



Computational Astrophysics

Violeta González Pérez

2022

Computational Astrophysics

Introduction to Computational Astrophysics

Concepts of High-Performance Computing

Numerical methods

Numerical Integration

Solving ordinary differential equations

Summary of Astrophysical processes

Boltzman equation for a system of

N-bodies

Gravity

(Magneto-)Hydrodynamics

Gravity Solver, Tree codes

Direct simulation

Eulerian methods: PM, AMR

Lagrangian methods: trees and multiple expansions

Hybrid methods: TreePM, (A)P3M

Gravity Solver, Grid codes

Eulerian methods: AMR

Lagrangian methods: SPH

What is your background?

Have you programmed before? What about C? and in parallel? What do you want out of this course? (the coding tutorials are highly adaptable!)

Computational Astrophysics: Lecturers

- Prof. Alexander Knebe, Mod. 8 316, alexander.knebe@uam.es
- Dr. Daniel Ceverino, Mod. 8 303, daniel.ceverino@uam.es
- Prof. Gustavo Yepes, Mod. 8 307, gustavo.yepes@uam.es
- Dr. Violeta González Pérez, Mod. 8 314, violeta.gonzalez@uam.es

Computational Astrophysics: Summary guide

- Course website: <http://popia.ft.uam.es/aknebe/page3/compastro>
- Theory on Thursdays (15pm to 17pm).
- Coding Tutorials on Fridays (12 to 14pm), except the first week.
- Classes will take place in Aula 01.15.SS.201
- Evaluation in 2 parts that need to be passed independently:
 - Attempt to solve 3 problems (50%):
 1. The Mandelbrot series.
 2. The difference between two distinct integration schemes for the equations of motion for two self gravitating bodies.
 3. A 1D code for solving the equations of gas dynamics using the Lagrangian SPH method.
 - Individual project (50%), it can consist of:
 - a) Using an existing professional code for the study of an astrophysical system (solar system, galaxy collision, cosmic structure formation).
 - b) Write your own code for approaching a physical phenomenon.
 - c) Literature research about one of the topics of the course.

Computational Astrophysics: Schedule

ACO classes 2021/22

day	date	time	teacher	topic	comments
Thu	24/03/2022	15-17	VGP	Introduction	
Fri	25/03/2022	12-14	VGP	HPC	
Thu	31/03/2022	15-17	VGP	Numerics Review	
Fri	01/04/2022	12-14	AK	Coding Tutorial	
Thu	07/04/2022	15-17	VGP	Physical Processes	
Fri	08/04/2022	12-14	AK	Coding Tutorial	Mandelbrot handout, Project discussion
Thu	14/04/2022		-----	-----	semana santa
Fri	15/04/2022		-----	-----	semana santa
Thu	21/04/2022	15-17	VGP	Tree Codes	
Fri	22/04/2022	12-14	AK	Coding Tutorial	Kepler handout, Mandelbrot solution
Thu	28/04/2022	15-17	DC	grid N-body	
Fri	29/04/2022	12-14	AK	Coding Tutorial	SPH handout, Kepler solution
Thu	05/05/2022	15-17	GY	Hydrodynamics	
Fri	06/05/2022	12-14	AK	Coding Tutorial	
Thu	12/05/2022	15-17	GY	Hydrodynamics	
Fri	13/05/2022	12-14	AK	Coding Tutorial	
Thu	19/05/2022	15-17	GY	Hydrodynamics	
Fri	20/05/2022	12-14	AK	Coding Tutorial	SPH discussion
Thu	26/05/2022		all	project presentations	
teachers	Alexander Knebe (AK), Violeta Gonzalez-Perez (VGP), Gustavo Yepes (GY), Daniel Ceverino (DC)				

Coding tutorials: weekly excersises

popia.ft.uam.es

Wunschzettel Dict-EN Dict-ES Astro UAM MAD Lifestyle Mac Mail Banking Misc Movies Newspaper Music Shopping Anja AK

Computational astrophysics

COMPUTATIONAL ASTROPHYSICS

HOME LECTURES **EXERCISES** PROJECT TEACHERS LINKS

← [back to Teaching](#)

The course is a mixture between actual class room lectures and hands-on coding exercises.

hands-on exercises:

- [Makefile](#)
- [hello world](#)
- [stdint](#)
- [foverflow](#)
- [pointer](#)
- [array 1D](#)
- [array 3D](#)
- [parallel recursion 1D](#)
- [improved parallel recursion 1D](#)
- [structures](#)
- [arrays of structures](#)
- [structure pointer](#)
- [function pointer](#)
- [index\(\) usage](#)
- [qsort\(\) usage](#)
- [I/O \(DarkMatter-Halos.txt\)](#)
- [read_mtree \(MergerTree.txt\)](#)
- [example for valgrind](#)

