



# **COMPUTACION I**

## **GRUPO 516-5**

PROF. ALEXANDER KNEBE

DEPARTAMENTO DE FISICA TEORICA

GRUPO DE ASTROFISICA

MODULO M – 8, DESPACHO 316

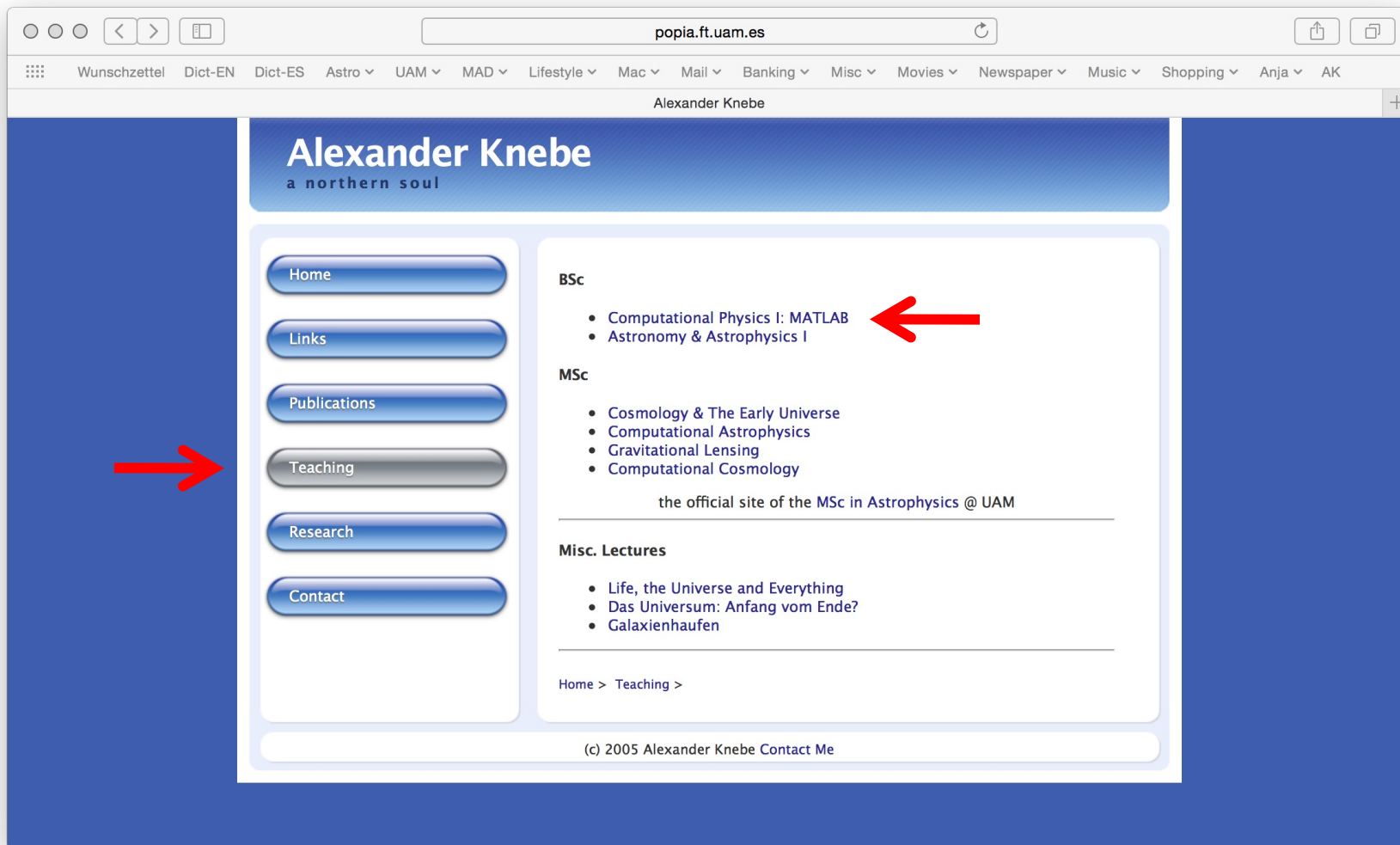


▪ **hands-on course: learning by doing!**

- 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



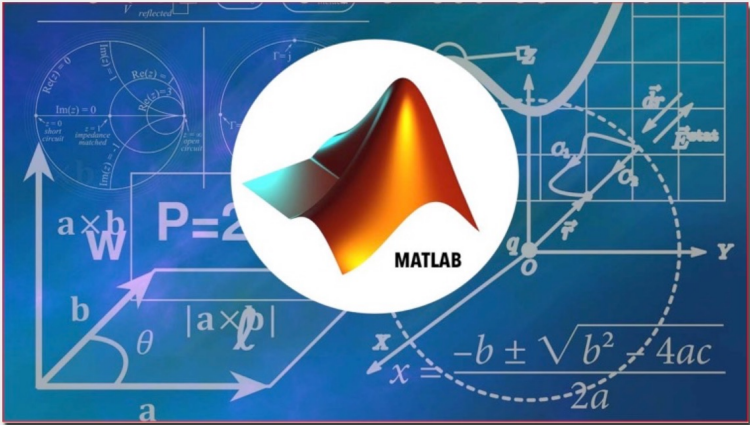
<http://popia.ft.uam.es/aknebe>

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

## Computacional Physics

HOME SCHEDULE UNITS PROJECT LINKS

← back to Teaching



This is a course for 1st year physics students aiming at...

- understanding how to use and program in a language relevant for scientific calculations,
- processing, interpreting, and presenting numerical data graphically,
- understanding how to use computers as a tool to solve analytical, algebraic and physical problems,
- developing to capability to model and implement a simple physical problem numerically in a computer, and

<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		
<i>christmas break</i>						
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)

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

U popia.ft.uam.es/Computacion/units.html

## Computacional Physics

HOME SCHEDULE **UNITS** PROJECT LINKS

← back to Teaching

Introduction.pdf: the overall introduction and motivation to the course as presented in class

Unit 1.pdf: basic numerical concepts

- variables and vectors ([10thElement.m](#))
- plotting and scripts ([sine.m](#), [linlog.m](#), [loglog.m](#), [errbars.m](#))
- numerical derivatives ([derivation.m](#), [derivation2.m](#))
- numerical integration ([integration.m](#), [integratecos.m](#))
- applications ([function.m](#), [cannonball.m](#), [trajectories.m](#), [harmosc.m](#), [TullyFisher.m](#))

Unit 2.pdf: Matrices & advanced plotting/scripting

- matrices ([Cgenerate.m](#), [Aextract.m](#), [Afindfill.m](#), [cannonball-colored.m](#))
- plotting scalar fields ([x2+y2.m](#), [sinxcosy.m](#), [potential2D.m](#), [potentials2D.m](#))
- plotting vector fields ([vectorfield2D.m](#), [vectorfield3D.m](#), [force2D.m](#), [force3D.m](#))
- rotations ([rotation2D.m](#), [rotsin.m](#), [rotation3D.m](#))
- functions ([use-statistic.m/statistic.m](#), [use-oplot.m/oplot.m](#), [use-ang2rad.m/ang2rad.m](#), [force2D-dist2D.m/dist2D.m](#), [log3.m](#))
- if-else-end clause ([cannonball-maximum.m](#), [sine-positive.m](#), [abs.m](#))
- while-loops ([simple-log3.m](#), [prime.m](#), [ranfraction.m](#))
- for-loops ([fibonacci.m](#), [my\\_sum.m](#), [my\\_find.m](#), [fac.m](#), [matrix.m](#))
- switch statement ([weekdays.m](#), [units.m](#))
- applications ([matrices.m](#), [gravity.m/derivative.m](#), [MaxwellBoltzmann.m/vecmax.m](#), [charges.m](#), [lissajous.m](#), [trajectory.m](#), [dharmosc.m/dho\\_x.m](#), [Lseries.m](#), [montecarlo.m](#))

Unit 3.pdf: Linear systems & root finding

- linear systems ([intersection.m](#), [suitcase.m](#), [cannonball-vinit.m](#), [horse.m](#), [unsolvable.m](#))

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

U popia.ft.uam.es/Computacion/units.html

## Computacional Physics

HOME SCHEDULE **UNITS** PROJECT LINKS

← back to Teaching

**Introduction.pdf:** the overall introduction and motivation to the course as presented in class

Unit 1.pdf: basic numerical concepts **this presentation**

- variables and vectors (10thElement.m)
- plotting and scripts (sine.m, linlog.m, loglog.m, errbars.m)
- numerical derivatives (derivation.m, derivation2.m)
- numerical integration (integration.m, integratecos.m)
- applications (function.m, cannonball.m, trajectories.m, harmosc.m, TullyFisher.m)

Unit 2.pdf: Matrices & advanced plotting/scripting

- matrices (Cgenerate.m, Aextract.m, Afindfill.m, cannonball-colored.m)
- plotting scalar fields (x2+y2.m, sinxcosy.m, potential2D.m, potentials2D.m)
- plotting vector fields (vectorfield2D.m, vectorfield3D.m, force2D.m, force3D.m)
- rotations (rotation2D.m, rotsin.m, rotation3D.m)
- functions (use-statistic.m/statistic.m, use-oplot.m/oplot.m, use-ang2rad.m/ang2rad.m, force2D-dist2D.m/dist2D.m, log3.m)
- if-else-end clause (cannonball-maximum.m, sine-positive.m, abs.m)
- while-loops (simple-log3.m, prime.m, ranfraction.m)
- for-loops (fibonacci.m, my\_sum.m, my\_find.m, fac.m, matrix.m)
- switch statement (weekdays.m, units.m)
- applications (matrices.m, gravity.m/derivative.m, MaxwellBoltzmann.m/vecmax.m, charges.m, lissajous.m, trajectory.m, dharmosc.m/dho\_x.m, Lseries.m, montecarlo.m)

Unit 3.pdf: Linear systems & root finding

- linear systems (intersection.m, suitcase.m, cannonball-vinit.m, horse.m, unsolvable.m)

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

U popia.ft.uam.es/Computacion/units.html

## Computacional Physics

HOME SCHEDULE **UNITS** PROJECT LINKS

← back to Teaching

Introduction.pdf: the overall introduction and motivation to the course as presented in class

**Unit 1.pdf: basic numerical concepts** **notes & exercises**

- windows and vectors (10thElement.m)
- plotting and scripts (sine.m, linlog.m, loglog.m, errbars.m)
- numerical derivatives (derivation.m, derivation2.m)
- numerical integration (integration.m, integratecos.m)
- applications (function.m, cannonball.m, trajectories.m, harmosc.m, TullyFisher.m)

**Unit 2.pdf: matrices & advanced plotting/scripting**

- matrices (Cgenerate.m, Aextract.m, Afindfill.m, cannonball-colored.m)
- plotting scalar fields (x2+y2.m, sinxcosy.m, potential2D.m, potentials2D.m)
- plotting vector fields (vectorfield2D.m, vectorfield3D.m, force2D.m, force3D.m)
- rotations (rotation2D.m, rotsin.m, rotation3D.m)
- functions (use-statistic.m/statistic.m, use-oplot.m/oplot.m, use-ang2rad.m/ang2rad.m, force2D-dist2D.m/dist2D.m, log3.m)
- if-else-end clause (cannonball-maximum.m, sine-positive.m, abs.m)
- while-loops (simple-log3.m, prime.m, ranfraction.m)
- for-loops (fibonacci.m, my\_sum.m, my\_find.m, fac.m, matrix.m)
- switch statement (weekdays.m, units.m)
- applications (matrices.m, gravity.m/derivative.m, MaxwellBoltzmann.m/vecmax.m, charges.m, lissajous.m, trajectory.m, dharmosc.m/dho\_x.m, Lseries.m, montecarlo.m)

**Unit 3.pdf: linear systems & root finding**

- linear systems (intersection.m, suitcase.m, cannonball-vinit.m, horse.m, unsolvable.m)



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

U popia.ft.uam.es/Computacion/units.html

## Computacional Physics

HOME SCHEDULE **UNITS** PROJECT LINKS

← back to Teaching

Introduction.pdf: the overall introduction and motivation to the course as presented in class

Unit 1.pdf: basic numerical concepts

- variables and vectors (10thElement.m)
- plotting and scripts (sine.m, linlog.m, loglog.m, errbars.m)
- numerical derivatives (derivation.m, derivation2.m)
- numerical integration (integration.m, integratecos.m)
- applications (function.m, cannonball.m, trajectories.m, harmosc, **TullyFisher.m**)

**solutions**

Unit 2.pdf: Matrices & advanced plotting/scripting

- matrices (Cgenerate.m, Aextract.m, Afindfill.m, cannonball-colored.m)
- plotting scalar fields (x2+y2.m, sinxcosy.m, potential2D.m, potentials2D.m)
- plotting vector fields (vectorfield2D.m, vectorfield3D.m, force2D.m, force3D.m)
- rotations (rotation2D.m, rotsin.m, rotation3D.m)
- functions (use-statistic.m/statistic.m, use-plot.m/plot.m, use-ang2rad.m/ang2rad.m, force2D-dist2D.m/dist2D.m, log3.m)
- if-else-end clause (cannonball-maximum.m, sine-positive.m, abs.m)
- while-loops (simple-log3.m, prime.m, ranfraction.m)
- for-loops (fibonacci.m, my\_sum.m, my\_find.m, fac.m, matrix.m)
- switch statement (weekdays.m, units.m)
- applications (matrices.m, gravity.m/derivative.m, MaxwellBoltzmann.m/vecmax.m, charges.m, lissajous.m, trajectory.m, dharmosc.m/dho\_x.m, Lseries.m, montecarlo.m)

