COURSE OUTLINE

(1) GENERAL

SCHOOL	ENGINEERING		
ACADEMIC UNIT	Department of Electrical and Electronics Engineering		
LEVEL OF STUDIES	Graduate (MSc)		
COURSE CODE	A.02	SEMESTER	01
COURSE TITLE	Scientific Computing and Mathematical Modeling		

INDEPENDENT TEACHING ACTIVITIES *if credits are awarded for separate components of the course, e.g. lectures, laboratory exercises, etc. If the credits are awarded for the whole of the course, give the weekly teaching hours and the total credits*	WEEKLY TEACHING HOURS	CREDITS
Lectures	3	6
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).		

COURSE TYPE *general background, special background, specialised general knowledge, skills development*	General Background
PREREQUISITE COURSES:	1. An undergraduate course on Mathematical Analysis 2. An undergraduate course on Introduction to Linear Algebra 3. An undergraduate course on programming (Matlab, Python, Julia, R, …) 4. An undergraduate course on Numerical Analysis (optional).
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek and English
IS THE COURSE OFFERED TO ERASMUS STUDENTS	YES
COURSE WEBSITE (URL)	

(2) LEARNING OUTCOMES

Learning outcomes

The course learning outcomes, specific knowledge, skills and competences of an appropriate level, which the students will acquire with the successful completion of the course are described.

Consult Appendix A

- *Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area*
- *Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning and Appendix B*
- *Guidelines for writing Learning Outcomes*

Upon successful completion of the course, students are expected to be able to:

- comprehend basic scientific programming methodologies for solving mathematical problems,
- implement solutions using the capabilities provided by modern scientific programming environments rather than programming them from scratch;
- understand the mathematical framework of the problem they want to solve,
- analyze the mathematical problem and choose the appropriate parameters to use,
- argue for the appropriate solution method,
- develop solutions by selecting and applying the appropriate tools provided by modern

- computing environments,
- analyze, evaluate and compare the solutions to other available,
- develop reports that present the calculations results and evaluate with arguments their correctness and quality characteristics.

General Competences

Taking into consideration the general competences that the degree-holder must acquire (as these appear in the Diploma Supplement and appear below), at which of the following does the course aim?

Search for, analysis and synthesis of data and information, with the use of the necessary technology	*Project planning and management*
Adapting to new situations	*Respect for difference and multiculturalism*
Decision-making	*Respect for the natural environment*
Working independently	*Showing social, professional and ethical responsibility and sensitivity to gender issues*
Team work	*Criticism and self-criticism*
Working in an international environment	*Production of free, creative and inductive thinking*
Working in an interdisciplinary environment	*……*
Production of new research ideas	*Others…*
	…….

- Analytical and synthetic work with complex Mathematical concepts to solve problems in basic fields of science and Engineering.
- Use of modern Mathematical Software for the implementation of solutions in scientific programming environments.
- Autonomous work.
- Teamwork.
- Ability to convert basic physical problems into corresponding mathematical-computer problems.
- Production of free, creative and inductive thinking.
- Analysis and synthesis of Mathematical processes with the use of the computer.
- Working in an interdisciplinary environment.
- Critical thinking and decision making depending on the solution of the Mathematical Problem.

(3) SYLLABUS

The course syllabus consists of the following units.

Unit 1: Mathematical Modeling
Deterministic and stochastic mathematical models. Mathematical modeling with dynamic systems and differential equations.

Unit 2: Introduction to Scientific Programming (S.P.), Modern S.P. Environments. Computer Errors
Solving mathematical problems in scientific programming environments (Matlab, Mathematica, Python, Fortran). Numerical and symbolic calculations on a computer. Double, quadruple and higher precision calculations. Numerical calculation errors on the computer.

Unit 3: Numerical Linear Algebra in S.P. environments
Numerical Linear Algebra Methodologies in an S.P. environment. (solving linear systems, factorizations of matrices, calculation of eigenvalues, SVD).

