UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO | DIPARTIMENTO DI INFORMATICA

Main information on the course

Course name	**Numerical Computing**	
Degree	Informatica	
Academic year	2023/24	
European Credit Transfer and Accumulation System (ECTS), in Italian Crediti Formativi Universitari (CFU)		6 CFU (each CFU corresponds to 25 hours (h) of student's time); CFU are of type T1, T2 or T3 T1 = 8 h lecture + 17 h individual study T2 = 15 h practice + 10 h individual study T3 = 25 h individual study
Settore Scientifico Disciplinare	MAT/08	
Course language	Italian	
Anno di corso	Second	
Periodo di erogazione	Second semester	
Obbligo di frequenza	It is highly recommended to attend classes	
Sito web del corso di studio	https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/informatica-270/laurea-triennale-in-informatica-d.m.-270-1	

Teacher(s)

Name and Surname	Angela Monti
email	angela.monti@uniba.it
phone	+39 080 5929748
office	
e-learning platform	Piattaforma e-learning UNIBA - https://elearning.uniba.it/
Teacher's homepage	
Office hours	By appointment (requested by email)

Syllabus

Course goals	The course aims to bridge the gap between mathematics and computer science by providing students with the fundamental tools to numerically solve mathematical problems by using computers.
Prerequisites/requirements	• Linear algebra (basic) • Elements of calculus (studied in Mathematical Analysis) • Programming skills
Course program	1. **Computer arithmetic and error analysis**: sources of error, machine representation of numbers, set of machine numbers, IEEE single and double precision, truncation and rounding techniques, absolute and relative error, conditioning: computer arithmetics and function evaluation. 2. **Roots finding algorithms**: conditioning, successive bisections method, convergence analysis, Taylor polynomial, Newton's method and other variants (chord, secant), successions defined by recurrence and general theory for one-step iterative methods with applications to root-finding. 3. **Fundamentals of linear algebra (part I):** matrices and vectors, matrix and vector operations, matrix determinant and its computation with Laplace and Sarrus rules. Structured matrices, invers matrix calculation and existence

	theorem. Linear systems. Cramer's method to determine solutions of linear systems. **4. Linear systems algorithms:** Upper and lower triangular linear systems, permutation matrices and properties. Gauss elimination algorithm. Theorem for LU factorization existence. Application to linear systems. Rank of a matrix. Generalization of LU factorization to rectangular matrices. Rouché-Capelli theorem and its applications. **5. Fundamentals of linear algebra (part II):** Vector spaces, subspaces, linear combination, generators of a vector space, linear dependency, basis for a vector space, dimension of a vector space. Scalar product, angle between vectors and perpendicularity. Kernel and image of a linear transformation. Under-determined and over-determined linear systems. Norms of vectors and matrices, conditioning number of a matrix and conditioning of linear systems, eigenvalues and eigenvectors, power method for computing eigenvalues. **6. Interpolation and approximation**: Power basis. Lagrange interpolation. Error for polynomial interpolation. Least squares method, linear and polynomial regression. **7. Programming in Matlab**: data, variables, algebraic operations and e elementary functions, predefined variables, logical and relational operators, *if-then-else*, *for* and *while,* Matlab script and functions, alghoritms: successive bisections method, Newton's method, Lagrange interpolation, LU factorization.
Books of reference	**1.** Atkinson K.E., An introduction to Numerical Analysis - 2nd Ed. John Wiley & Sons **2.** Bini D., Capovani M., Menchi O., Metodi numerici per l'algebra lineare - Zanichelli **3.** Quarteroni A., Saleri F., Gervasio P., Calcolo Scientifico - Esercizi e problemi risolti con MATLAB e Octave - 5a edizione. Springer Italia.
Notes to the books	Books will be integrated by lecture notes, exercise, exam examples and alghoritms provided by the teacher.

Organization of the didactic activities

Hours			
Total	Lectures	Practice sessions	Individual study
150	32	30	88
CFU/ETCS			
6	4	2	

Teaching methods

Standard lectures and lab sessions in class.

Expected learning outcomes

Knowledge and understanding	• Learn techniques and methodologies for numerical programming, especially in the context of numerical linear algebra. • Understand and illustrate issues related to the use of computers to solve numerical problems.
Applying knowledge and understanding	• Optimization of algorithms with respect to features of the problem and computing resources availability. • Development, documentation and testing of software and capability of interpretation of results.
Other skills	*Making judgements* Identify suitable methods for each specific problem. *Communication* Describe in a rigorous way and with proper language the problem, the method used to solve it and its main features. *Learning skills* Apply techniques and methods to slightly different problems.

Assessment	
Assessment methods	Oral exam on all the explained topics, including both theoretical parts (definitions, theorems and proofs) and practical exercises. Some mathematical problems will be required to be numerically solved by using the presented and implemented numerical algorithms.
Evaluation criteria	• *Knowledge and understanding*: students must show the understanding of main techniques for developing numerical software and they must be able to describe the main methods illustrated during the course. • *Autonomy of judgment*: students must show to be able to evaluate mean features of each method and to be able of compare performances of different methods • *Communication skills*: students must be able to present in an effective way the outcomes of their work on programming and testing numerical methods. • *Capacities to continue learning*: students must show to be able to apply main numerical technique to slightly different problems with respect to those illustrated during the course.
Measurements and final grade	Students must show to understand the main issues related to solving numerical problems, to develop methods and study their property. Students must be able to implement methods, test them and present, in an effective way, results from execution and testing. For the final mark will be also considered the ability to present in a correct and effective way methods and outcomes from programming tasks.

Further information	