

Méthodes de calcul appliqué à la chimie

- Acquis d'apprentissage UE :
 - Appliquer des méthodes numériques standards ou des logiciels existant pour résoudre des problèmes fondamentaux ou annexes, liés à des activités de recherche scientifique
 - Être actif dans la recherche de méthodes de résolution numérique existantes et adaptées à des problèmes auxquels les chimistes sont confrontés
 - Contenu de l'UE :
 - Équations différentielles ordinaires (résolutions numériques et applications cinétiques)
 - Équations aux dérivées partielles (différences finies, problèmes de diffusion)
 - Systèmes d'équations non linéaires (méthode de Newton-Raphson)
 - Problèmes aux valeurs propres (applications à des problèmes de relaxation et de population)
 - Approximation par moindres carrés linéaires et non-linéaires (application à la déconvolution)
 - Approximation de Tchébyshev
 - Modélisation et visualisation de molécules
 - Minimisation et problèmes conformationnels
 - Compétences préalables
 - Connaissance de base d'un langage de programmation
 - Bases des mathématiques
 - Exercices et applications : codes écrits ou à écrire principalement en Python, avec les librairies générales matplotlib, numpy et scipy, ainsi que d'autres librairies spécialisées, notamment en chimie
 - Types d'évaluations : Examen oral sur base d'un travail approfondi sur un des chapitres du cours ou un thème additionnel
-

Synopsis

Pré-requis mathématiques

Base de la programmation

Méthodes numériques de base

- Systèmes d'équations linéaires
 - Diagonalisation et triangularisation
 - Décomposition LU en matrices triangulaires
- Intégration numérique
 - Simpson et quadratures gaussiennes
- Résolutions d'équations du type $f(x) = 0$
 - Équations polynomiales
 - Recherche dichotomique
 - Méthode de la sécante
 - Méthode de Newton-raphson

Méthodes numériques usuelles

Équations différentielles ordinaires

- Résolutions numériques des ODE, principe de discrétisation, méthode d'Euler
- Améliorations et méthodes de Runge-Kutta
- Runge-Kutta d'ordre 4
- Contrôle du pas d'intégration
- Méthodes predictor-corrector
- Méthodes d'extrapolation (Richardson, Burlish-Stoer)
- applications :
 - équations de cinétique chimique
 - Équation logistique
 -  [Modèle de Verhulst](#),  [dynamique des populations](#) et non-linéarité
 - Bifurcation, doublements de périodes et transition vers le chaos
 - Réactions chimiques oscillantes : Belousov-Zhabotinsky, Brusselator, Oregonator
 - Modèle proie-prédateur
 - Attracteur étrange, modèle atmosphérique de Lorenz

Équations aux dérivées partielles

- Domaine d'application des équations : équation de diffusion, équation d'ondes, équations de Navier-Stokes
- Types de traitements numériques
- Différences finies et problèmes de diffusion
- Schémas classiques de différences finies
 - Résolutions stationnaires
 - Résolutions dépendantes du temps
- Méthodes explicites, critère de (ou d'in)stabilités et méthodes implicites

Problèmes aux valeurs propres

applications à des problèmes de relaxation et de population, analyse de modes normaux de vibration, PCA (principal component analysis),...

Systèmes d'équations non linéaires

Méthode de Newton-Raphson

Approximation par moindres carrés linéaires et non-linéaires

application à la déconvolution

Approximations de Tchébyshev

Modélisation et visualisation de molécules

Minimisation

problèmes conformationnels

Thèmes additionnels

- Bioinformatique et algorithmes spécifiques
 - Chimie
 - calculs quantiques, de minimisation, de mécanique moléculaire
 - représentations
 - Data science, statistiques (bibliothèque Python Pandas,...)
 - Time series analysis
 - Machine learning (Scikit-learn,...)
 - Data visualization
 - boxplot, 3D, animations, graphes,...
 - Senseurs et interfaçage, Arduino, Raspberry Pi, IoT
 - Simulations
 - Agent base modelling et systèmes complexes
 - Automates cellulaires
 - Simpy,...
 - Traitement d'image
 - particle tracking,...
-

Références

- Bioinformatique
 - [Biopython](#)
 - [Rosalind.info](#), plateforme d'apprentissage par la résolution de problèmes
- Machine Learning
 - Scikit-learn
 - [12 Algorithms Every Data Scientist Should Know](#)

From:
<https://dvillers.umons.ac.be/wiki/> - Didier Villers, UMONS - wiki

Permanent link:
<https://dvillers.umons.ac.be/wiki/teaching:methcalchim:start?rev=1480060896>

Last update: 2016/11/25 09:01