Unit 3.pdf: Linear systems & root finding

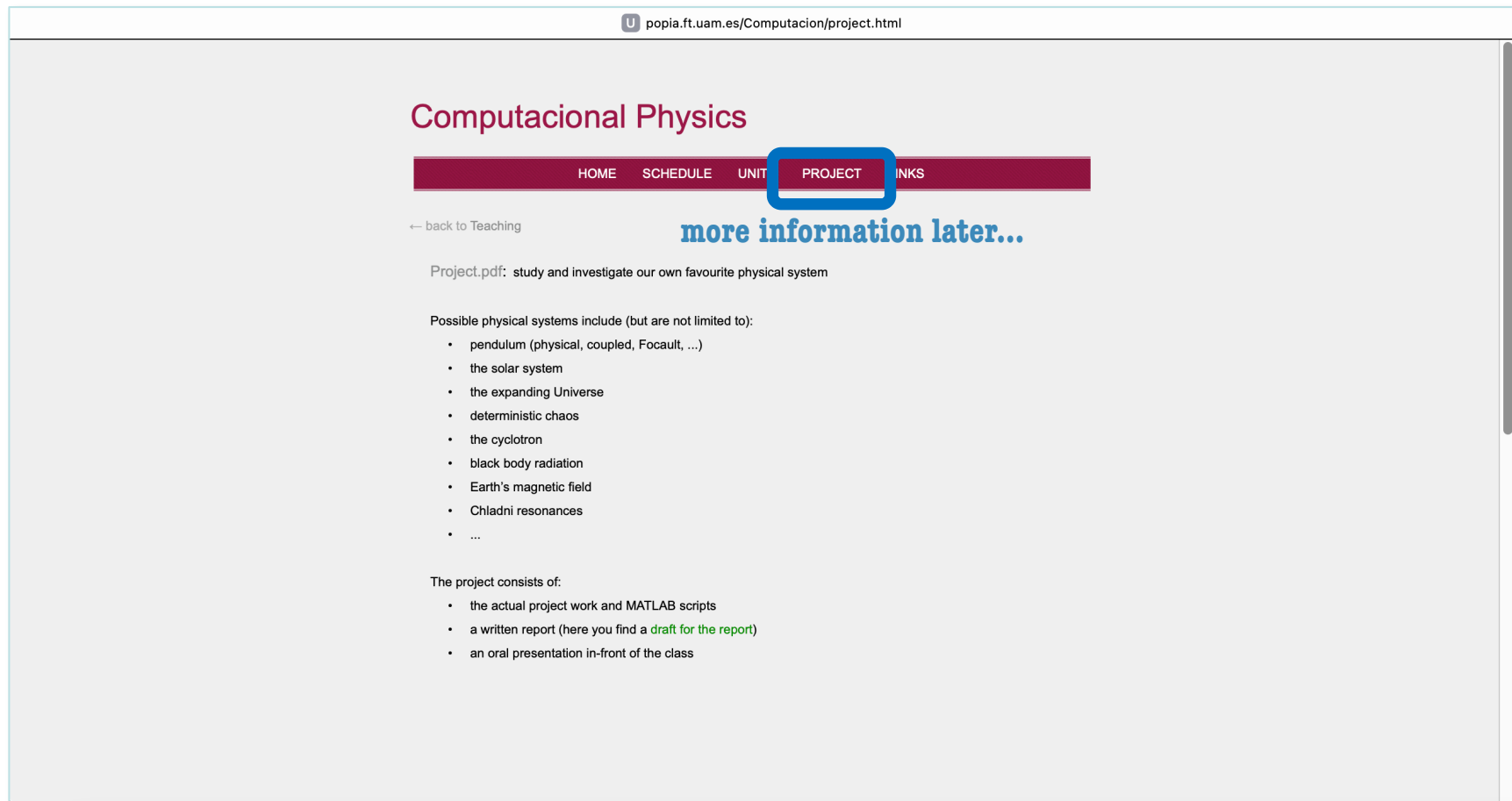
- linear systems (intersection.m, suitcase.m, cannonball-vinit.m, horse.m, unsolvable.m)

PROF. ALEXANDER KNEBE

DEPARTAMENTO DE FISICA TEORICA

GRUPO DE ASTROFISICA

MODULO M – 8, DESPACHO 316

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

U popia.ft.uam.es/Computacion/project.html

## Computacional Physics

HOME SCHEDULE UNIT **PROJECT** LINKS

← back to Teaching

**more information later...**

Project.pdf: study and investigate our own favourite physical system

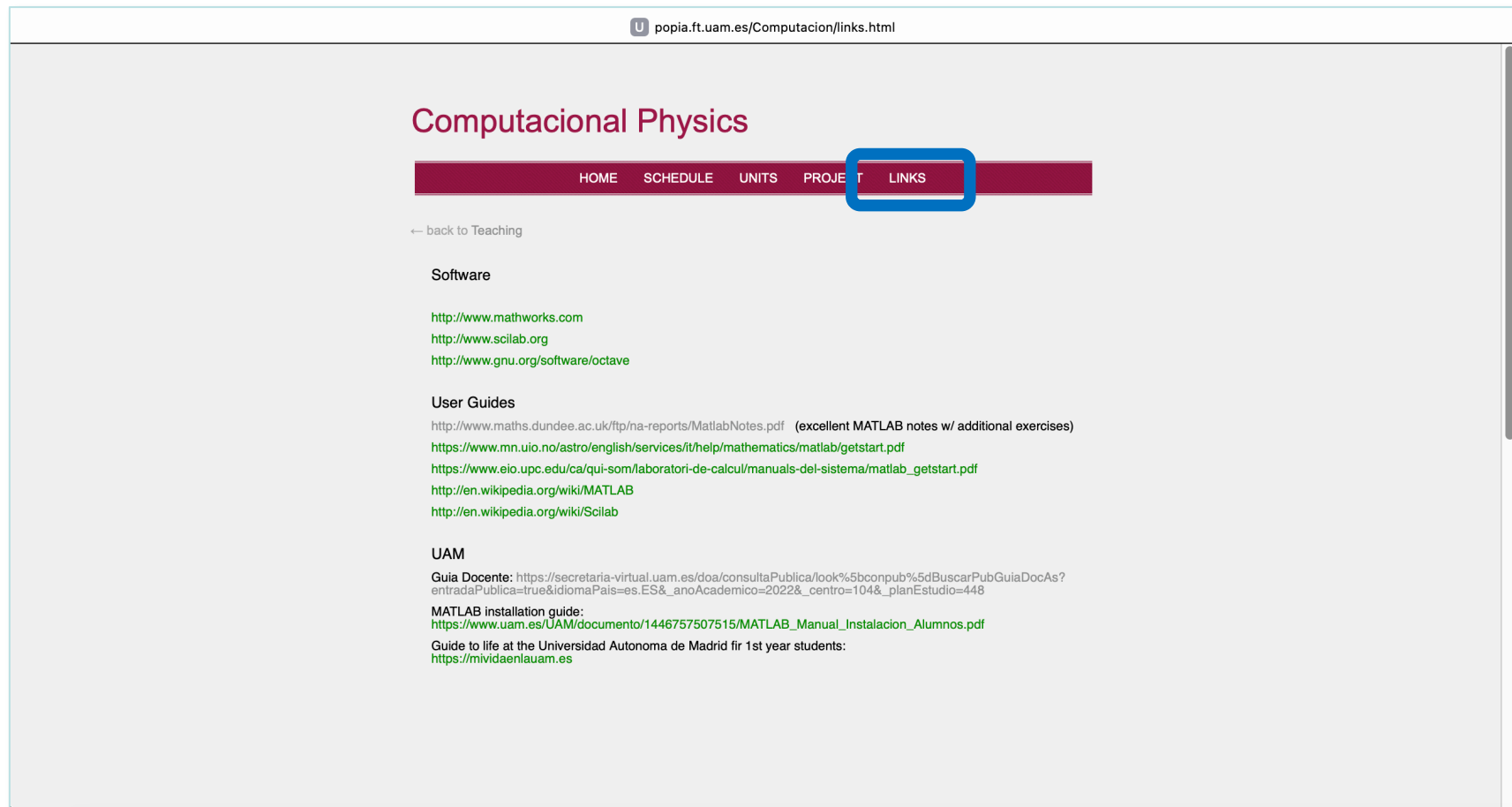
Possible physical systems include (but are not limited to):

- pendulum (physical, coupled, Foucault, ...)
- the solar system
- the expanding Universe
- deterministic chaos
- the cyclotron
- black body radiation
- Earth's magnetic field
- Chladni resonances
- ...

The project consists of:

- the actual project work and MATLAB scripts
- a written report (here you find a [draft for the report](#))
- an oral presentation in-front of the class

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



The screenshot shows a web browser window with the address bar displaying `popia.ft.uam.es/Computacion/links.html`. The page title is "Computacional Physics". A dark red navigation bar contains the following menu items: HOME, SCHEDULE, UNITS, PROJE T, and LINKS. The "LINKS" item is highlighted with a blue rectangular box. Below the navigation bar, there is a link "← back to Teaching". The page is organized into three sections: "Software", "User Guides", and "UAM".

**Software**

- <http://www.mathworks.com>
- <http://www.scilab.org>
- <http://www.gnu.org/software/octave>

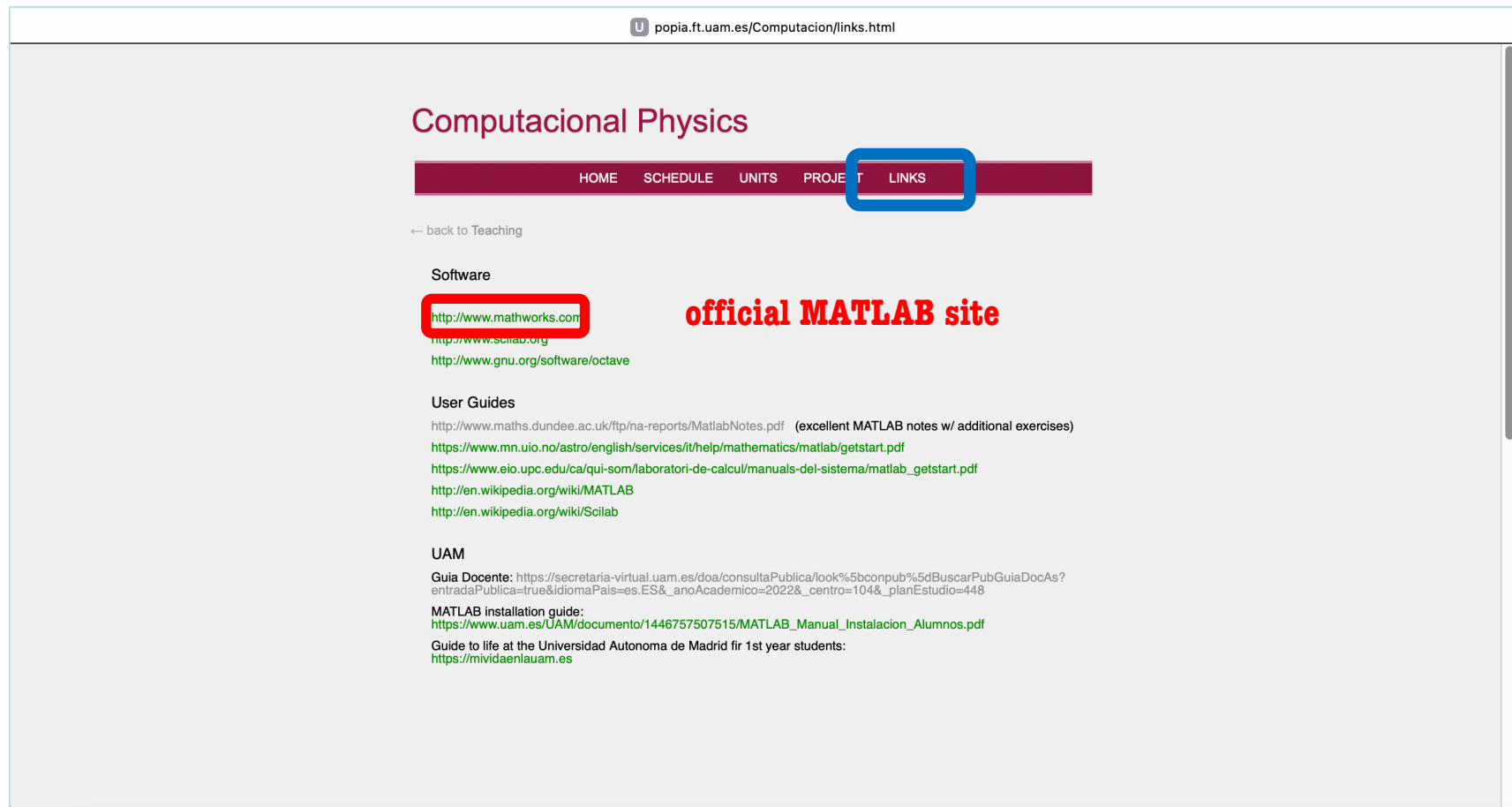
**User Guides**

- <http://www.maths.dundee.ac.uk/ftp/na-reports/MatlabNotes.pdf> (excellent MATLAB notes w/ additional exercises)
- <https://www.mn.uio.no/astro/english/services/it/help/mathematics/matlab/getstart.pdf>
- [https://www.eio.upc.edu/ca/qui-som/laboratori-de-calcul/manuals-del-sistema/matlab\\_getstart.pdf](https://www.eio.upc.edu/ca/qui-som/laboratori-de-calcul/manuals-del-sistema/matlab_getstart.pdf)
- <http://en.wikipedia.org/wiki/MATLAB>
- <http://en.wikipedia.org/wiki/Scilab>

