Prof. Alexander Knebe
Departamento de Fisica Teorica
Grupo de Astrofisica
Modulo M – 8, Despacho 316

Computacion I

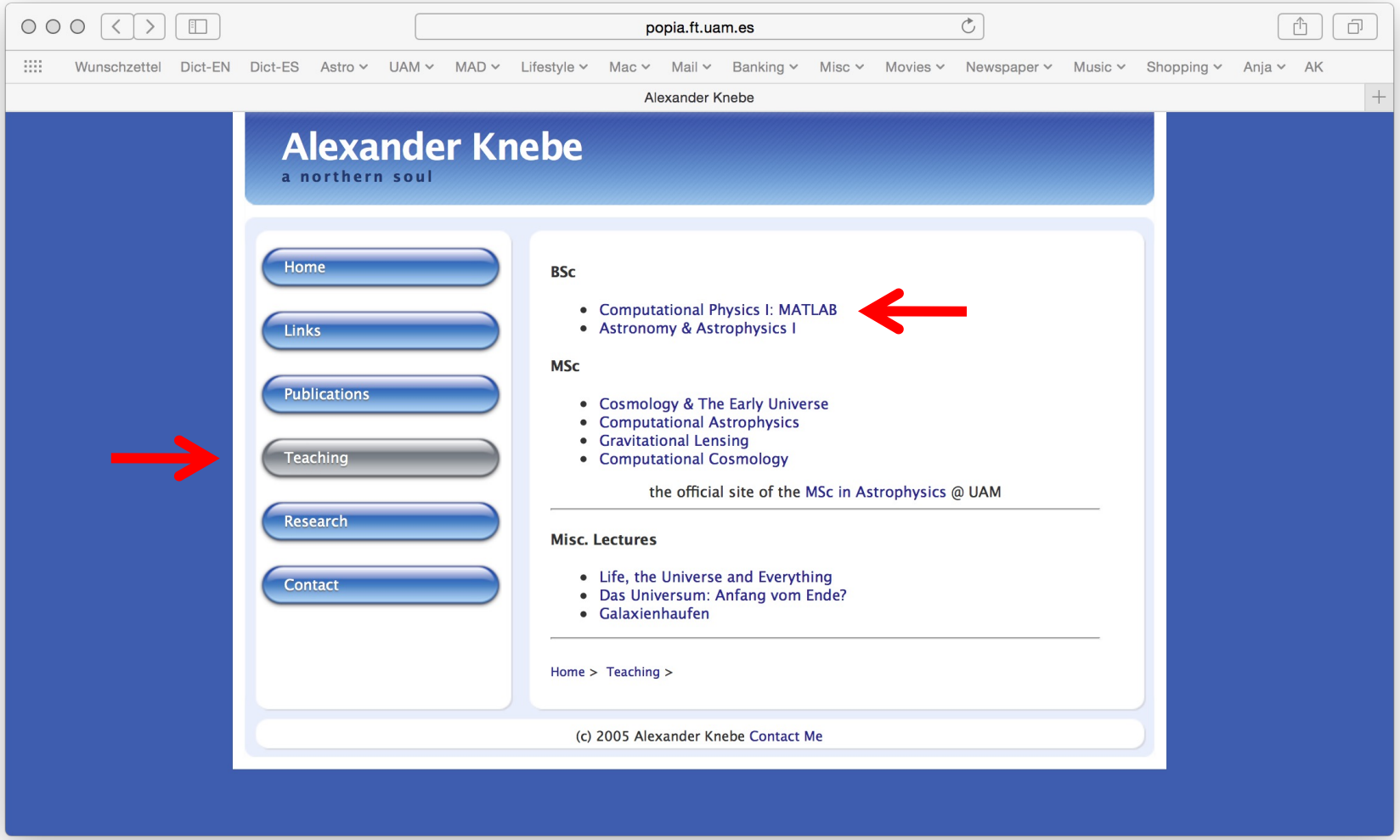
Grupo 516-5

PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

- **hands-on course: learning by doing!**
  - explanation of basic concepts, but
  - working individually on exercises
  - whenever you need help: **ASK**!

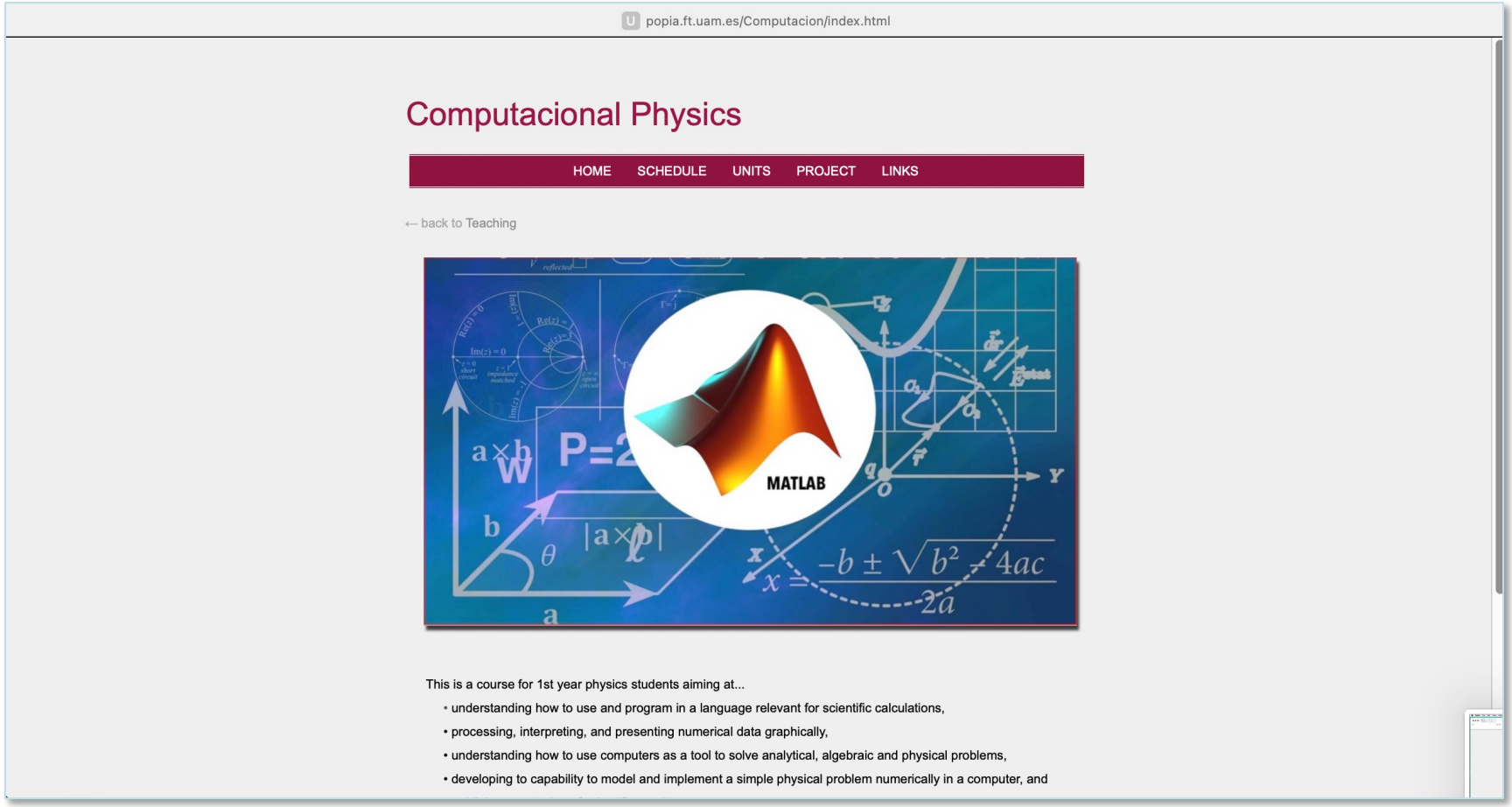
PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe



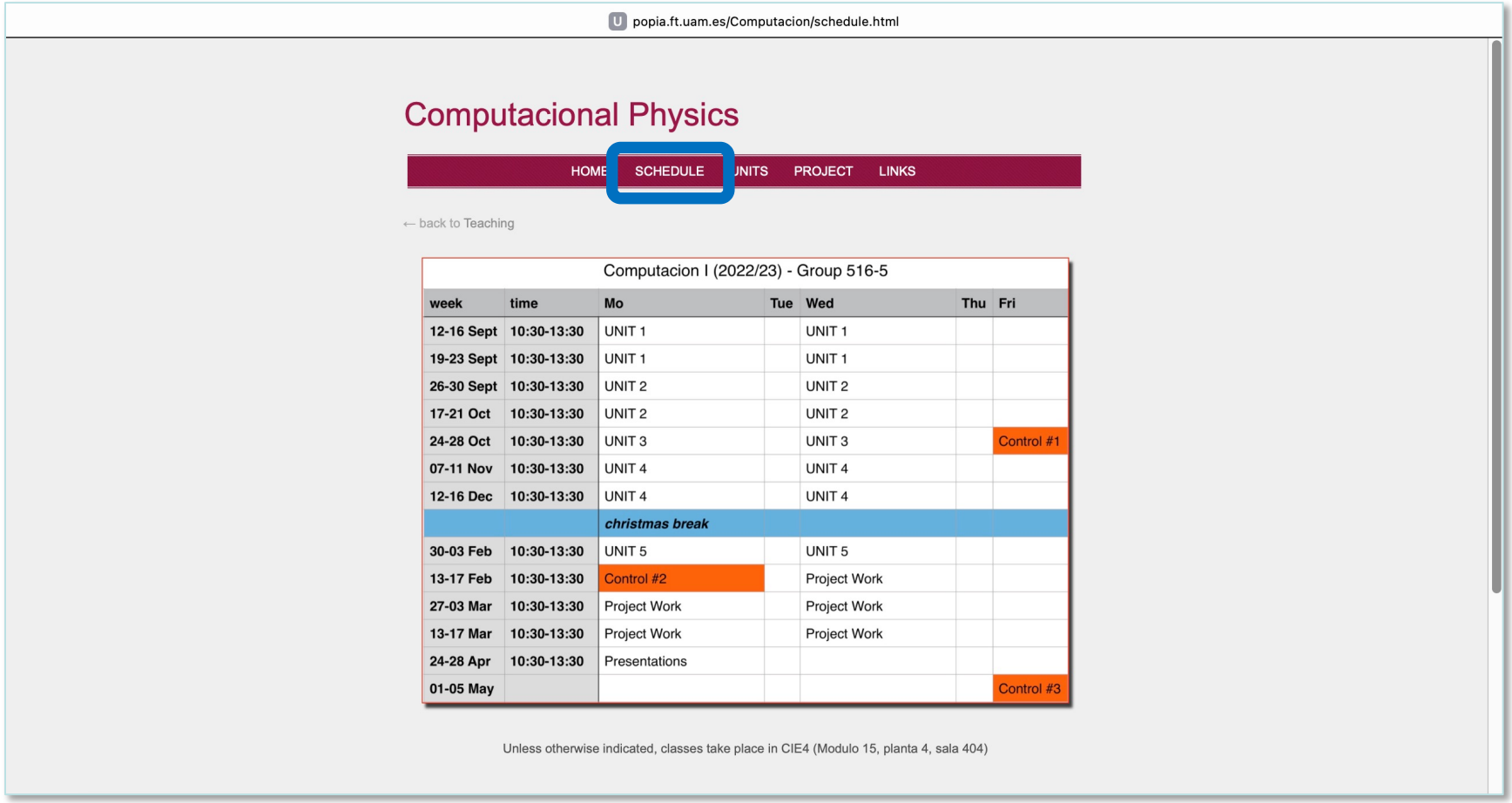
PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe



PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe



popia.ft.uam.es/Computacion/schedule.html

Computacional Physics

HOME | SCHEDULE | UNITS | PROJECT | LINKS

← back to Teaching

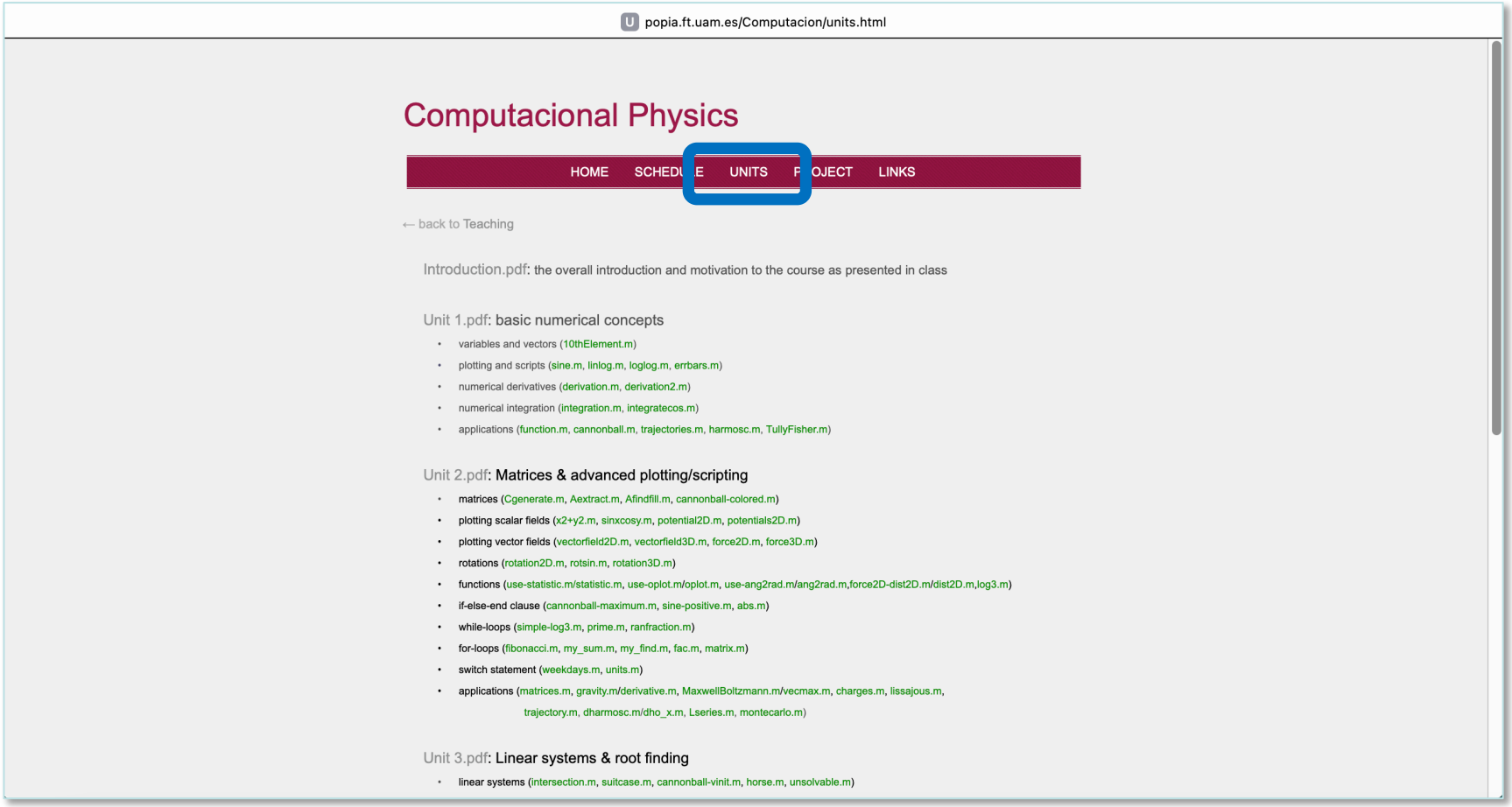
Computacion I (2022/23) - Group 516-5

week	time	Mo	Tue	Wed	Thu	Fri
12-16 Sept	10:30-13:30	UNIT 1		UNIT 1		
19-23 Sept	10:30-13:30	UNIT 1		UNIT 1		
26-30 Sept	10:30-13:30	UNIT 2		UNIT 2		
17-21 Oct	10:30-13:30	UNIT 2		UNIT 2		
24-28 Oct	10:30-13:30	UNIT 3		UNIT 3		Control #1
07-11 Nov	10:30-13:30	UNIT 4		UNIT 4		
12-16 Dec	10:30-13:30	UNIT 4		UNIT 4		
		christmas break				
30-03 Feb	10:30-13:30	UNIT 5		UNIT 5		
13-17 Feb	10:30-13:30	Control #2		Project Work		
27-03 Mar	10:30-13:30	Project Work		Project Work		
13-17 Mar	10:30-13:30	Project Work		Project Work		
24-28 Apr	10:30-13:30	Presentations				
01-05 May						Control #3

Unless otherwise indicated, classes take place in CIE4 (Modulo 15, planta 4, sala 404)