[utility.c / utility.h](#)

→ solutions to in-class exercises

Evaluation: attempt to write code for 3 problems (50%)

In order to pass this subject you need to attempt the following 3 exercises:

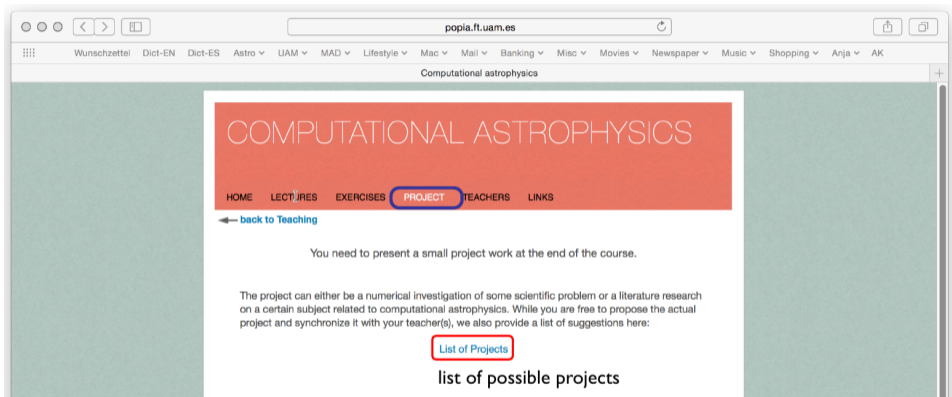
hello world
stdint
overflow
pointer
array 1D
array 3D
parallel recursion 1D
improved parallel recursion 1D
structures
arrays of structures
structure pointer
function pointer
index() usage
qsort() usage
I/O (DarkMatterHaloes.txt)
read_mtree (MergerTree.txt)
example for valgrind

utility.c / utility.h
my.c / my.h

homework exercises:
Mandelbrot code (mandelbrot.pdf)
Kepler problem (kepler.pdf)
SPH code (sph.pdf)

→ homework exercises
and their solutions

Evaluation: individual projects (50%)



The screenshot shows a web browser window with the URL `popia.ft.uam.es`. The page title is "Computational astrophysics". The navigation menu includes: HOME, LECTURES, EXERCISES, PROJECT (highlighted with a blue circle), TEACHERS, and LINKS. Below the menu is a blue link: [← back to Teaching](#). The main content area contains the following text:

You need to present a small project work at the end of the course.

The project can either be a numerical investigation of some scientific problem or a literature research on a certain subject related to computational astrophysics. While you are free to propose the actual project and synchronize it with your teacher(s), we also provide a list of suggestions here:

[List of Projects](#) (highlighted with a red circle)

list of possible projects

You can also come up with your own project. Talk to us!

Evaluation: retake exams

Students will only be permitted to attend the retake exam if they fail one or both of the evaluable parts (excercises and project).

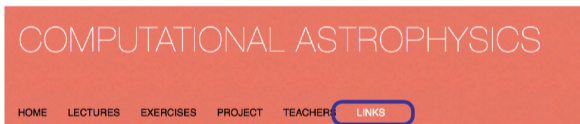
The retake exam will be a written exam, lasting 2 hours. No books will be permitted.

Coding in C

```
#include <stdio.h>
int main() {
    // printf() displays the string inside quotation
    printf("Hello, World! \n");
    return 0;
}
```

```
> gcc -o p hello.c
> ./p
```

To compile, in general, you will be using the **Makefile** provided in the course website:
<http://popia.ft.uam.es/aknebe/page3/files/ComputationalAstrophysics/exercises/Makefile>.



← back to Teaching

Programming Guides (by Steffen Knollmann):

C tutorial
parallel programming guide



excellent
programming guides

Books

"Computer Simulations using Particles", R.W. Hockney & J.W. Eastwood
"Gravitational N-Body Simulations: Tools and Algorithms", S. Aarseth
"Riemann Solvers and Numerical Methods for Fluid Dynamics", E. Toro

Coding in C: set-up

Throughout this course, you are going to do hands-on coding in C. Thus, you will need to install on your laptop:

- The **gcc** compiler:
 - Mac: <http://hpc.sourceforge.net/>.
 - Other OS: <https://gcc.gnu.org/>.
- A way to write your code:
 - In Linux and Mac you have already available a Terminal application and an editor (vi, emacs, gedit, etc.).
 - In Windows you could install the Windows Subsystem for Linux (WLS), use Visual Studio or an other integrated development environment (IDE) or even use the virtual linux in the UAM virtual PCs (you will need the UAM VPN) plus OneDrive;
<https://servidorlibreuam.com/pc-virtual-de-la-uam/>.

What tools are you going to use?

Do you know what you're going to use? Do you want to clarify any of the above now?