**UAM**

- Guia Docente:** [https://secretaria-virtual.uam.es/doa/consultaPublica/look%5Bconpub%5dBuscarPubGuiaDocAs?entradaPublica=true&idiomaPais=es.ES&\\_anoAcademico=2022&\\_centro=104&\\_planEstudio=448](https://secretaria-virtual.uam.es/doa/consultaPublica/look%5Bconpub%5dBuscarPubGuiaDocAs?entradaPublica=true&idiomaPais=es.ES&_anoAcademico=2022&_centro=104&_planEstudio=448)
- MATLAB installation guide:** [https://www.uam.es/UAM/documento/1446757507515/MATLAB\\_Manual\\_Instalacion\\_Alumnos.pdf](https://www.uam.es/UAM/documento/1446757507515/MATLAB_Manual_Instalacion_Alumnos.pdf)
- Guide to life at the Universidad Autonoma de Madrid fir 1st year students:** <https://mividaenlauam.es>

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



The screenshot shows a web browser window with the address bar displaying `popia.ft.uam.es/Computacion/links.html`. The page title is "Computacional Physics". A navigation bar contains links for HOME, SCHEDULE, UNITS, PROJE T, and LINKS. The LINKS link is highlighted with a blue box. Below the navigation bar, there is a "← back to Teaching" link. The main content is organized into sections: "Software" with links to <http://www.mathworks.com> (highlighted with a red box), <http://www.scilab.org>, and <http://www.gnu.org/software/octave>; "User Guides" with links to <http://www.maths.dundee.ac.uk/ftp/na-reports/MatlabNotes.pdf> (noted as "excellent MATLAB notes w/ additional exercises"), <https://www.mn.uio.no/astro/english/services/it/help/mathematics/matlab/getstart.pdf>, [https://www.eio.upc.edu/ca/qui-som/laboratori-de-calcul/manuals-del-sistema/matlab\\_getstart.pdf](https://www.eio.upc.edu/ca/qui-som/laboratori-de-calcul/manuals-del-sistema/matlab_getstart.pdf), <http://en.wikipedia.org/wiki/MATLAB>, and <http://en.wikipedia.org/wiki/Scilab>; and "UAM" with links to [https://secretaria-virtual.uam.es/doi/consultaPublica/look%5bconpub%5dBuscarPubGuiaDocAs?entradaPublica=true&idiomaPais=es.ES&\\_anoAcademico=2022&\\_centro=104&\\_planEstudio=448](https://secretaria-virtual.uam.es/doi/consultaPublica/look%5bconpub%5dBuscarPubGuiaDocAs?entradaPublica=true&idiomaPais=es.ES&_anoAcademico=2022&_centro=104&_planEstudio=448), [https://www.uam.es/UAM/documento/1446757507515/MATLAB\\_Manual\\_Instalacion\\_Alumnos.pdf](https://www.uam.es/UAM/documento/1446757507515/MATLAB_Manual_Instalacion_Alumnos.pdf), and <https://mividaenlauam.es>.

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

U popia.ft.uam.es/Computacion/links.html

## Computacional Physics

HOME SCHEDULE UNITS PROJECTS **LINKS**

← back to Teaching

### Software

<https://www.mathworks.com>  
<http://www.scilab.org>  
<http://www.gnu.org/software/octave>

### free MATLAB clones

### User Guides

<http://www.maths.dundee.ac.uk/ftp/na-reports/MatlabNotes.pdf> (excellent MATLAB notes w/ additional exercises)  
<https://www.mn.uio.no/astro/english/services/it/help/mathematics/matlab/getstart.pdf>  
[https://www.eio.upc.edu/ca/qui-som/laboratori-de-calcul/manuals-del-sistema/matlab\\_getstart.pdf](https://www.eio.upc.edu/ca/qui-som/laboratori-de-calcul/manuals-del-sistema/matlab_getstart.pdf)  
<http://en.wikipedia.org/wiki/MATLAB>  
<http://en.wikipedia.org/wiki/Scilab>

### UAM

**Guia Docente:** [https://secretaria-virtual.uam.es/doa/consultaPublica/look%5Bconpub%5dBuscarPubGuiaDocAs?entradaPublica=true&idiomaPais=es.ES&\\_anoAcademico=2022&\\_centro=104&\\_planEstudio=448](https://secretaria-virtual.uam.es/doa/consultaPublica/look%5Bconpub%5dBuscarPubGuiaDocAs?entradaPublica=true&idiomaPais=es.ES&_anoAcademico=2022&_centro=104&_planEstudio=448)

**MATLAB installation guide:**  
[https://www.uam.es/UAM/documento/1446757507515/MATLAB\\_Manual\\_Instalacion\\_Alumnos.pdf](https://www.uam.es/UAM/documento/1446757507515/MATLAB_Manual_Instalacion_Alumnos.pdf)

**Guide to life at the Universidad Autonoma de Madrid fir 1st year students:**  
<https://mividaenlauam.es>

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

U popia.ft.uam.es/Computacion/links.html

## Computacional Physics

HOME SCHEDULE UNITS PROJECTS **LINKS**

← back to Teaching

### Software

- <http://www.mathworks.com>
- <http://www.scilab.org>
- <http://www.gnu.org/software/octave>

### User Guides

- <http://www.maths.dundee.ac.uk/ftp/na-reports/MatlabNotes.pdf> (excellent MATLAB notes w/ additional exercises)
- <https://www.mn.uio.no/astro/english/services/it/help/mathematics/matlab/getstart.pdf>
- [https://www.eio.upc.edu/ca/qui-som/laboratori-de-calcul/manuals-del-sistema/matlab\\_getstart.pdf](https://www.eio.upc.edu/ca/qui-som/laboratori-de-calcul/manuals-del-sistema/matlab_getstart.pdf)
- <http://en.wikipedia.org/wiki/MATLAB>
- <http://en.wikipedia.org/wiki/Scilab>

### UAM

Guia Docente: [https://secretaria-virtual.uam.es/doi/consultaPublica/look%5Bconpub%5dBuscarPubGuiaDocAs?entradaPublica=true&idiomaPais=es\\_ES&anoAcademico=2022&centro=104&planEstudio=448](https://secretaria-virtual.uam.es/doi/consultaPublica/look%5Bconpub%5dBuscarPubGuiaDocAs?entradaPublica=true&idiomaPais=es_ES&anoAcademico=2022&centro=104&planEstudio=448)

**MATLAB installation guide:**  
[https://www.uam.es/UAM/documento/1446757507515/MATLAB\\_Manual\\_Instalacion\\_Alumnos.pdf](https://www.uam.es/UAM/documento/1446757507515/MATLAB_Manual_Instalacion_Alumnos.pdf)

Guide to life at the Universidad Autonoma de Madrid fir 1st year students:  
<https://mividaenlauam.es>

**...use an UAM licence!**

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

U popia.ft.uam.es/Computacion/links.html

## Computacional Physics

HOME SCHEDULE UNITS PROJECTS LINKS

← back to Teaching

### Software

- <http://www.mathworks.com>
- <http://www.scilab.org>
- <http://www.gnu.org/software/octave>

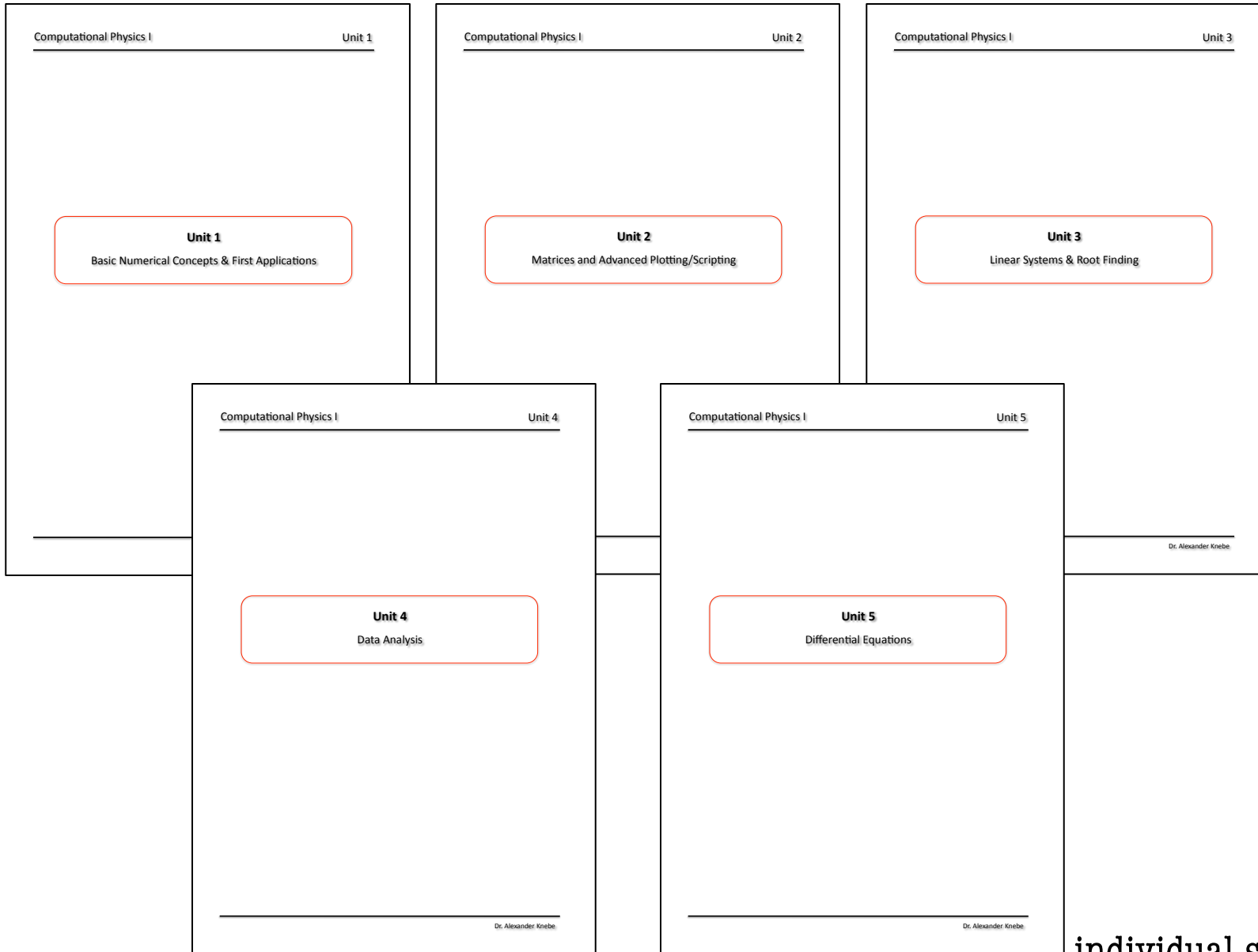
### User Guides

- <http://www.maths.dundee.ac.uk/ftp/na-reports/MatlabNotes.pdf> (excellent MATLAB notes w/ additional exercises)
- <https://www.mn.uio.no/astro/english/services/it/help/mathematics/matlab/getstart.pdf>
- [https://www.eio.upc.edu/ca/qui-som/laboratori-de-calcul/manuals-del-sistema/matlab\\_getstart.pdf](https://www.eio.upc.edu/ca/qui-som/laboratori-de-calcul/manuals-del-sistema/matlab_getstart.pdf)
- <http://en.wikipedia.org/wiki/MATLAB>
- <http://en.wikipedia.org/wiki/Scilab>

### UAM

- Guia Docente:** [https://secretaria-virtual.uam.es/doa/consultaPublica/look%5Bconpub%5dBuscarPubGuiaDocAs?entradaPublica=true&idiomaPais=es.ES&\\_anoAcademico=2022&\\_centro=104&\\_planEstudio=448](https://secretaria-virtual.uam.es/doa/consultaPublica/look%5Bconpub%5dBuscarPubGuiaDocAs?entradaPublica=true&idiomaPais=es.ES&_anoAcademico=2022&_centro=104&_planEstudio=448)
- MATLAB installation guide:** [https://www.uam.es/UAM/documento/1446757507515/MATLAB\\_Manual\\_Instalacion\\_Alumnos.pdf](https://www.uam.es/UAM/documento/1446757507515/MATLAB_Manual_Instalacion_Alumnos.pdf)
- Guide to life at the Universidad Autonoma de Madrid fir 1st year students:** <https://mividaenlauam.es>

**useful 'welcome guide' for 1<sup>st</sup>-year students**



individual sets of exercises!