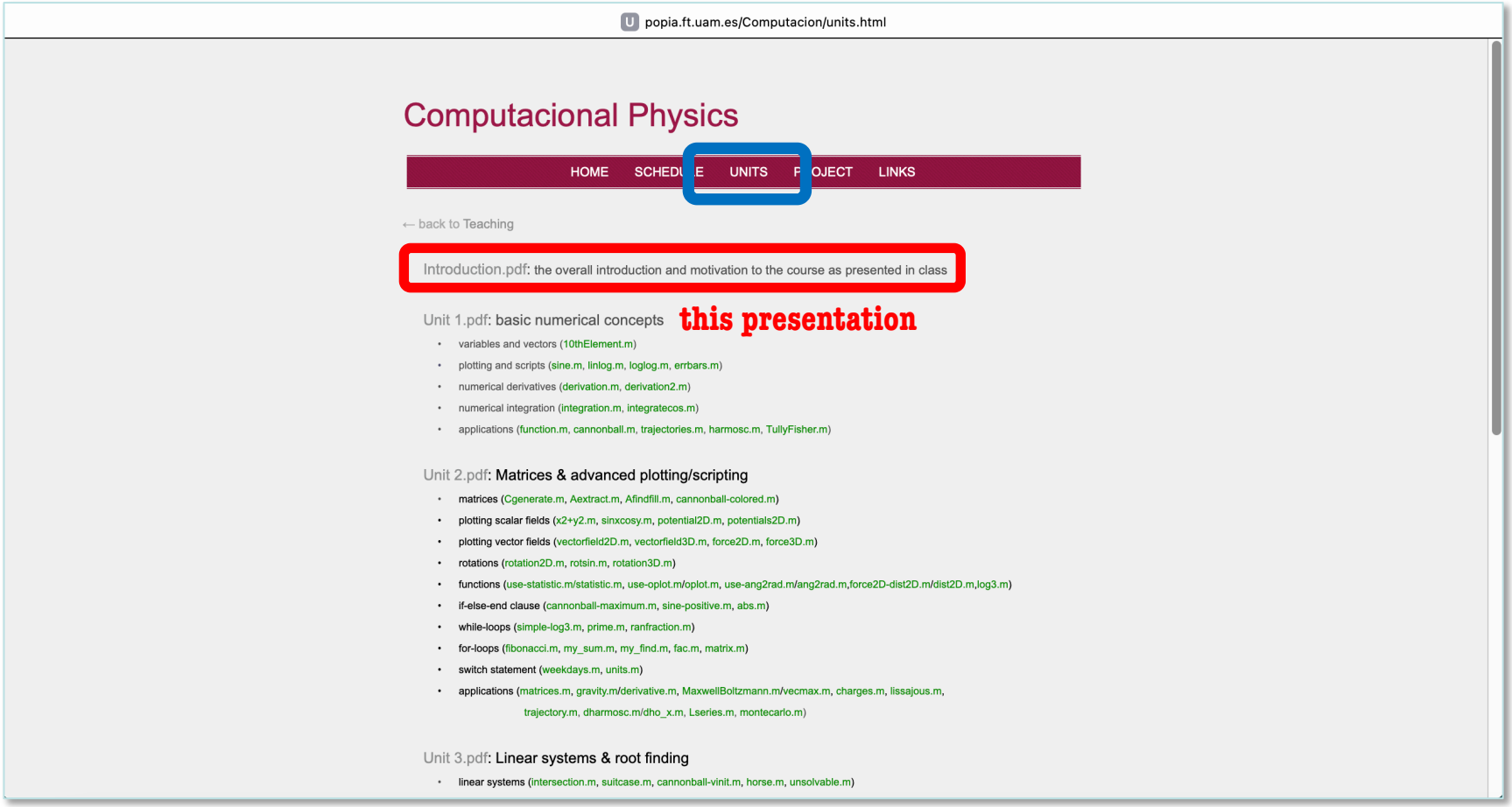
PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe



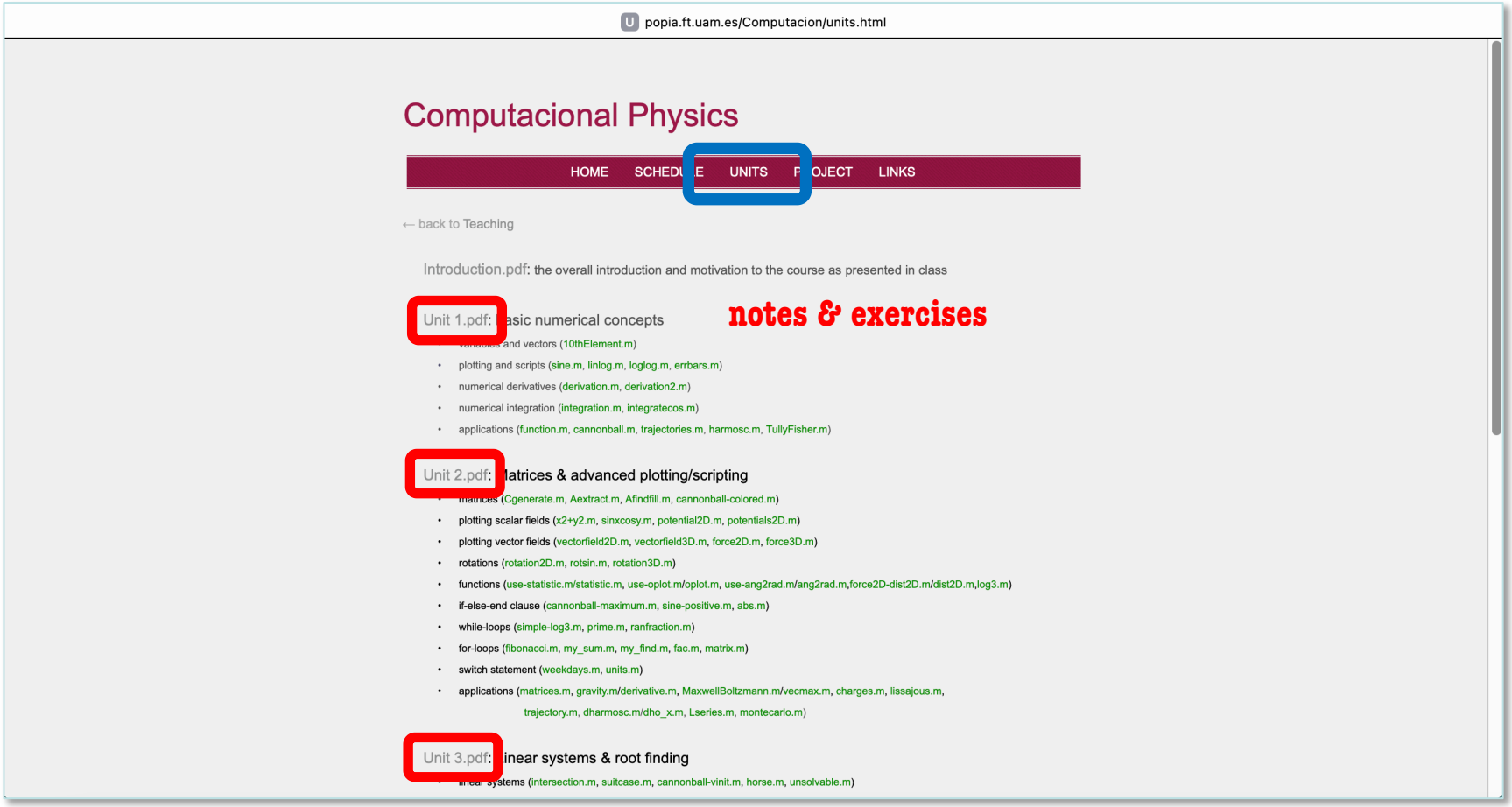
PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe



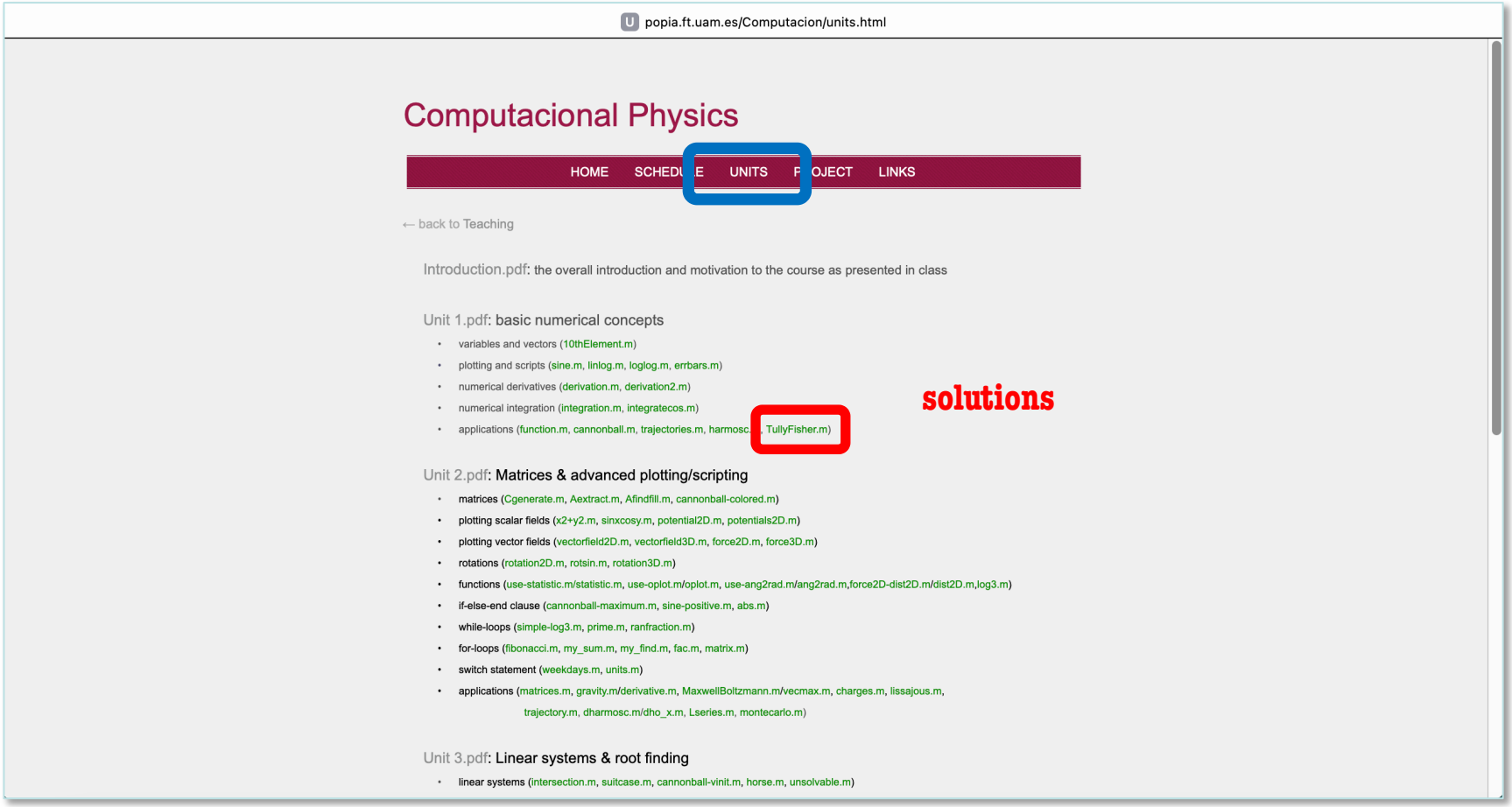
PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe



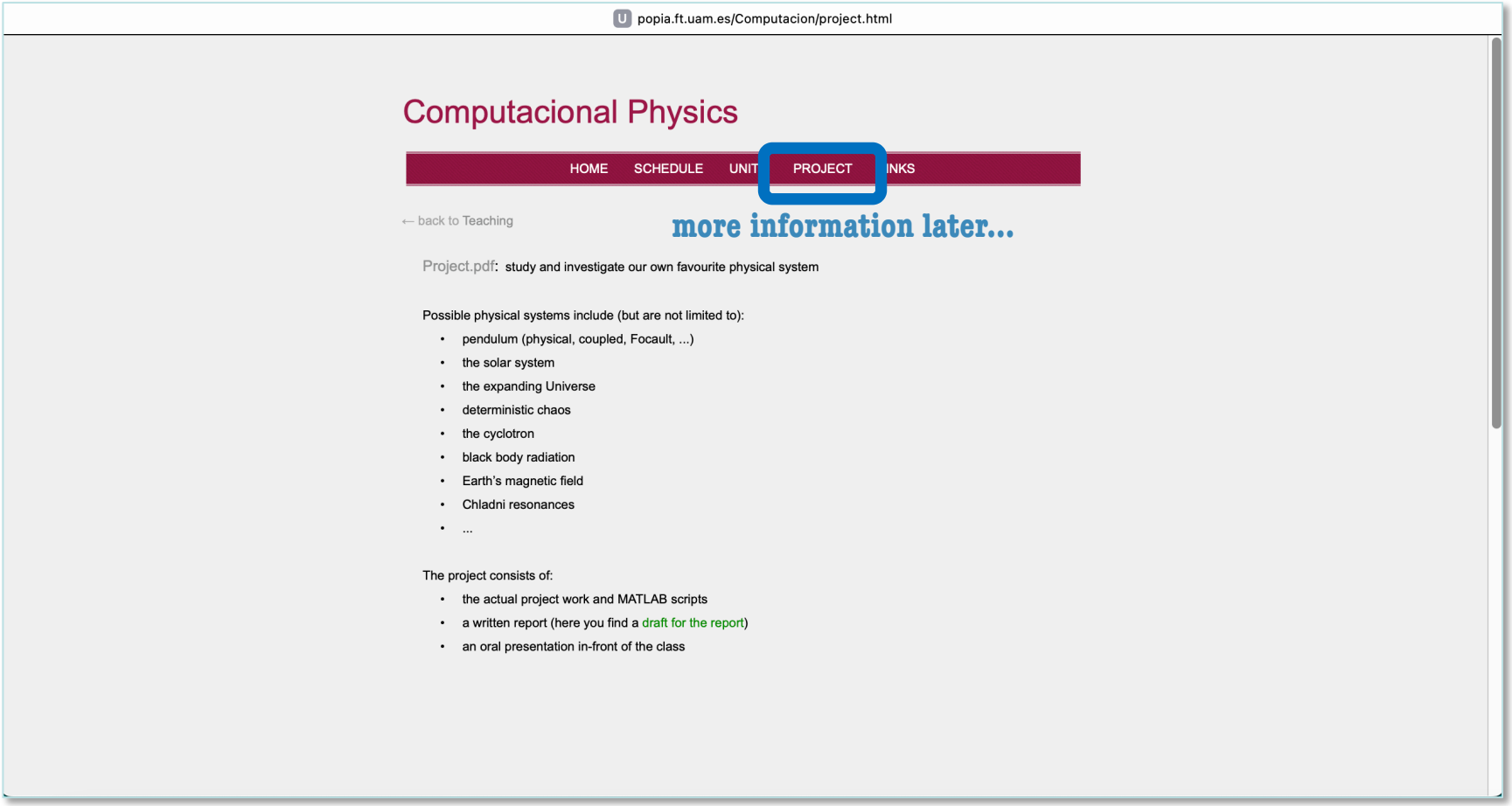
PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe



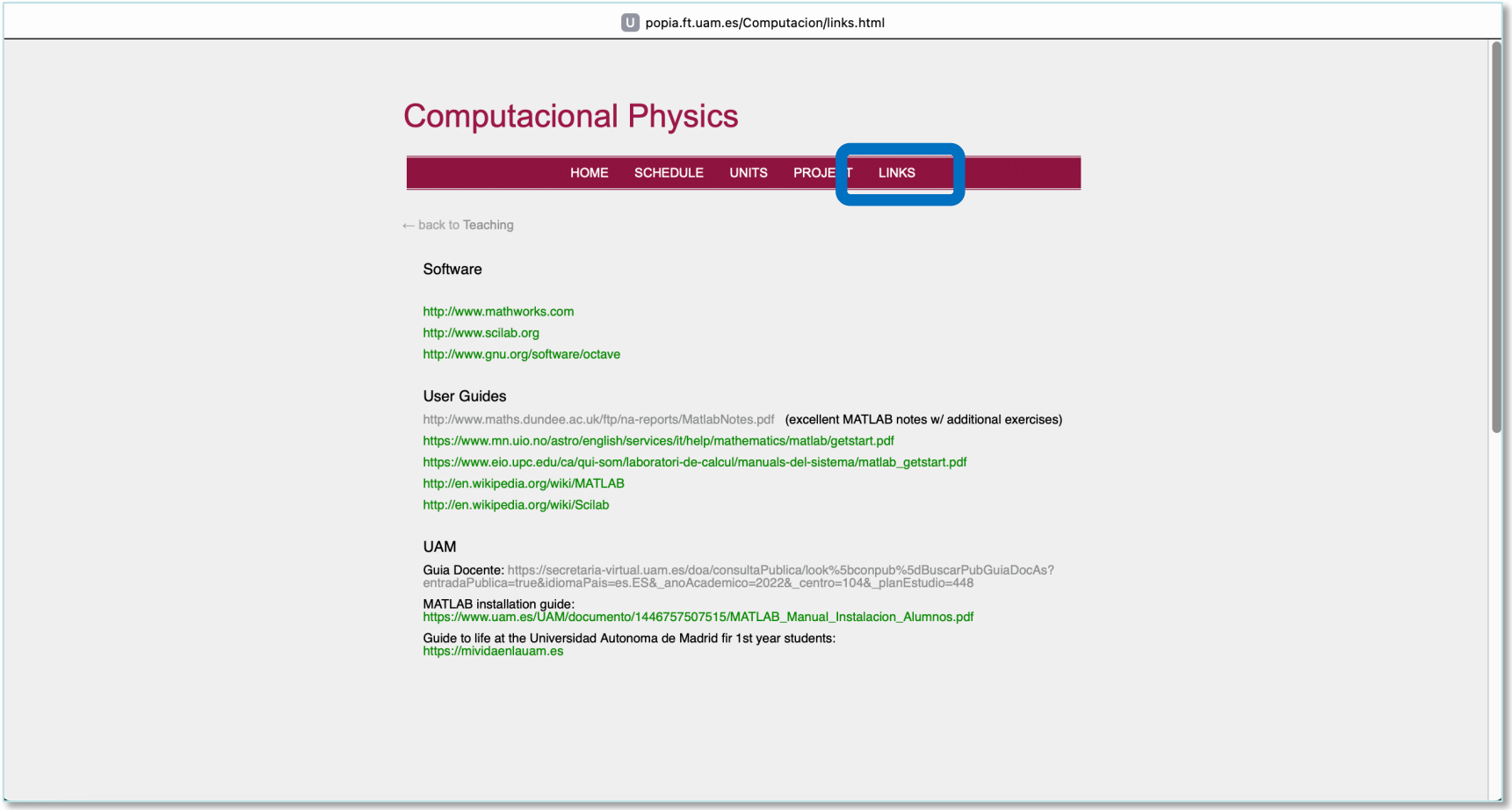
PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe



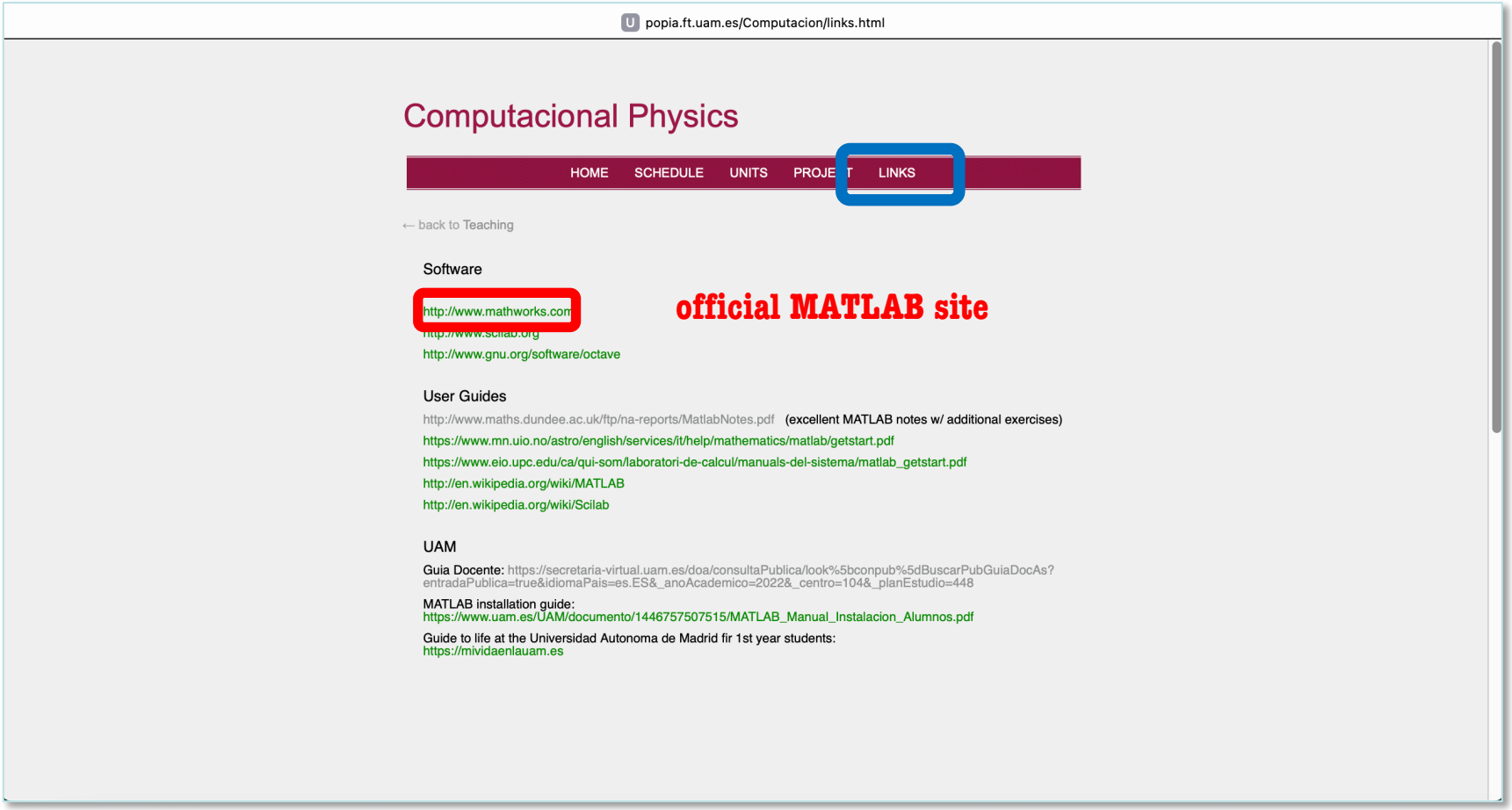
PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe



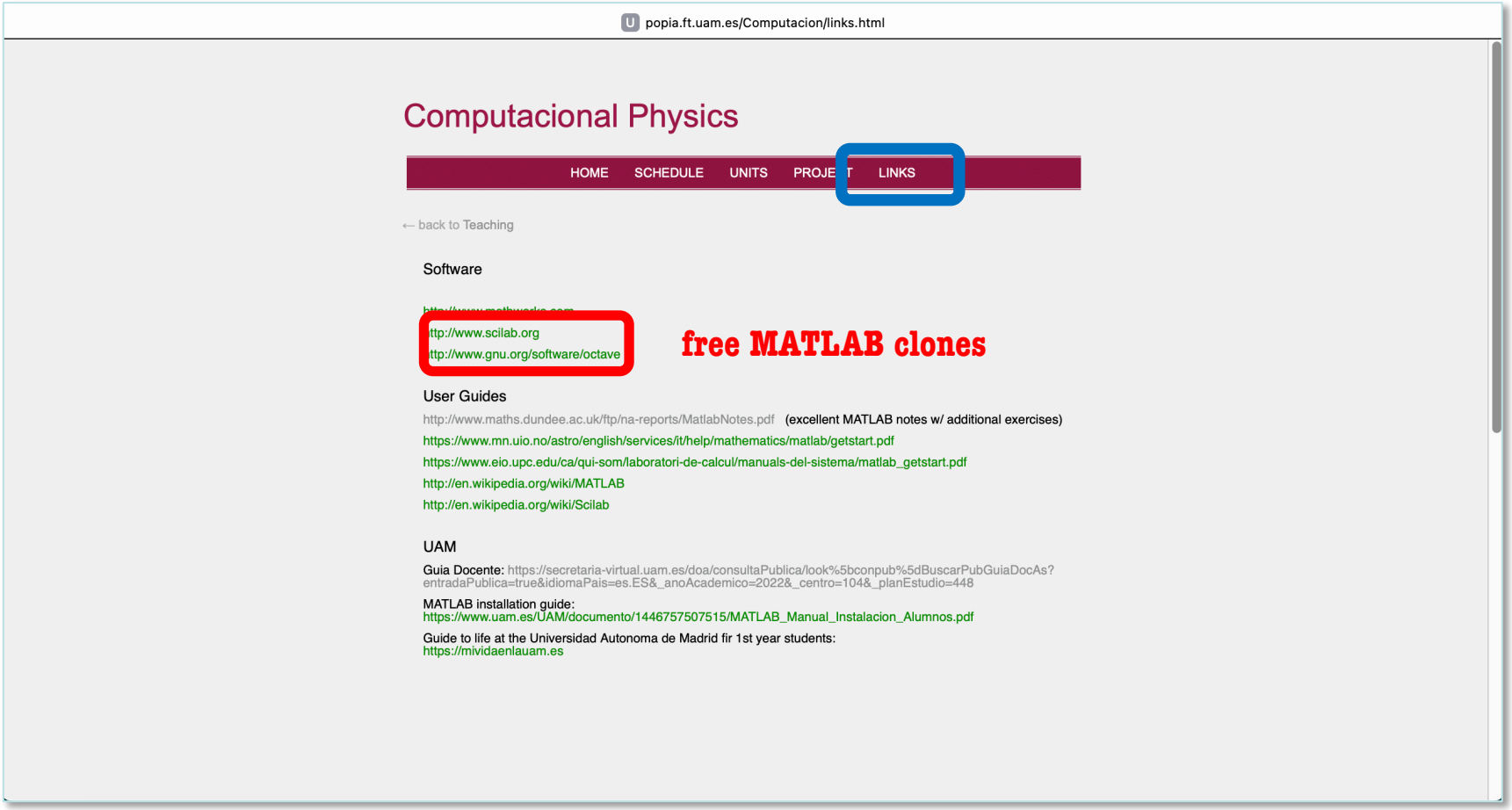
PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe



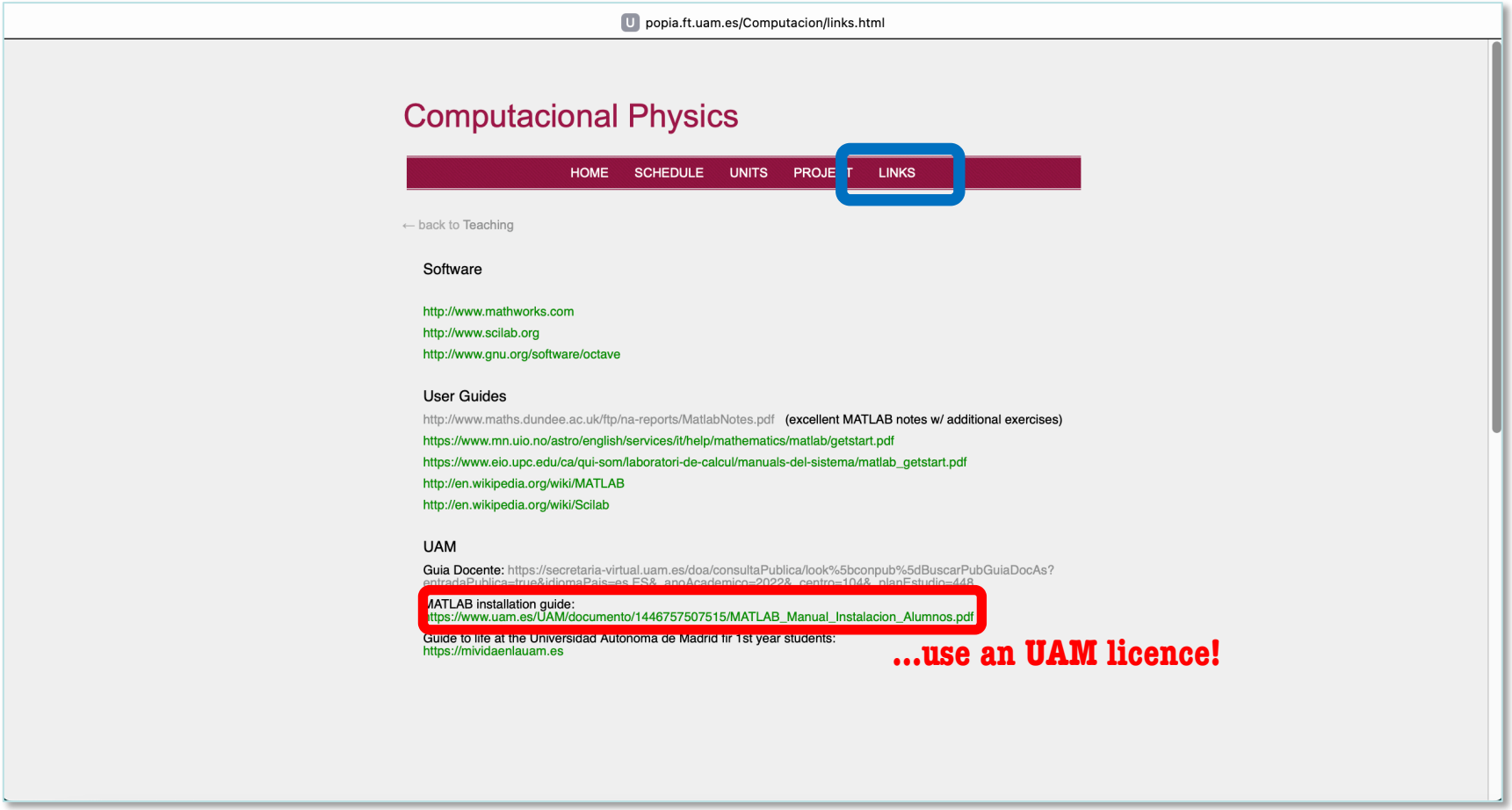
PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe



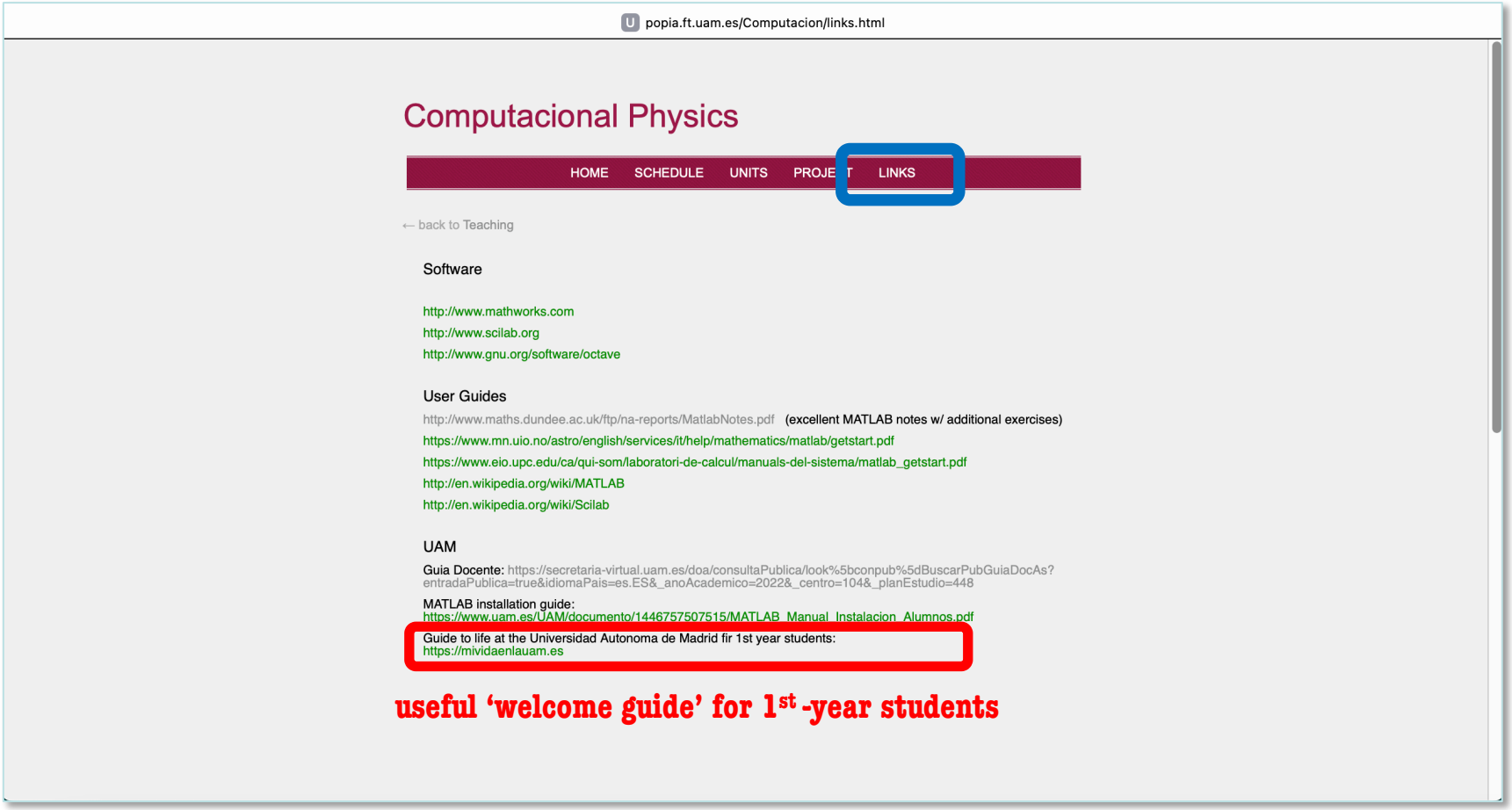
PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

