



Principali informazioni sull'insegnamento	
Denominazione dell'insegnamento	<i>Metodi Numerici per l'Informatica</i>
Corso di studio	<i>Informatica e Tecnologie per la Produzione del Software</i>
Anno di corso	<i>3°</i>
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS):	6
SSD	<i>MAT 08</i>
Lingua di erogazione	<i>ITALIANO</i>
Periodo di erogazione	<i>2° semestre</i>
Obbligo di frequenza	<i>No, ma fortemente consigliata</i>

Docente	
Nome e cognome	Antonella Falini; Francesca Mazzia
Indirizzo mail	Antonella.falini@uniba.it ; francesca.mazzia@uniba.it ;
Telefono	
Sede	<i>Dipartimento di Informatica, V piano</i>
Sede virtuale	
Ricevimento (giorni, orari e modalità)	Venerdì dalle 10:00 alle 12:00 Il ricevimento in altri giorni è possibile previo appuntamento per email.

Syllabus	
Obiettivi formativi	Il corso si propone di fornire conoscenze numeriche di base nell'ambito dell'approssimazione monovariata e bivariata e nella trattazione numerica delle matrici (sparse e non, mediante opportune fattorizzazioni), dando maggior rilievo all'ambito applicativo nei contesti di analisi-delle immagini e di dati di serie temporali.
Prerequisiti	<i>Calcolo Numerico, Matematica Discreta, Analisi</i>
Contenuti di insegnamento (Programma)	<ol style="list-style-type: none"><i>1. Cenni di analisi bivariata (2D): grafico di una funzione in due variabili, derivate parziali, gradiente, matrice Hessiana, punti critici e loro natura. Polinomio di Taylor per funzioni bivariate.</i><i>2. Interpolazione spline in 1D e in 2D: interpolazione polinomiale a tratti, costruzione della spline lineare, quadratica e cubica. Applicazioni all'analisi delle immagini</i><i>3. Regressione lineare bivariata: costruzione del piano di migliore approssimazione e applicazioni in data science</i><i>4. Decomposizione ai valori singolari: Caratterizzazione e proprietà. Applicazioni nell'ambito dell'imgae-processing</i><i>5. Ottimizzazione mediante meta-euristiche: introduzione al concetto di metaeuristica, presentazione della "swarm intelligence", metodo della particle swarm optimization e sue varianti principali. Applicazioni allo studio di serie di dati temporali.</i><i>6. Algoritmi per matrici sparse.</i>
Testi di riferimento	<ul style="list-style-type: none"><i>Uri M. Ascher, Chen Greif, A First Course in Numerical Methods, SIAM, 2011</i><i>Michel Gendreau, Jean-Yves Potvin, Handbook of Metaheuristics, third edition, Springer 2019.</i>
Note ai testi di riferimento	<i>Slides, codici e dispense fornite dal docente.</i>

Organizzazione della didattica	
Ore	



Totali	Didattica frontale	Pratica (laboratorio, campo, esercitazione, altro)	Studio individuale
150	32	30	88
CFU/ETCS			
6	4	2	

Metodi didattici	
	<i>Lezioni frontali ed esercitazioni/laboratorio in aula.</i>

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	<ul style="list-style-type: none">• Conoscere le tecniche e i metodi per la programmazione numerica finalizzati alla risoluzione di problemi nell'ambito delle discipline matematiche ed affini, con particolare enfasi ai problemi fondamentali nell'ambito dell'algebra lineare.• Comprendere e saper illustrare le problematiche relative dell'uso del calcolatore per la risoluzione di problemi matematici
Conoscenza e capacità di comprensione applicate	<ul style="list-style-type: none">• Capacità di risolvere problemi matematici mediante algoritmi ottimizzati dal punto di vista del costo computazionale e della stabilità.• Sviluppo delle capacità di programmare, documentare e testare algoritmi numerici, interpretandone correttamente i risultati.• Sviluppo delle capacità di risolvere problemi matematici usando problem solving environments.
Competenze trasversali	<ul style="list-style-type: none">• <i>Autonomia di giudizio</i><ul style="list-style-type: none">○ Saper individuare il metodo numerico più idoneo per risolvere numericamente un problema matematico tra quelli trattati nel corso.• <i>Abilità comunicative</i><ul style="list-style-type: none">○ Saper definire in modo rigoroso i problemi matematici trattati nel corso e saper esporre i relativi metodi numerici, delineandone le proprietà fondamentali.• <i>Capacità di apprendere in modo autonomo</i><ul style="list-style-type: none">○ Capacità di studiare e risolvere problemi numerici simili ma non necessariamente uguali a quelli affrontati durante le lezioni.

Valutazione	
Modalità di verifica dell'apprendimento	<i>L'esame consiste nello sviluppo di codice scientifico sugli argomenti trattati durante il corso e sulla sua discussione</i>
Criteri di valutazione	<ul style="list-style-type: none">• <i>Conoscenza e capacità di comprensione:</i><ul style="list-style-type: none">○ Lo studente dovrà mostrare di avercompreso le tecniche di base per lo sviluppo di software numerico• <i>Conoscenza e capacità di comprensione applicate:</i><ul style="list-style-type: none">○ Lo studente deve saper mettere in pratica anche attraverso lo sviluppo di codici di calcolo scientifico le conoscenze teoriche apprese• <i>Autonomia di giudizio:</i><ul style="list-style-type: none">○ Lo studente deve essere in grado di selezionare il metodo numerico adeguato al problema in esame ed effettuare confronti.• <i>Abilità comunicative:</i><ul style="list-style-type: none">○ Lo studente dovrà mostrare di essere in grado di presentare in maniera efficace i risultati del proprio lavoro• <i>Capacità di apprendere:</i><ul style="list-style-type: none">○ Lo studente dovrà mostrare di essere in grado di applicare le tecniche studiate anche in contesti leggermente differenti da quelli illustrati durante il corso.
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	Lo studente deve saper individuare il metodo numerico più idoneo per risolvere numericamente un problema matematico tra quelli trattati nel corso, conoscere le tecniche e i metodi per la programmazione numerica finalizzati alla sua risoluzione.



