market
 4. Python
 5. Apache web server
 6. Firefox
 7. Android
 8. MOOCs, you can see what other people do ...
 9. Ubuntu pre-installed ...
- ▶ it exists
- ▶ its up to you whether to use it or not ...
- ▶ but leave other people to make their choices ...

Education

a glimpse on education ...

Branislav Nušić, ...

“Autobiografija” (Autobiography) ...

copyright expired?

[https://klubcitalaca.files.wordpress.com/2010/12/
branislav-nusic-autobiografija.pdf](https://klubcitalaca.files.wordpress.com/2010/12/branislav-nusic-autobiografija.pdf)

... a section on education ...

... and what have changed?

... technology! other than that, there is no change!

teaching ...

- ▶ education reform ...
- ▶ emergence of “**practicums**,” although the meaning is ...
- ▶ translated: small classes intended to provide “practical” knowledge (taught by ...)
- ▶ which did not exist before ...
- ▶ and we had to create them ...
- ▶ retell a book you have partially read only to teach something that you do not practice otherwise?
- ▶ **maybe I should teach something I actually do?**
- ▶ not a rocket science, related to practice, useful, ...
- ▶ careful!
- ▶ Department of Electronics (not others!), 30–50 students
- ▶ **you asked for it!**

“Software Tools in Electronics,” 2010.

motivation:

- ▶ problems with licenses
- ▶ poor understanding of the legal side of the problem
- ▶ **absolute necessity to be 100% legal**
- ▶ poor awareness of accumulating data in proprietary formats
- ▶ visual appearance of a number of students' theses ...
- ▶ an opportunity to spend some time to learn to use some software packages ...
- ▶ since this usually starts too late ...
- ▶ ... and never ends!
- ▶ ... and it was necessary to create some new practicum offerings ...
- ▶ enough arguments!

Software Tools in Electronics (SAE)

With tremendous help of my students!

#1: Mirela Simonović

“we should state . . .”

and not so much enthusiasm among colleagues . . .

elective course!

potential for impassioned half-informed disputants . . .

- ▶ ideologically sensitive
- ▶ hits some interests:
in favor for someone, but not for everyone
- ▶ even aesthetics, but in matters of taste, there can be no disputes
- ▶ nowadays mostly resolved
- ▶ existence cannot be disputed
- ▶ solution: **elective course!**
- ▶ however: not a single tiniest problem **with students**

“goals and outcomes ...”

1. coping with GNU/Linux OS, Ubuntu
2. **command line!** shell scripts (will be useful!)
3. documentation and presentations, \LaTeX
4. drawing schematics, Xcircuit, $\frac{1}{28} \approx 3.57\%$ of the course
5. visualizing data, gnuplot and matplotlib
6. numerical mathematics, GNU Octave and PyLab
7. symbolic computing, Maxima (wxMaxima), SymPy
8. general purpose programming, **Python**

... and all of this should fit in 14 times 2 hours!

course content . . .

easy course; easy?

learning on examples the students are familiar with

alternative: reading manuals?

frequently above 1000 pages!

like learning a foreign language by reading a dictionary!

nothing to memorize: everything is “open book, open notes”

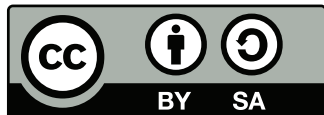
two tools wherever meaningful: in that manner, mastering the third would be easy!

$\frac{27}{28} \approx 96.43\%$ of the course fits any engineering major,
mathematics, physics, chemistry, . . . suitable to anyone who needs
computers, type text, do some math, and programming . . .

honestly, did not resonate with my own school, the ETF

available!

<http://tnt.etf.rs/~oe4sae/>



take it, use it, teach it, translate it,
more than welcome!

back to work!

- ▶ introduction about the class, two hours, $\frac{2}{28} \approx 7.14\%$
- ▶ what is free software and how come that it exists?
- ▶ a story about licensing ...
 - ▶ four freedoms
 - ▶ **“copyleft”, GPL** (ingenious invention!)
 - ▶ “permissive”, BSD
- ▶ related areas, CC
- ▶ the only ideologically sensitive class (double class)
- ▶ still enough time “to drop it”

Ubuntu

- ▶ new environment ...
- ▶ I insist on default settings on the machines
- ▶ Unity
- ▶ frequently, smooth and easy
- ▶ before we reach the command line ...
- ▶ but now, students are getting better even in that (why?)
- ▶ commands ...
- ▶ **and shell scripts!**
- ▶ which is the first facing of programming in this class ...
- ▶ let's create and run few programs, C and Python
- ▶ get acquainted with gedit
- ▶ lasts for 4 hours, $\frac{4}{28} \approx 14.29\%$

- ▶ T_EX, Donald Knuth, Stanford, 1978.
 - ▶ “Science is what we understand well enough to explain to a computer. Art is everything else we do.”
 - ▶ “A mathematical formula should never be “owned” by anybody! Mathematics belong to God.”
- ▶ L^AT_EX, Leslie Lamport, 1984.
- ▶ markup language
- ▶ WYSIWYM
- ▶ back home **used by** mathematicians and physicists!!!
- ▶ old, but very vivid and in development!
- ▶ after L^AT_EX, mastering HTML is easy . . . concept is the same!
- ▶ that would be “the second one,” but there is no time . . .

another quote ...

- ▶ “How \LaTeX changed the face of Mathematics”, an E-interview with Leslie Lamport, the author of \LaTeX
- ▶ Q: Is \LaTeX hard to use?
- ▶ A: It’s easy to use—if you’re one of the 2% of the population who thinks logically and can read an instruction manual. The other 98% of the population would find it very hard or impossible to use.
- ▶ insulting or motivating?
- ▶ fortunately, all of my students are within the 2%

this is not for everyone ...

if you need:

$$\ln \left(\frac{1+x}{1-x} \right) = 2 \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1}$$

$$\int \frac{1}{\sqrt{1-x^2}} dx = \arcsin(x) + C$$

$$DPF(k, \phi) = \frac{\sqrt{3} (k \cos \phi + 16)}{\sqrt{768 + 27 k^2 + 96 k \cos \phi - 24 k^2 (\cos \phi)^2}}$$

L^AT_EX is for you! Otherwise: **LibreOffice!**

so, fits our purpose!

my students are in “the 2%” !!!

you have a feeling that you program, not that you create text

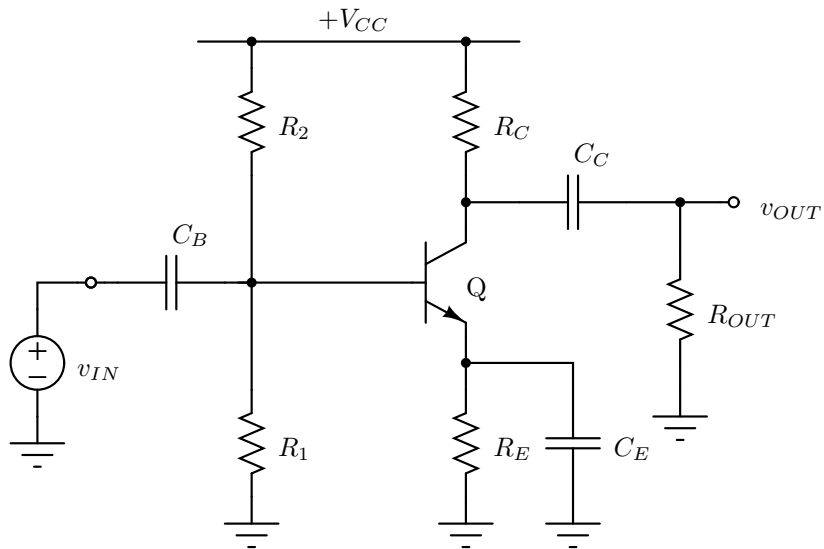
and that's exactly what I like:

I don't like creating text, I like programming!

students agree with 50% of this!

lasts for $3\frac{1}{2}$ weeks, 7 hours, $\frac{5}{28} \approx 17.86\%$ of the course

XCircuit



XCircuit + L^AT_EX

- ▶ not really a user-friendly program . . .
- ▶ but with a wonderful output!
- ▶ “interoperability”
- ▶ **very** educational, shell scripts, integrating skills . . .
- ▶ difficult, requires some understanding, but useful!
- ▶ experience: success!
- ▶ lasts for 1 hour, $\frac{1}{28} \approx 3.57\%$ of the course!
- ▶ this is only for electronics majors,
but XCircuit could be used in other ways . . .

Beamer

Beamer is used to create this presentation ...

but this is plain Beamer, default ...

lasts for 2 hours, $\frac{2}{28} \approx 7.14\%$ of the course

interesting: foreign universities, templates

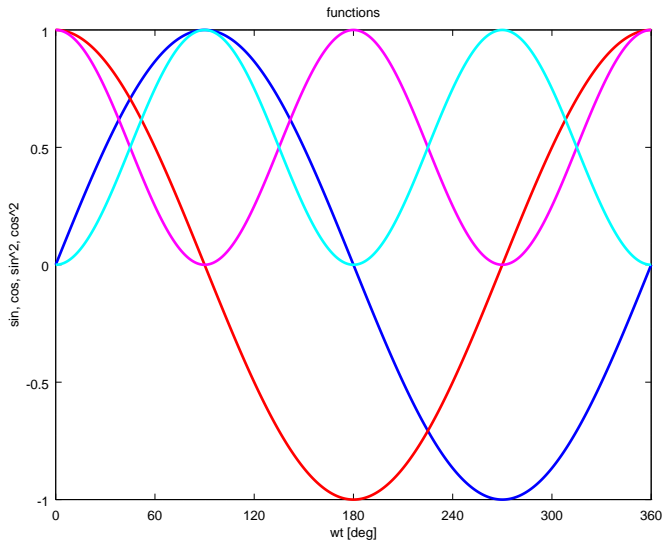
some examples:

<http://www.hartwork.org/beamer-theme-matrix/>

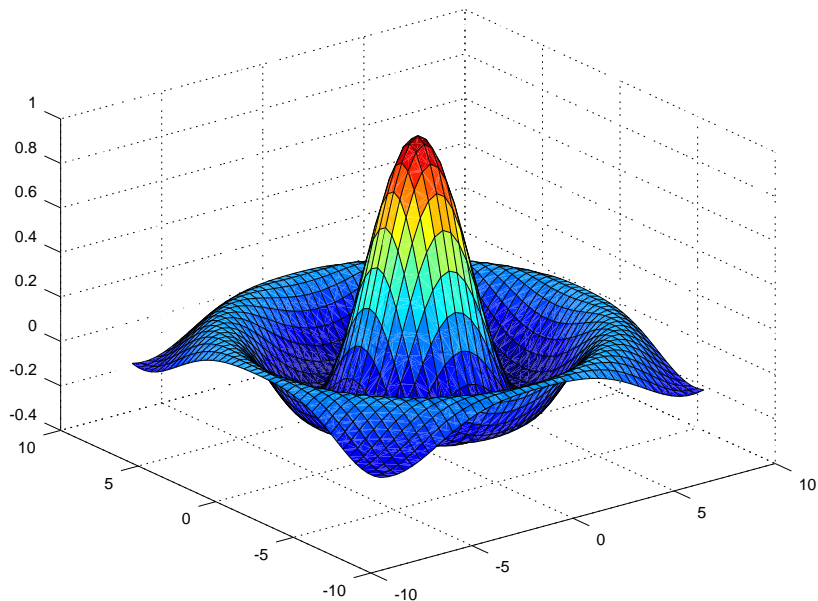
GNU Octave

- ▶ numerical computing (versus “symbolic computation”)
- ▶ interpreter, programming
- ▶ data visualization . . .
- ▶ “interoperability”
- ▶ language “clone”, one of
- ▶ language upgrade to the alternative . . .
- ▶ until recently under windows did not look nice
- ▶ slower, but I do not find it as a problem
- ▶ manual 972 pages long!
- ▶ lasts for 2 hours, $\frac{2}{28} \approx 7.14\%$ of the course

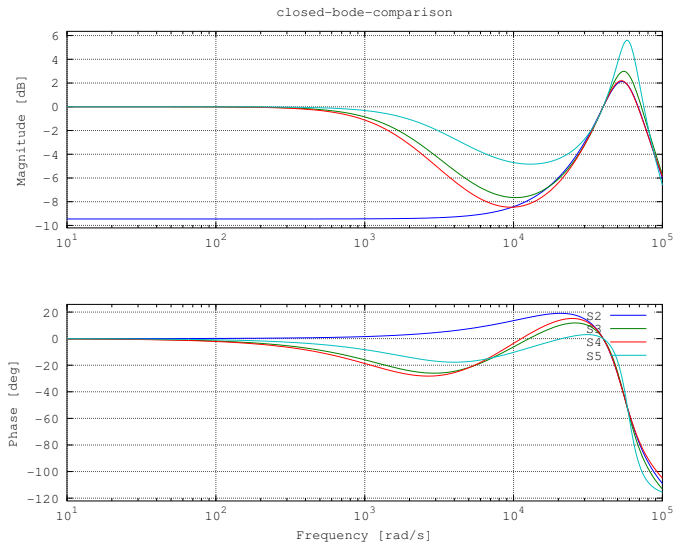
a picture for its own sake



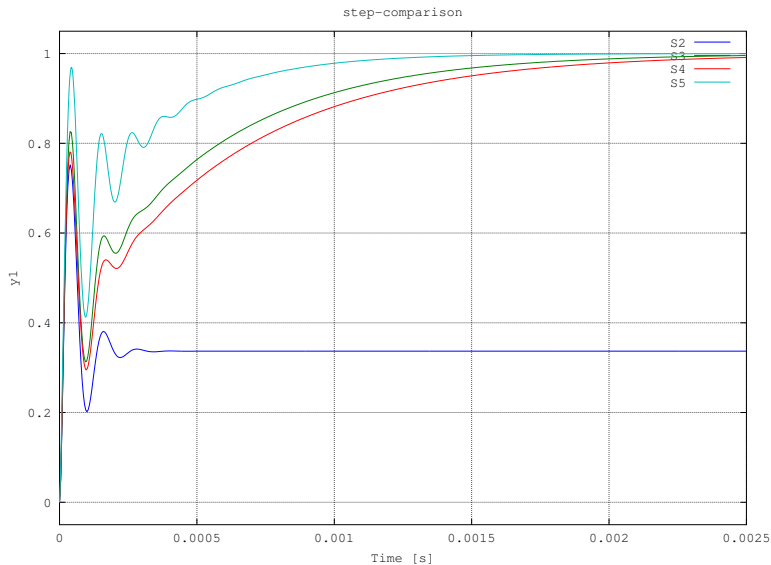
and another one!