## Computational Physics I

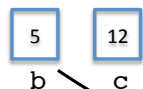
## Unit 1

### 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()`

## Computational Physics I

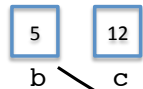
## Unit 1

### 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!**

## explanations:

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

#### > 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()`

## Computational Physics I

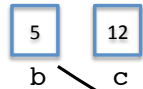
## Unit 1

### 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()`

## exercises:

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

## Computational Physics I

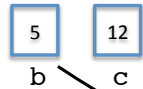
## Unit 1

### 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()`

- 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

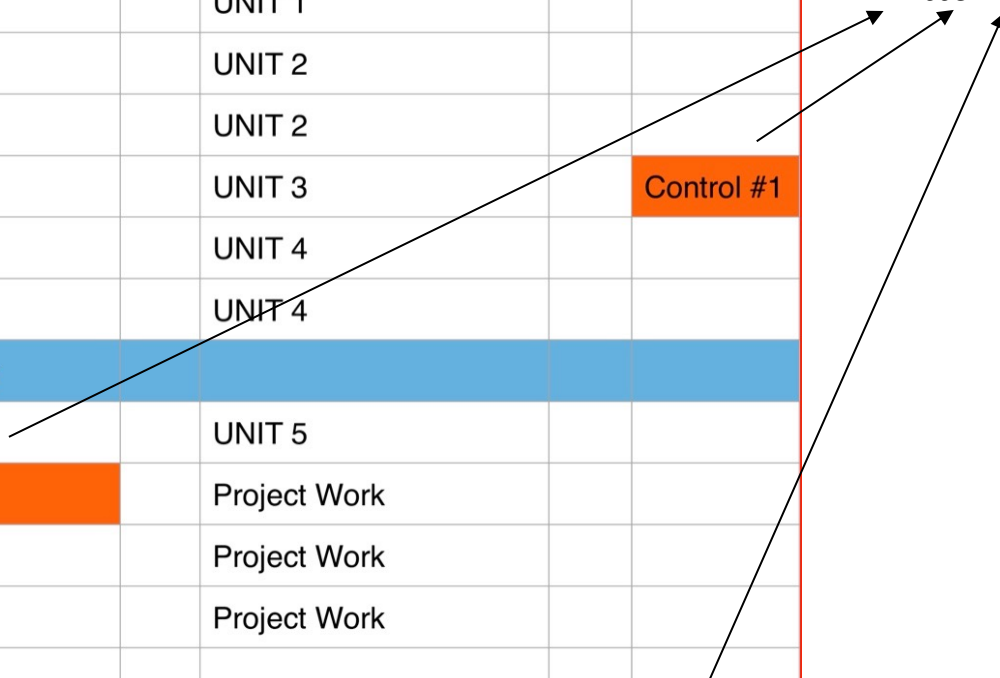
Computacion I (2022/23) - Group 516-5

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		
		<i>christmas break</i>				
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		
		<i>christmas break</i>				
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		
		<i>christmas break</i>				
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%
------------	-----

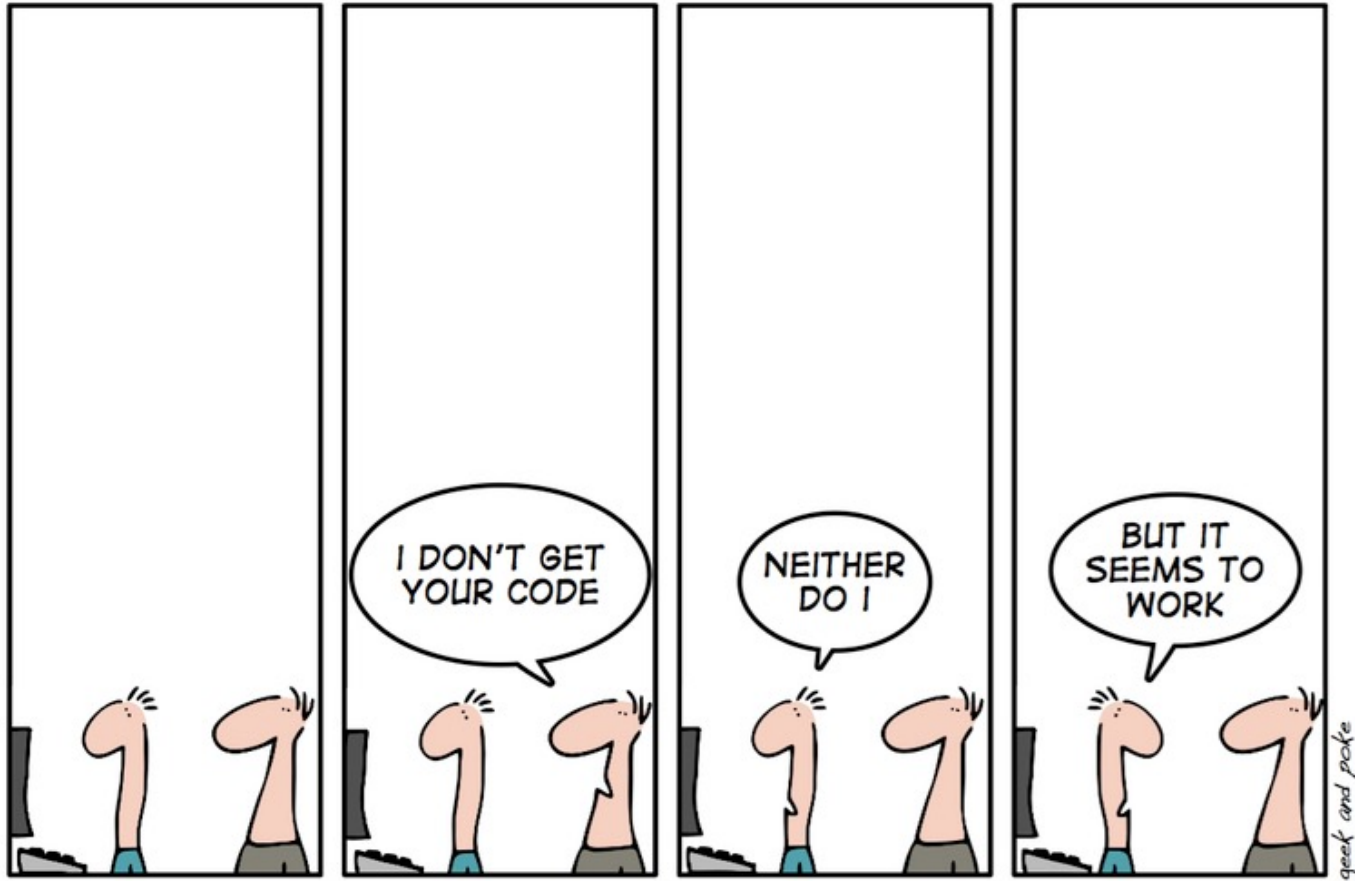
Control #2	20%
------------	-----

Control #3	30%
------------	-----

Project	40%	(programs, written report, oral presentation)
---------	-----	---

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





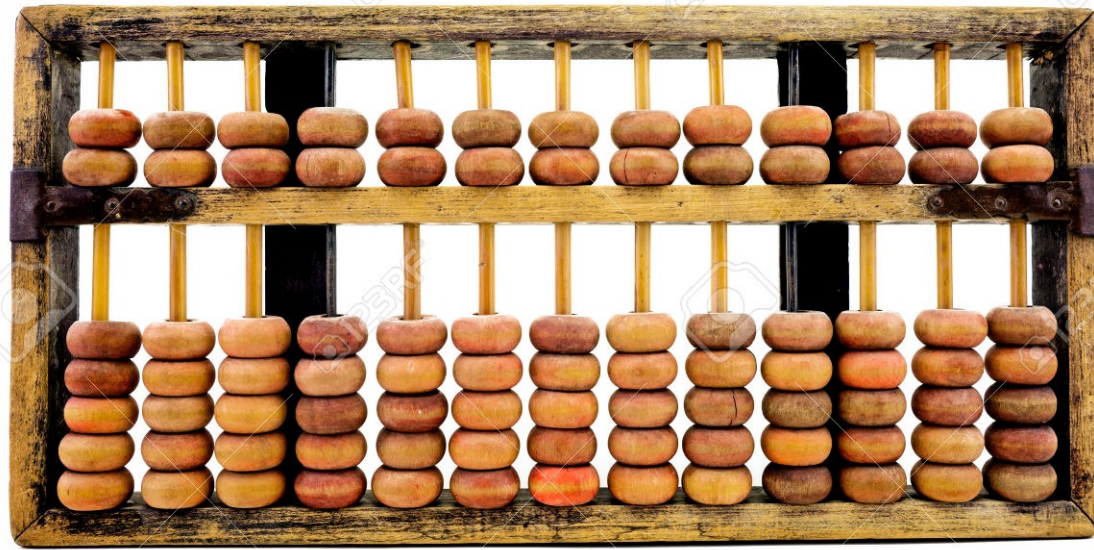
THE ART OF PROGRAMING

how and when did it all begin?



Eurasia

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

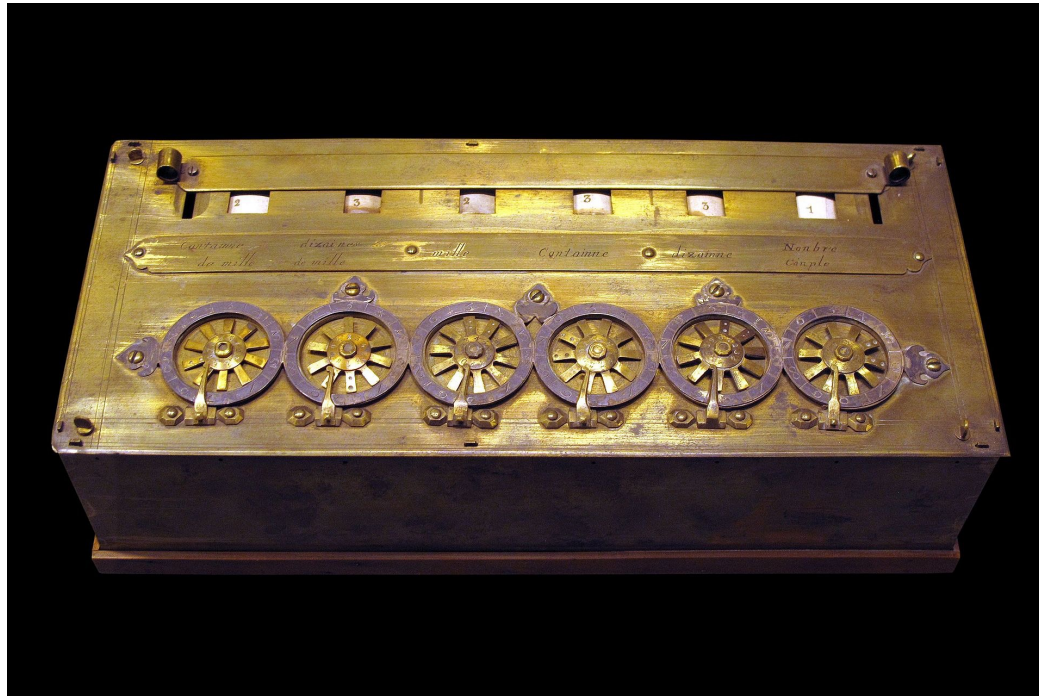


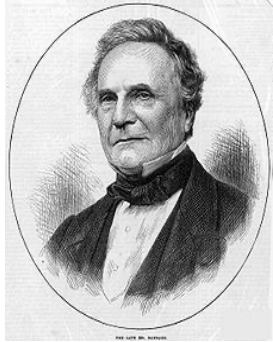
Abacus



Blaise Pascal

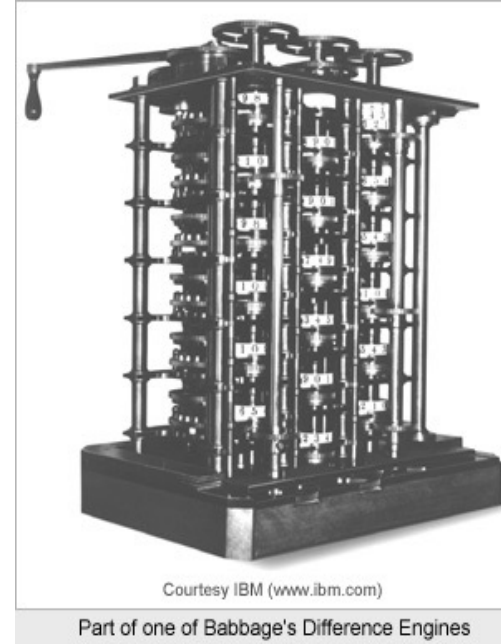
- mechanical calculator
- addition/subtraction
- multiplication/division by successive addition/subtraction
- 5 digit accuracy

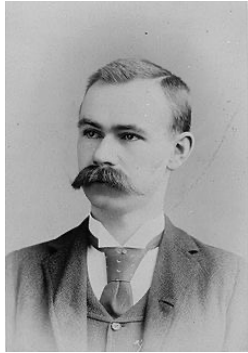




Charles Babbage

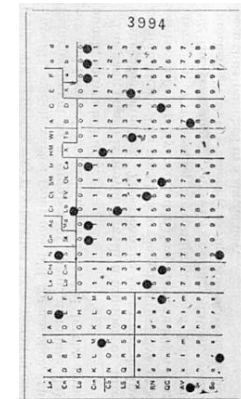
- 4000 parts
- 3 tons
- 3m x 2 m
- 31 digits accuracy
- steam driven
- “difference engine”:
  - designed to tabulate polynomials
  - calculates 2<sup>nd</sup> order differences



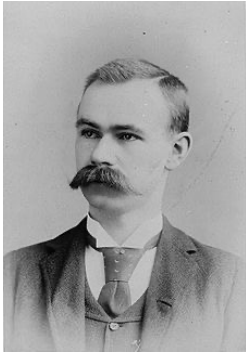


Herman Hollerith

- 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

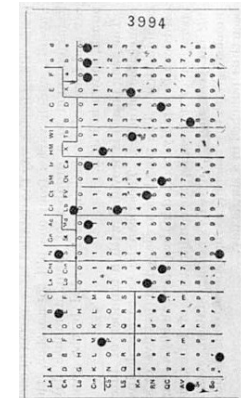


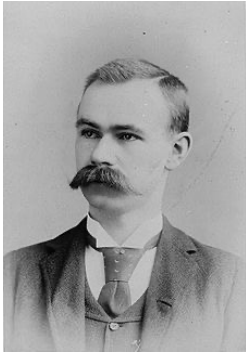




Herman Hollerith

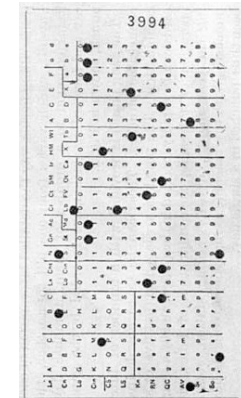
- developed a machine to read punch cards
- machines was used for 1890 census in the USA
- census took only one year to evaluate (prev. 8 years!)
- company name: Computing Tabulating Recording Corporation
- his company was renamed in 1924 to ...





