UM4MET11 – Calcul Scientifique (Scientific Computing)

Guillaume Michel

2025-06-23 12:00:31 +0200

Informations générales

Title (EN)	Scientific Computing
Titre (FR)	Calcul Scientifique
Nom du ou de la responsable de l'UE	Guillaume Michel
Nombre d'heures de cours / Amount of class hours	1
Volume h TD / Amount of exercise hours	0
Volume h TP / Amount of practical work hours	20
Volume h Projet / Amount of project hours	16
ECTS	3
Semestre	Automne (S1)
Semester	Sept-Jan (S1)
Langue	Français/Anglais
Language	Français/Anglais
Localisation	campus PMC
Code de l'UE	UM4MET11

Informations pédagogiques

Contenu (FR)

Objectifs Ce cours introduit l'outil numérique dans le contexte de la mécanique. Dans un premier temps, de nombreux problèmes concrets de mécanique sont mis en équation, résolus puis analysés avec Python. Différentes techniques d'implémentation sont présentées et comparées pour rendre l'étudiant capable de choisir la plus adéquate. Ces acquis sont ensuite exploités au cours d'un projet centré sur la spécialité de l'étudiant

---

Contenu

Partie 1 : Introduction aux Notebooks Jupyter

- Introduction à Python et aux Notebooks Jupyter
- Présentations des librairies Numpy et Matlplotlib

Partie 2 : La méthode d'Euler et ses raffinements

- Résolution d'EDO via la méthode d'Euler et généralisations (explicit midpoint, Runge Kutta)
- Ordre d'une méthode numérique
- Utilisation des spectres de puissance pour déterminer la fréquence d'un phénomène physique

Partie 3 : Algèbre linéaire

- Présentations de la librairies Scipy
- Résolution d'un système d'équations linéaires de grande taille, matrices creuses, méthodes itératives
- Décomposition LU
- Calcul de valeurs propres, vecteurs propres

Partie 4 : Optimisation

- Introduction au calcul symbolique (librairie Sympy)
- Recherche des zéros d'une fonction non linéaire
- Recherche d'un minimum/maximum local et interprétation dans le contexte de l'énergie potentielle

---

Déroulé

1e bimestre

- semaine 1 : CM de présentation générale, 1h
- semaine 2 : TP1, 4h
- semaine 3 : TP2, 4h
- semaine 4 : TP3 (1ere partie), 4h
- semaine 5 : TP3 (2nd partie), 4h
- semaine 6 : TP4, 4h
- semaine 7 : présentation des projets, 2h
- semaine 8 : suivi des projets, 2h

2e bimestre

- semaine 1 : suivi des projets, 2h
- semaine 2 : suivi des projets, 2h
- semaine 3 : suivi des projets, 2h
- semaine 4 : suivi des projets, 2h
- semaine 5 : suivi des projets, 2h
- semaine 6 : suivi des projets, 2h
- semaine 7 ou ultérieurement : soutenances des projets

Content (EN)

Objectifs

Objectives This course introduces scientific computing in the context of mechanics. A large number of mechanical problems are modeled, solved, and analyzed using Python. Various implementation techniques are presented and compared to enable students to choose the most appropriate one. These skills are then applied in a project focused on the student's area of specialization.

Content

Part 1: Introduction to Jupyter Notebooks

- Introduction to Python and Jupyter Notebooks
- Overview of the NumPy and Matplotlib libraries

Part 2: Euler's Method and Its Refinements

- Solving ODEs using Euler's method and its generalizations (explicit midpoint, Runge-Kutta)
- Order of a numerical method
- Using power spectra to determine the frequency of a physical phenomenon

Part 3: Linear Algebra

- Overview of the Scipy libraries
- Solving large linear systems, sparse matrices, iterative methods
- LU decomposition
- Eigenvalue and eigenvector computation

Part 4: Optimization

- Introduction to symbolic computation (Sympy library)
- Finding the zeros of a nonlinear function
- Finding and interpreting a local minimum/maximum in the context of potential energy

