Programació Científica i Computació d'Alt Rendiment (250COD006)

Informació general

Centre docent	ETSECCPB
Departaments	Departament d'Enginyeria Civil i Ambiental (DECA)
Crèdits	5.0 ECTS
Titulacions	MÀSTER UNIVERSITARI EN ENGINYERIA COMPUTACIONAL I ASSISTIDA PER DADES (pla 2026)
Curs	2026/27

Idioma majoritari per grup

- Grup 10Q1 Anglès (Q1)

Professorat de l'assignatura

Professorat responsable: Rubén Zorrilla Martínez
Professorat: Riccardo Rossi Bernecoli, Rubén Zorrilla Martínez

Objectius formatius

Aquest curs introdueix els principis i les pràctiques necessàries per desenvolupar programari científic eficient, escalable i mantenible per a aplicacions d'enginyeria computacional i assistida per dades. Se centra en la implementació de mètodes numèrics mitjançant paradigmes de programació moderns i tècniques de computació d'alt rendiment dins d'un ecosistema basat en Python.

El curs cobreix tot el flux de treball de l'enginyeria computacional, des del disseny de programari i la robustesa numèrica fins a l'optimització del rendiment i l'execució paral·lela. Els temes centrals inclouen l'àlgebra lineal densa i esporàdica d'alt rendiment i conceptes d'enginyeria del rendiment com ara la jerarquia de memòria, la vectorització i el perfilatge. Aquests s'estenen als paradigmes de computació paral·lela, incloent-hi el paral·lelisme de memòria compartida, la programació de memòria distribuïda amb MPI i l'acceleració per GPU.

A més, el curs aborda els reptes moderns de l'enginyeria assistida per dades, com ara la gestió de conjunts de dades a gran escala mitjançant enfocaments fora de nucli (out-of-core) i distribuïts, i la integració de fluxos de treball de simulació amb canonades de processament de dades.

A través d'una combinació de sessions teòriques i tasques pràctiques, els estudiants desenvolupen la capacitat de dissenyar, implementar, analitzar i optimitzar solucions computacionals per a problemes d'enginyeria a gran escala.

Escola de Camins

Hores totals de dedicació de l'estudiantat

		Hores	Percentatge
Aprenentatge dirigit	Grup gran	45.0 h	100.00 %
	Grup mitjà	0.0 h	0.00 %
	Grup petit/Laboratori	0.0 h	0.00 %
	Activitats dirigides	0.0 h	0.00 %
Aprenentatge autònom		80.0 h	

Temari

Paradigmes de programació i introducció a Python

- Programació procedural vs orientada a objectes (OOP)
- Conceptes de programació funcional
- OOP en codis de computació a gran escala (encapsulació, abstracció)
- Patrons funcionals en càlcul numèric (mapa, reducció, inmutabilitat)
- Llenguatges compilats vs interpretats
- Tipus estàtics vs dinàmics
- Introducció a la sintaxi y entorns virtuals de Python

Complexitat algorítmica & estructures de dades

- Complexitat algorítmica & estructures de dades
- Disseny modular amb Python (funcions, classes, tipus)
- Visualització amb Python (matplotlib i PyVista)

Desenvolupament de codi professional

- Control de versions
- Desenvolupament professional: qualitat del codi i revisió del codi
- Estratègies de test (integració contínua i desenvolupament basat en tests
- Entorns de programació assistits per IA (IDEs)

Precisió numèrica & estabilitat en simulació a gran escala

- Aritmètica en coma flotant
- Condicionament dels problemes vs estabilitat algorítmica
- Complexitat algorítmica en fluxos de treball de simulació
- Jerarquia de memòria i comportament de la memòria cache
- Model Roofline
- Vectorització i disposició de les dades en memòria
- Identificació de colls d'ampolla en bucles tipus elements finits

Âlgebra lineal d'alt rendiment

- Conceptes de BLAS/LAPACK subjacents a NumPy
- Operacions denses eficients i avaluació del rendiment (benchmarking)
- Formats de matrius disperses (CSR, COO)
- Assemblatge eficient de matrius disperses
- Anàlisi de l'empremta de memòria
- Breu introducció als mètodes de Krylov i a la seva implementació
- Estratègies de precondicionament
- Solucionadors iteratius: convergència versus escalabilitat
- Solucionadors iteratius: robustesa numèrica

Introducció al paralelisme

- Llei d'Amdahl
- Escalabilitat forta versus escalabilitat feble
- Paral·lelisme en CPU i disposició de les dades en memòria: paral·lelisme en memòria compartida vs paral·lelisme en memòria distribuïda
- Paral·lelisme en GPU

Paralelisme en memòria compartida

- Paralelisme en memòria compartida en Python (GIL i multiprocessing, fonamentals de asyncio, task orchestration)
- Treball de SMP

Paralelisme en memòria distribuida amb MPI

- Estructures de dades distribuides
- Patrons de comunicació
- Producte matriu-vector distribuit

Paralelisme en GPU

- Introducció a l'acceleració en GPU amb PyTorch
- Treball de GPU

Resum de paralelisme

- Sessió pràctica per a treballar en els projectes de DMP, MPI i GPU.