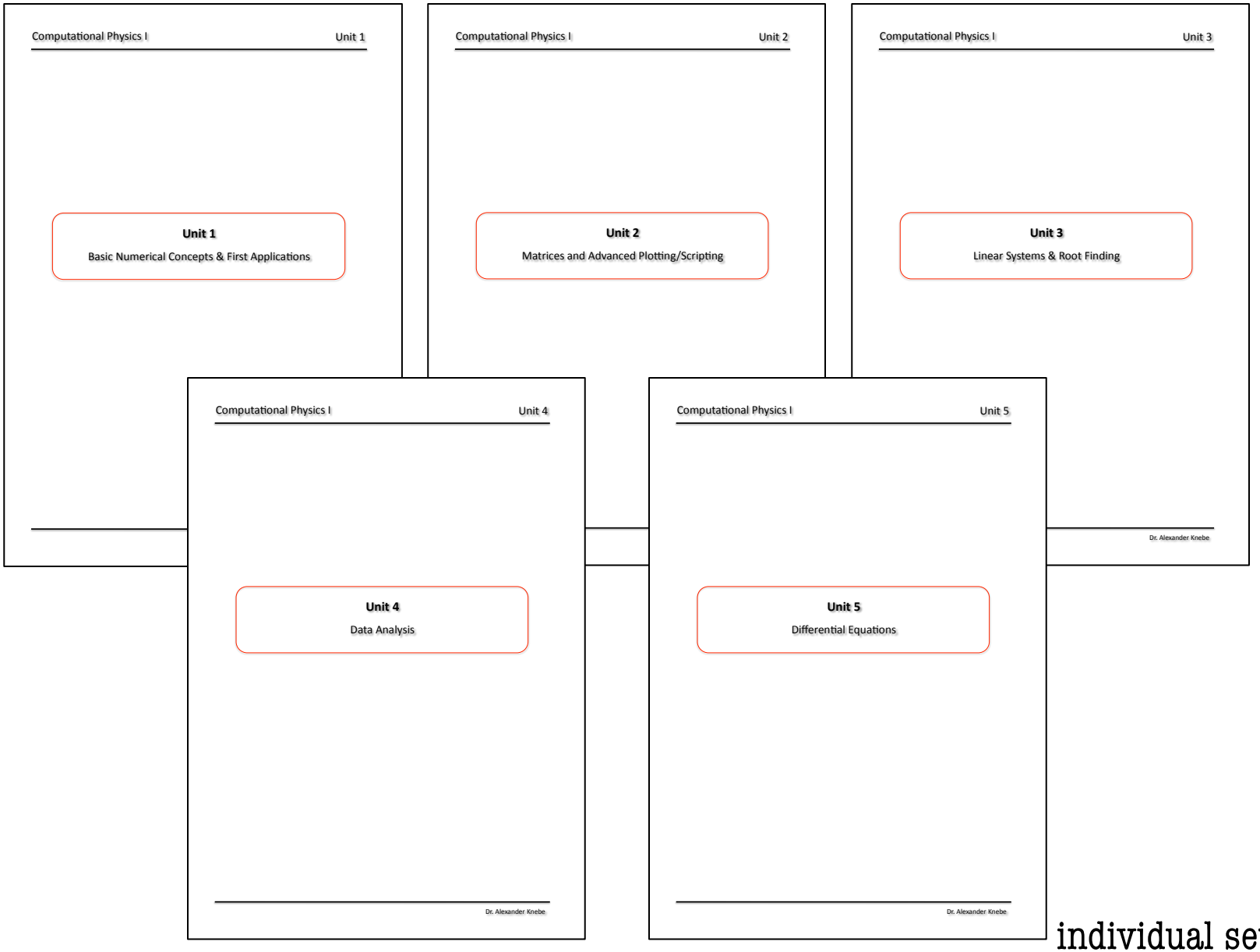
http://popia.ft.uam.es/aknebe



PROF. ALEXANDER KNEBE
DEPARTAMENTO DE FISICA TEORICA
GRUPO DE ASTROFISICA
MODULO M – 8, DESPACHO 316

http://popia.ft.uam.es/aknebe





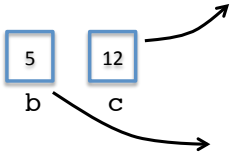
individual sets of exercises!

Computational Physics I — Unit 1

MATLAB — *variables*

- MATLAB can store numbers in variables

```
>> a=4
>> b=5
>> c=12
>> a+b/c
```



a "piece" of computer memory (RAM) will be called "c" and is reserved for storing any number, e.g. "12"

a "piece" of computer memory (RAM) will be called "b" and is reserved for storing any number, e.g. "5"

you can choose whatever name you prefer for variables:
ALWAYS USE MEANINGFUL VARIABLE NAMES!

➢ **exercise:**

- repeat your previous calculations utilizing variables this time…

➢ **exercise:**

- why is
  `>>4=a`
  not working?

- useful commands (continued…)

```
>> clear variable
>> clear all
>> who
>> clc
```

(try `help` on them, i.e. `help clear`)

➢ **exercise:**

- are there any predefined variables in MATLAB?
- what happens when you use MATLAB's variables as your own?
- how can you recover MATLAB's values?

➢ **exercise:**

- define variables that act as conversion factors between degrees and radians to allow for a simple usage of the trigonometric functions with degrees (e.g. call those variables `deg2rad` and `rad2deg`)
- check the precision of the results when subsequently applying, e.g. `sin()` and `asin()`

Day 2 — Dr. Alexander Knebe

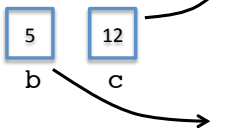
individual sets of exercises!

Computational Physics I Unit 1

MATLAB *variables*

- MATLAB can store numbers in variables

```
>> a=4
>> b=5
>> c=12
>> a+b/c
```



a "piece" of computer memory (RAM) will be called "c" and is reserved for storing any number, e.g. "12"

a "piece" of computer memory (RAM) will be called "b" and is reserved for storing any number, e.g. "5"

you can choose whatever name you prefer for variables:
ALWAYS USE MEANINGFUL VARIABLE NAMES!

➢ **exercise:**

- repeat your previous calculations utilizing variables this time…

➢ **exercise:**

- why is
  `>>4=a`
  not working?

- useful commands (continued…)

```
>> clear variable
>> clear all
>> who
>> clc
```

(try `help` on them, i.e. `help clear`)

➢ **exercise:**

- are there any predefined variables in MATLAB?
- what happens when you use MATLAB's variables as your own?
- how can you recover MATLAB's values?

➢ **exercise:**

- define variables that act as conversion factors between degrees and radians to allow for a simple usage of the trigonometric functions with degrees (e.g. call those variables `deg2rad` and `rad2deg`)
- check the precision of the results when subsequently applying, e.g. `sin()` and `asin()`

Day 2 Dr. Alexander Knebe

explanations:

(mostly) presented in class,
but try the examples yourself, too

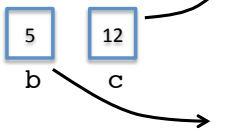
individual sets of exercises!

Computational Physics I — Unit 1

MATLAB — *variables*

- MATLAB can store numbers in variables

```
>> a=4
>> b=5
>> c=12
>> a+b/c
```



a "piece" of computer memory (RAM) will be called "c" and is reserved for storing any number, e.g. "12"

a "piece" of computer memory (RAM) will be called "b" and is reserved for storing any number, e.g. "5"

you can choose whatever name you prefer for variables:
ALWAYS USE MEANINGFUL VARIABLE NAMES!

➢ **exercise:**

- repeat your previous calculations utilizing variables this time…

➢ **exercise:**

- why is
  `>>4=a`
  not working?

- useful commands (continued…)

```
>> clear variable
>> clear all
>> who
>> clc
```

(try `help` on them, i.e. `help clear`)

➢ **exercise:**

- are there any predefined variables in MATLAB?
- what happens when you use MATLAB's variables as your own?
- how can you recover MATLAB's values?

➢ **exercise:**

- define variables that act as conversion factors between degrees and radians to allow for a simple usage of the trigonometric functions with degrees (e.g. call those variables `deg2rad` and `rad2deg`)
- check the precision of the results when subsequently applying, e.g. `sin()` and `asin()`

Day 2 — Dr. Alexander Knebe

exercises:

to be done by each and every one,
but if in trouble, ask!

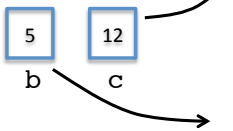
individual sets of exercises!

Computational Physics I — Unit 1

MATLAB — *variables*

- MATLAB can store numbers in variables

```
>> a=4
>> b=5
>> c=12
>> a+b/c
```



a "piece" of computer memory (RAM) will be called "c" and is reserved for storing any number, e.g. "12"

a "piece" of computer memory (RAM) will be called "b" and is reserved for storing any number, e.g. "5"

you can choose whatever name you prefer for variables:
ALWAYS USE MEANINGFUL VARIABLE NAMES!

➢ **exercise:**

- repeat your previous calculations utilizing variables this time…

➢ **exercise:**

- why is
  `>>4=a`
  not working?

- useful commands (continued…)

```
>> clear variable
>> clear all
>> who
>> clc
```

(try `help` on them, i.e. `help clear`)

➢ **exercise:**

- are there any predefined variables in MATLAB?
- what happens when you use MATLAB's variables as your own?
- how can you recover MATLAB's values?

➢ **exercise:**

- define variables that act as conversion factors between degrees and radians to allow for a simple usage of the trigonometric functions with degrees (e.g. call those variables `deg2rad` and `rad2deg`)
- check the precision of the results when subsequently applying, e.g. `sin()` and `asin()`

Day 2 **guideline for schedule!**

Dr. Alexander Knebe

individual sets of exercises!

- Content
- Itinerary
- Motivation
- MATLAB

- **Content**
- Itinerary
- Motivation
- MATLAB

- Objective:
  - to understand how to use and program (in a language relevant for scientific calculations)
  - to process, interpret, and present numerical data graphically
  - to understand how to use computers as a tool to solve physical problems
  - to develop to capability to model and implement a simple physical problems
  - public presentation of scientific results

- Objective:
  - to understand how to use and program (in a language relevant for scientific calculations)
  - to process, interpret, and present numerical data graphically
  - **to understand how to use computers as a tool to solve physical problems**
  - to develop to capability to model and implement a simple physical problems
  - public presentation of scientific results

- Unit 1: Introduction and Basic Concepts
- Unit 2: Matrices and Functions
- Unit 3: Root Finding
- Unit 4: Data Analysis & Statistics, I/O
- Unit 5: Solving Differential Equations

- Project: Physical Systems
  - some examples given/proposed on website...
  - ...but preferably:

your very own project!

- Content
- **Itinerary**
- Motivation
- MATLAB

Computacion I (2022/23) - Group 516-5

week	time	Mo	Tue	Wed	Thu	Fri
12-16 Sept	10:30-13:30	UNIT 1		UNIT 1		
19-23 Sept	10:30-13:30	UNIT 1		UNIT 1		
26-30 Sept	10:30-13:30	UNIT 2		UNIT 2		
17-21 Oct	10:30-13:30	UNIT 2		UNIT 2		
24-28 Oct	10:30-13:30	UNIT 3		UNIT 3		Control #1
07-11 Nov	10:30-13:30	UNIT 4		UNIT 4		
12-16 Dec	10:30-13:30	UNIT 4		UNIT 4		
		christmas break				
30-03 Feb	10:30-13:30	UNIT 5		UNIT 5		
13-17 Feb	10:30-13:30	Control #2		Project Work		
27-03 Mar	10:30-13:30	Project Work		Project Work		
13-17 Mar	10:30-13:30	Project Work		Project Work		
24-28 Apr	10:30-13:30	Presentations				
01-05 May						Control #3

Computacion I (2022/23) - Group 516-5

week	time	Mo	Tue	Wed	Thu	Fri
12-16 Sept	10:30-13:30	UNIT 1		UNIT 1		
19-23 Sept	10:30-13:30	UNIT 1		UNIT 1		
26-30 Sept	10:30-13:30	UNIT 2		UNIT 2		
17-21 Oct	10:30-13:30	UNIT 2		UNIT 2		
24-28 Oct	10:30-13:30	UNIT 3		UNIT 3		Control #1
07-11 Nov	10:30-13:30	UNIT 4		UNIT 4		
12-16 Dec	10:30-13:30	UNIT 4		UNIT 4		
		christmas break				
30-03 Feb	10:30-13:30	UNIT 5		UNIT 5		
13-17 Feb	10:30-13:30	Control #2		Project Work		
27-03 Mar	10:30-13:30	Project Work		Project Work		
13-17 Mar	10:30-13:30	Project Work		Project Work		
24-28 Apr	10:30-13:30	Presentations				
01-05 May						Control #3

written exams (3-4 hours)

Computacion I (2022/23) - Group 516-5

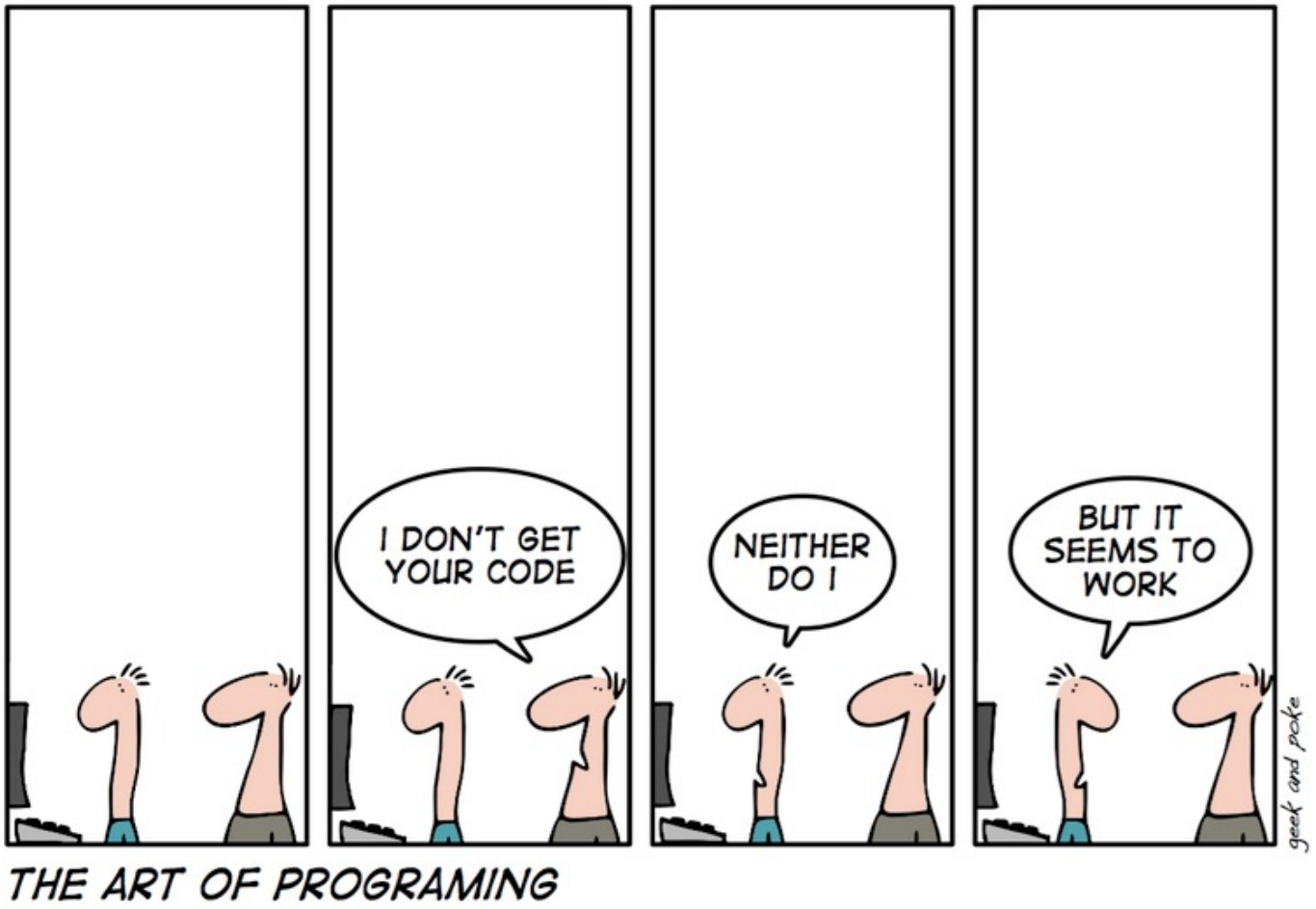
week	time	Mo	Tue	Wed	Thu	Fri
12-16 Sept	10:30-13:30	UNIT 1		UNIT 1		
19-23 Sept	10:30-13:30	UNIT 1		UNIT 1		
26-30 Sept	10:30-13:30	UNIT 2		UNIT 2		
17-21 Oct	10:30-13:30	UNIT 2		UNIT 2		
24-28 Oct	10:30-13:30	UNIT 3		UNIT 3		Control #1
07-11 Nov	10:30-13:30	UNIT 4		UNIT 4		
12-16 Dec	10:30-13:30	UNIT 4		UNIT 4		
		christmas break				
30-03 Feb	10:30-13:30	UNIT 5		UNIT 5		
13-17 Feb	10:30-13:30	Control #2		Project Work		
27-03 Mar	10:30-13:30	Project Work		Project Work		
13-17 Mar	10:30-13:30	Project Work		Project Work		
24-28 Apr	10:30-13:30	Presentations				
01-05 May						Control #3

studying a physical system, incl.
- written report, and
- oral presentation

- evaluation

Control #1	10%	
Control #2	20%	
Control #3	30%	
Project	40%	(programs, written report, oral presentation)

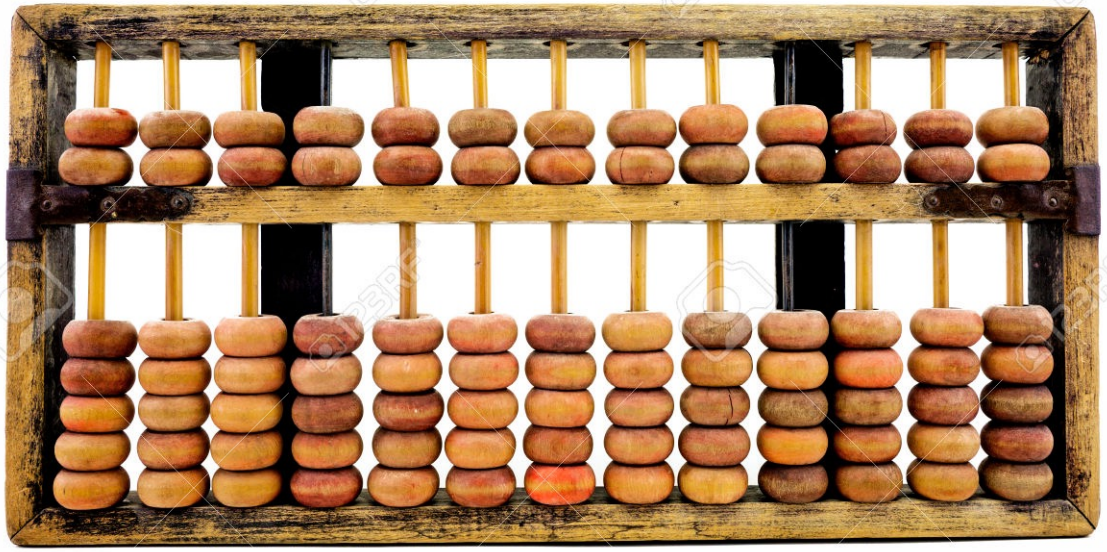
- Content
- Itinerary
- **Motivation**
- MATLAB


I DON'T GET YOUR CODE
NEITHER DO I
BUT IT SEEMS TO WORK
geek and poke
THE ART OF PROGRAMING

how and when did it all begin?

