UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

CONSIGLIO INTERCLASSE IN MATEMATICA

General information		
Academic subject	***Mathematical Computing Laboratory***	
Degree course	*Mathematics* (***Degree Class:*** *L-35)*	
Academic Year	*2021/2022*	
European Credit Transfer and Accumulation System (ECTS)	7	
Language	*Italian*	
Academic calendar (starting and ending date)	*Second semester*	
Attendance	*no*	

Professor/ Lecturer		
Name and Surname	Pierluigi Amodio (group A)	Felice Iavernaro (group B)
E-mail	pierluigi.amodio@uniba.it	felice.iavernaro@uniba.it
Telephone	+39 080 5442703	+39 080 5442703
Department and address	*Dip. Matematica piano IV, stanza 2*	*Dip. Matematica piano IV, stanza 2*
Virtual headquarters	*Microsoft Teams*	*Microsoft Teams*
Tutoring (time and day)	Information available at the url: https://www.dm.uniba.it/members/amodio/ricevimento	Information available at the url: https://www.dm.uniba.it/members/iavernaro/ricevimento

---

Syllabus	
Learning Objectives	Acquiring some knowledge about the main properties and issues related to the use of finite arithmetic as opposed to real arithmetic. Acquiring the basic tools to operate in Matlab and Sage environments, with special attention to scrutured programming.
Course prerequisites	The knowledge gained in the course “Computer Science”, classical analysis of one and several variables, fundamental linear algebra.
Contents	1. **INTRODUCTION TO SCIENTIFIC COMPUTING AND ERROR ANALYSIS.** Mathematical models and numerical methods, errors sources, the process of the numerical approach to solve problems, computational environments, some languages for the scientific computing, problem solving environments: MATLAB, SAGE. Representing real numbers in a computer, IEEE standard, single and double precision. Truncating and rounding techniques. Absolute and relative errors. Machine precision. Floating-point operations. Errors propagations. Conditioning of a problem. Stability of an algorithm. Computational complexity. 2. **MATLAB.** Introduction to Matlab, the language, script and function files. Built-in functions in Matlab. The workspace. Introduction to graphics in one and two dimensions. Some Matlab examples about rounding errors. Handle vectors and matrices in Matlab. Basic operations involving vector and matrices. Implementation of some numerical algorithms in Matlab: approximation of derivatives of functions by means of divided difference formulae, using the Taylor polynomial to approximate transcendent functions, Laplace formula, Cramer rule and related computational cost. Examples on unstable algorithms.

	3. SAGE. Programming basics, graphs of functions, derivation, integration, linear and nonlinear equations. Sequences, discrete dynamical systems, linear and nonlinear difference equations applied to biology, medicine, finance. Logistic equation and bifurcation diagram. Hints on the solution of differential equations. Iterated functions systems, fractals. Elementary operations in vector spaces. Generating random vectors and matrices. Orthogonal matrices, group actions, visualization of orbits generated by the standard action of the groups O(3) e O(2) on R^3. Linear transformations and solution of some classical problems in linear algebra. Explicit construction of affinities or isometries. Classification of planar isometries: examples of a procedure that splits an isometry in axial symmetries and of a procedure that classifies a given isometry. Examples on the computation of the group of symmetries of a finite set of points. Construction of projectivities. Visualization of the five non-degenerate quadrics of the Euclidean space. Euclidean algorithm, extended Euclidean algorithm, Bezout coefficients, algorithms to find the first n prime numbers, algorithms for the prime factorization of a natural number.
Books and bibliography	• *Handout on machine arithmetic, available at the url https://elearning-mat.hosting.uniba.it* • *Handouts supplied during the course lectures.* • *Uri M. Ascher and Chen Greif, A First Course on Numerical Methods, SIAM, 2011.*
Additional materials	*Handouts, notes and Matlab codes will be made available through an e-learning platform. Log-in information will be provided at the course starting days*

Work schedule			
Hours			
Total 70	Lectures 30	Hands on (Laboratory, working groups, seminars, field trips) 40	Out-of-class study/self-study 105
ECTS			
7	*Lectures: 3*	*Exe/lab: 4*	
Teaching strategy			
	Lectures and exercise sessions. Exercise sessions in the Computer Centre		
Expected learning outcomes			
Knowledge and understanding on:	○ Understanding and being able to explain issues related to the use of a computer for solving elementary mathematical problems.		
Applying knowledge and understanding on:	○ Acquiring skills in programming, testing numerical algorithms and consistently interpreting computer results.		
Soft skills	• *Making informed judgments and choices* ○ Being able to provide rigorous definitions and analysis of the principal aspects of finite arithmetic. • *Communicating knowledge and understanding* ○ Being able to communicate with computers ☺.		

	• *Capacities to continue learning* ○ Capability of studying and solving, both numerically and symbolically, problems similar, but not necessarily equivalent, to those faced during the teaching activities.

Assessment and feedback	
Methods of assessment	*The exam consists in solving two exercises, in Matlab and Sage, on a computer and an oral test which includes a discussion of the Matlab codes prepared during the course lectures.*
Evaluation criteria	• *Knowledge and understanding* ○ Identification of the main differences between real and machine arithmetic. ○ Ability to compare methods that solve the same problem, in terms of stability and computational efficiency. • *Applying knowledge and understanding* ○ Discussion of the codes and examples performed; correct interpretation of the results obtained. • *Autonomy of judgment* ○ Collection and processing of data, encoding of the algorithm and interpretation of the results obtained. • *Communication skills* ○ Clarity, also in terms of formalism, in the description and coding of the numerical methods, as well as ability to effectively present the numerical tests carried out. • *Capacities to continue learning* ○ Numerical implementation and discussion of more elaborate problems than those presented during the lectures.
Criteria for assessment and attribution of the final mark	The achievement of eligibility takes into account the judgment achieved by the student in three partial tests: ○ Matlab test: correct implementation on the computer of the algorithm administered during the exam. ○ Sage test: correct implementation on the computer and discussion of the problem proposed during the preparation of the exam. Theoretical test: theoretical discussion on machine arithmetic and error analysis.
Additional information	