



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	<i>Calcolo Numerico</i>	
Corso di studio	<i>Informatica e Tecnologie per la produzione del Software</i>	
Anno Accademico	2025/26	
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	6 CFU	
Settore Scientifico Disciplinare	MATH-06/A	
Lingua di erogazione	Italiano	
Anno di corso	Secondo	
Periodo di erogazione	2° semestre	
Obbligo di frequenza	La frequenza è fortemente raccomandata	
Sito web del corso di studio	https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea	

Docenti	
Nome e cognome	Francesca Mazzia
Indirizzo mail	Francesca.mazzia@uniba.it
Telefono	0805443291
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.565, 5 ^a piano
Sede virtuale	Piattaforma e-learning UNIBA - https://elearning.uniba.it/
Sito web del docente	https://archimede.uniba.it/~mazzia/
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Venerdì dalle 11:30 alle 13:30, Previo appuntamento per email, per concordare un giorno e/o un orario diverso

Syllabus	
----------	--



Obiettivi formativi	Il corso si propone come raccordo costruttivo fra la matematica e l'informatica, fornendo allo studente gli strumenti specifici di base per risolvere i problemi matematici usando il calcolatore mettendo in evidenza i rischi che si corrono con un uso ingenuo delle risorse di calcolo. Il corso presenta i metodi fondamentali per risolvere numericamente problemi matematici, mettendo in evidenza gli aspetti computazionali. Il programma comprende i metodi iterativi per equazioni non lineari, i metodi diretti per sistemi lineari, interpolazione, approssimazione, integrazione e calcolo di autovalori.
Prerequisiti	Relazioni, Funzioni, Principio di Induzione e Ricorsività, Matrici trattati nel corso di Matematica Discreta; Tutti gli argomenti trattati nel corso di Analisi Matematica; Conoscenze di base di programmazione trattate nel corso di Programmazione e Laboratorio di Informatica.
Contenuti di insegnamento (Programma)	Il corso presenta i metodi fondamentali per risolvere numericamente problemi matematici, mettendo in evidenza gli aspetti computazionali e implementativi. Il programma comprende i metodi iterativi per equazioni non lineari, i metodi diretti per sistemi lineari, interpolazione, approssimazione e calcolo di autovalori. <u>1. Introduzione al Calcolo Scientifico</u> Modelli matematici e metodi numerici, sorgenti di errori, il processo di risoluzione numerica, efficienza, testing, errori computazionali, ambienti computazionali, linguaggi per il calcolo scientifico, problem solving environments: MATLAB, PYTHON. Ore lezione frontale: 1 Ore esercitazione in aula: 0 <u>2. Argomenti preliminari.</u> Polinomi. Polinomio di Taylor. Calcolo approssimato delle derivate di una funzione. Errore Assoluto e Relativo. Numeri Complessi. Introduzione a Python e Matlab, il linguaggio, file di tipo script e function. Funzioni predefinite in NumPy e in Matlab. Introduzione alla grafica. Ore lezione frontale: 4 Ore esercitazione in aula: 3 <u>3. Analisi dell'errore.</u> Rappresentazione dei numeri. IEEE singola e doppia precisione. Troncamento e Arrotondamento. Precisione di macchina. Operazioni con i numeri di macchina. Cancellazione di cifre significative. Condizionamento di un problema. Stabilità degli algoritmi. Propagazione degli errori. Esempi Python e Matlab sugli errori di arrotondamento. Ore lezione frontale: 5 Ore esercitazione in aula: 5 <u>4. Elementi di algebra lineare</u> Matrici e vettori. Operazioni con matrici. Sistemi lineari. Prodotto esterno fra vettori. Forma di Echelon, rango. Dipendenza lineare di vettori. Norme di vettori e matrici. Spazi vettoriali lineari. Sottospazi vettoriali. Spazio vettoriale generato da un insieme di vettori. Base di uno spazio vettoriale. Nucleo e immagine di una trasformazione lineare. Inversa di una matrice.