Eurasia

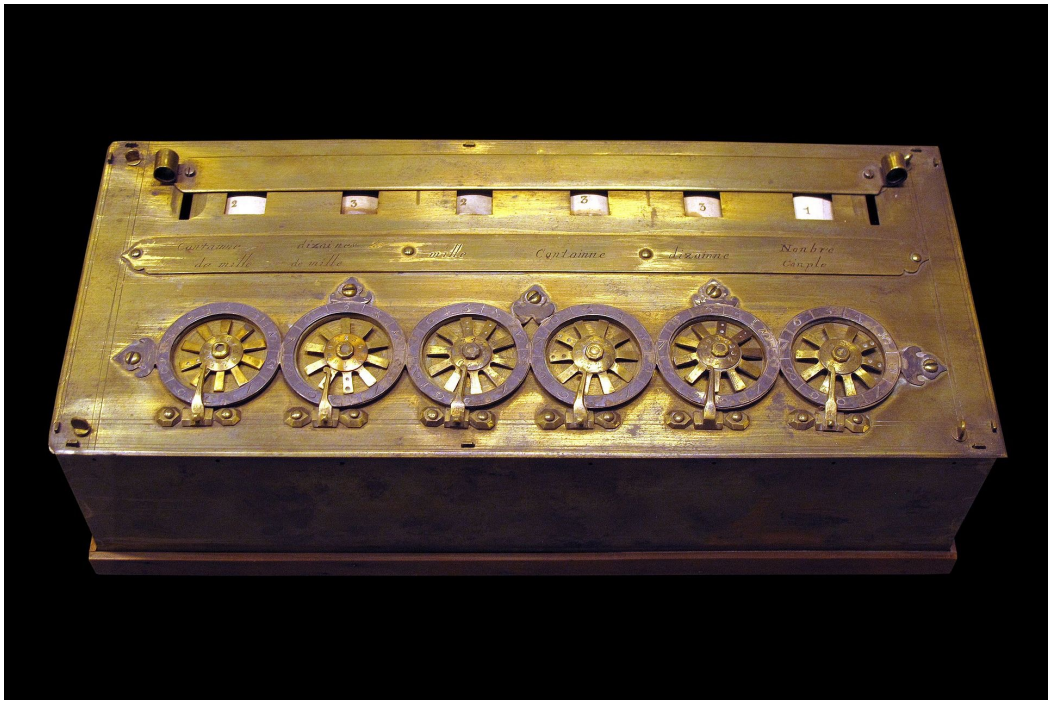
- suitable for any base system
- developed all across the Eurasian continent thousands of years BC

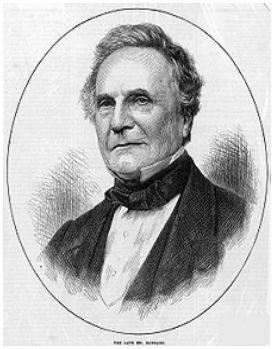


Abacus

Blaise Pascal

- mechanical calculator
- addition/subtraction
- multiplication/division by successive addition/subtraction
- 5 digit accuracy

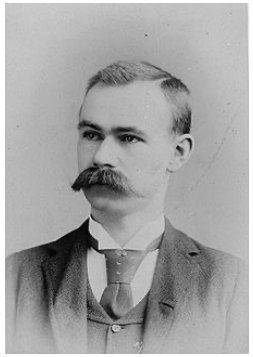




Charles Babbage

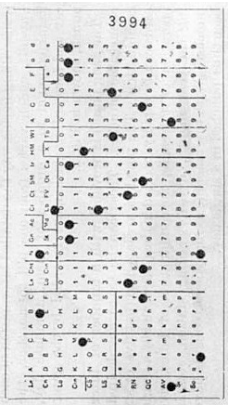
- 4000 parts
- 3 tons
- 3m x 2 m
- 31 digits accuracy
- steam driven
- “difference engine”:
  - designed to tabulate polynomials
  - calculates 2nd order differences

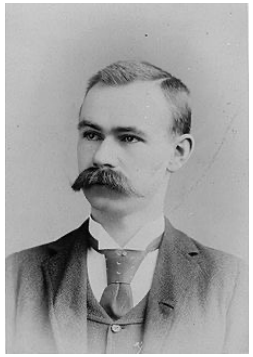
Part of one of Babbage's Difference Engines



Herman Hollerith

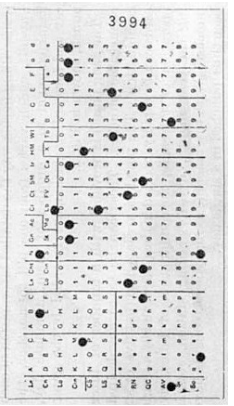
- developed a machine to read punch cards
- machines was used for 1890 census in the USA
- census took only one year to evaluate (prev. 8 years!)
- company name: Computing Tabulating Recording Corporation

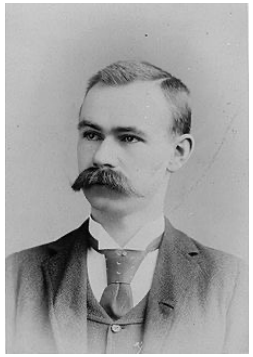




Herman Hollerith

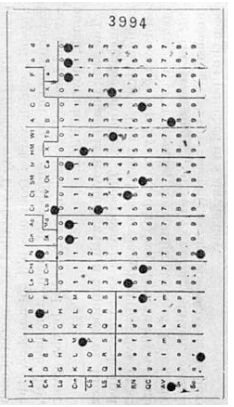
- developed a machine to read punch cards
- machines was used for 1890 census in the USA
- census took only one year to evaluate (prev. 8 years!)
- company name: Computing Tabulating Recording Corporation
- his company was renamed in 1924 to ...





Herman Hollerith

- developed a machine to read punch cards
- machines was used for 1890 census in the USA
- census took only one year to evaluate (prev. 8 years!)
- company name: Computing Tabulating Recording Corporation
- his company was renamed in 1924 to **International Business Machines (IBM)**



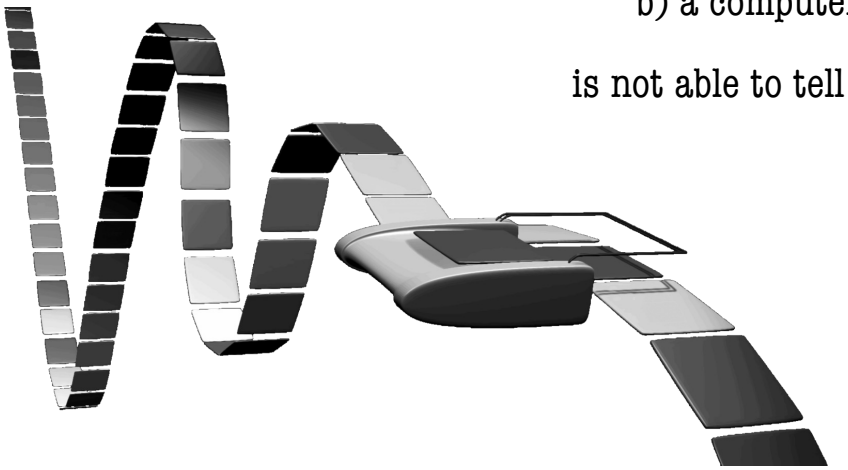
Alan Turing

- Turing Machine:
  - theoretical concept only:
    - reading instructions from printed symbols on a tape
    - result printed on back of tape
  - foundation for theories about computing
- Turing Test:
  - if a human interacting with

    a) another human

    b) a computer

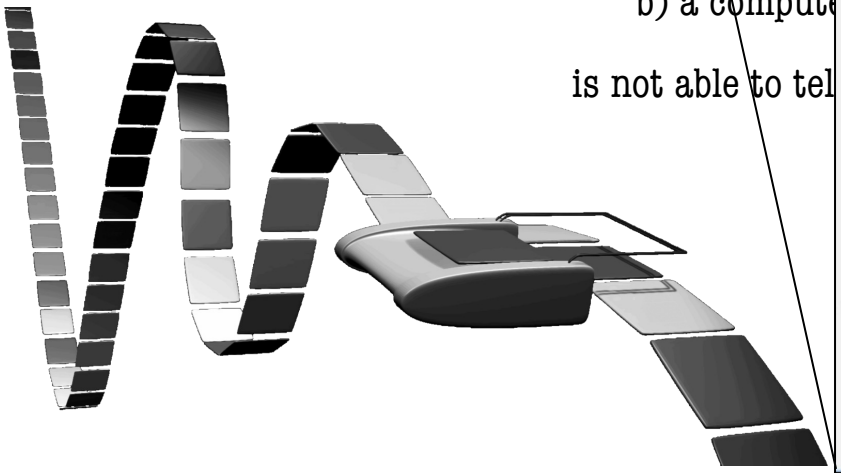
    is not able to tell the difference, the computer is said to “think”

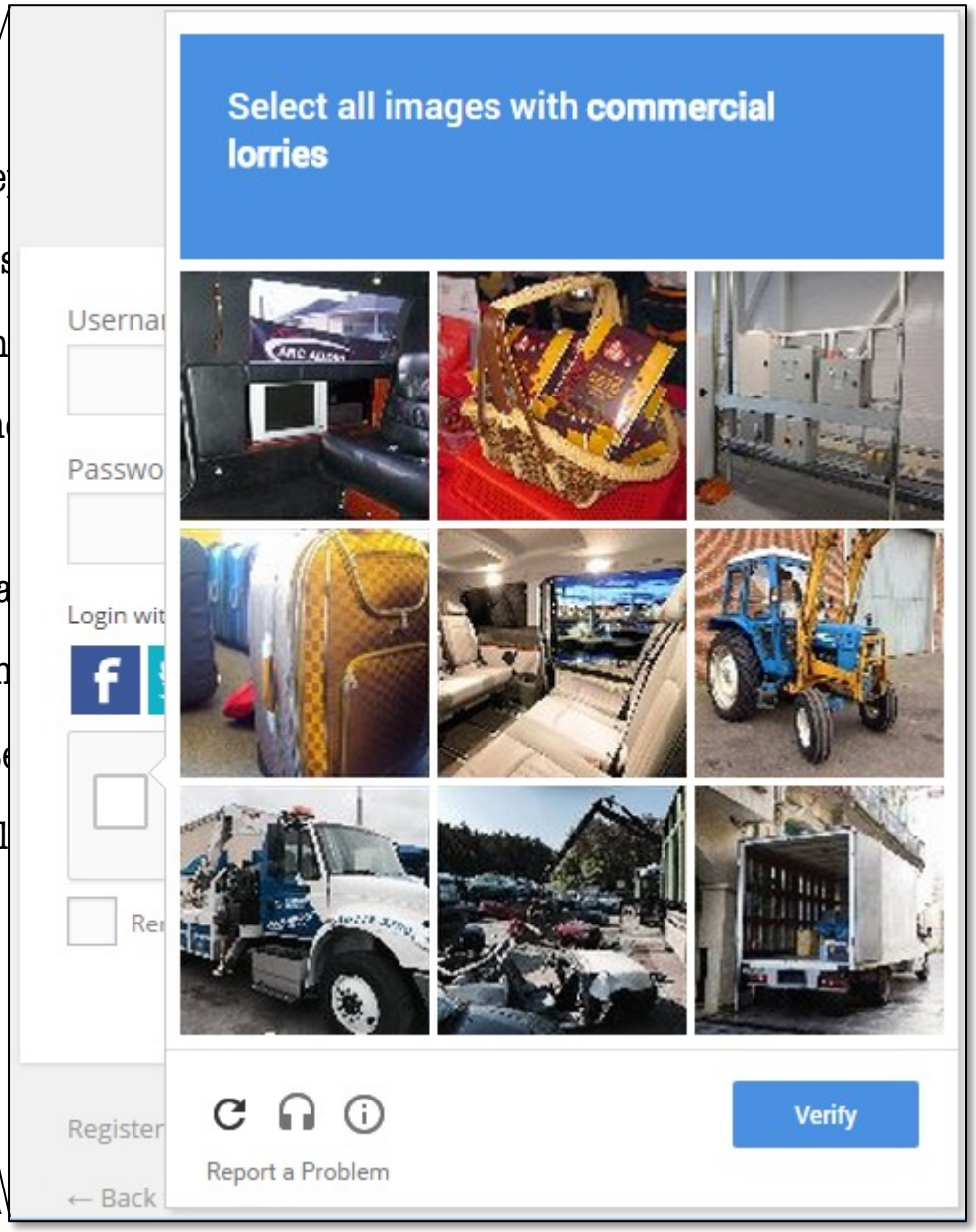


Alan Turing

- Turing Machine:
  - theoretical conce
    - reading ins
    - result prin
  - foundation for th
- **Turing Test**:
  - if a human intera
    - a) another h
    - b) a compute

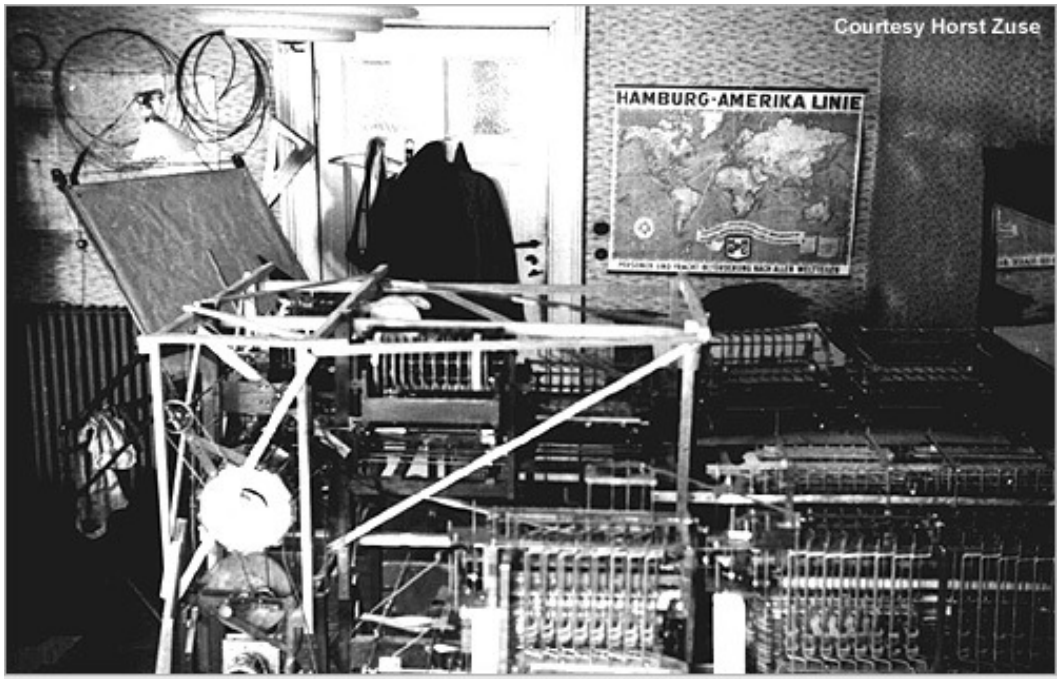
is not able to tel





Konrad Zuse

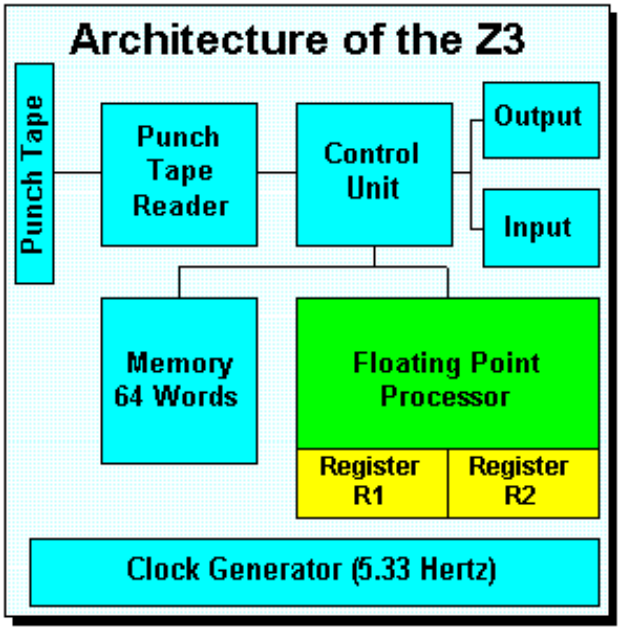
- Z1:
  - 30000 parts
  - calculations were performed in binary (while input and displayed in decimal...)
  - floating point operations
  - freely programmable via punch cards

