UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Physical Modelling and Simulations

2627-1-E3004Q004

Obiettivi

Questo corso intende fornire agli studenti le basi della programmazione e dei metodi computazionali per la fisica; gli studenti impareranno a modellizzare e simulare semplici problemi fisici, ottenendo predizioni e analizzando i risultati.

Obiettivi secondo i Descrittori di Dublino:
2. Conoscenza e capacità di comprensione applicate
3. Autonomia di giudizio
5. Capacità di apprendere

Contenuti sintetici

• Introduzione al Bash e agli script Bash.
• Introduzione alla programmazione, linguaggi interpretati vs linguaggi compilati. Python e librerie fondamentali: NumPy, SciPy, Matplotlib. Accenni al calcolo simbolico con SimPy.
• Accenni di programmazione a oggetti e al calcolo parallelo
• Metodi numerici: interpolazione, approssimazione, zeri, differenziazione e integrazione.
• Algebra lineare: operazioni elementari con matrici, sistemi lineari, equazioni agli autovalori.
• Equazioni differenziali ordinarie.
• Modellizzare semplici problemi fisici: esempi dalla meccanica classica.
• Dinamica gravitazionale di sistemi a più corpi.
• Metodi di Monte Carlo per simulazioni.

Programma esteso

Parte 1: Basi di programmazione e calcolo numerico

Introduzione al Bash: shell, comandi Unix, creare e manipolare file e cartelle, variabili di ambiente e alias.
Creare ed eseguire uno script in Bash, primi passi nel coding: variabili, cicli, condizioni, funzioni.

Introduzione ai linguaggi di programmazione. Python, accenni di C++; creare ed eseguire un codice.
Tipi di variabili e operazioni fondamentali. Leggere e scrivere file, definire funzioni, importare moduli. Liste e manipolazioni di liste.
Moduli specifici
NumPy: funzioni fondamentali, numeri pseudocasuali, array multidimensionali.
Matplotlib: realizzare e manipolare immagini, 2D e 3D plotting.
Accenni alla programmazione a oggetti in Python: classi, membri, metodi, ereditarietà.

Metodi numerici per trovare gli zeri: bisezione, metodo di Newton ed estensioni, metodo della secante. Implementazioni in Python con SciPy.
Interpolazione in una e due dimensioni. Implementazione con SciPy.
Derivate prime e seconde con differenze finite. Implementazione numerica, np.gradient
Metodi numerici per l'integrazione: metodo dei rettangoli, metodo dei trapezi, formula di Cavalieri-Simpson, metodo di Gauss. Implementazione con SciPy.integrate.

Operazioni con array e matrici: somme, differenze, moltiplicazione righe per colonne. Risolvere sistemi lineari, calcolare il determinante di una matrice e ottenere la matrice inversa. Diagonalizzazione, calcolare autovalori e autovettori con NumPy.

Calcolo Simbolico : utilizzo di Simpy per operazioni elementari

Risolvere equazioni differenziali ordinarie: metodi di Eulero e Runge Kutta. Condizioni al contorno e Shooting.

Parte 2: modellizzazione e simulazione
Risoluzione di problemi di fisica elementare usando metodi numerici:
• **cinematica del punto materiale**: derivazione numerica delle traiettorie per moti in una, due e tre dimensioni.
• **Meccanica**: equazioni del moto per il pendolo semplice e smorzato
• pendolo doppio
• sistemi meccanici in presenza di attrito
• moto di rotolamento

Modellizzare problemi complessi:
• sistemi collisionali a molte particelle (esempio del gas in una scatola e calcolo di osservabili fisiche come temperatura e pressione)
• sistemi non collisionali a molte particelle: il potenziale gravitazionale di un sistema autogravitante
• Tempi computazionali. Accenno al calcolo parallelo

Rudimenti di metodi Monte-Carlo: cenni storici, algoritmo di Metropolis-Hastings. L'integrazione numerica con metodi Monte-Carlo.

Prerequisiti

Si richiedono conoscenze matematiche di base, funzioni elementari e calcolo algebrico: equazioni, disequazioni, trigonometria.

Modalità didattica

Le lezioni saranno erogate sia da remoto, in modalità asincrona, per 30 ore, sia in presenza per 18 ore. Ogni lezione prevede una parte teorica di lezione frontale (30 ore), sia una parte di programmazione guidata (18).

Materiale didattico

Dispense delle lezioni ed esempi di codici forniti dal docente, volta per volta.
Testo suggerito per la programmazione in Python: “Introduction to Python Programming by OpenStack”, reperibile liberamente online https://openstax.org/details/books/introduction-python-programming (CC license)

Periodo di erogazione dell'insegnamento

Primo semestre

Modalità di verifica del profitto e valutazione

La modalità di verifica del profitto consiste in un esame scritto dove verranno valutate le competenze di programmazione attraverso un esercizio (da scrivere a mano) estratto dagli esercizi forniti durante l'anno; ci saranno poi 4 domande a risposta multipla la cui risposta deve essere motivata ed 1 domanda di teoria. L'accesso all'orale è subordinato al raggiungimento di un punteggio minimo di 16 su 30. L'orale consiste nella presentazione di un progett con relazione finale, dove verranno inoltre verificate le competenze di programmazione e modellizzazione acquisite durante il corso. Non sono previste prove parziali.

Orario di ricevimento

Ogni giorno, possibilmente con richiesta di appuntamento via email

Sustainable Development Goals

ISTRUZIONE DI QUALITÁ