	Ore lezione frontale: 4 Ore esercitazione in aula: 6 5. <u>Algoritmi per la soluzione di sistemi lineari</u> Il Python per l'algebra lineare, Memorizzazioni di vettori e matrici, operazioni sulle matrici. Sistemi triangolari inferiori e superiori. Matrici di permutazione e proprietà. Algoritmo di eliminazione di Gauss. Problematiche di stabilità. Teorema di esistenza della fattorizzazione LU con pivot. Studio del condizionamento di un sistema lineare. Studio del residuo. Esempi di codici Python/Matlab, documentazione e testing, confronti con il software esistente. Ore lezione frontale: 6 Ore esercitazione in aula: 6 6. <u>Calcolo degli autovalori.</u> Autovalori ed autovettori. Metodo delle potenze. Applicazione: google page rank. Esempi di codici Python. Ore lezione frontale: 2 Ore esercitazione in aula: 2 7. <u>Calcolo degli zeri di funzione</u> Metodo delle bisezioni. Convergenza. Criteri di arresto e stime dell'errore. Ordine di convergenza. Iterazione funzionale. Teorema di contrazione. Il metodo di Newton. Criteri di stop. Metodi quasi newtoniani. Metodo della direzione costante. Metodo della falsa posizione. Il metodo delle secanti. Metodo di Brent. Errore accuratezza e numero di condizione. Esempi di codici Python/Matlab, documentazione e testing, confronti con il software esistente. Ore lezione frontale: 6 Ore esercitazione in aula: 3 8. <u>Interpolazione, Approssimazione e Integrazione.</u> Base delle potenze. Interpolazione di Lagrange. Interpolazione di Newton. Differenze divise. Interpolazione con nodi coincidenti. Interpolazione di Hermite. Errore nell'interpolazione polinomiale. Scelta dei nodi per l'interpolazione. Studio del condizionamento. Applicazione per il calcolo degli integrali definiti. Approssimazione ai minimi quadrati nel discreto. Fitting di dati: polinomio di migliore approssimazione nel senso dei minimi quadrati. Retta di regressione lineare. Esempi di codici Python/Matlab, documentazione e testing, confronti con il software esistente. Ore lezione frontale: 5 Ore esercitazione in aula: 5
Testi di riferimento	<i>Uri M. Ascher, Chen Greif, A First Course in Numerical Methods, SIAM, 2011 (testo di riferimento)</i> Gli studenti che lo desiderano possono ottenere i testi in prestito dalla Biblioteca. Può convenire verificarne la disponibilità mediante il Sistema Bibliotecario di Ateneo https://opac.uniba.it/easyweb/w8018/index.php? e contattare la biblioteca per concordare il prestito.
Note ai testi di riferimento	I libri di testo sono integrati con le slide fornite dal docente ed eventuale materiale di approfondimento, resi disponibili sulla piattaforma ADA (vedi sopra "sede virtuale"). Sulla medesima piattaforma e/o sul canale Teams della classe sono anche disponibili i codici degli esempi numerici sviluppati a lezione.



Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Esercitazione	Studio individuale
62 ore	32 ore	30 ore	88 ore
CFU/ETCS			
6 CFU	4 CFU	2 CFU	

Metodi didattici	
	Lezioni frontali ed esercitazioni in aula

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	○ 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	○ 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	Autonomia di giudizio



	○ Saper individuare il metodo numerico più idoneo per risolvere un problema matematico tra quelli trattati nel corso. Abilità comunicative ○ Saper definire in modo rigoroso i problemi matematici trattati nel corso e saper esporre i relativi metodi numerici, delineandone le proprietà fondamentali. Capacità di apprendere in modo autonomo ○ Capacità di studiare e risolvere problemi numerici simili ma non necessariamente uguali a quelli affrontati durante le lezioni.
--	---

Valutazione	
Modalità di verifica dell'apprendimento	Durante le lezioni verranno discussi, in modo partecipato, diversi quesiti ed esercizi simili per tipologia a quelli comunemente somministrati durante gli esami e gli esoneri. La finalità è duplice: monitorare in tempo reale lo stato di preparazione degli studenti frequentanti, perfezionandone la preparazione in vista dell'esame o degli esoneri; agevolare lo studio in itinere degli aspetti pratici della disciplina, motivando concretamente i corsisti a sostenere l'esame in tempi brevi, sfruttando possibilmente la modalità degli esoneri. Sono previste due prove parziali scritte per facilitare la preparazione all'esame: la prima durante l'interruzione delle lezioni a metà corso, la seconda a fine corso. Entrambe le date sono concordate, nei limiti consentiti, con gli studenti frequentanti. Di seguito sono elencati ulteriori vantaggi riservati ai frequentanti: - La partecipazione alle prove parziali consente di avere informazioni sulle parti di programma che vanno approfondite e quindi migliorare la preparazione all'esame finale. - La prova parziale viene valutata complessivamente durante l'esame orale al momento del recupero degli argomenti considerati insufficienti, in modo da migliorare la valutazione complessiva. - Le prove parziali sono prese in considerazione per i tre appelli che seguono la fine delle attività didattiche. Lo studente che non ha sostenuto le prove parziali sosterrà l'esame orale completo in cui dimostrerà la conoscenza degli argomenti teorici e la capacità di risolvere esercizi. Tale conoscenza può essere dimostrata anche risolvendo esercizi per iscritto. Tutti gli studenti all'orale presenteranno e discuteranno in sede di esame la soluzione degli esercizi da svolgere in Python che gli saranno stati assegnati la settimana precedente l'esame orale. Gli esercizi proposti sono simili a quelli svolti a lezione e disponibili sulle piattaforme Teams e/o ADA.
Criteri di valutazione	● Conoscenza e capacità di comprensione: ○ Dimostrare un'adeguata comprensione e competenza sui contenuti dell'insegnamento. ● Conoscenza e capacità di comprensione applicate:



	○ Conoscenza delle possibili applicazioni dei concetti teorici e adeguata capacità di implementare gli algoritmi proposti. ● Autonomia di giudizio: ○ Lo studente dovrà mostrare di essere in grado di valutare le principali caratteristiche di ciascun metodo e di saper effettuare confronti tra i diversi metodi. ● Abilità comunicative: ○ adeguata capacità espositiva dei contenuti studiati, ○ adeguata capacità di analisi e sintesi ● Capacità di apprendere: ○ 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	Il voto finale è attribuito in trentesimi. L'esame si intende superato quando il voto finale è maggiore o uguale a 18. Nella valutazione della prova orale e nell'attribuzione del voto finale varrà la seguente scala di valutazione dell'apprendimento: ● Voto insufficiente (<18): Conoscenze frammentarie e superficiali dei contenuti, errori nell'applicare i concetti, esposizione carente ● Voto 18-20: Conoscenze dei contenuti sufficienti ma generali, esposizione semplice, incertezze nell'applicazione di concetti teorici ● Voto 21-23: Conoscenze dei contenuti appropriate ma non approfondite, capacità di applicare i concetti teorici, capacità di presentare i contenuti in modo semplice. ● Voto 24-25: Conoscenze dei contenuti appropriate e ampie, discreta capacità di applicazione delle conoscenze, capacità di presentare i contenuti in modo articolato. ● Voto 26-27: Conoscenze dei contenuti precise e complete, buona capacità di applicare le conoscenze, capacità di analisi, esposizione chiara e corretta ● Voto 28-29: Conoscenze dei contenuti ampie, complete ed approfondite, buona applicazione dei contenuti, buona capacità di analisi e di sintesi, esposizione sicura e corretto ● Voto 30 e 30 e lode: Conoscenze dei contenuti molto ampie, complete ed approfondite, capacità ben consolidata di applicare i contenuti, ottima capacità di analisi, di sintesi e di collegamenti interdisciplinari, padronanza di esposizione
Altro	Si suggerisce agli studenti di affidarsi esclusivamente alle informazioni/comunicazioni fornite sui siti ufficiali del Dipartimento di Informatica, ovvero sui gruppi social solo se costituiti e amministrati esclusivamente dai docenti dei relativi insegnamenti: ● https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea ● https://www.uniba.it/it/ricerca/dipartimenti/informatica ● https://elearning.uniba.it/ I programmi degli insegnamenti sono disponibili qui: ● https://elearning.uniba.it/



Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei Regolamenti didattici e manifesti degli studi disponibili nel sito:

- <https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea>

Si suggerisce agli studenti di diffidare delle informazioni e dei materiali circolanti su siti o gruppi social non ufficiali, poiché spesso sono risultati non affidabili, non corretti o incompleti. Per ogni dubbio, chiedere un incontro al docente secondo le modalità previste per il ricevimento.

- link al corso sulla piattaforma e-learning UNIBA

<https://elearning.uniba.it/course/view.php?id=2079>

- link alla pagina Teams del corso:

<https://teams.microsoft.com/l/team/19%3A9365eec0a899460a87a563283a8604be%40thread.tacv2/conversations?groupId=ff098863-08e7-4520-98db-28e5ea6a73ce&tenantId=c6328dc3-afdf-40ce-846d-326ead86d49>

codice di accesso:

84yx59m



Main information on the course

Course name	Numerical Methods for Data Science	
Degree	<i>Informatica e Tecnologie per la produzione del Software</i>	
Academic year	2025/26	
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	MATH-06/A	
Course language	Italian	
Anno di corso	second	
Periodo di erogazione	2 nd 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/corsi-di-laurea	

Teachers

Name and Surname	Francesca Mazzia
email	francesca.mazzia@uniba.it
phone	0805443291
office	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Room n.565, 5 th floor
e-learning platform	Piattaforma e-learning UNIBA - https://elearning.uniba.it/
Teacher's homepage	https://archimede.uniba.it/~mazzia/
Office hours	Fridays 11:30 am-13:030 pm. The students that require a different time can schedule a meeting via email.