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

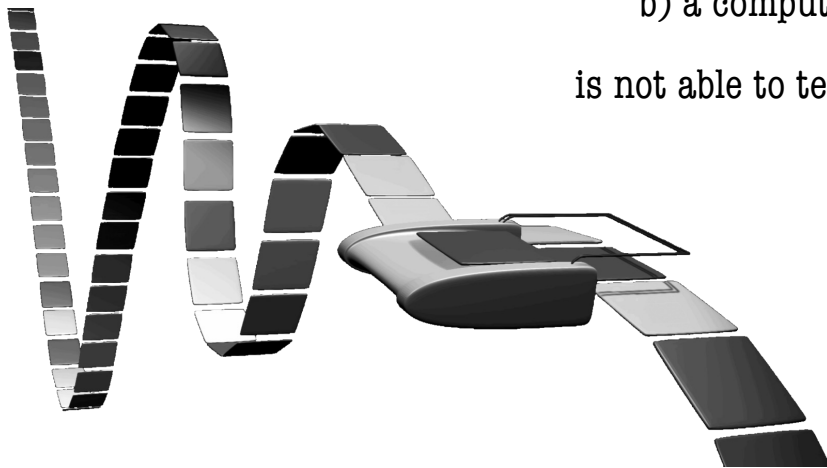
- Turing Test:

- if a human interacting with

- a) another human

- b) a computer

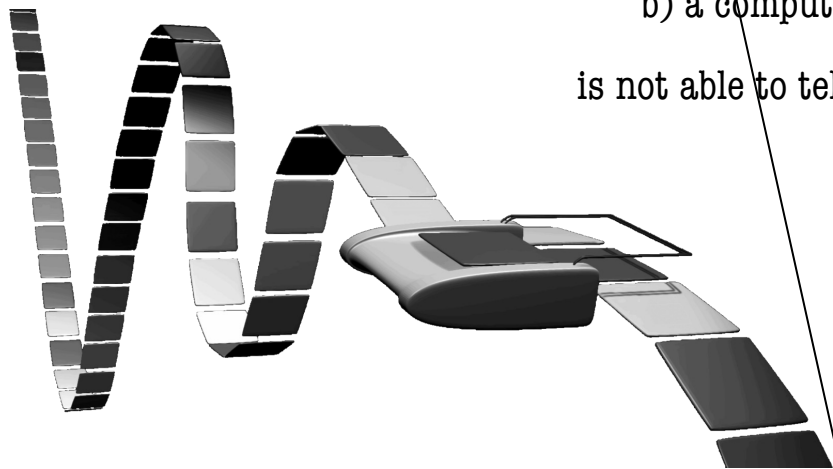
is not able to tell the difference, the computer is said to “think”





Alan Turing

- Turing Machine:
  - theoretical conce
  - reading ins
  - result prin
  - foundation for the
- Turing Test:
  - if a human intera
  - a) another h
  - b) a compute
  - is not able to tel



Select all images with commercial lorries

Username

Password

Login with

Remember me

Register

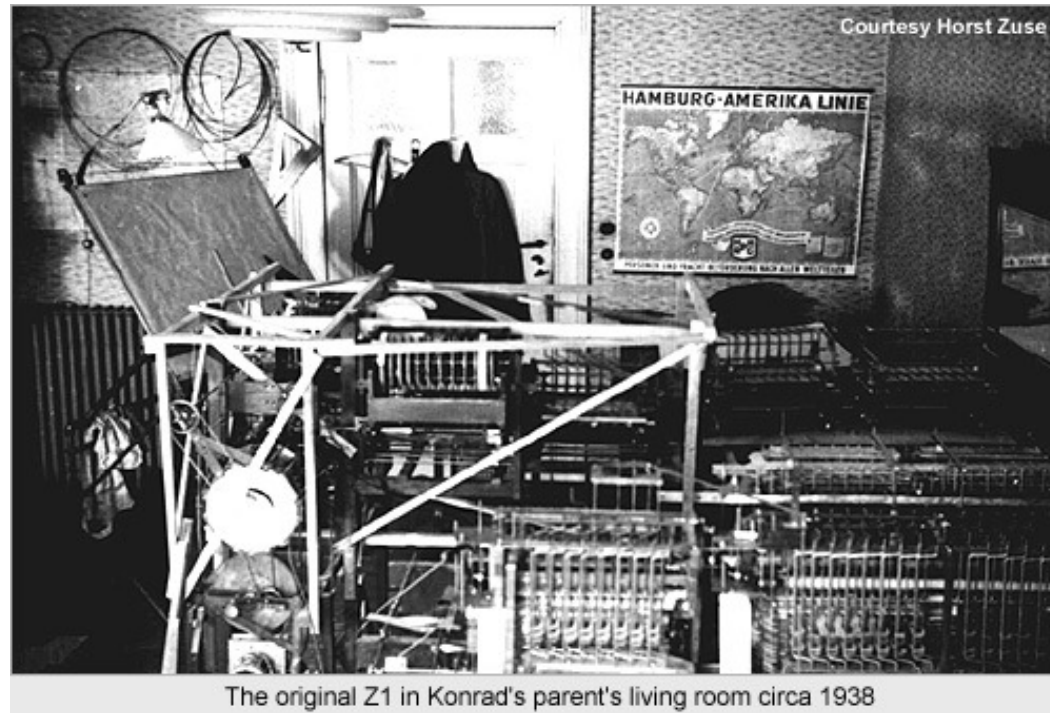
Report a Problem

Verify



Konrad Zuse

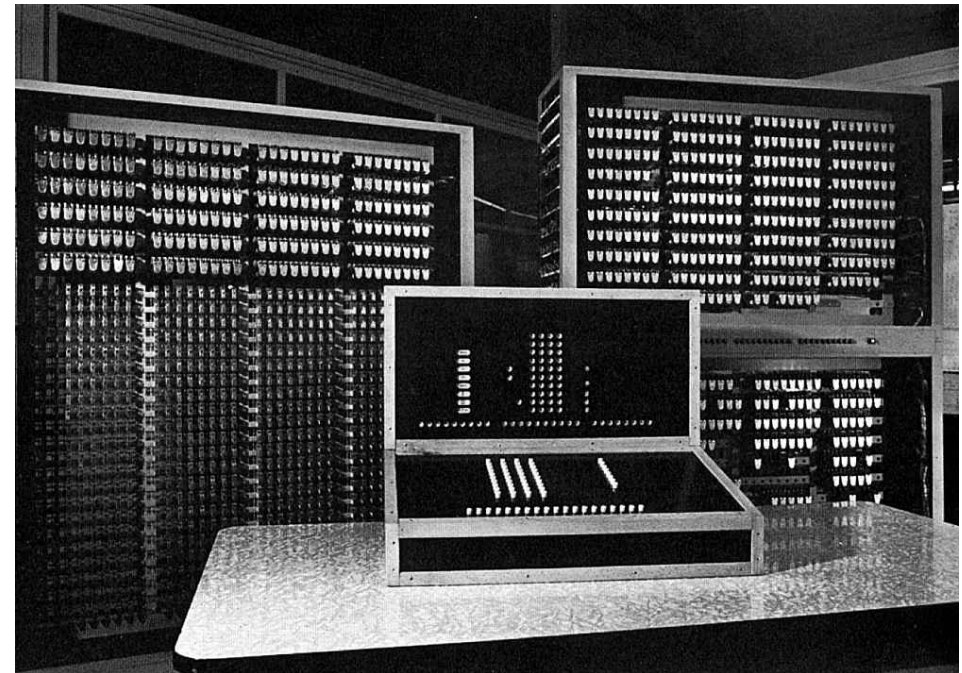
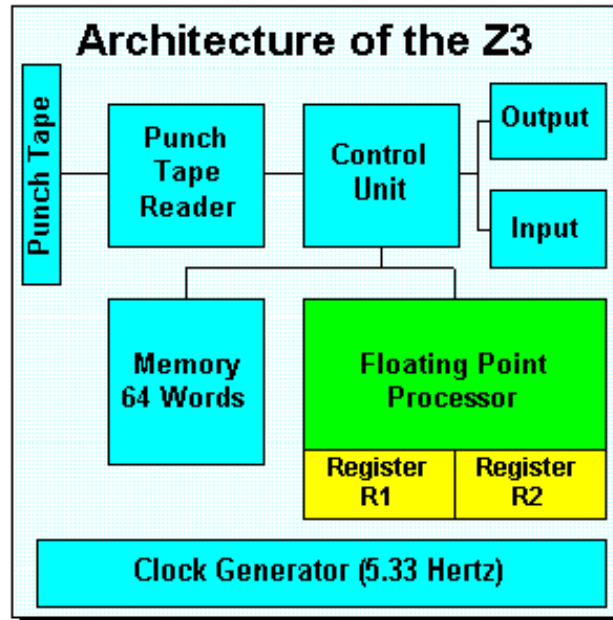
- Z1:
  - 30000 parts
  - calculations were performed in binary (while input and displayed in decimal...)
  - floating point operations
  - freely programmable via punch cards





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:  $440 \times 10^{15}$  flop/s (= 440 PetaFlops)
  - freely programmable (not via punch cards...)
  - operation system: Linux (Red Hat)

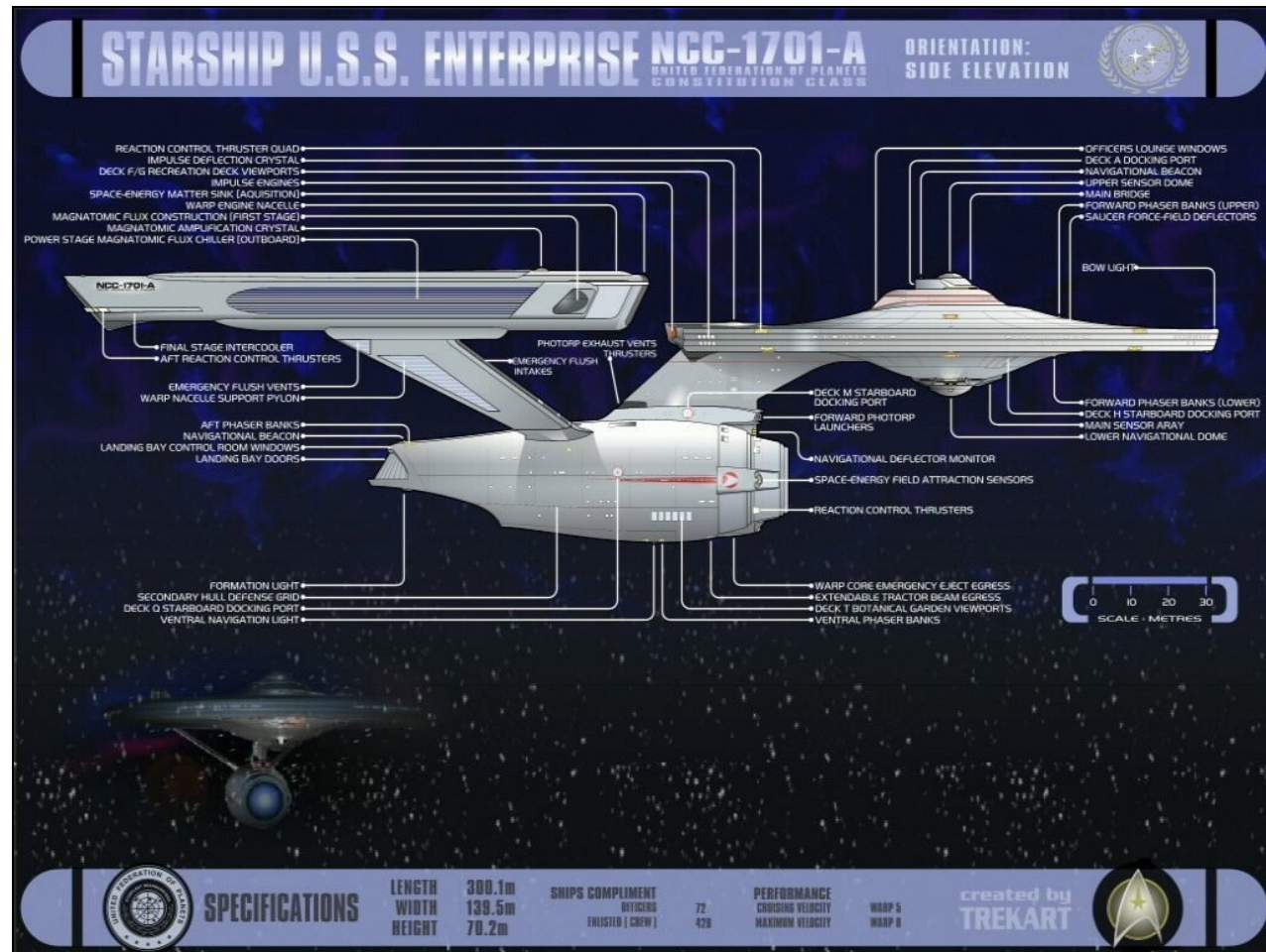


***RIKEN Center for Computational Science in Kobe***

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



*Torre Girona Chapel, Barcelona*

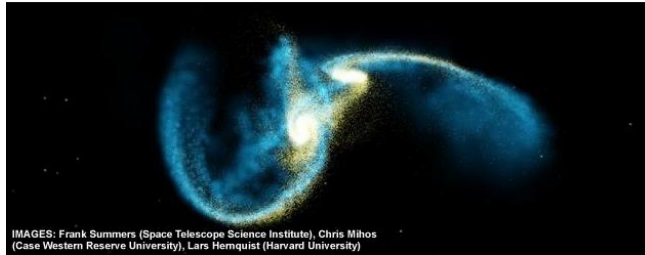
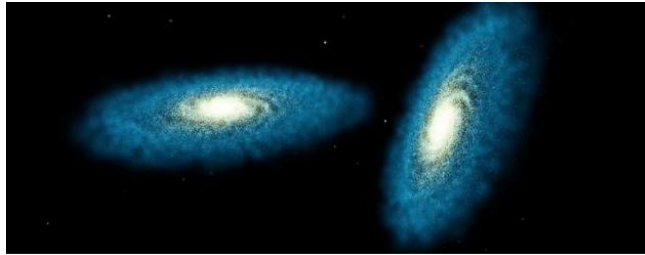


...but what about the science?