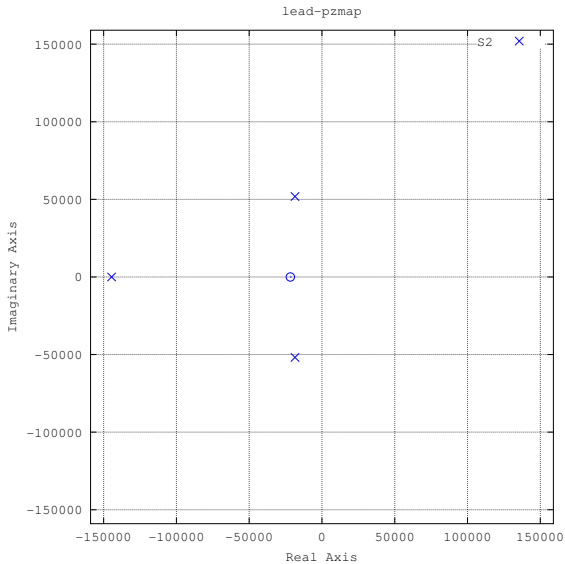
toolboxes, control, Bode plot



toolboxes, control, step response



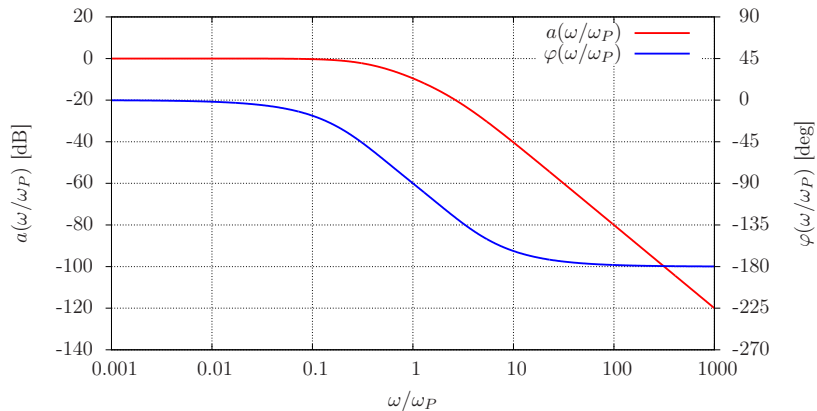
toolboxes, control, pole zero map



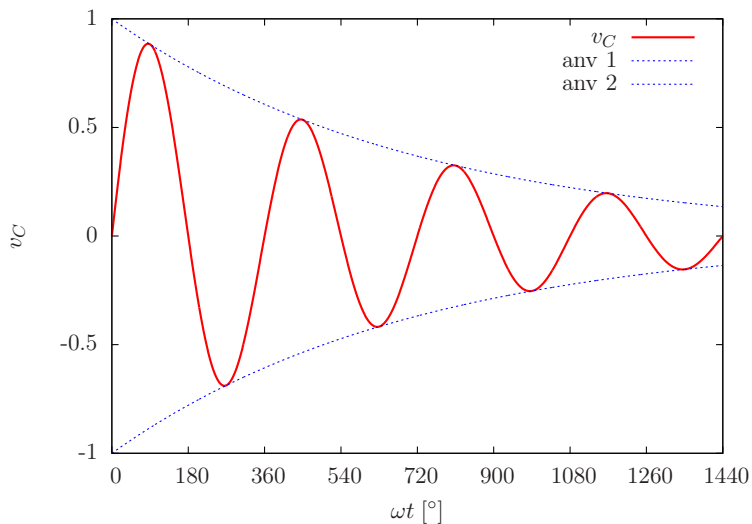
gnuplot

- ▶ data visualization
- ▶ “command line”
- ▶ **awful interface and beautiful result!**
- ▶ backend for GNU Octave, Maxima, wxMaxima, R, ...
- ▶ takes time to master ... (about two hours of work!)
- ▶ high quality output
- ▶ command line abstraction, already built up, useful!
- ▶ lasts for 2 hours, $\frac{2}{28} \approx 7.14\%$ of the course
- ▶ actually, somewhat longer ...
- ▶ just examples here, some code a little bit later ...

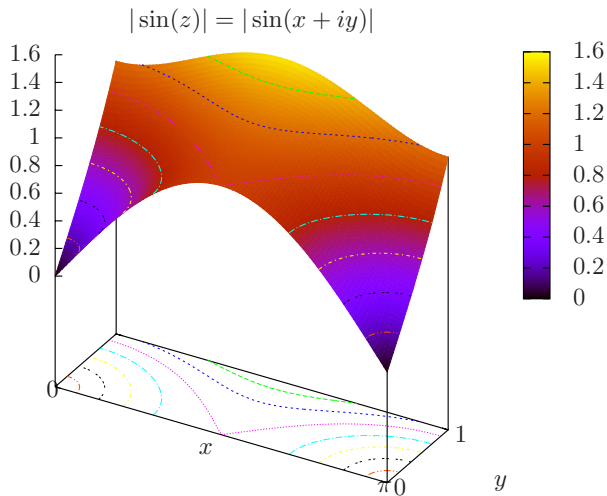
gnuplot, frequency response, repeated pole



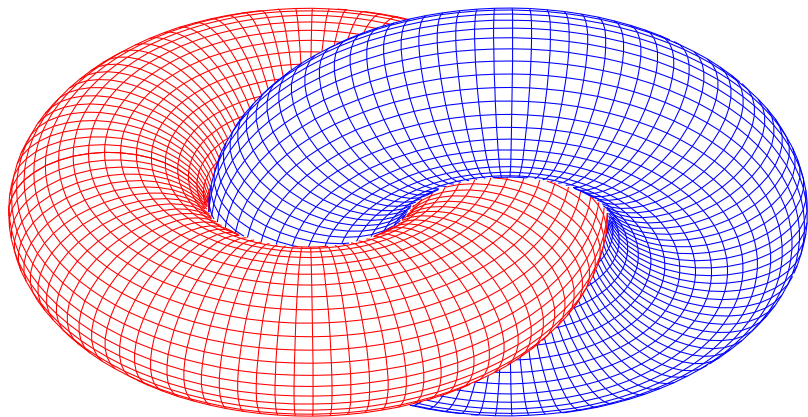
gnuplot, impulse response, pole pair



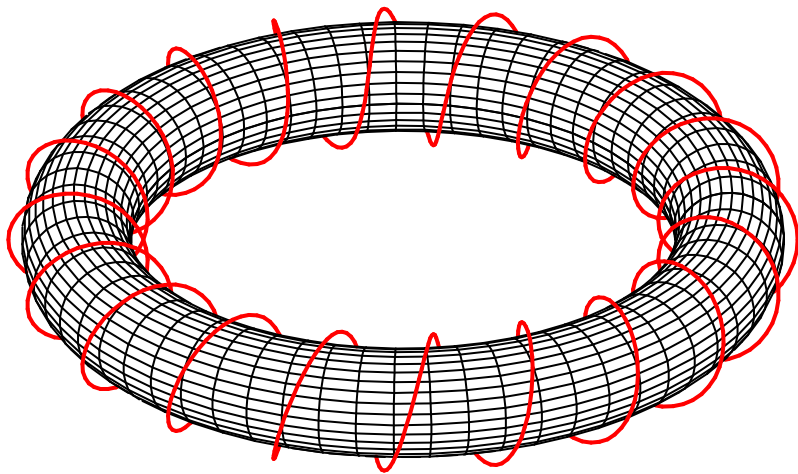
gnuplot, $\sin(z)$, complex function, modulus



