COURSE OUTLINE

(1) GENERAL

SCHOOL	ENGINEERING		
ACADEMIC UNIT	ELECTRICAL AND COMPUTER ENGINEERING DEPT.		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	**ECE_K611**	**SEMESTER**	**6**
COURSE TITLE	Computational Methods for Engineers		

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	2	
Seminars / Practice exercises	1	
Laboratory	1	
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (4).	4	5

COURSE TYPE *general background, special background, specialised, general knowledge, skills development*	General background
PREREQUISITE COURSES:	No. It is suggested that students have already attended: • Calculus I (ECE_K110), • Linear Algebra (ECE_K120), • Differential Equations (ECE_K220) and • Data Structures & Algorithms (ECE_K320)
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek
IS THE COURSE OFFERED TO ERASMUS STUDENTS	No
COURSE WEBSITE (URL)	https://eclass.uop.gr/courses/ECE611/

(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*

The goal of this course is to introduce and familiarize students with computer methods for solving engineering problems. The course covers fundamental numerical analysis methods and their application to electrical engineering problems using scientific code.

After successfully completing the course, students will be able to:

- Recognize the utility of graphs and apply fundamental graph algorithms (e.g. shortest paths).
- Recognize the difference between analytical and numerical solutions.
- Solve numerical problems involving the root finding of nonlinear algebraic equations, as well as linear and nonlinear algebraic equation systems.

- Understand why and where optimization occurs in engineering problem solving, as well as how to select and apply the appropriate method.
- Understand how to interpolate (estimate) the value of a function between two known values and how to fit curves.
- Understand how to approximate derivatives and definite integrals.
- Understand how to solve initial and boundary value problems numerically.
- To develop scientific code in MATLAB/Octave that use numerical techniques to solve electrical engineering problems.

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…*
	…….

- Search for, analysis and synthesis of data and information, with the use of the necessary technology
- Working independently
- Team work
- Knowledge application in practice
- Production of new research ideas

(3) SYLLABUS

Short Course Description

Theory units – lectures:

Unit 1 – Lecture 1

Graph Algorithms: Representation of graphs, strongly connected components, Minimum Spanning trees (MST), the algorithms of Kruskal and Prim. Single-source shortest paths, the Bellman-Ford algorithm, Dijkstra's algorithm

Unit 2 – Lecture 2

Calculations and errors: Representation of real numbers, absolute and relative error, roundoff errors, truncation errors and Taylor series, error propagation, data uncertainty, interval arithmetic.

Unit 3 – Lectures 3-4

Computational methods for determining roots: Bracketing methods, open methods, roots of polynomials, multiple roots. Systems of nonlinear equations (Newton-Raphson).

Unit 4 – Lectures 5-6

Computational Methods for Solving Systems of Equations: Gauss elimination, LU decomposition and matrix inversion, special matrices and Gauss-Seidel.

Unit 5 – Lectures 7-8

Computational Methods for Optimization: One-Dimensional Unconstrained Optimization, Golden-Section search, parabolic interpolation, Newton's method, Brent's method. Multidimensional Unconstrained Optimization, direct methods, gradient methods, Newton's method.

Unit 6 – Lecture 9
Computational Methods for Curve Fitting: Least-Squares Regression, Interpolation, Lagrange interpolating polynomials, Spline interpolation.

Unit 7 – Lecture 10
Numerical differentiation and integration: Newton-Cotes Integration Formulas (trapezoidal rule, Simpson's 1/3 and 3/8 rule), Romberg integration, Gauss quadrature. Finite-divided-difference formulas (the forward, backward and centered formulas), Richardson extrapolation.

Unit 8 – Lectures 11-12
Computational Methods for Solving ODEs: Euler's method, Taylor's method, Heun's Method, Runge-Kutta (RK) methods. Solution of systems of simultaneous ordinary differential equations. Stiffness and Multistep Methods (Adams-Bashforth, Adams-Multon). Boundary-Value and Eigenvalue Problems: The shooting method, Finite-Difference Methods, Eigenvalue problems.

Unit 9 – Lecture 13
Computational Methods for Solving PDEs. Finite Difference: Elliptic Equations, The Liebmann Method. Finite Difference: Parabolic Equations, Crank-Nicolson method, the ADI scheme.

Laboratory exercises
The computational methods of the theory are applied to real-world problems of the Electrical Engineer in each Unit using the MATLAB / Octave software.

(4) TEACHING and LEARNING METHODS - EVALUATION

DELIVERY *Face-to-face, Distance learning, etc.*	Face to face in class and in the laboratory. Distance learning support via e-Class system.
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY *Use of ICT in teaching, laboratory education, communication with students*	• Course notes for theory lectures at e-Class. • Support of the learning process with e-Class platform (for announcements, note distribution, additional material, resources, bibliography, and final examinations, etc.) • During lectures a projector and presentations in electronic form are used, which are also posted in the eclass platform. • During the lectures, a computer is used to write and execute code. • Utilization of open source software Octave.
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.*	

Activity	***Semester workload***
Theory lectures	26
Problem solving	13
Problem solving in the laboratory with the aid of specialized software	13
Problem solving and report writing	33

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	Independent study of lectures and bibliography research	40
	Course Total	***125 hours (5 ECTS)***
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.*	A. Written final exam that includes: • Solving exercises • Multiple choice questions • Short answer questions B. Preparation of laboratory exercises. <u>Remarks:</u> • The final grade results from the weighting of the theory and work grades with coefficients determined at the beginning of the semester and announced to the students via e-class. Indicatively it will be about 70% - 30% • Laboratory exercises and assignments are submitted electronically and students are asked to take an oral exam on them. • The exam material and the evaluation process are communicated to the students in the lecture hall and in the e-class. • The evaluation is carried out in Greek.	

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

- Αριθμητικές Μέθοδοι για Μηχανικούς, Chapra S. και Canale R., Εκδόσεις Α. Τζιόλα & Υιοί Α.Ε., 7η έκδοση, 2018.
- Αριθμητικές Μέθοδοι για Μηχανικούς και Επιστήμονες, Gilat Amos και Subramaniam Vish, Broken Hill Publishers, 1η έκδοση, 2021.
- Αριθμητικές Μέθοδοι και Εφαρμογές για Μηχανικούς, Σαρρής Ι. και Καρακασίδης Θ., Εκδόσεις Α. Τζιόλα & Υιοί Α.Ε., 4η έκδοση, 2017.
- Αριθμητικές Μέθοδοι για Προβλήματα Μηχανικής, Εφαρμογές με χρήση MATLAB, Νταουτίδης Π., Μαστρογεωργόπουλος Σ. και Σιδηροπούλου Ε., Εκδόσεις Ανικούλα, 2η έκδοση, 2016.
- Ζάχος, Ε., Παγουρτζής, Α., Σούλιου, Θ. 2015. Αλγόριθμοι Γράφων. [Κεφάλαιο Συγγράμματος]. Στο Ζάχος, Ε., Παγουρτζής, Α., Σούλιου, Θ. 2015. Θεμελίωση επιστήμης υπολογιστών. [ηλεκτρ. βιβλ.] Αθήνα:Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. κεφ 6. Διαθέσιμο στο: http://hdl.handle.net/11419/5468

- Related academic journals:

- SIAM Journal on Scientific Computing (SISC)
- Journal of Scientific Computing