# COMPUTATIONAL PHYSICS

---



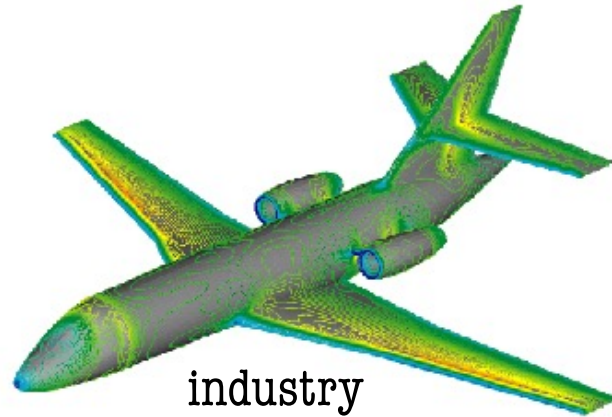
IMAGES: Frank Summers (Space Telescope Science Institute), Chris Mihos (Case Western Reserve University), Lars Hemquist (Harvard University)

astrophysics

biology

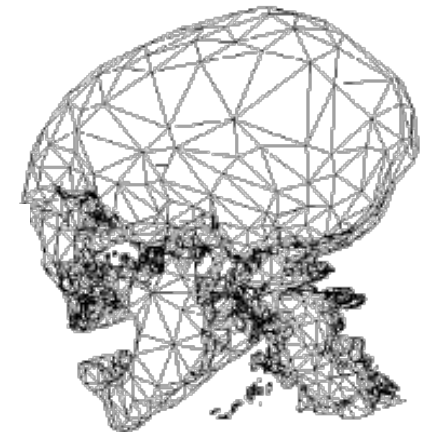


stock market



industry

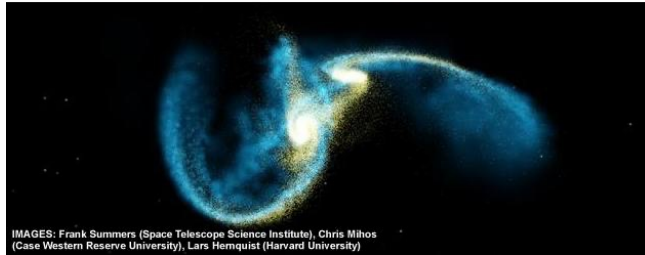
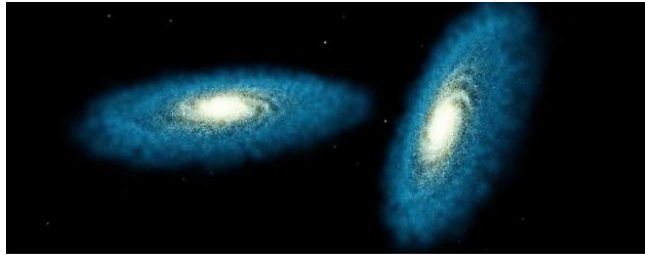
entertainment



medicine

# COMPUTATIONAL PHYSICS

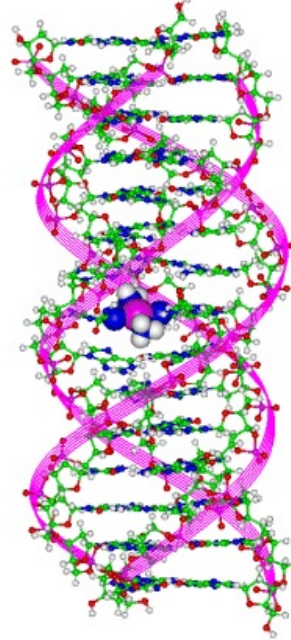
---



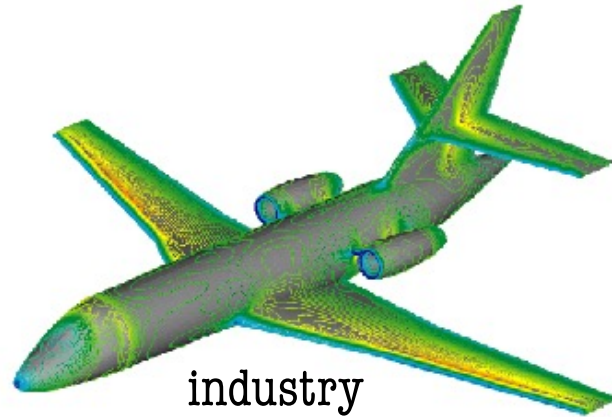
IMAGES: Frank Summers (Space Telescope Science Institute), Chris Mihos (Case Western Reserve University), Lars Hemquist (Harvard University)

*astrophysics*

biology

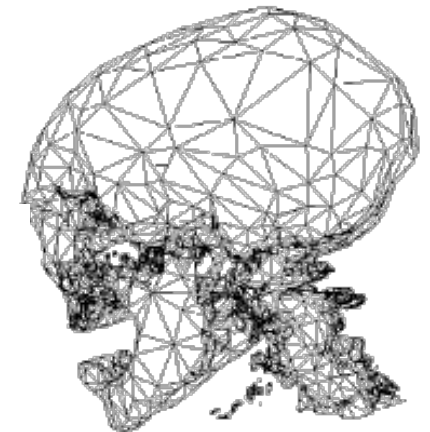


stock market



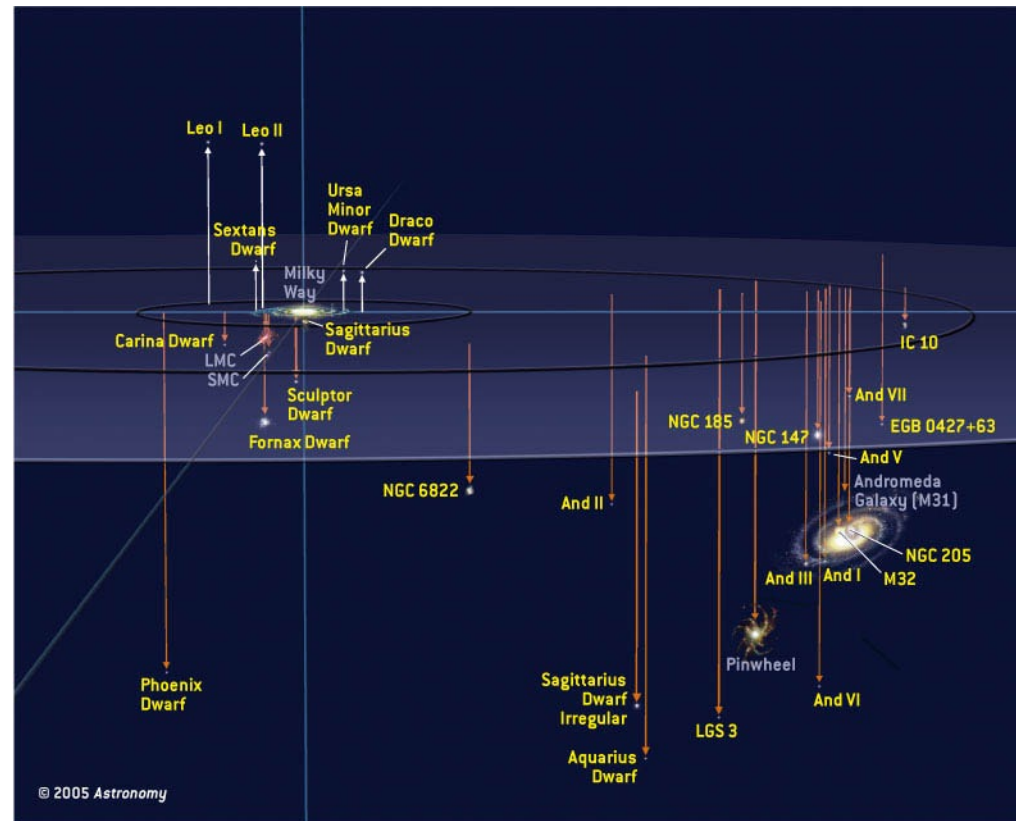
industry

entertainment



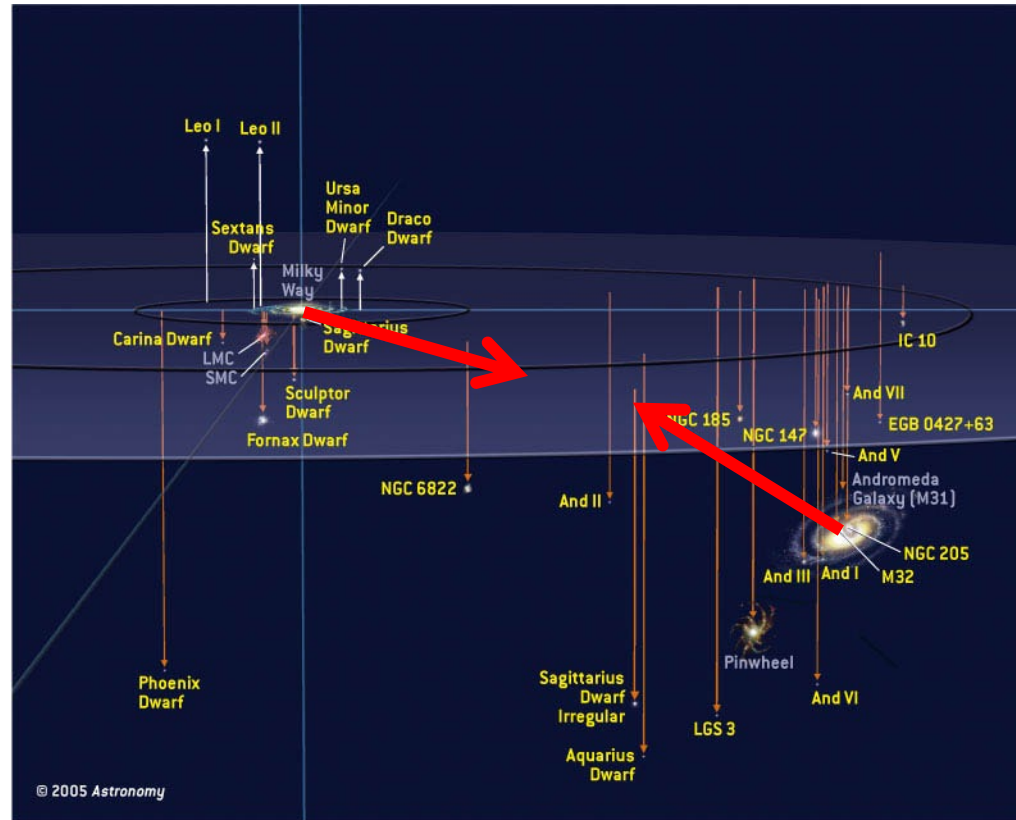
medicine

# The collision of our Milky Way with the Andromeda Galaxy!





The collision of our Milky Way with the Andromeda Galaxy!



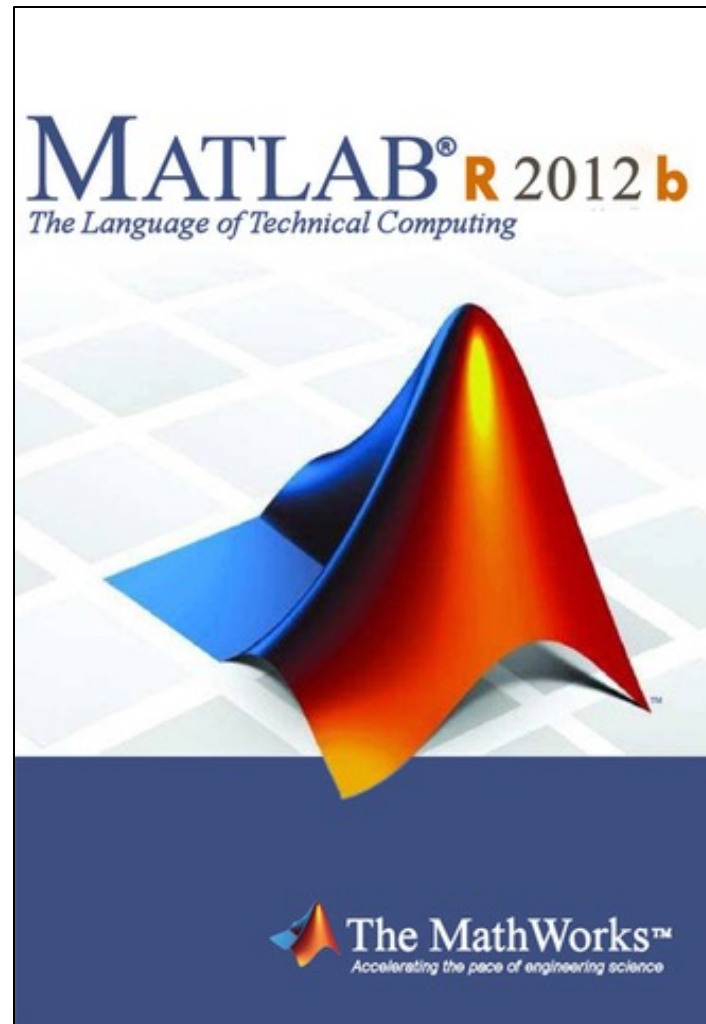
The collision of our Milky Way with the Andromeda Galaxy!



(courtesy Arman Khalatyan)

...and what about “Computacion I”?