gnuplot, tori, current density and magnetic field



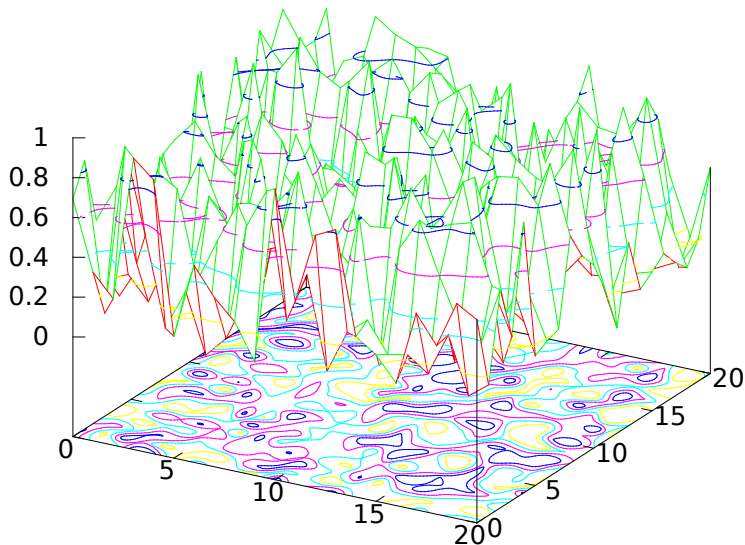
gnuplot, hysteresis curve test coil



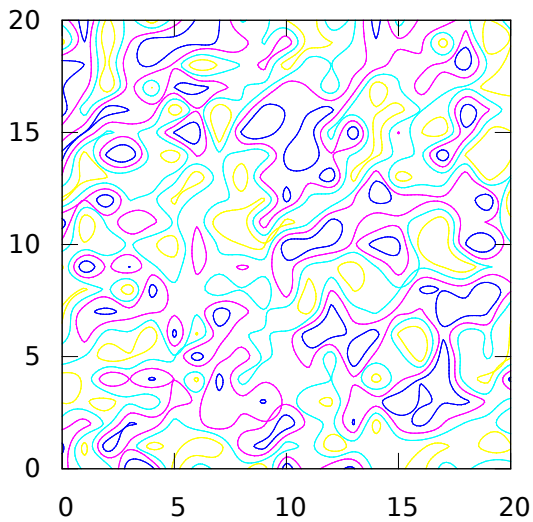
gnuplot, 3D, a **non-engineering** example ...

```
set contour both
set hidden3d
set cntrparam cubicspline
unset key
set terminal pdfcairo size 10cm, 8cm
set ztics 0.2
set output "s3d.pdf"
splot "z" matrix with lines
```

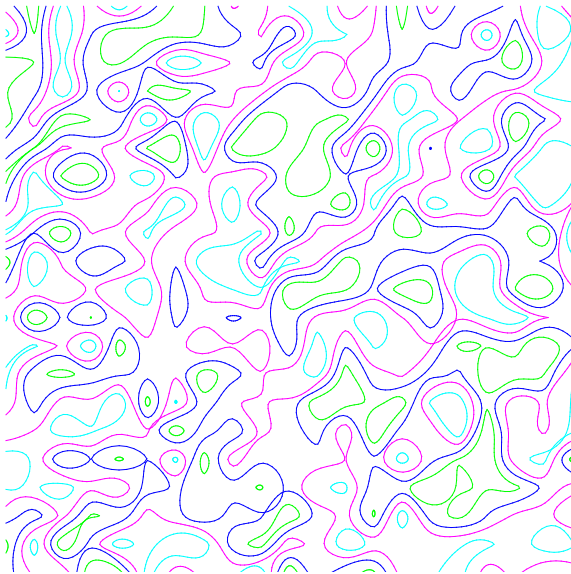
gnuplot, 3D, output, students made some fun of me ...



gnuplot, 3D, map



gnuplotism!



galleries ...

uniform gallery

normal gallery

ingredients: 1. random terrain, GNU Octave

```
z = rand(21);  
save z z
```

ingredients: 2. figure, gnuplot

```
set contour
unset surface
set view map
set cntrparam cubicspline
unset xtics
unset ytics
unset key
set size square
set terminal pdf size 10cm, 10cm
set output "slika.pdf"
set rmargin at screen 1
set lmargin at screen 0
set bmargin at screen 0
set tmargin at screen 1
unset border
splot "z" matrix with lines
```

ingredients: 3. repeat loop, bash script

```
#!/bin/bash

for i in {1..100}
do
    octave slika.m &> /dev/null
    gnuplot slikapng.gp
    mv slika.png slika$i.png
done
```


in the case it was worth something ...

“artistic property” rights?

1. random number generator?
2. programmer? (which one?)
3. proposer? (a student, Bojan Ilić)

fortunately, without any financial value!

Maxima and wxMaxima

- ▶ symbolic computing
- ▶ extremely heretical concept ...
- ▶ ... but not disputed any more!
- ▶ Maxima (ex Macsyma, 1968-1982)
- ▶ manual 1094 pages
- ▶ wxMaxima ...
- ▶ examples ...
- ▶ an example, Taylor
- ▶ “interoperability”
- ▶ we’ll get back to symbolic computing later on!
- ▶ lasts for 2 hours, $\frac{2}{28} \approx 7.14\%$ of the course

Python

- ▶ somewhere, the first language: MIT, UC Boulder, ...
- ▶ available since 1991 (“he pushes for something new”)
- ▶ interpreter
- ▶ **very** popular!
- ▶ youtube, Google, ...
- ▶ modern language!
- ▶ modular!
- ▶ “gluing language”
- ▶ very general purpose ...
- ▶ diverse applications!
- ▶ numerous MOOCs!!!
- ▶ personally, I recommend MIT Grimson-Guttag MOOCs!

one book ...

How to Think Like a Computer Scientist Learning with Python

Allen Downey, Jeffrey Elkner, Chris Meyers

Green Tea Press, Wellesley, Massachusetts

...and one quote:

page 90:

... One of the characteristics of algorithms is that they do not require any intelligence to carry out. They are mechanical processes in which each step follows from the last according to a simple set of rules.

In our opinion, it is embarrassing that humans spend so much time in school learning to execute algorithms that, quite literally, require no intelligence.

On the other hand, the process of designing algorithms is interesting, intellectually challenging, and a central part of what we call programming.

“particle man, particle man, ...”

Relatives

- ▶ “Google man”
- ▶ “toolbox man”

are gone. We need different education models.

Python for Programmers

- ▶ lasts for 4 hours, $\frac{4}{28} \approx 14.29\%$ of the course
- ▶ for students already familiar with programming, just the language
- ▶ fairly complete coverage ...
- ▶ environments: python, IDLE, ipython
- ▶ exam topic: create a table and write it to text file

PyLab and SymPy

- ▶ PyLab + SymPy lasts for only 2 hours, $\frac{2}{28} \approx 7.14\%$ of the course
- ▶ PyLab: NumPy, SciPy, matplotlib, IPython: “interoperability”
- ▶ the same as GNU Octave, but with modern language!
- ▶ **I believe that the future is here!** ... and in Julia!
- ▶ **very** nice data visualization, matplotlib!
- ▶ matplotlib 1.5.1, manual 2864 pages!
- ▶ definitely, a new style of learning, I remember the way we had to “learn” COBOL in high school ...