Unit 4: Methodologies of approximation of functions and scientific data in S.P. environments.
Interpolation and Approximation of functions and data. Interpolatory Procedures. Least Squares Approximation. Statistical processing and data analysis methodologies.

Unit 5: Optimization Methodologies in S.P. Environments
Optimization Methodologies with or without conditions. Finding minimum of cost functions with classical or differential-evolutionary algorithms. Solving equations of non-linear systems.

Unit 6: Differentiation, Integration, Differential Equations
Numerical Integration and Differentiation. Numerical Solution of Ordinary Differential Equations. Methodologies of solving Partial Differential Equations.

Unit 7: Introduction of parallel computation in modern S.P. Environments

(4) TEACHING and LEARNING METHODS - EVALUATION

DELIVERY *Face-to-face, Distance learning, etc.*	Distance Learning (Synchronous, MS Teams)
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY *Use of ICT in teaching, laboratory education, communication with students*	• MS Teams for distance learning classes (weekly) • E-class for course content support and teacher-student-class communication • Mathematical software and tools (Matlab, Mathematica, Python, Fortran) for the subjects taught.
TEACHING METHODS *The manner and methods of teaching are described in detail.* *Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, placements, clinical practice, art workshop, interactive teaching, educational visits, project, essay writing, artistic creativity, etc.* *The student's study hours for each learning activity are given as well as the hours of non-directed study according to the principles of the ECTS*	see table below
STUDENT PERFORMANCE EVALUATION *Description of the evaluation procedure* *Language of evaluation, methods of evaluation, summative or conclusive, multiple choice questionnaires, short-answer questions, open-ended questions, problem solving, written work, essay/report, oral examination, public presentation, laboratory work, clinical examination of patient, art interpretation, other* *Specifically-defined evaluation criteria are given, and if and where they are accessible to students.*	Student performance evaluation comes form: • Participation in the educational process and contribution to discussions that take place (20% of the final grade) • Assignment average (best 3 out of a total of 4, 40% of final grade). Assignments are submitted via eclass. • Final written exam using a computer (40% of final grade).

Activity	*Semester workload*
Lectures	39
Study learning content	61
Bibliography study, solution of exercises	30
Projects preparation	30
Final Exam preparation	20
Course total	***180***

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

- Numerical Analysis, Burden R., Faires J. D, Brooks\Cole.
- A First Course in Numerical Analysis, A. Ralston, Ph. Rabinowitz, Mc Graw Hill.
- Numerical Methods using Matlab, J. Mathews, K. Fink, Pearson Prentice Hall.
- Applied Numerical Analysis, C. Gerald, P. O. Wheatley, Addison Wesley.
- Applied Numerical Analysis Using Matlab, L. Fausett, Pearson Prentice Hall.
- Numerical Methods for Engineers, With Software and Programming Applications Fourth Edition, S.C. Chapra, R.P. Canale, MC Geaw Hill, 2002
- Numerical Python, Scientific Programming and Data Science Applications with Numpy, Scipy and Matplotlib, R. Johansson, Apress
- Practical Numerical and Scientific Computing with MATLAB and Python", 1st edition, Eihab B. M. Bashie, CRC Press "
- Learning Scientific Programming with Python, Christias Hill

Related Scientific Journals:

- SIAM Journal on Numerical Analysis
- International Journal for Numerical Methods in Engineering
- Applied Numerical Mathematics
- Journal of Computational and Applied Mathematics
- Numerical Algorithms
- Numerische Mathematik
- Scientific Programming

TOOLS

- Matlab: https://www.mathworks.com/products/matlab.html
- Mathematica: https://www.wolfram.com/
- Wolfram Alpha: https://www.wolframalpha.com/
- Python: https://www.python.org/
- scipy: https://scipy.org/
- Julia: https://julialang.org/
- R: https://www.r-project.org/

WEBSITES

- https://scipython.com/
- https://earthlab.colorado.edu/blog/what-scientific-programming-and-why-it-rocks
- https://sciprog.center/
- https://www.opensourceforu.com/2011/05/what-is-scientific-programming/