Principali informazioni sull'insegnamento	
Titolo insegnamento	Calcolo Numerico
Corso di studio	Informatica e Tecnologie per la Produzione del Software
Crediti formativi	6
Denominazione inglese	Numerical Computing
Obbligo di frequenza	No
Lingua di erogazione	Italiano

Docente responsabile	Roberto Garrappa	roberto.garrappa@uniba.it

Dettaglio credi formativi	Ambito disciplinare	SSD	Crediti
	MATEMATICA	MAT/08	6

Modalità di erogazione	
Periodo di erogazione	I semestre
Anno di corso	2017/2018
Modalità di erogazione	Lezioni frontali Esercitazioni in aula (anche con ausilio di computer)

Organizzazione della didattica	
Ore totali	150
Ore di corso	62
Ore di studio individuale	88

Calendario	
Inizio attività didattiche	24/09/18
Fine attività didattiche	11/01/19

Syllabus	
Prerequisiti	Elementi di base di algebra lineare; calcolo differenziale e integrale per funzioni di una variabile; elementi di programmazione. L'insegnamento di Analisi Matematica è propedeutico all'insegnamento di Calcolo numerico
Risultati di apprendimento previsti (declinare rispetto ai Descrittori di Dublino) (si raccomanda che siano coerenti con i risultati di	• *Conoscenza e capacità di comprensione* Lo studente al termine del corso dovrà aver acquisito le tecniche di base per lo sviluppo di metodi numerici e lo studio delle loro principali proprietà. • *Conoscenza e capacità di comprensione*

apprendimento del CdS, compreso i risultati di apprendimento trasversali)	*applicate* Lo studente dovrà essere in grado di saper individuare i metodi più appropriati per la risoluzione di alcuni specifici problemi numerici anche in rapporto all'accuratezza richiesta ed alle risorse di calcolo a disposizione. Dovrà essere quindi in grado di passare dallo sviluppo ed analisi dei metodi alla loro implementazione, al loro utilizzo ed ai relativo test di correttezza ed accuratezza. • *Autonomia di giudizio* Lo studente, sulla base dell'analisi dei metodi studiati, dovrà essere in grado di fornire una valutazione di pro e contro di ciascun metodo in modo da effettuare scelte consapevoli nella risoluzione dei problemi. Dovrò inoltre essere in grado di osservare e valutare la bontà dei risultati forniti dalle prove numeriche effettuate per mezzo dei codici presentati dal docente in aula o realizzati dallo stesso studente. • *Abilità comunicative* Lo studente dovrà acquisire la capacità di descrivere per ciascun problema studiato la natura del problema stesso, le difficoltà nella sua risoluzione al calcolatore e le modalità con cui tali difficoltà sono affrontate. Dovrà inoltre acquisire la capacità di presentare in maniera efficace i risultati delle proprie esperienze numeriche. • *Capacità di apprendere* *Sulla base dei problemi, dei metodi studiati e della loro analisi, lo studente dovrà acquisire la capacità di elaborare in proprio strategie risolutive da applicare a nuovi problemi.*
Contenuti di insegnamento	Introduzione al calcolo Scientifico. Aritmetica dei calcolatori ed analisi degli errori. Elementi di algebra lineare e risoluzione numerica di sistemi di equazioni lineari. Calcolo di autovalori. Approssimazione di funzioni mediante interpolazione. Approssimazione ai minimi quadrati. Calcolo degli zeri di funzione. Risoluzione numerica di integrali definiti mediante formule di quadratura.

Programma	1. Introduzione al Calcolo Scientifico. La risoluzione di problemi numerici al calcolatore, sorgenti di errore e misurazione degli errori, il processo di risoluzione numerica, efficienza, testing, errori computazionali, ambienti computazionali, linguaggi per il calcolo scientifico. 2. Il linguaggio di programmazione Python Introduzione al linguaggio, file di tipo script e function. Funzioni predefinite in NumPy. Introduzione alla grafica.