Depuració eficient i anàlisi de performància

- Experiència pràctica amb un cicle de vida de desenvolupament "corromput"
- Eines de depuració (debugging)
- Colls d'ampolla en operacions de reducció
- Experiència pràctica amb eines de perfilatge (profiling)

Gestió de conjunts de dades extremadament grans

- Arranjaments fora de memòria (out-of-core arrays)
- Estratègies de particionament en blocs (chunking)
- Grafs de tasques i avaluació diferida (lazy evaluation)
- Planificador distribuït de Dask
- Paradigmes MPI vs Dask
- Entrada/sortida científica eficient (HDF5)

Metodologia docent

El curs adopta un enfocament interactiu i pràctic en què els conceptes teòrics s'integren estretament amb la implementació. Per tant, cada sessió combina:
- Explicacions conceptuals sobre programació científica i principis de computació d'alt rendiment.
- Demostracions de programació en directe, on les idees clau es desenvolupen pas a pas, mostrant com els conceptes teòrics es tradueixen en implementacions computacionals eficients.
- Exercicis pràctics, que els estudiants duen a terme durant la sessió, i que permeten l'aplicació immediata dels conceptes utilitzant Python.

Una part significativa del curs s'imparteix mitjançant quaderns interactius (per exemple, Google Colab), que permeten l'experimentació, la reproducibilitat i la iteració ràpida sense una configuració complexa. Aquest enfocament facilita l'aprenentatge actiu i permet als estudiants explorar variacions dels mètodes en temps real.

Al llarg del curs, els estudiants treballen en tasques pràctiques i en un projecte progressiu. Aquestes activitats posen èmfasi en l'anàlisi de rendiment, l'escalabilitat i la integració de components numèrics i basats en dades.

El treball autònom complementa les activitats a l'aula, i se centra a completar els deures, millorar les implementacions i analitzar el rendiment computacional, fomentant l'autonomia i una comprensió més profunda.

Mètode de qualificació

(*) El calendari d'avaluació i el mètode de qualificació s'aprovaran abans de l'inici de curs.

L'avaluació del curs es basa en una avaluació contínua i un projecte final:
- Avaluació contínua (40 %): avaluació mitjançant treballs pràctics i tasques manuals. Els criteris d'avaluació inclouen la correcció, la qualitat del codi i les consideracions bàsiques de rendiment.
- Projecte final (45 %): desenvolupament d'un projecte computacional que integri els conceptes del curs, com ara mètodes numèrics, informàtica paral·lela i optimització del rendiment. Els lliurables del projecte inclouen el codi font (sota control de versions), així com un informe tècnic que inclogui una anàlisi de rendiment i una demostració dels resultats. Els criteris d'avaluació inclouen la correcció tècnica, l'escalabilitat, l'anàlisi de rendiment i la qualitat del disseny del programari.
- Defensa oral (15 %): avaluació oral individual basada en el projecte final, que avalua la comprensió dels mètodes implementats, les decisions de disseny i de paral·lelització, i la capacitat d'analitzar el rendiment.

Horari d'atenció

Les sessions de consulta s'organitzaran en un horari convenient tant per a l'estudiant com per al professor.

Bibliografia

Bàsica

- Dongarra, Jack ... [et al.]. Sourcebook of Parallel Computing. Amsterdam [etc.]: Morgan Kaufmann, 2003. ISBN 1558608710.
- Eijkhout, Victor. Introduction to High-Performance Scientific Computing. 3rd ed. 2022.
- Scott, L.R.; Clark, T.; Bagheri, B. Scientific Parallel Computing. Princeton, New Jersey ; Oxford: Princeton University Press, 2005. ISBN 069111935X.
- Dongarra, Jack ... [et al.]. Numerical Linear Algebra on High-Performance Computers. Philadelphia: SIAM, 1998. ISBN 0898714281.