PyLab and SymPy

- ▶ SymPy, symbolic computing
- ▶ version 1.0, manual 1785 pages
- ▶ another project with the future ...
- ▶ again, advantage of a modern language!
- ▶ examples are the same as for wxMaxima, on purpose
- ▶ more of the same ...
- ▶ is there a problem to learn a third tool of this kind?
- ▶ the concept is important, not a specific tool

exam?

- ▶ this is the worst part!
- ▶ essentially, three midterms:
 1. \LaTeX + XCircuit
 2. GNU Octave + gnuplot + wxMaxima
 3. Python + PyLab + SymPy
- ▶ midterms are open book, open notes
- ▶ automated problem generation, a program!

and that would be all about the class ...

number of students:

1. 2010/2011: **19**
2. 2011/2012: **12**
3. 2012/2013: **23**
4. 2013/2014: **44**
5. 2014/2015: **61**
6. 2015/2016: **89**

Applications

application: virtual instruments

- ▶ automated measurement systems
- ▶ specialized instruments created using general purpose instruments
- ▶ oscilloscope → samples
- ▶ PySerial for acquisition (scope limitation, no TCP/IP)
- ▶ compute goal quantities, digital signal processing, NumPy
 1. power
 2. root-mean-square values (RMS)
 3. power factor (PF)
 4. total harmonic distortion (THD)
 5. ...
- ▶ form tables and diagrams (matplotlib)
- ▶ automated report generation (Python, L^AT_EX)
- ▶ “interoperability!”

application: virtual instruments

report, 6-pulse

report, 12-pulse

application: animations

rectifier, animation

approximation, animation

buck converter, open loop, animation

buck converter, regulated, animation

synchronous buck converter, animation

application: automated generating of test problems

the first midterm, example

combination space of $15^3 = 3375$ possibilities (at least!)

which for about 100 test sets is quite acceptable ...

as I said, the worst part is the exam ...

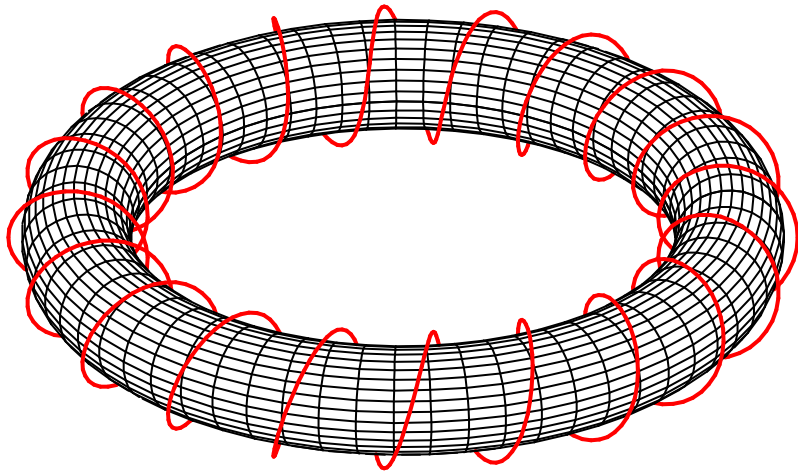
since it has to be spread over several days ...

and students are quick to get the pattern ...

but this does not bother me.