	3. Aritmetica dei calcolatori Rappresentazione dei numeri. IEEE singola e doppia precisione. Troncamento e Arrotondamento. Precisione di macchina. Errore assoluto e relativo. Operazioni con i numeri di macchina. Cancellazione di cifre significative. Condizionamento di un problema. Stabilita' degli algoritmi. Propagazione degli errori. Esperienze numeriche in Python sugli errori di arrotondamento e la loro propagazione. 3. Algoritmi per la soluzione di sistemi lineari 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. Esperienze numeriche in Python: documentazione e testing, confronti con il software esistente. 4. Calcolo degli autovalori. Condizionamento del problema del calcolo degli autovalori. Metodo delle potenze e delle potenze inverse. Applicazione: google page rank. Esempi di codici Python. 5. Interpolazione Base delle potenze. Interpolazione di Lagrange. Interpolazione di Newton. Differenze divise. Interpolazione con nodi coincidenti. Errore nell'interpolazione polinomiale. Scelta dei nodi per l'interpolazione. Studio del condizionamento. Interpolazione lineare a tratti. Spline. Approssimazione ai minimi quadrati nel discreto. Esempi di codici Python, documentazione e testing, confronti con il software esistente. 6. Calcolo degli zeri di funzione Metodo delle bisezioni. Convergenza. Criteri di arresto e stime dell'errore. Ordine di convergenza. Iterazione funzionale. Teorema di contrazione. Ordine di convergenza. Il metodo di Newton. Criteri di stop. Metodi quasi newtoniani. Metodo della direzione costante. Metodo della falsa posizione. Il metodo delle secanti. Errore accuratezza e numero di condizione. Esempi di codici Python, documentazione e testing, confronti con il

	software esistente. 7. Calcolo degli integrali. Metodo dei trapezi, di Simpson, dei trapezi composto, di Simpson composto per il calcolo di integrali definiti. Stime dell'errore.
Testi di riferimento	Uri M. Ascher, Chen Greif, A First Course in Numerical Methods, SIAM, 2011 (testo di riferimento) Michael R. Overton, Numerical Computing with IEEE Floating Point Arithmetic, SIAM 2001 (lettura consigliata) James F. Epperson, Introduzione all'analisi numerica, teoria, metodi, algoritmi. McGraw-Hill, Milano, 2003 (lettura consigliata) G. Dahlquist, A. Bjork, Numerical Methods in Scientific Computing, Volume 1, SIAM, 2008 (lettura consigliata)
Note ai testi di riferimento	I testi di riferimento sono integrati con gli appunti delle lezioni e con dispense e/o slide distribuite dal docente.
Metodi didattici	Durante il corso il docente distribuirà alcune dispense e le slide utilizzate durante le lezioni.
Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)	L'esame consiste in una prova orale, nel quale sono fatte domande sulla teoria e sugli algoritmi numerici; inoltre si commentano i programmi in Python prodotti dagli studenti come esercitazioni e corredati di una documentazione scritta in cui sono riassunti i commenti sui risultati numerici, organizzati in modo da poter avere una facile lettura di essi. Per gli studenti che hanno almeno l'80% di presenze del numero di ore di lezioni sono previste prove in itinere, la prima durante l'interruzione delle lezioni a metà corso ed la seconda a fine corso. Tali prove in itinere saranno effettuate sotto forma di domane scritte (essenzialmente domande sulla parte teorica con possibilità di qualche esercizio).
Criteri di valutazione (per ogni risultato di apprendimento atteso su indicato, descrivere cosa ci si aspetta lo studente conosca o sia in grado di fare e a quale livello al fine di dimostrare che un risultato di apprendimento è stato raggiunto e a quale livello)	Gli studenti devono mostrare di essere in grado di comprendere le principali problematiche relative alla risoluzione di problemi numerici, allo sviluppo di metodi ed allo studio delle loro proprietà. Gli studenti dovranno essere quindi in grado di implementare i metodi studiati, di testarli e di presentare in maniera efficace i risultati ottenuti.
Altro	