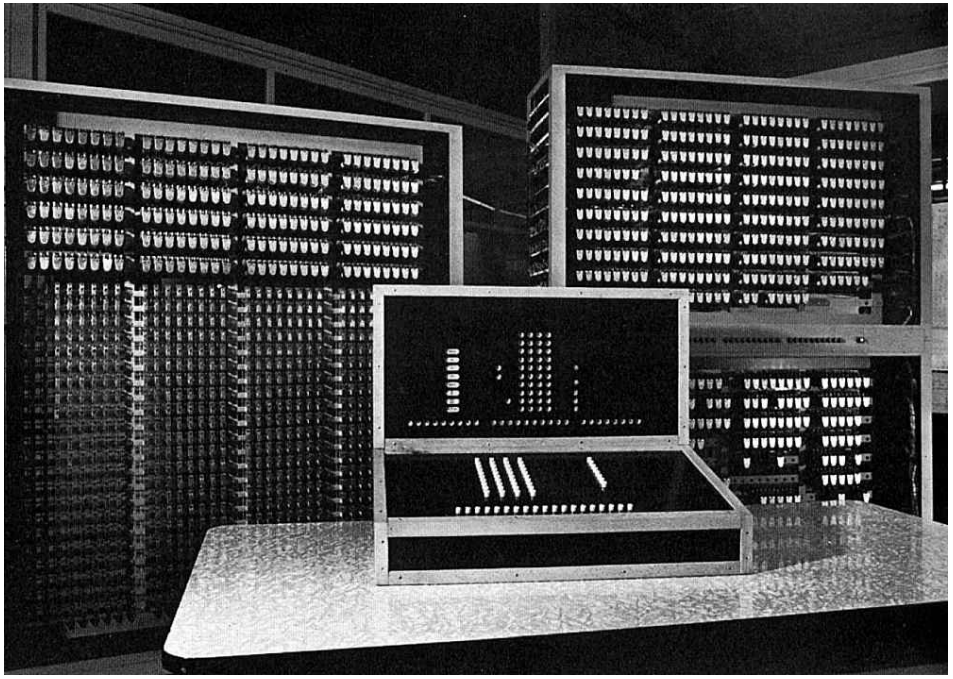


The original Z1 in Konrad's parent's living room circa 1938

Konrad Zuse

- Z3:
  - 5.3 Hz, 22bit, 176 bytes memory, 2600 relays
  - speed: 0.8sec/+ and 3sec/* => 0.3 flop/s
  - floating point operations
  - freely programmable via punch cards





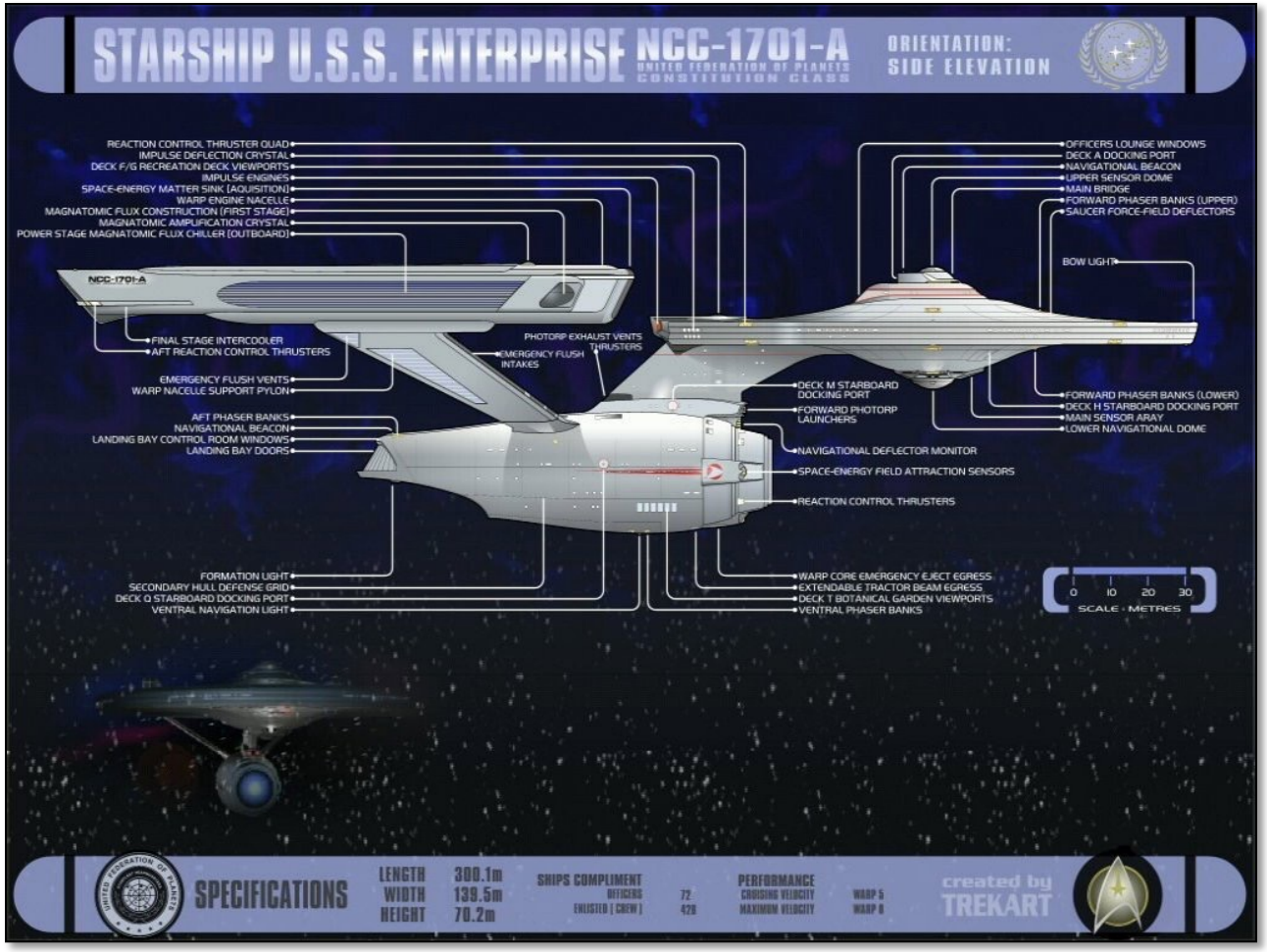
...and what about today's computers?

- Fugaku “Mount Fuji” (Japan), #1 in 11/2020:
  - 7,630,848 cores, 5000TB RAM
  - speed: 440 x 10^{15} flop/s (= 440 PetaFlops)
  - freely programmable (not via punch cards...)
  - operation system: Linux (Red Hat)

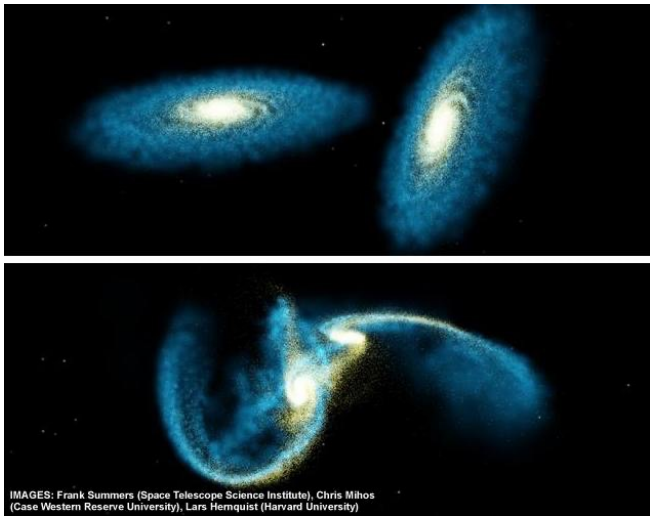
RIKEN Center for Computational Science in Kobe

- MareNostrum (Spain), #42 in 11/2020:
  - 153.216 cores, 41TB RAM
  - speed: 7 Pflop/s
  - operating system: Linux (Suse)

Torre Girona Chapel, Barcelona

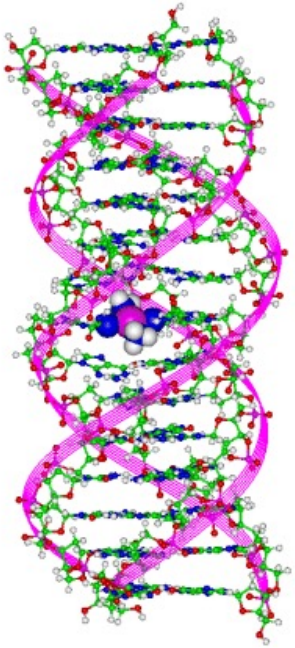


...but what about the science?

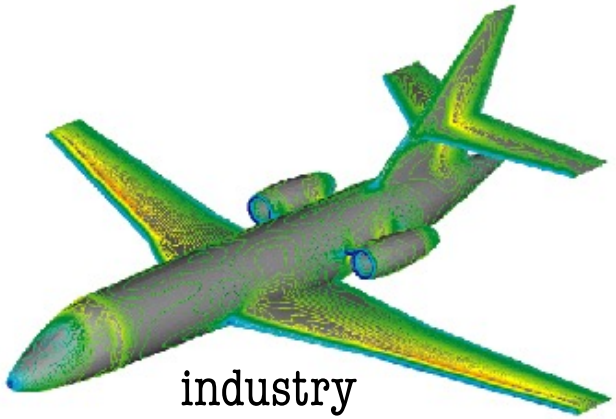


astrophysics

biology

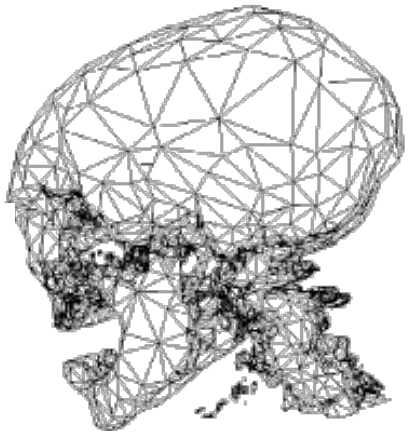


stock market

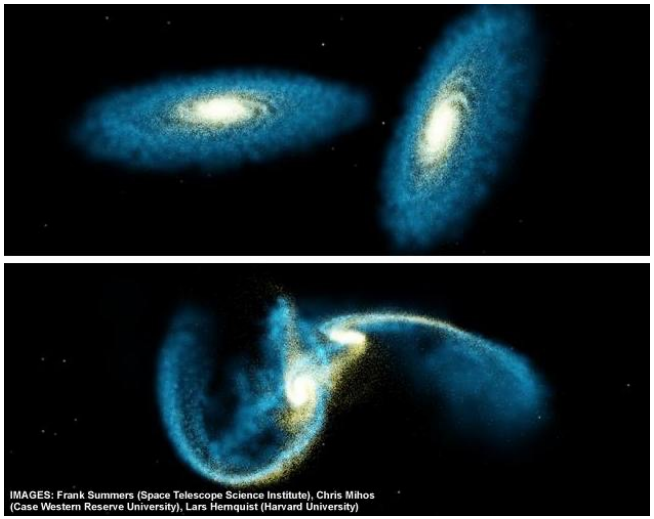


industry

entertainment

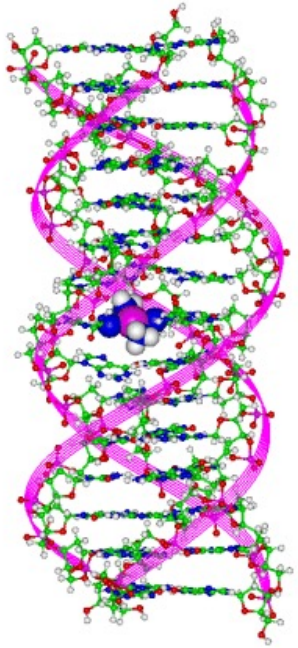


medicine

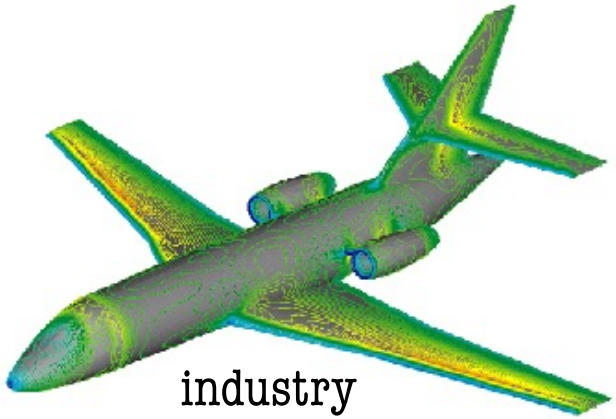


astrophysics

biology

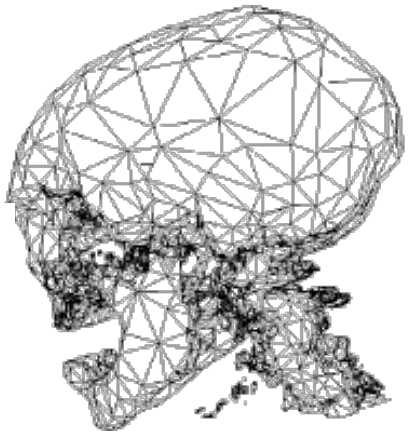


stock market



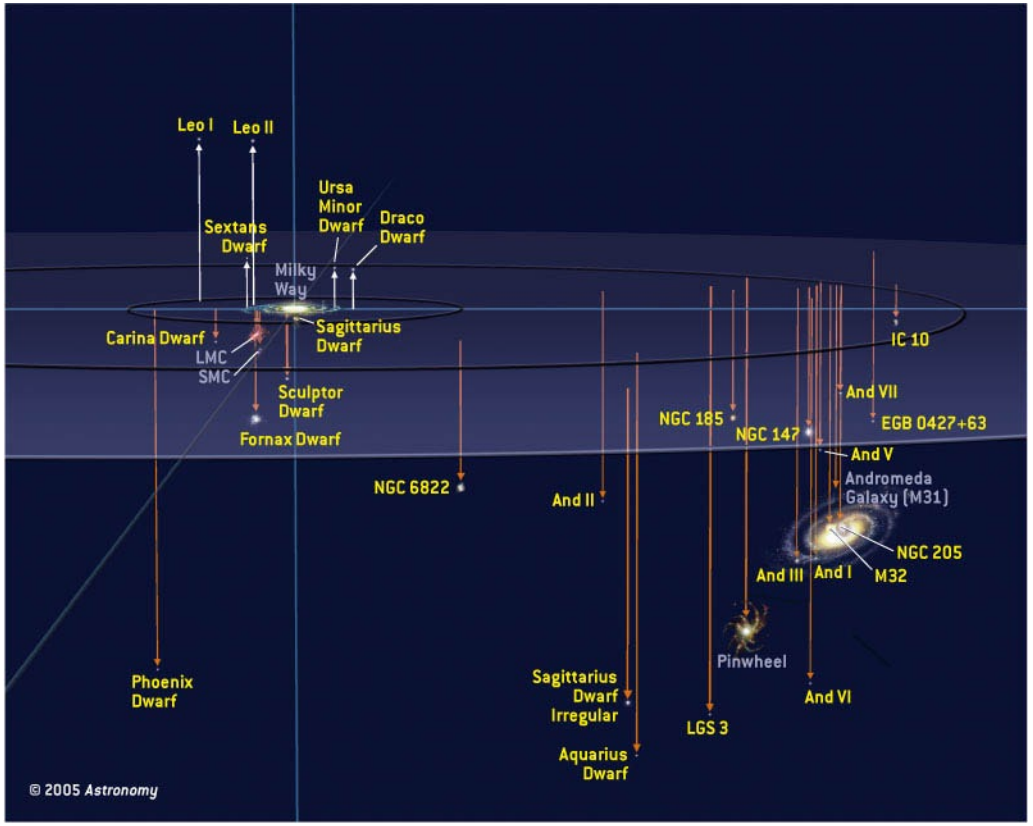
industry

entertainment

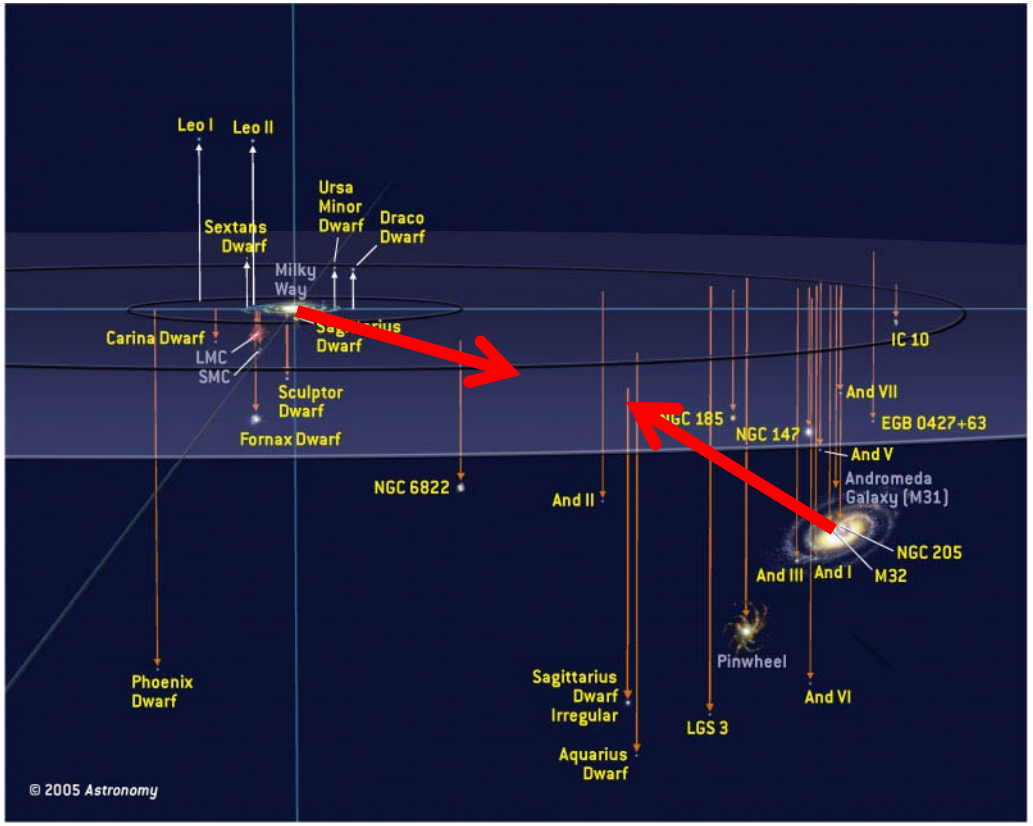


medicine

The collision of our Milky Way with the Andromeda Galaxy!