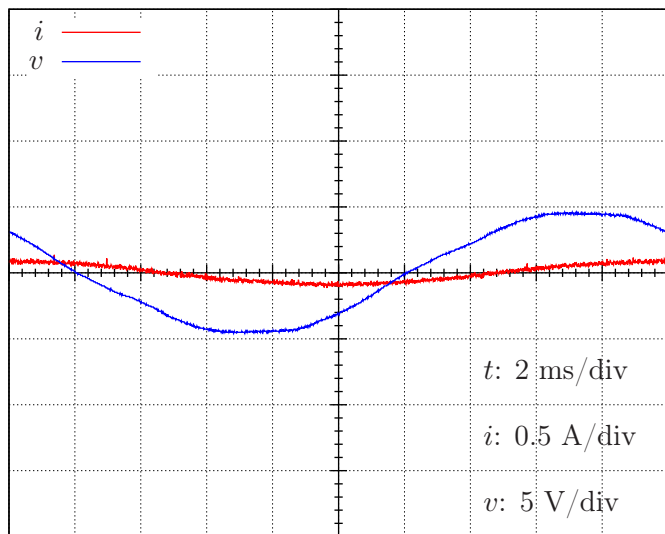
magnetizing curves, hysteresis

test coil



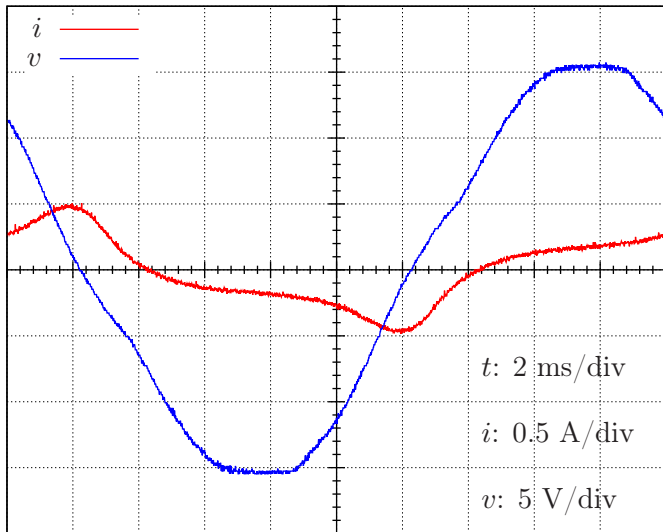
magnetizing curves, hysteresis

voltage and current, # 1



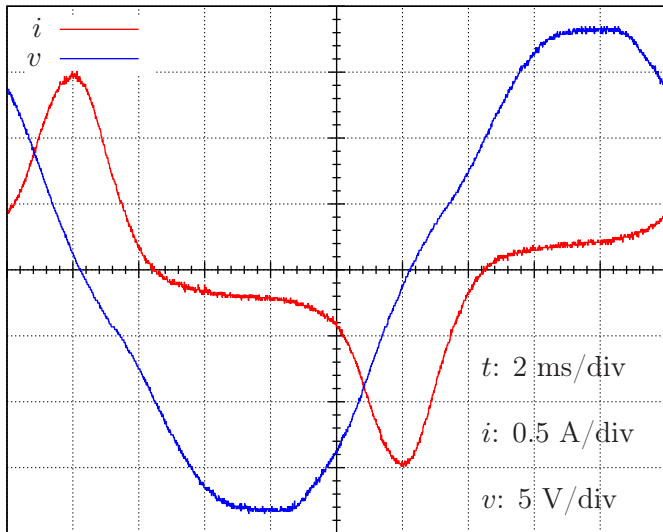
magnetizing curves, hysteresis

voltage and current, # 2



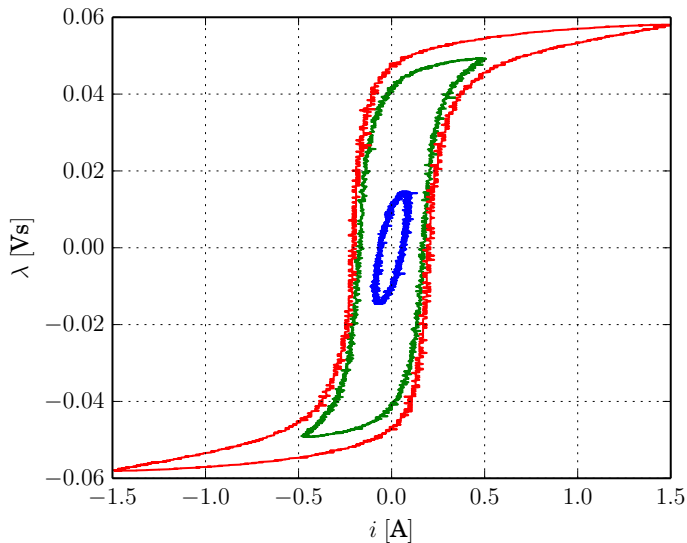
magnetizing curves, hysteresis

voltage and current, # 3



magnetizing curves, hysteresis

magnetizing curves



What Could not be
Covered?

Julia

- ▶ the problem with Python might be speed ...
- ▶ <http://julialang.org/>
- ▶ high-performance numerical and scientific computing
- ▶ performance like statically compiled languages
- ▶ rich libraries, good interfacing with other languages ...
- ▶ built-in package manager
- ▶ ...
- ▶ **worth considering!**

Python versus Julia

a recent research paper, program execution time, circuit simulation, repeated tests, average simulation time ...

	100 kpts		250 kpts		1 Mpts
	Python	Julia	Python	Julia	Julia
computer #1	14.79	5.97	145.04	7.77	18.89
computer #2	47.89	5.12	333.20	16.95	38.89
computer #3	41.98	13.79	339.36	17.66	41.38
computer #4	93.86	15.88	592.46	23.55	70.03
computer #5	776.09	72.59	2256.77	96.74	259.81

SPICE

from <https://en.wikipedia.org/wiki/SPICE>:

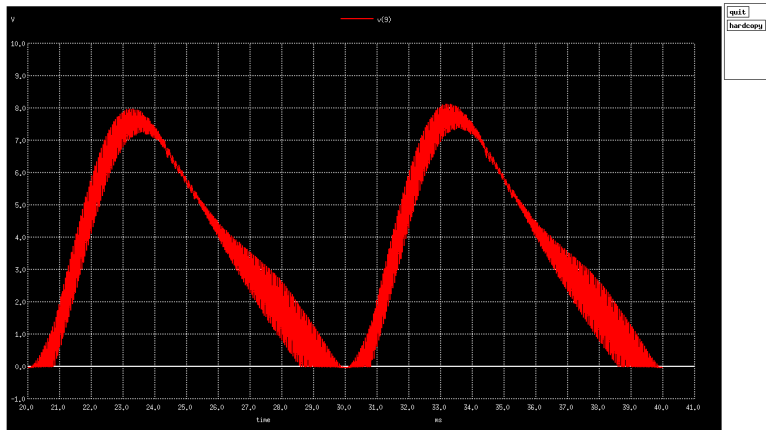
SPICE (Simulation Program with Integrated Circuit Emphasis)[1][2] is a general-purpose, **open source** analog electronic circuit simulator. It is a program used in integrated circuit and board-level design to check the integrity of circuit designs and to predict circuit behavior.

<http://bwrcs.eecs.berkeley.edu/Courses/IcBook/SPICE/>

I use Ngspice, <http://ngspice.sourceforge.net/>, mixed-mode circuit simulator

GUI? Needed?

Ngspice, just an output figure ...



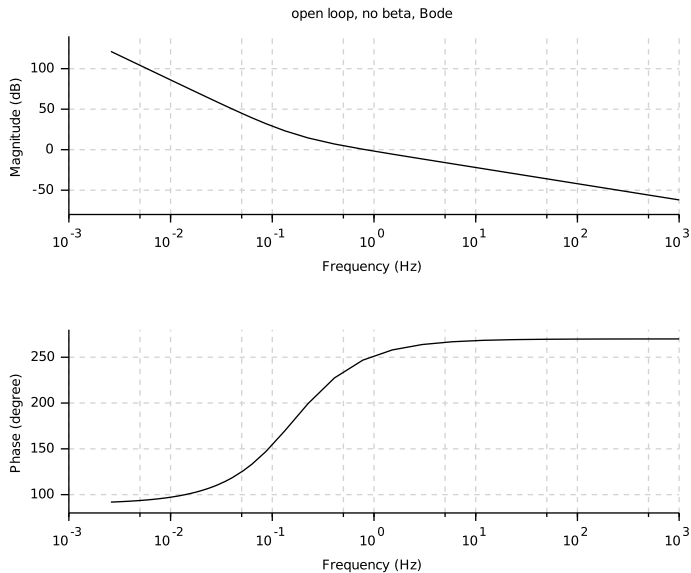
QUCS, another one

- ▶ <http://qucs.sourceforge.net/>
- ▶ completely new simulator
- ▶ pretty much MW oriented, distributed parameters covered
- ▶ very nice GUI
- ▶ worth keeping an eye on ...
- ▶ but there are more: https://en.wikipedia.org/wiki/Comparison_of_EDA_software

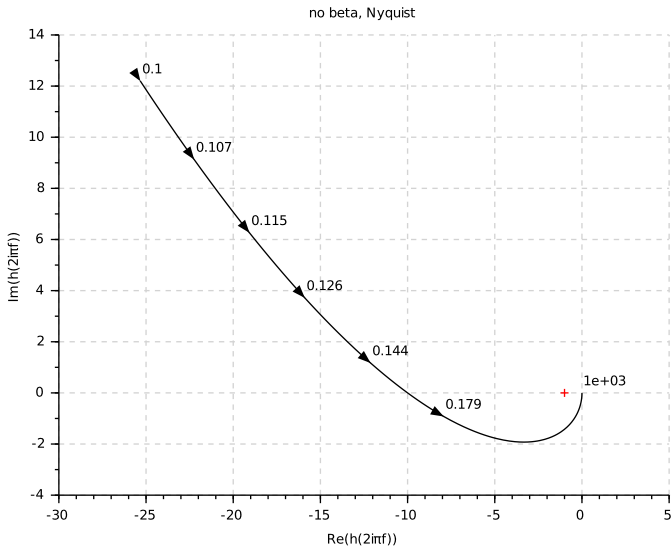
Scilab

- ▶ another numerical computation tool
- ▶ with a good support for signal processing and control theory
- ▶ that is what I use it for
- ▶ has XCOS, GUI for control systems
- ▶ not that I am a fan of GUIs . . .
- ▶ but I use it for control system design