---

Schedule

1st Term

- Week 1 : Introductory lecture, 1h
- Week 2 : Lab session 1, 4h
- Week 3 : Lab session 2, 4h
- Week 4 : Lab session 3 (part 1), 4h
- Week 5 : Lab session 3 (part 2), 4h
- Week 6 : Lab session 4, 4h
- Week 7 : Project presentations, 2h
- Week 8 : Project follow-up, 2h

2e Term

- Week 1 : Project follow-up, 2h
- Week 2 : Project follow-up, 2h
- Week 3 : Project follow-up, 2h
- Week 4 : Project follow-up, 2h
- Week 5 : Project follow-up, 2h
- Week 6 : Project follow-up, 2h
- Week 7 or later: Project defenses

Mots clés (FR)

Méthodes numériques, Python, Différence finie, Runge Kutta, Algèbre linéaire, Optimisation

Keywords (EN)

Numerical methods, Python, Finite difference, Runge Kutta, Linear algebra, Optimization

Prérequis (FR)

Notions de base en programmation (boucles, types de variables) et leurs implémentations en Python, mise en équation d'un système linéaire à nombre fini de degrés de liberté, résolution dans les régimes libre et forcé

Pre-requisites (EN)

Basic concepts in programming (loops, variable types) and their implementation in Python, formulation of a linear system with a finite number of degrees of freedom, solution in free and forced regimes.

Modalité d'evaluation

Épreuve	Pondération
Examens individuels	40%
Rendu et présentation du projet	60%

---

1. Examens individuels (40%)

- **Durée** : 30 minutes par test
- **Modalité** : présentiel, sur papier, sans accès à internet, sans document
- **Sujet** :
  - Porte sur le TP de la semaine précédente (méthodes numériques et phénomènes physiques)
  - Questionnaire à choix multiples ou extraits de code à analyser

---

2. Projet (60%)

- **Travail individuel ou en binôme**
- Sujet défini avec l'enseignant (des exemples sont fournis par l'enseignant référent)
- Séances de suivi hebdomadaires

Rendus :

- **Présentation orale** : présentation courte et séance de questions
- **Rapport écrit** : notebook Jupyter fonctionnel, comportant la présentation du sujet ainsi que son analyse numérique

Assessment

Evaluation Type	Weight
Individual exams	40%
Project report and presentation	60%

---

1. Individual exams (40%)

- **Duration** : 30 minutes per test
- **Format** : In-person, on paper, no internet or documents allowed
- **Content** :
  - Based on the previous week's practical work (numerical methods and physical phenomena)
  - Multiple-choice questions or code excerpts to analyze

---

2. Project report and presentation (60%)

- **Individual or pair work**
- Topic defined with the instructor (examples are provided by the supervising instructor)
- Weelky follow-up sessions

Deliverables :

- **Oral presentation** : Short presentation followed by a Q&A session
- **Written report** : Functional Jupyter notebook including a presentation of the topic and its numerical analysis

Acquis d'Apprentissage Visés

À l'issue du cours, l'étudiant sera capable de :

1. Modéliser un problème mécanique pour expliciter l'objet de la résolution numérique.
2. Connaître différentes méthodes numériques (d'optimisation, de résolution d'EDO, de problèmes d'algèbre linéaire) et choisir la plus adaptée.
3. Etre autonome dans sa programmation en Python, savoir exploiter la documentation pour utiliser de nouvelles fonctions et utiliser les messages d'erreur pour corriger son code.
4. Interpréter les résultats obtenus.

Learning outcomes

By the end of the course, students will be able to:

1. Model a mechanical problem and specify what needs to be numerically solved.
2. Understand various numerical methods (optimization, solving ODEs, linear algebra problems) and choose the most suitable one.
3. Work independently with Python programming, make use of documentation to learn new functions, and use error messages to debug code
4. Interpret the obtained results.

Bibliographie

- Supports des cours de L1 et L2 reportés sur Moodle
- Documentation officielle de Scipy, Numpy, Matplotlib
- Course materials from L1 and L2 available on Moodle
- Official documentation of SciPy, NumPy, and Matplotlib

Version PDF