1. OVERVIEW

Subject area	Scientific Computing II
Degree	Bachelor's Degree in Physics
School/Faculty	Architecture, Engineering and Design
Year	First
ECTS	6 ECTS
Type	Compulsory
Language(s)	Spanish
Delivery mode	On campus
Semester	Second semester

2. INTRODUCTION

Scientific Computing II is a compulsory subject area on the Bachelor's Degree in Physics at Universidad Europea de Madrid. It is the natural continuation of the subject area Scientific Computing I, which lays the foundations for algorithmic thought and basic data structures, and both subject areas provide a solid basis in computational physics.

Scientific Computing II focuses on developing numerical methods by using specialist Python libraries, such as NumPY, Matplotlib and others, which are extremely powerful in terms of calculation. It also refers to the Matlab platform as an alternative work environment. The main aim of the subject area is to solve problems using numerical methods through algorithms in a specific programming language, to understand their strengths, to understand the method and how it has been implemented. The content is based on topics such as solving equations, data interpolation and fitting, numerical differentiation and integration, solving ordinary differential equations, etc. The selection of numerical methods in each topic leans towards their relevance in relation to scientific and technical problems.

3. SKILLS AND LEARNING OUTCOMES

Key skills (CB, by the acronym in Spanish):

- CG02. Ability to plan and perform independent work when managing projects associated with different areas of physics.
- CB03. Students have the ability to gather and interpret relevant data (usually within their study area) to form opinions which include reflecting on relevant social, scientific or ethical matters.
- CB05. Students have developed the learning skills necessary to undertake further study in a much more independent manner.

Transversal skills (CT, by the acronym in Spanish):

- CT05. Problem solving: Be able to critically evaluate information, separate complex situations into their constituent parts, recognise patterns, and consider alternatives, different approaches and perspectives in order to find optimal solutions and negotiate efficiently.

- CT06. Adaptability: Being able to accept, appreciate and integrate different positions, being able to adapt one's own approach as required by the situation, as well as working effectively in ambiguous situations.

Specific skills (CE, by the acronym in Spanish):

- CE05. To understand and know how to use the mathematical and numerical methods used in physics and in handling experimental data.
- CE07. To use the most suitable electronic instruments and IT tools to study physical problems and search for solutions.

Learning outcomes (RA, by the acronym in Spanish):

- RA01. To be able to rigorously analyse experimental measures using integrated numerical software packages.
- RA02. To know how to properly apply the method of least squares in the context of scientific data processing.
- RA03. To know a broad range of elementary numerical analysis algorithms applicable to algebra, as well as differentiation and integration algorithms.

The following table shows how the skills developed in the subject area match up with the intended learning outcomes:

Skills	Learning outcomes
CG02, CB03, CE05, CE07	**RA01.** To be able to rigorously analyse experimental measures using integrated numerical software packages.
CB03, CT05, CT06, CE05, CE07	**RA02.** To know how to properly apply the method of least squares in the context of scientific data processing.
CG02, CB05, CT05, CT06, CE05, CE07	**RA03.** To know a broad range of elementary numerical analysis algorithms applicable to algebra, as well as differentiation and integration algorithms.

4. CONTENTS

The subject is organised into six learning units which, in turn, are divided into topics (four or five topics depending on the unit):

1. Mathematical software packages (Python + NumPY/MATLAB).
2. Data fitting and interpolation. Global and local methods. Least squares.
3. Algebraic equations.
4. Eigenvalues and eigenvectors.
5. Differential initial value problems: Euler and Runge–Kutta methods.
6. Numerical differentiation and integration. Finite differences.

A more detailed description of the syllabus is provided below.

- **UNIT 1. Introduction and numerical representation:** Number representation in computers, IEEE Standard for Floating-Point Arithmetic (IEEE 754), errors due to rounding and approximation, machine epsilon. Recap of programming in Python, use of the NumPY library.

- **UNIT 2. Curve fitting with least squares**: Linear least squares, fitting a straight line, linearisation, curve fitting using normal equations, polynomial curve fitting, multi-variable curve fitting.
- **UNIT 3. Finding roots of non-linear equations**. Bisection method, false position method, Newton method, secant method. Systems of non-linear equations, Newton–Raphson method.
- **UNIT 4. Interpolation methods:** Polynomial interpolation, Vandermonde matrix, Legendre polynomials, Newton's divided differences interpolation polynomial, natural cubic spline interpolation and Hermite polynomials.
- **UNIT 5. Numerical differentiation and integration**: Numerical differentiation, divided difference formulas, numerical integration using the trapezoidal rule, Simpson's rule and Gaussian quadrature.
- **UNIT 6. Numerical resolution of ordinary differential equations**: ODE with Taylor, 2nd Order Runge–Kutta and 1 4th Order Runge–Kutta. Applications.

5. TEACHING/LEARNING METHODS

The types of teaching/learning methods are as follows:

- Collaborative learning: Students learn to collaborate with other people (classmates and professors) in order to find creative, comprehensive and constructive solutions to questions and problems that arise from the given case studies, using all relevant knowledge and material resources available.
- Problem-based learning: Students are given problems and asked to solve them, working individually or in groups.
- Lectures: Presentations by the professor with the necessary technological tools to maximise comprehension of the learning content.
- Workshop-based learning: Students acquire knowledge through learning to use the tools and equipment needed in their profession. In other words, "learning by doing".
- Guided academic activities: Individual and group work that is more independent, including information searches, written summaries, debates and the public defence of projects.

6. LEARNING ACTIVITIES

The types of learning activities, plus the amount of time spent on each activity, are as follows:

On campus:

Learning activity	Number of hours
Lectures	26
Oral presentations of projects and debates	2
Report writing	8
Assessment	6
Practical activities (problems, written work, projects, workshops and/or lab work)	38
Tutorials	16
Independent working	54

TOTAL	150

7. ASSESSMENT

The assessment systems, plus their weighting in the final grade for the subject area, are as follows:

Assessment system	Weighting
Individual on-campus knowledge tests (theory and/or practice)	50%
Oral defence	5%
Submission of group and/or individual reports, written work, projects or exercises	35%
Performance observation	10%

On the Virtual Campus, when you open the subject area, you'll find details of your assessable tasks, including the submission dates and assessment procedures for each task.

8. BIBLIOGRAPHY

The reference material for the subject area is as follows:

- Jaan Kiusallas (2013). Numerical Methods in Engineering with Python 3. Cambridge University Press
- B.S. Grewal (2018). Numerical Methods in Engineering and Science. Mercury Learning and Information
- Abhijit Kar Gupta (2018). Scientific Computing in Python. Amazon Media