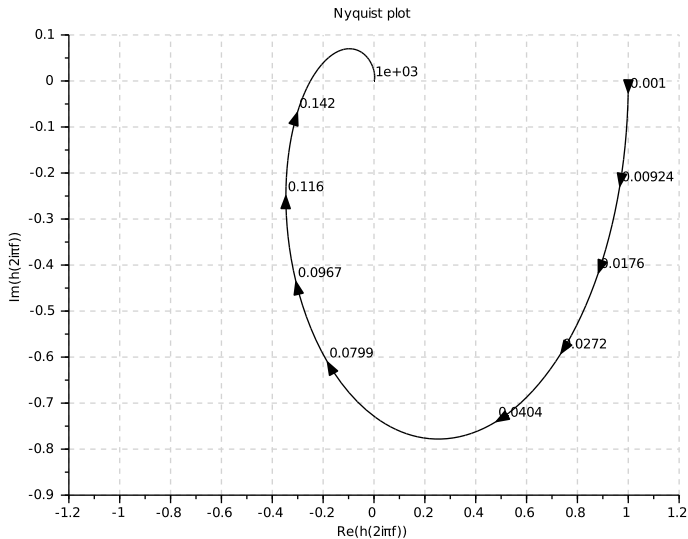
Scilab, figure 1



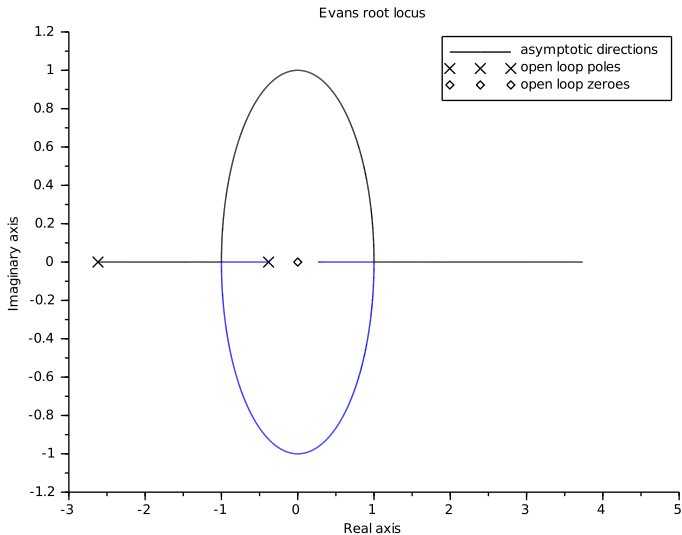
Scilab, figure 2



Scilab, figure 3



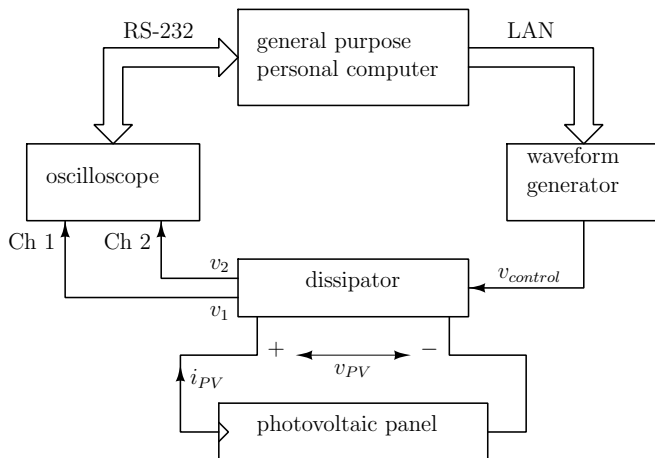
Scilab, figure 4



- ▶ <http://www.sagemath.org/>
- ▶ huge project ...
- ▶ eclectic ...
- ▶ covered with books, [Sage for Undergraduates](#)
- ▶ aims to be comprehensive ...
- ▶ seems to have future ...