The collision of our Milky Way with the Andromeda Galaxy!

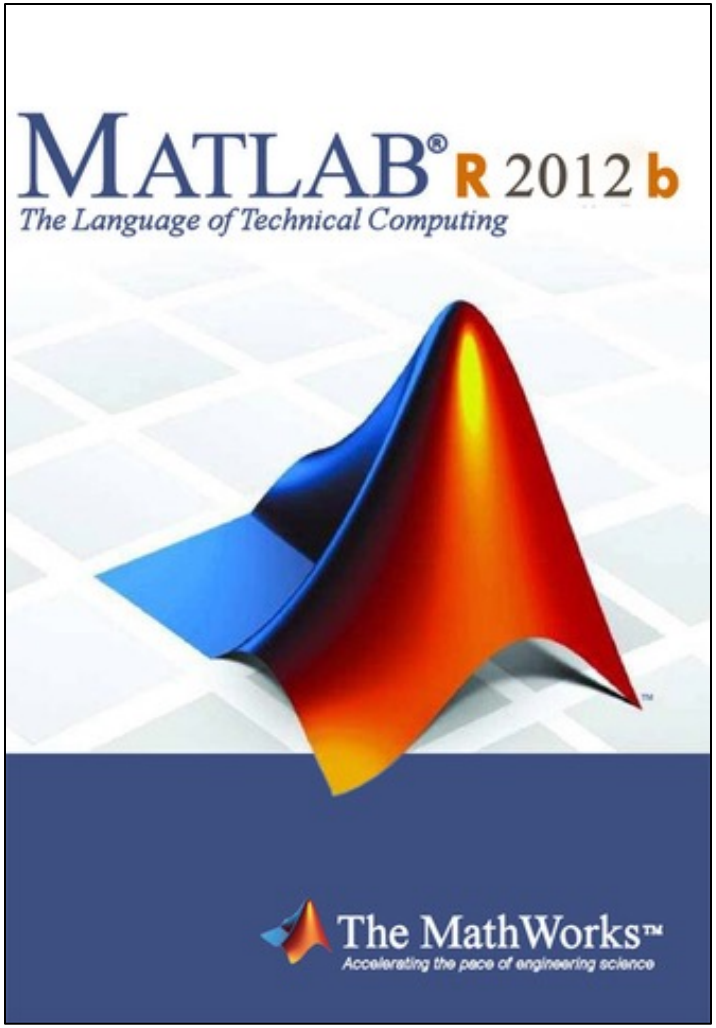


The collision of our Milky Way with the Andromeda Galaxy!

(courtesy Arman Khalatyan)

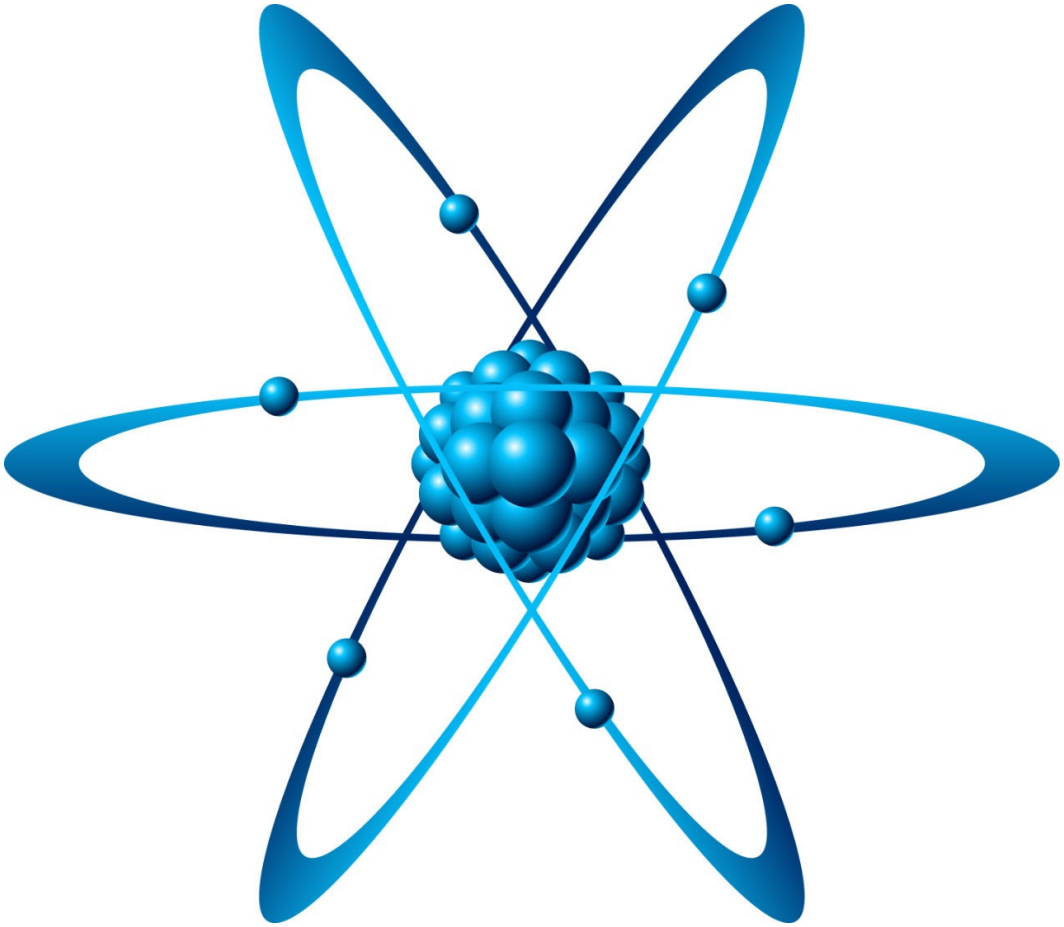
...and what about “Computacion I”?

- Content
- Itinerary
- Motivation
- **MATLAB**

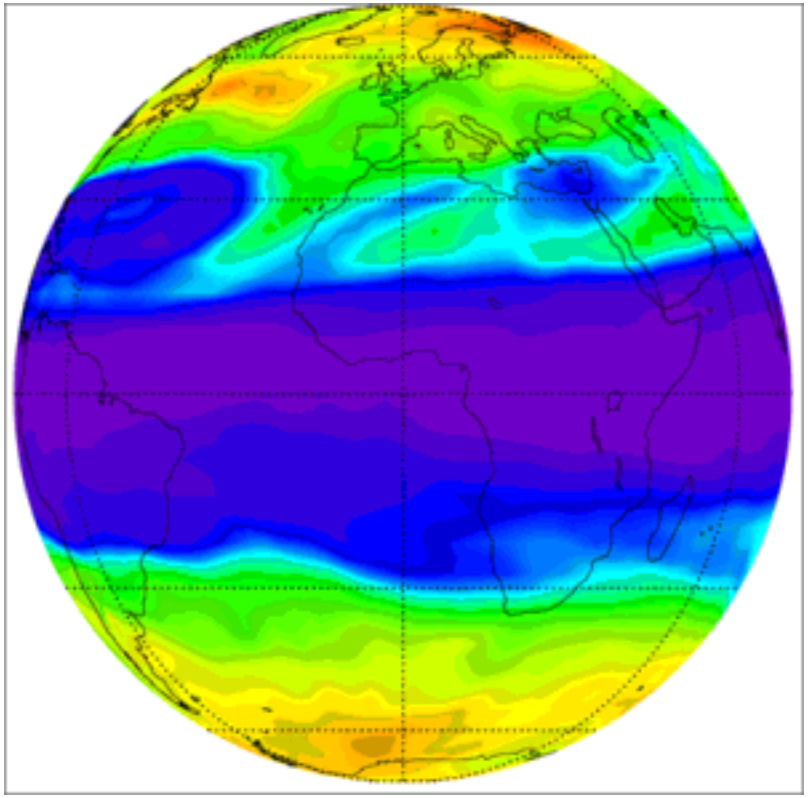

MATLAB® R 2012 b
The Language of Technical Computing
The MathWorks™
Accelerating the pace of engineering science

- MATLAB = **MAT**rix **LAB**oratory
- interactive program for computation and visualisation
- array processing language
- workhorse to better understand physics, but **not** a programming language