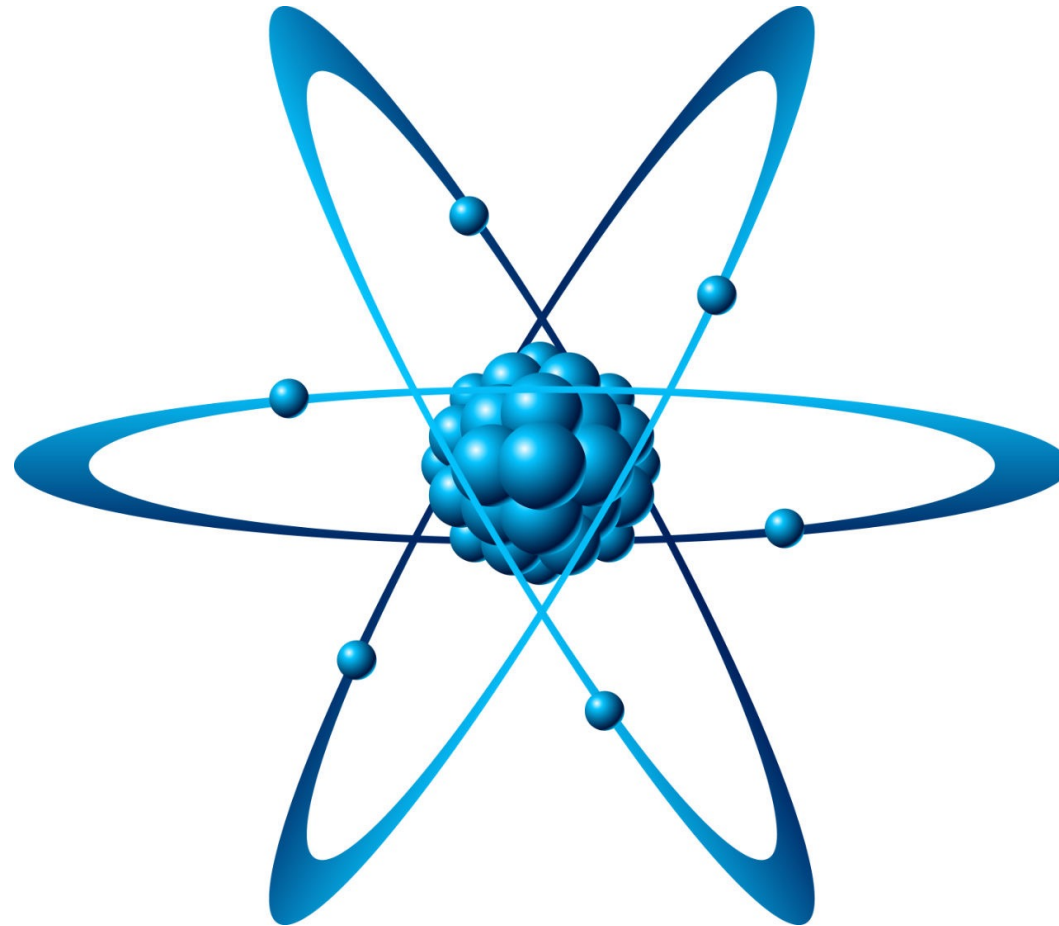
- Content
- Itinerary
- Motivation
- **MATLAB**



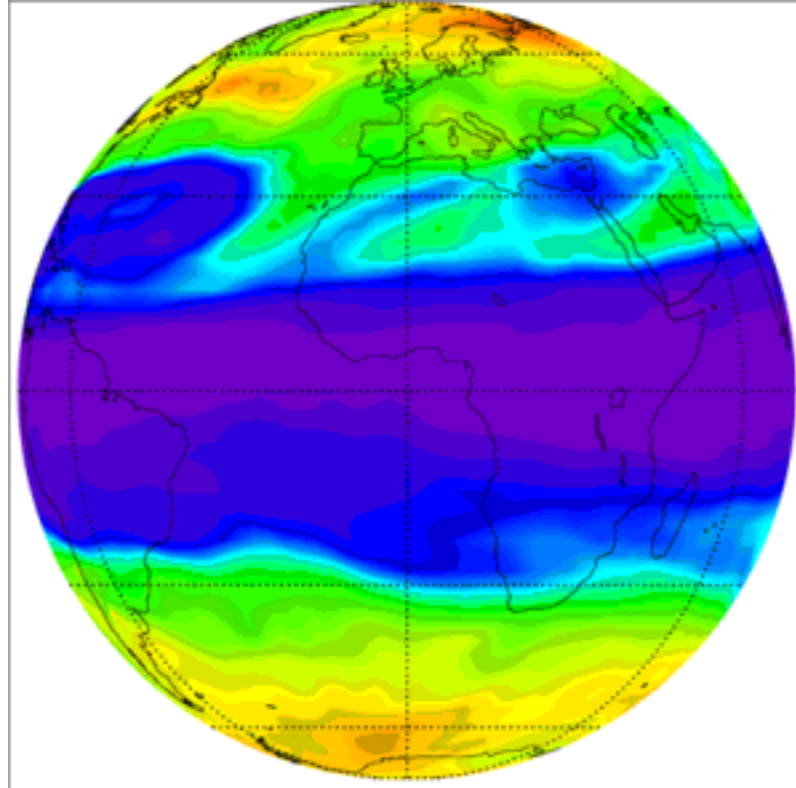


- MATLAB = **MAT**rix **LAB**oratory
- interactive program for computation and visualisation
- array processing language
  