Syllabus

Course goals	<i>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.</i>
Prerequisites/requirements	<i>Linear algebra; differential and integral calculus; programming skills. The exam of Analisi Matematica is mandatory to take this exam.</i>
Course program	The course presents the fundamental methods for numerically solving mathematical problems, highlighting computational and implementation aspects. The program includes iterative methods for nonlinear equations, direct methods for linear systems, interpolation, approximation, and eigenvalue computation. 1. <u>Introduction to Scientific Computing.</u> Mathematical models and numerical methods, sources of errors, the numerical solution process, efficiency, testing, computational errors, computational environments, languages for scientific computing, problem-solving environments: MATLAB, PYTHON. Lecture hours: 1 Classroom exercise hours: 0



	2. <u>Preliminary Topics.</u> Polynomials. Taylor Polynomial. Approximate calculation of function derivatives. Absolute and Relative Error. Complex Numbers. Introduction to Python and Matlab, the language, script and function files. Predefined functions in NumPy and Matlab. Introduction to graphics. Lecture hours: 4 Classroom exercise hours: 3 3. <u>Error Analysis.</u> Representation of numbers. IEEE single and double precision. Truncation and Rounding. Machine precision. Operations with machine numbers. Significant digit cancellation. Conditioning of a problem. Algorithm stability. Error propagation. Python and Matlab examples on rounding errors. Lecture hours: 5 Classroom exercise hours: 5 4. <u>Elements of Linear Algebra.</u> Matrices and vectors. Matrix operations. Linear systems. Outer product of vectors. Echelon form, rank. Linear dependence of vectors. Norms of vectors and matrices. Linear vector spaces. Subspaces. Vector space generated by a set of vectors. Basis of a vector space. Kernel and Image of a linear transformation. Inverse of a matrix. Lecture hours: 4 Classroom exercise hours: 6 5. <u>Algorithms for Solving Linear Systems.</u> Python/Matlab for linear algebra, storage of vectors and matrices, matrix operations. Lower and upper triangular systems. Permutation matrices and properties. Gaussian elimination algorithm. Stability issues. Theorem on the existence of LU factorization with pivoting. Study of the conditioning of a linear system. Residual study. Examples of Python/Matlab codes, documentation and testing, comparisons with existing software. Lecture hours: 6 Classroom exercise hours: 6 6. <u>Eigenvalue Computation.</u> Eigenvalues and eigenvectors. Power method. Application: Google PageRank. Examples of Python codes. Lecture hours: 2 Classroom exercise hours: 2 7. <u>Root-Finding.</u> Bisection method. Convergence. Stopping criteria and error estimates. Order of convergence. Functional iteration. Contraction theorem. Newton's method. Stop criteria. Quasi-Newton methods. Constant direction method. False position method. Secant method. Brent's method. Error accuracy and condition number. Examples of Python/Matlab codes, documentation and testing, comparisons with existing software. Lecture hours: 6 Classroom exercise hours: 3 8. <u>Interpolation, Approximation, and Integration.</u> Power basis. Lagrange interpolation. Newton interpolation. Divided differences. Interpolation with coincident nodes. Hermite interpolation. Error in polynomial interpolation. Choice of nodes for interpolation. Conditioning study. Application to definite integral calculation. Least squares approximation in discrete form. Data fitting: best polynomial approximation in the least squares sense. Linear regression line. Examples of Python/Matlab codes, documentation and testing, comparisons with existing software. Lecture hours: 5 Classroom exercise hours: 5
Books of reference	The topics of the course can be studied in more details by consulting the following:



	Uri M. Ascher, Chen Greif, A First Course in Numerical Methods, SIAM, 2011		
Notes to the books	Textbooks are integrated with the slides provided by the instructor and any supplementary materials, made available on the ADA platform (see above "virtual classroom") or on Teams paltform. The platforms also provide access to the codes of the numerical examples developed during the lessons.		
Organization of the didactic activities			
Hours			
Total	Lectures	Practice sessions	Individual study
62 hours	32 hours	30 hours	88 hours
CFU/ETCS			
6 CFU	3 CFU	3 CFU	

Teaching methods	
	Standard lectures and exercises/lab sessions

Expected learning outcomes	
Knowledge and understanding	○ Learn techniques and methodologies for numerical programming, especially in the context of numerical linear algebra. ○ Understanding 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. ○ Problem solving skills development.
Other skills	<i>Making judgements</i> ○ identify suitable methods for each specific problem. <i>Communication</i> ○ Describing in a rigorous way and with proper language the problem, the method used to solve it and its main features. <i>Learning skills</i>



- Applying techniques and methods to slightly different problems.