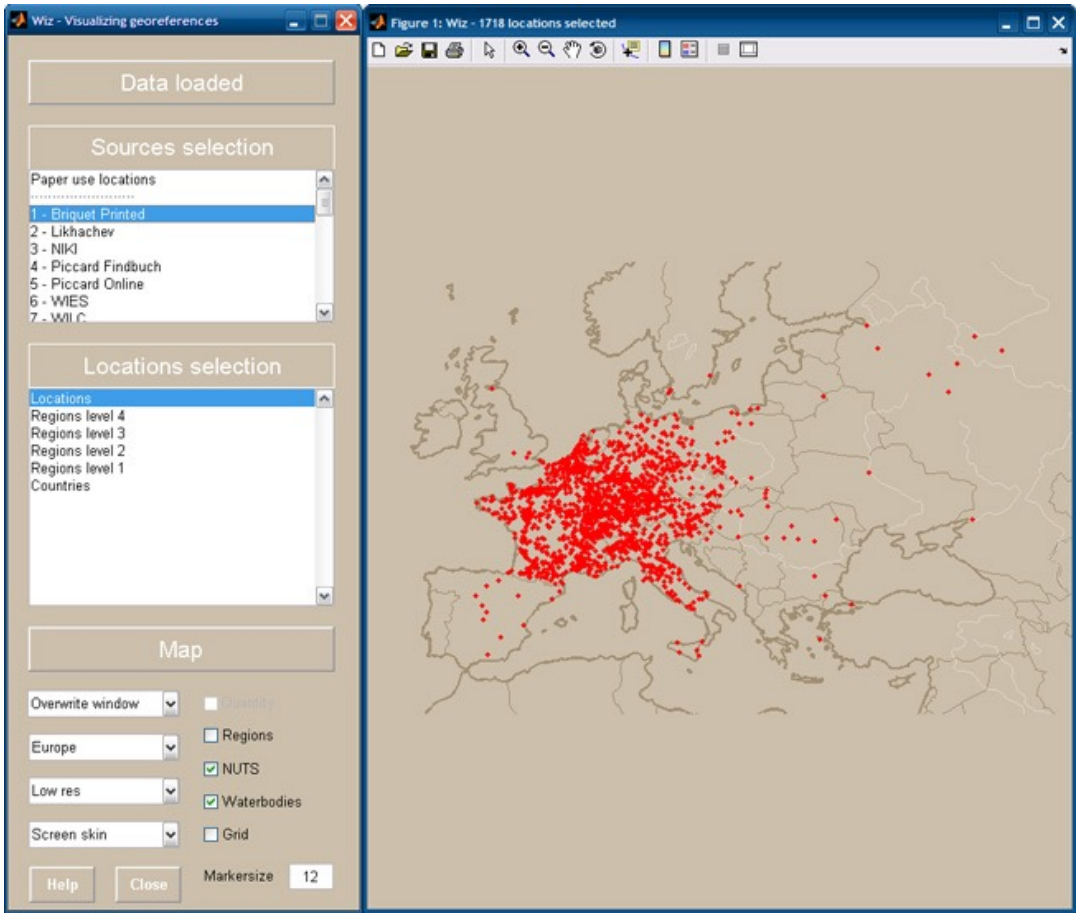
- applications in...physics



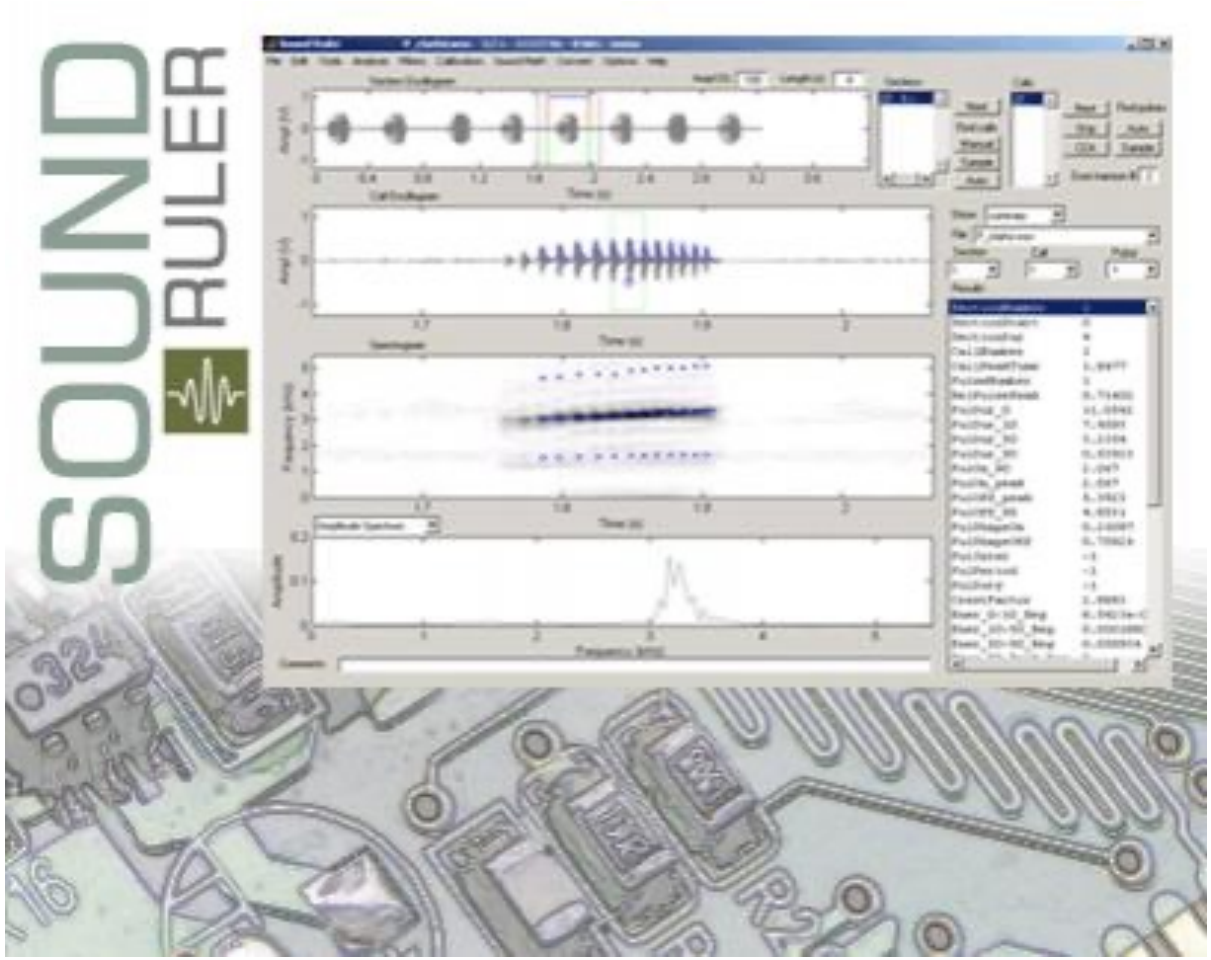
- applications in...meteorology



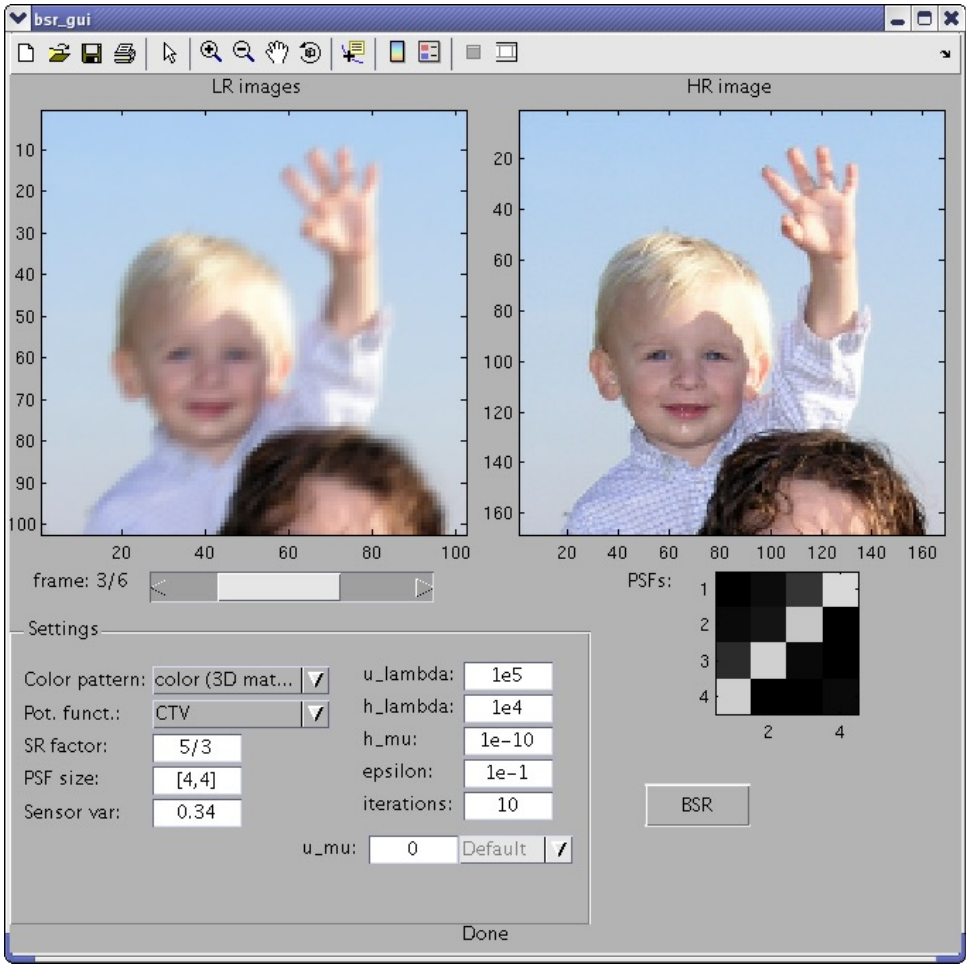
- applications in...population studies



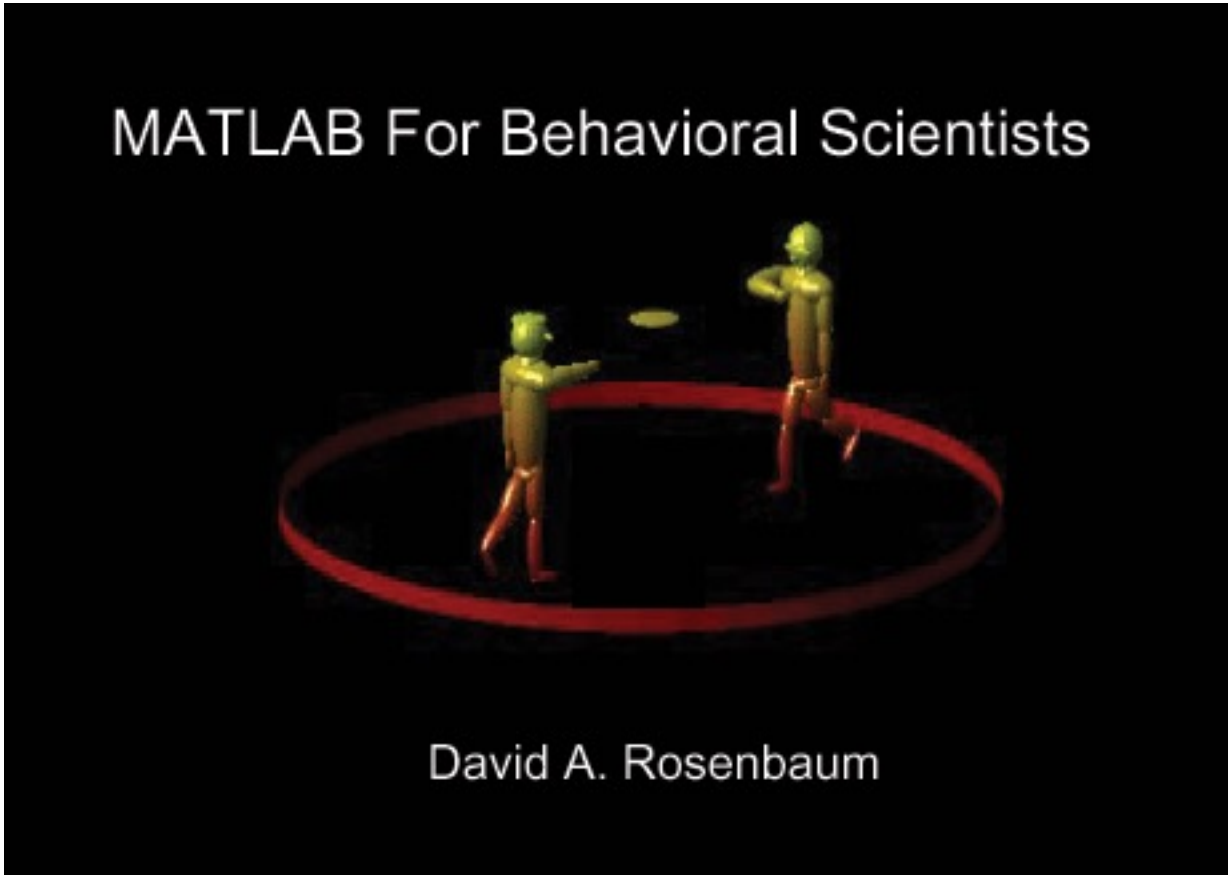
- applications in...sound effects



- applications in...image processing



- applications in...psychology

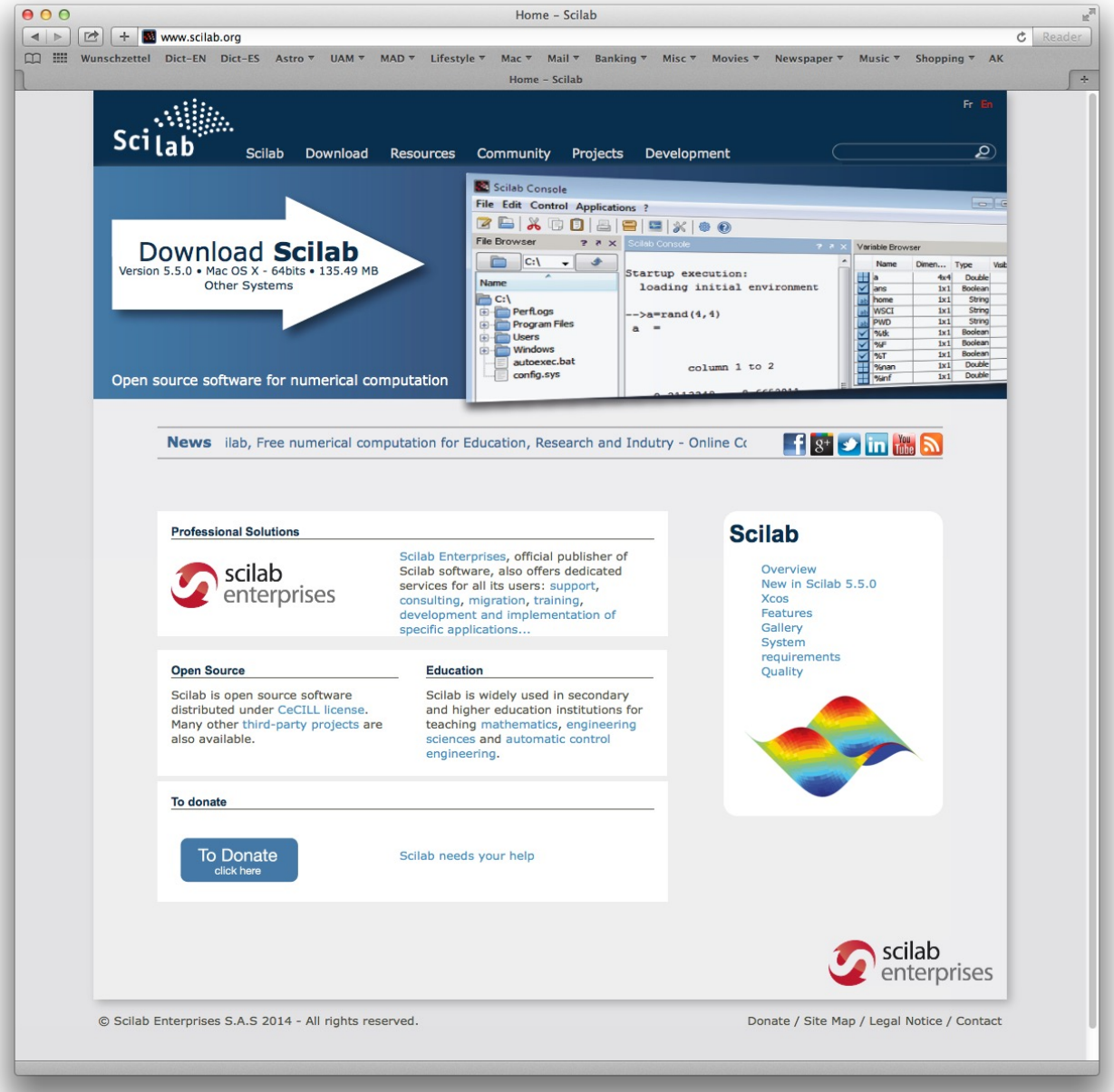


- licensed software!*

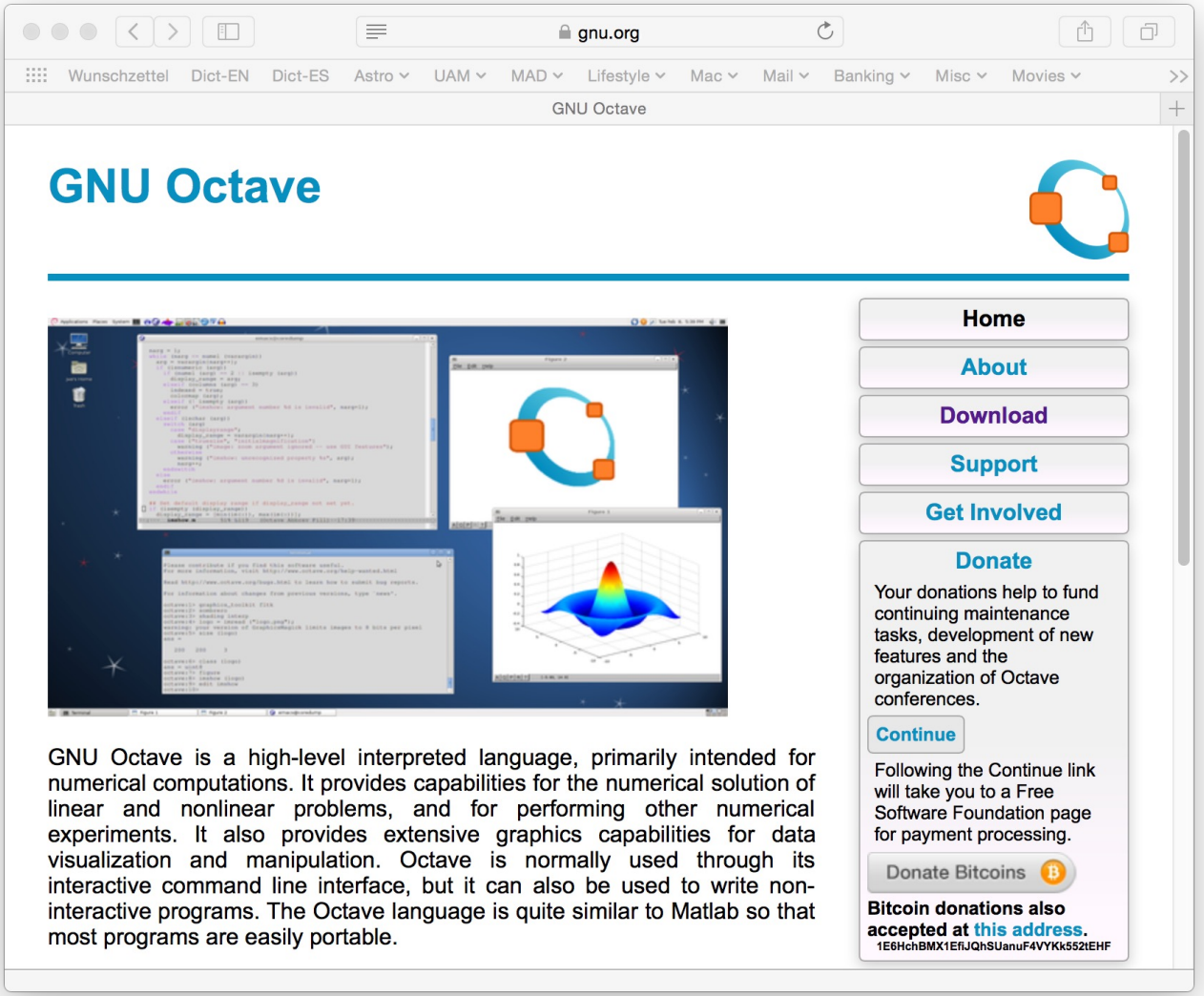
***but it is possible for UAM students to use the campus licence:**

https://www.uam.es/uam/media/doc/1606860596895/matlab-manual-instalacion-alumnos.pdf

- open source software:



- open source software:




START

1. logon to computer:

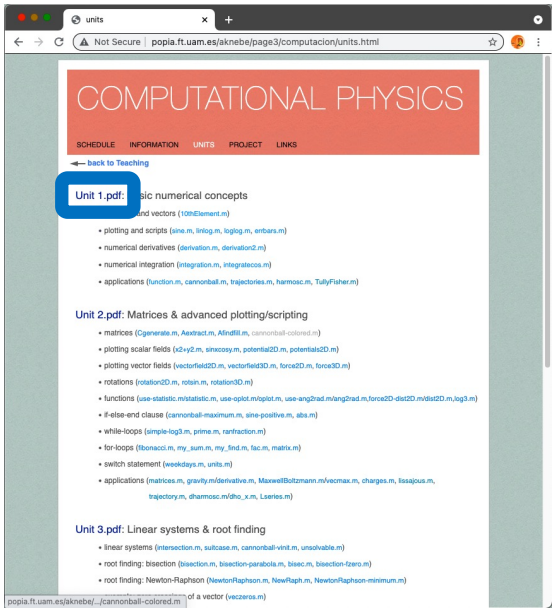
   user = nombre.apellido@estudiante.uam.es
   password = same as for email

2. open MATLAB

3. open/download **Unit1.pdf** from website

4. start with exercises on your own

http://popia.ft.uam.es/aknebe



1. logon to computer:

   user = nombre.apellido@estudiante.uam.es
   password = same as for email

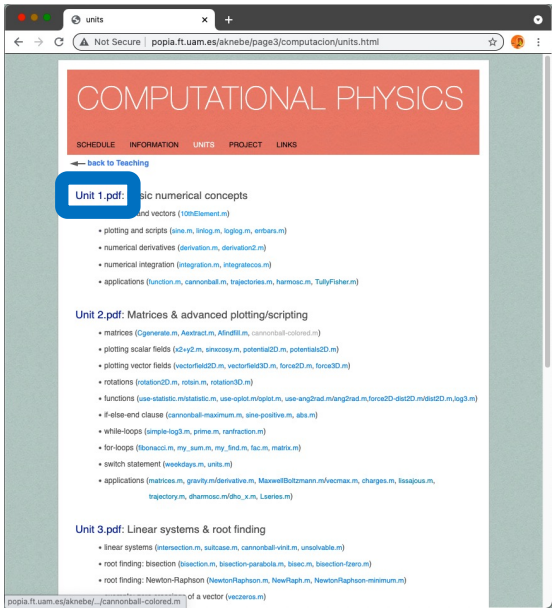
any problems? ⇨ contact CAU

2. open MATLAB

3. open/download **Unit1.pdf** from website

4. start with exercises on your

http://popia.ft.uam.es/aknebe



- general advice:
  - save everything you do!
  - CIE4 (Modulo 15) has free access to work anytime*
  - best to use your own laptop (SCILAB/Octave or MATLAB)

*Guia Docente: 6ECTS = 150h = 70h in class + **80h homework**