Applications in Research

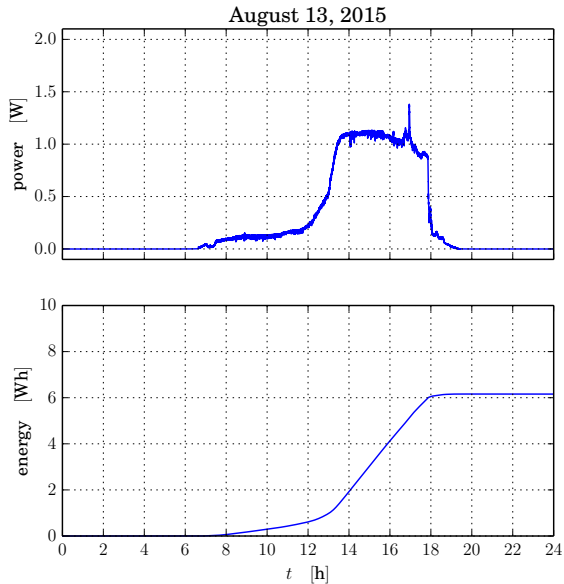
Solar Power Harvester



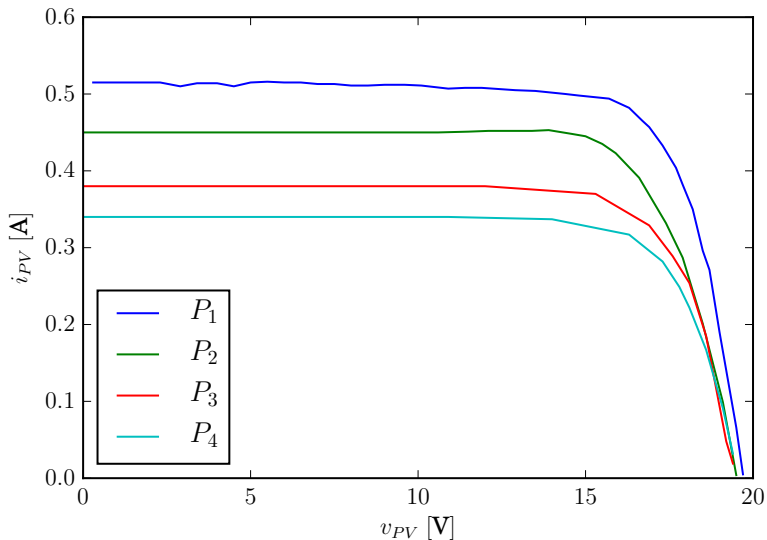
Solar Power Harvester



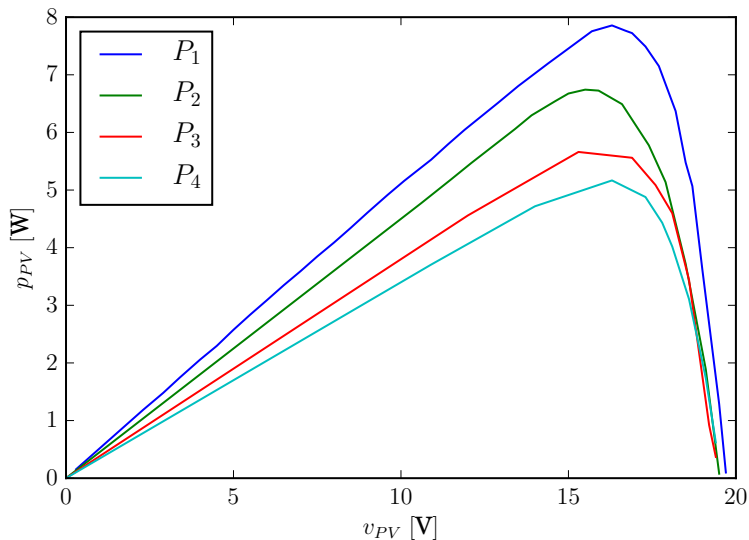
Solar Power Harvester



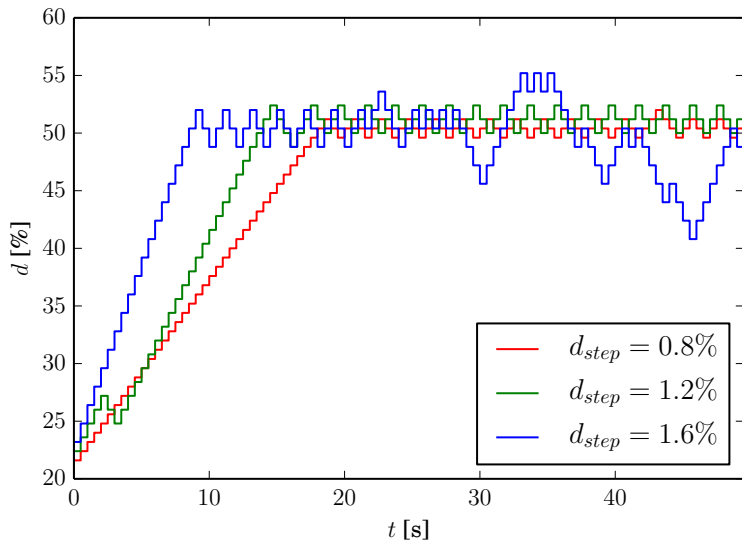
Maximum Power Point Tracking System



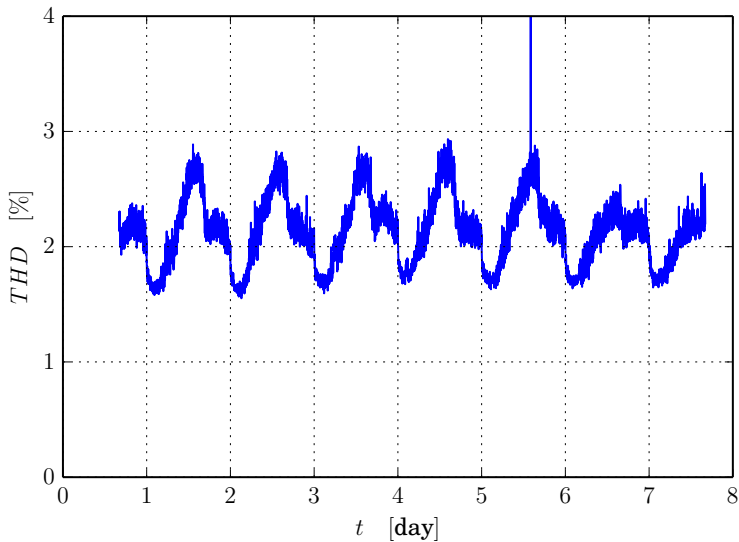
Maximum Power Point Tracking System



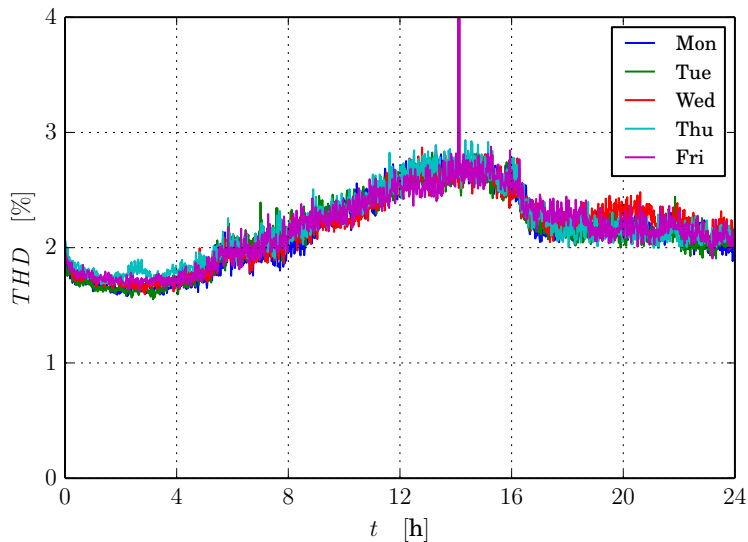
Maximum Power Point Tracking System



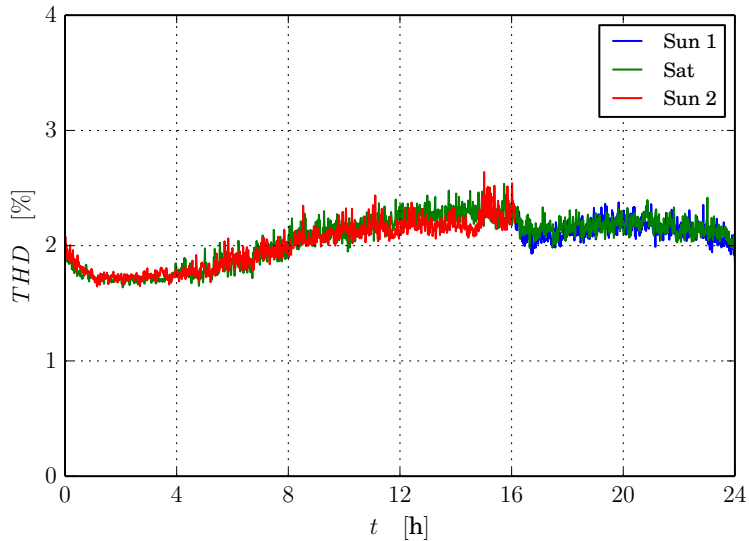
THD Data Logger



THD Data Logger



THD Data Logger



Conclusions

Conclusions

- ▶ motivation and experiences regarding a new course, “practicum” which teaches how to use software tools for:
 1. documenting and presenting, \LaTeX
 2. drawing circuit schematics, $1/28 \approx 3.57\%$
 3. data visualization (gnuplot, matplotlib)
 4. numeric computing (GNU Octave, PyLab)
 5. symbolic computing (wxMaxima, SymPy)
 6. general purpose programming (Python)
- ▶ exclusive application of free software
- ▶ “two tools” policy (Nehru)
- ▶ some tools mentioned which the class cannot cover ...
- ▶ up to this point, without hemlock ...

Conclusions, application

- ▶ free software as applied in:
 1. measurements and documenting (teaching)
 2. creating animation (teaching)
 3. automatic generation of test sets (teaching)
 4. creating this presentation
 5. running the solar harvester
 6. running the MPPT system
 7. running the THD logging system
 8. ...

Conclusions, experiences

- ▶ **it works!**
- ▶ void of licensing problems
- ▶ user generated data stored in open formats
- ▶ “interoperability” of the programs
- ▶ rapid progress!
- ▶ owner controls the computer
- ▶ void of problems with viruses
- ▶ takes some time to adjust and requires some knowledge
- ▶ ... thus, it is important to build good habits at the right time
- ▶ everything I know about free software I learned over the Internet
- ▶ **students master the subject without any problem!**

Conclusions: why did I do this?

- ▶ I use free software
- ▶ it exists
- ▶ I find it useful, as many other people do
- ▶ I did not create it
- ▶ I got it as a gift
- ▶ I believe that the very least thing I should do is to share my experience with you
- ▶ I believe that it is educationally valuable
- ▶ I believe that it carries a message of good will
- ▶ I believe we need that message implemented ...
- ▶ ... at least in software!