Assessment	
Assessment methods	During the lessons, various questions and exercises similar in type to those commonly administered during exams and exemptions will be discussed in a participatory manner. The aim is twofold: to monitor the real-time preparation status of attending students, enhancing their preparation for the exam or exemptions; to facilitate ongoing study of the practical aspects of the discipline, actively motivating participants to take the exam promptly, possibly utilizing the exemption mode. Two written partial tests are scheduled to facilitate exam preparation: the first during the mid-course break, and the second at the end of the course. Both dates are agreed upon, within permissible limits, with attending students. Below are further advantages reserved for attendees: • Participation in partial tests provides information on the parts of the program that need further study, thus improving preparation for the final exam. • The partial test is assessed overall during the oral exam when recovering insufficient topics, aiming to enhance the overall evaluation. • Partial tests are considered for the three exam sessions following the end of didactic activities. Students who have not taken the partial tests will undergo a complete oral exam, demonstrating their knowledge of theoretical topics and their ability to solve exercises. This knowledge can also be demonstrated by solving written exercises. All students during the oral exam will present and discuss the solution of Python exercises assigned to them the week before the oral exam. The proposed exercises are similar to those solved during lessons and available on the Teams and/or ADA platforms.
Evaluation criteria	• Knowledge and understanding: ○ Demonstrate adequate comprehension and competency in the course content. • Applied knowledge and understanding: ○ Understanding possible applications of theoretical concepts and adequate ability to implement proposed algorithms. • Judgement autonomy: ○ Identify suitable methods for each specific problem. • Communicative skills:



	○ Adequate ability of analysis and synthesis. describing in a rigorous way and with proper language the problem, the method used to solve it and its main features. • Learning ability: ○ Student 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	The final grade is assigned on a scale of thirty. The exam is considered passed when the final grade is greater than or equal to 18. In the evaluation of the oral exam and in the assignment of the final grade, the following learning assessment scale will be used: • Insufficient (<18): Fragmented and superficial knowledge of the contents, errors in applying concepts, deficient presentation. • 18-20: Sufficient but general knowledge of the contents, simple presentation, uncertainties in applying theoretical concepts. • 21-23: Appropriate but not in-depth knowledge of the contents, ability to apply theoretical concepts, ability to present the contents in a simple manner. • 24-25: Appropriate and extensive knowledge of the contents, fair ability to apply knowledge, ability to present the contents coherently. • 26-27: Precise and comprehensive knowledge of the contents, good application of knowledge, analytical ability, clear and correct presentation. • 28-29: Broad, comprehensive, and in-depth knowledge of the contents, good application of contents, good analytical and synthesis skills, confident and correct presentation. • 30 and "lode" (with praise): Very broad, comprehensive, and in-depth knowledge of the contents, well-established ability to apply the contents, excellent analytical, synthesis, and interdisciplinary connection skills, mastery of presentation.
Further information	It is recommended that students rely exclusively on information/communications provided on the official websites of the Department of Computer Science, or on social groups only if they are established and managed exclusively by the instructors of the respective courses: • https://www.uniba.it/en/research/departments/computer-science/teaching/degree-programs • https://www.uniba.it/en/research/departments/computer-science • https://elearning.uniba.it/ The course programs are available here: • https://elearning.uniba.it/ The information that all students should know is written in the Didactic Regulations and study manifests available on the website: • https://www.uniba.it/en/research/departments/computer-science/teaching/degree-programs/degree-courses Students are advised to be wary of information and materials circulating on unofficial websites or social groups, as they often prove to be unreliable, incorrect, or incomplete. For any doubts, students should request a meeting with the instructor according to the procedures provided for office hours.
	• Link to the course on the UNIBA e-learning platform: https://elearning.uniba.it/course/view.php?id=2079 • Link to the course Teams page:



<https://teams.microsoft.com/l/team/19%3A9365eec0a899460a87a563283a8604be%40thread.tacv2/conversations?groupId=ff098863-08e7-4520-98db-28e5ea6a73ce&tenantId=c6328dc3-afdf-40ce-846d-326ead86d49>

codice di accesso:

84yx59m