	Saper definire in modo rigoroso i problemi matematici trattati nel corso e saper esporre i relativi metodi numerici, delineandone le proprietà fondamentali.
Altro	

General information	
Academic subject	Numerical Methods for Informatics
Degree course	Informatica e Tecnologie per la Produzione del Software
Academic Year	3rd
European Credit Transfer and Accumulation System (ECTS)	6
Language	Italian
Academic calendar (starting and ending date)	Second semester
Attendance	Not mandatory but strongly recommended.

Professor/ Lecturer	
Name and Surname	Antonella Falini; Francesca Mazzia
E-mail	Antonella.falini@uniba.it ; francesca.mazzia@uniba.it ;
Telephone	
Department and address	Computer Science Department, V floor
Virtual headquarters	
Tutoring (time and day)	Friday from 10 am to 12 pm

Syllabus	
Learning Objectives	The aim of the course is to provide numerical basic knowledge on univariate and bivariate approximation techniques and numerical treatment of matrices (either sparse or full via using suitable factorizations algorithms), with the main objective being the application context to image-analysis and time-series.
Course prerequisites	Numerical Methods, Discrete Mathematics and Analysis
Contents	<ol style="list-style-type: none">Bivariate analysis(2D): graph of a function in two variables; partial derivatives; gradients, Hessian matrix; critical points and their nature. Taylor polynomial for bivariate functions.Spline interpolation in 1D and 2D: piece-wise polynomial interpolation, linear, quadratic and cubic spline constructions. Applications to image-analysis.Bivariate linear regression: best plane approximation and applications to data science problems.Singular Value Decomposition: Characterization and properties. Applications to image-processing.Metaheuristic based optimization: Introduction of metaheuristics, presentation of swarm intelligence algorithms: particle swarm optimization and its variation. Applications to time-series analysis.Sparse matrices algorithms.
Books and bibliography	<ul style="list-style-type: none">Uri M. Ascher, Chen Greif, A First Course in Numerical Methods, SIAM, 2011Michel Gendreau, Jean-Yves Potvin, Handbook of Metaheuristics, third edition, Springer 2019.
Additional materials	Slides, codes and lecture notes provided during the class

Work schedule	
---------------	--



Total	Lectures	Hands on (Laboratory, working groups, seminars, field trips)	Out-of-class study hours/ Self-study hours
Hours			
150	32	30	88
ECTS			
6	4	2	
Teaching strategy		<i>Standard lecture and lab/exercise sessions</i>	
Expected learning outcomes			
Knowledge and understanding on:	<ul style="list-style-type: none"> ○ Learning techniques of scientific programming ○ Applications of theoretical knowledge to practical scenarios. 		
Applying knowledge and understanding on:	<ul style="list-style-type: none"> ○ Development, documentation and testing of software ○ Correct interpretation of the produced results ○ Problem solving skills. 		
Soft skills	<ul style="list-style-type: none"> ● <i>Making informed judgments and choices</i> <ul style="list-style-type: none"> ○ Identify the suitable method according to the problem at hand ● <i>Communicating knowledge and understanding</i> <ul style="list-style-type: none"> ○ Describing in a rigorous way and with proper language the problem, the method used to solve it and its main features. ● <i>Capacities to continue learning</i> <ul style="list-style-type: none"> ○ Applying techniques and methods to slightly different problems. 		

Assessment and feedback	
Methods of assessment	<i>Development and discussion of scientific codes about the topics learned during the course.</i>
Evaluation criteria	<ul style="list-style-type: none"> ● <i>Knowledge and understanding</i> <ul style="list-style-type: none"> ○ Students should show the understanding of the main techniques for developing numerical software and they should be able to describe the main methods illustrated during the course ● <i>Applying knowledge and understanding</i> <ul style="list-style-type: none"> ○ Students should be able to write scientific codes to solve practical problems. ● <i>Autonomy of judgment</i> <ul style="list-style-type: none"> ○ Students must show to be able to evaluate main features of each method and to be able to compare performances of different methods ● <i>Communicating knowledge and understanding</i> <ul style="list-style-type: none"> ○ Students must be able to present in an effective way the outcomes of their work on programming and testing numerical methods ● <i>Communication skills</i> <ul style="list-style-type: none"> ○ Students must be able to adopt the rigorous and technical language to scientifically describe the problem at hand. ● <i>Capacities to continue learning</i> <ul style="list-style-type: none"> ○ Students must show to be able to apply main numerical techniques to slightly different problems with respect to those illustrated during the course.
Criteria for assessment and attribution of the final mark	The students must be able to select the best method to numerically solve the problems at hand. Moreover, suitable programming and algorithmic skills are required to efficiently develop suitable software. Students should be able to adopt rigorous language to properly describe the treated numerical methods.
Additional information	