- workhorse to better understand physics, but **not** a programming language

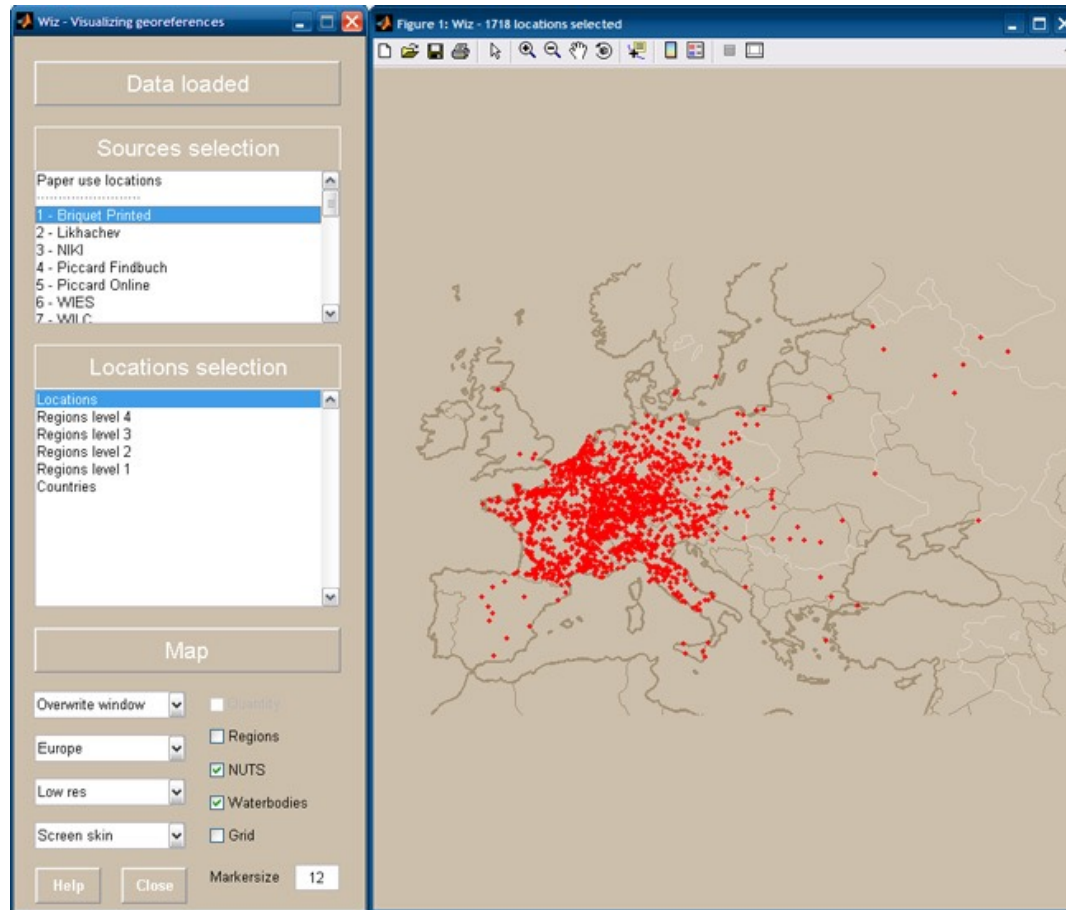
- applications in...physics



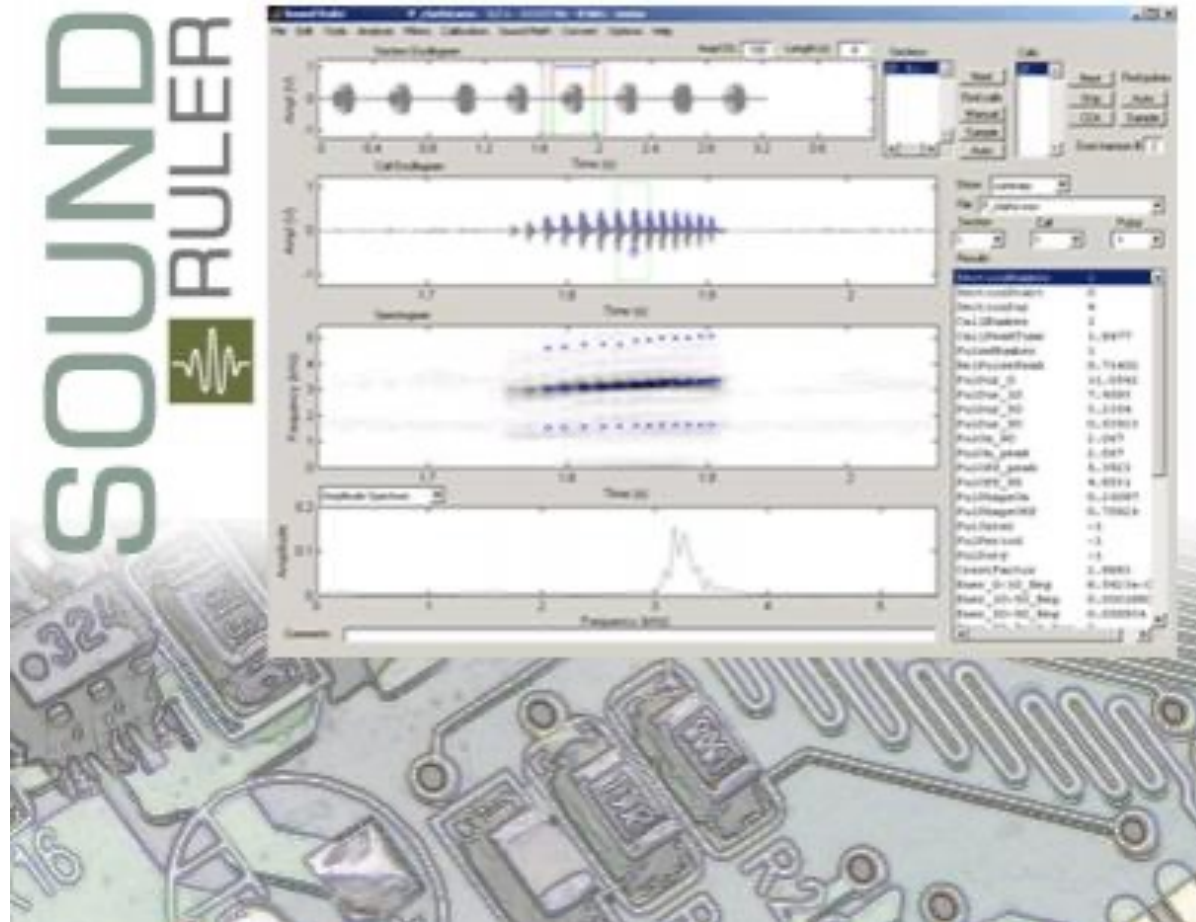
- applications in...meteorology



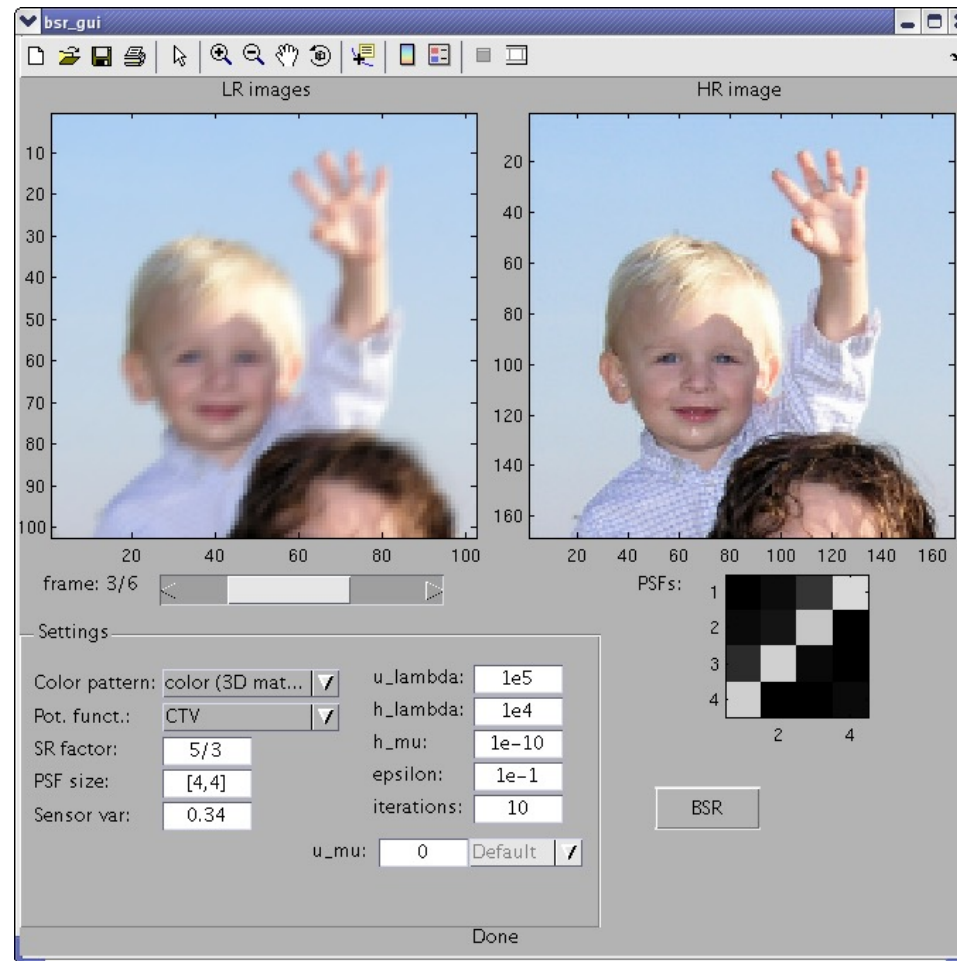
- applications in...population studies



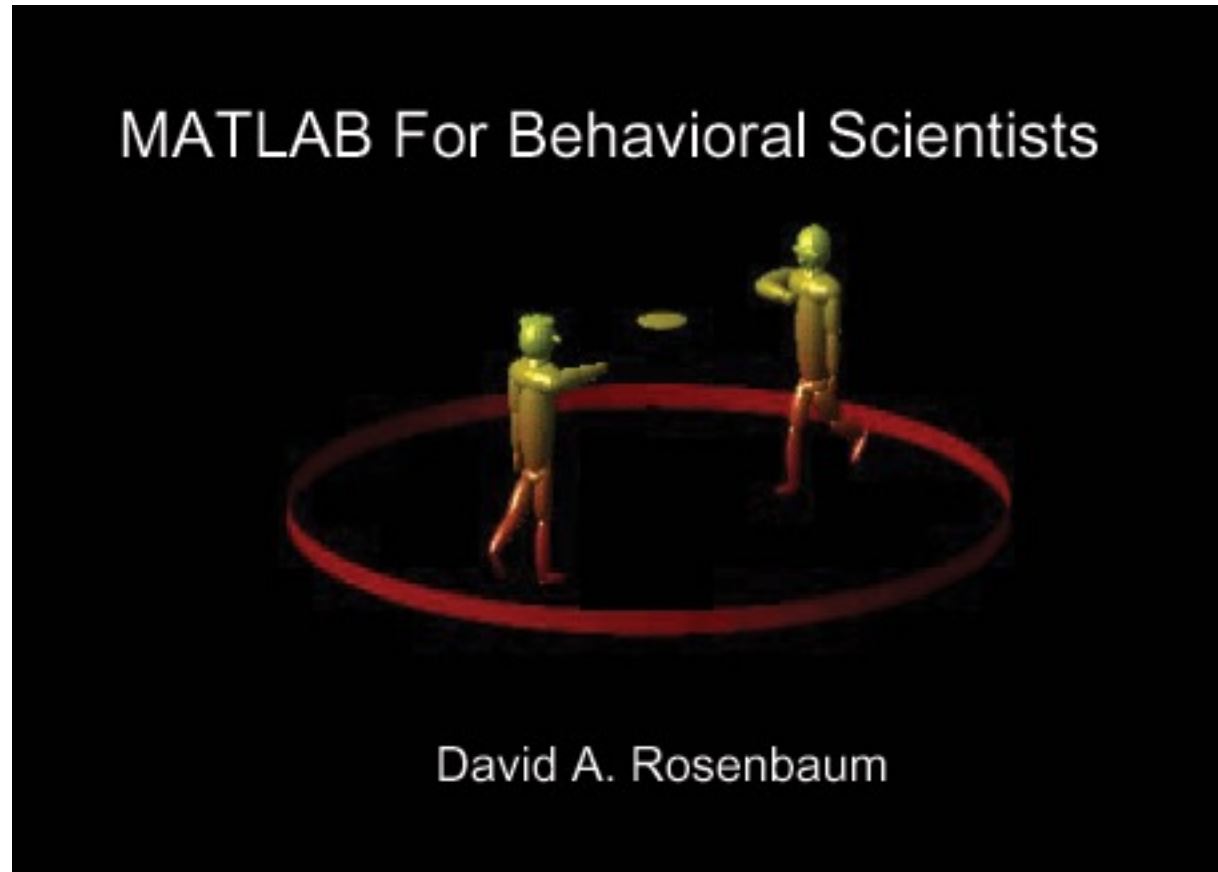
- applications in...sound effects



- applications in...image processing



- applications in...psychology



- licensed software!\*



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

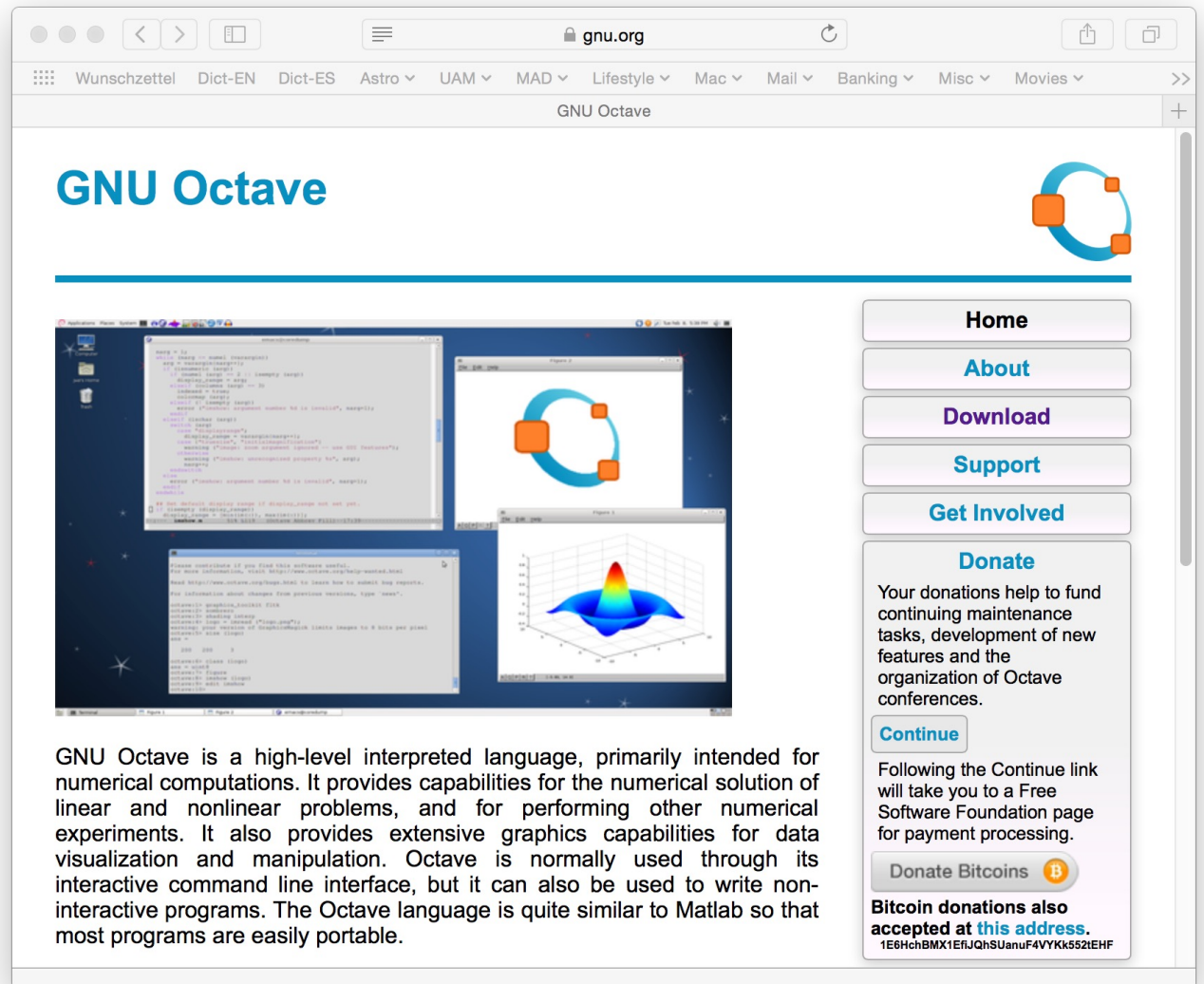
<https://www.uam.es/uam/media/doc/1606860596895/matlab-manual-instalacion-alumnos.pdf>



- open source software:

The screenshot shows the Scilab website homepage. At the top, there is a navigation menu with links for Scilab, Download, Resources, Community, Projects, and Development. A large blue banner features the Scilab logo and a white arrow pointing to a 'Download Scilab' button. Below the button, it specifies 'Version 5.5.0 • Mac OS X - 64bits • 135.49 MB Other Systems' and states 'Open source software for numerical computation'. To the right of the banner is a 'Scilab Console' window showing a file browser and a variable browser. The console displays the startup execution process, including 'loading initial environment' and a command 'a = rand(4,4)'. Below the banner, there is a 'News' section with a link to 'ilab, Free numerical computation for Education, Research and Indury - Online Cx' and social media icons. The main content area is divided into several sections: 'Professional Solutions' with the Scilab Enterprises logo and a description of their services; 'Open Source' and 'Education' sections providing details about the software's license and its use in academic institutions; and a 'To donate' section with a 'To Donate click here' button and the text 'Scilab needs your help'. On the right side, there is a 'Scilab' section with a list of links: 'Overview', 'New in Scilab 5.5.0', 'Xcos', 'Features', 'Gallery', 'System requirements', and 'Quality', accompanied by a colorful wave logo. At the bottom right, the Scilab Enterprises logo is displayed. The footer contains the copyright notice '© Scilab Enterprises S.A.S 2014 - All rights reserved.' and a list of links: 'Donate / Site Map / Legal Notice / Contact'.

- open source software:




**GNU Octave**

Home  
About  
Download  
Support  
Get Involved  
Donate

Your donations help to fund continuing maintenance tasks, development of new features and the organization of Octave conferences.

Continue

Following the Continue link will take you to a Free Software Foundation page for payment processing.

Donate Bitcoins 

Bitcoin donations also accepted at [this address](https://www.gnu.org/donations).  
1E6HchBMX1EfiJQhSUanuF4VYKk552iEHF

GNU Octave is a high-level interpreted language, primarily intended for numerical computations. It provides capabilities for the numerical solution of linear and nonlinear problems, and for performing other numerical experiments. It also provides extensive graphics capabilities for data visualization and manipulation. Octave is normally used through its interactive command line interface, but it can also be used to write non-interactive programs. The Octave language is quite similar to Matlab so that most programs are easily portable.



# 1. logon to computer:

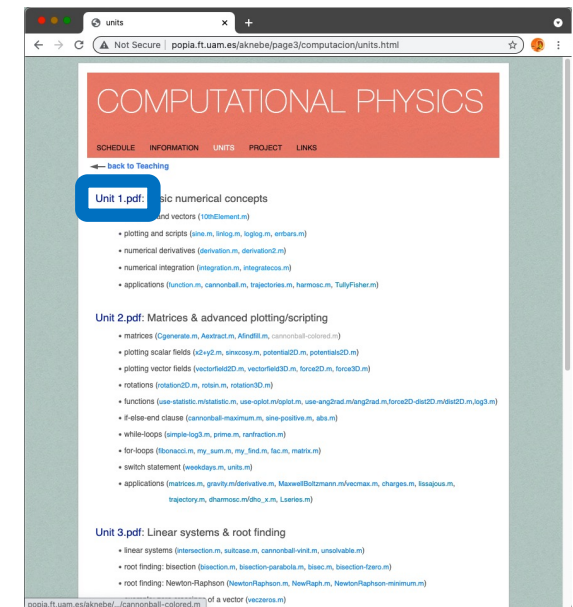
user = nombre.apellido@estudiante.uam.es  
password = same as for email

# 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:

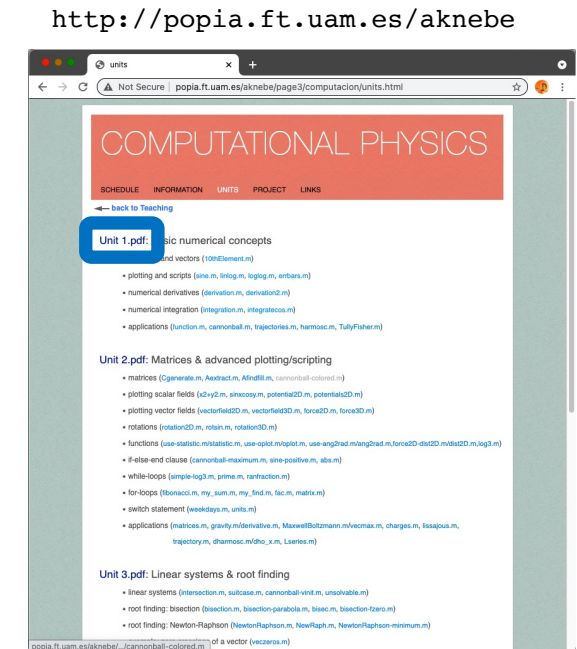
user = nombre.apellido@estudiante.uam.es  
 password = same as for email

any problems? ⇒ contact CAU

# 2. open MATLAB

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

# 4. start with exercises on your



- 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**