HAI507I – Calcul formel et scientifique

Bruno Grenet

Université de Montpellier – Faculté des Sciences

Administratif

Enseignant·e·s

- Pascal Giorgi[1] : resp. ; CM ; TP gpes C & B+CMI
- Bruno Grenet[1] : co-resp. ; CM ; TP gpe A
- Armelle Perret du Cray[1] : TP gpe D

CM et TP

- 4 CM : deux cette semaine (dont aujourd'hui !), 19 oct. et 16 nov.
- 10 TP de 3h : le lundi à 15h (C & D) ou le mercredi à 15h (A & B+CMI)

Modalité de contrôle des connaissances

- 1 note d'examen final (avec 2ème chance)
- 1 note de contrôle continu (examen écrit / rendu de TP)
- *Règle du max* : NoteUE = max(NoteExamen; 70% NoteExamen + 30% NoteCC)

https://moodle.umontpellier.fr/course/view.php?id=22734 (chercher : « HAI507I »)

1. pascal.giorgi@umontpellier.fr, bruno.grenet@umontpellier.fr, armelle.perret-du-cray@umontpellier.fr

Pédagogie

Objectif du cours : **savoir utiliser un système de calcul formel et scientifique**

- Pas : « connaître par cœur toutes les fonctionnalités du logiciel »
- Ni : « connaître par cœur un sous-ensemble des fonctionnalités »
- Mais : « connaître les principes et savoir chercher dans la doc / sur internet »

Pédagogie

Objectif du cours : **savoir utiliser un système de calcul formel et scientifique**

- Pas : « connaître par cœur toutes les fonctionnalités du logiciel »
- Ni : « connaître par cœur un sous-ensemble des fonctionnalités »
- Mais : « connaître les principes et savoir chercher dans la doc / sur internet »

Apprentissage par le TP

- Découverte d'un sujet sur un ou plusieurs TP
- Travail en autonomie (avec recherche de doc, etc.)
- Enseignant·e pour accompagner
- Utilisation du logiciel SageMath, basé sur Python

Cours : bases pour comprendre les TPs

- Quelques principes de base
- Quelques notions mathématiques ou informatiques nécessaires

1. Organisation

2. Python et SageMath

3. Jupyter

Présentation générale de SageMath

- Logiciel de calcul mathématique créé en 2005 par William Stein
- Alternative **libre** à Maple, Mathematica, Matlab, Magma, etc. (licence GPL)

Principes

- Basé sur des bibliothèques de calcul rapides (GMP, Pari, GAP, NTL, etc.)
- « Ne pas réinventer la roue, mais construire ~~une voiture~~ un vélo »
- Interface commune aux bibliothèques, basée sur python

Langages de programmation

- Langage principal : Python
- Autres langages utilisés dans les sources : cython, C, C++, Fortran

Rappels (?) de Python : types

Types de bases

- `int` : `3+4`, `7*8`, `12%5`, `17//3`, ...
- `float` : `3.2+4.1`, `2/3`, ...
- `str` : `'abcd'`, `"calcul !"`, ...
- Note : `int` et `float` (quasiment) inutiles dans SageMath !

Types construits

- Listes : `[1, 2, 3]`, mutable
- Tuples : `(1, 2, 3)`, immuable
- Dictionnaires : `{1:'a', 2:'b', 3:'c'}`

En Python, tout est objet !

Rappels (?) de Python : structures

```
while i < n:
    i += 1
    print(i)

for elt in [1, 2, 3, 4]:
    print(elt)

def fonction(x, y, z):
    return x + y - z
```

```
if x == 1 and y < 2:
    print(x+y)
else:
    print(x-y)
```

Passages à la ligne et indentation significatifs

Boucles `for`

- `for <var> in <iterable>` : liste, tuple, chaîne, dictionnaire, etc.
- Boucle entière : `range(a, b)` est un itérateur entre `a` et `b-1`

Revoir le langage Python si nécessaire !

SageMath

- SageMath est une bibliothèque Python
- SageMath est une distribution de paquets Python et autres
- SageMath est un système interactif de calcul
- SageMath est une communauté de développeurs (chercheurs en info. et maths)

SageMath

- SageMath est une bibliothèque Python
- SageMath est une distribution de paquets Python et autres
- SageMath est un système interactif de calcul
- SageMath est une communauté de développeurs (chercheurs en info. et maths)

SageMath

- SageMath est une bibliothèque Python
- SageMath est une distribution de paquets Python et autres
- SageMath est un système interactif de calcul
- SageMath est une communauté de développeurs (chercheurs en info. et maths)

Bon alors c'est quoi, en pratique ?

1. Bibliothèque : plein de classes, fonctions, etc. pour des objets mathématiques
2. Distribution : en lançant SageMath, on charge ces bibliothèques
3. Système : basé sur Python, mais surcouche (ex : `R.<x> = QQ[]`)

SageMath : démonstration

SageMath et Python

Tout code Python est valide dans SageMath

- Utilisation de la syntaxe Python
- Programmes Python, qui utilisent les fonctionnalités SageMath

SageMath et Python

Tout code Python est valide dans SageMath

- Utilisation de la syntaxe Python
- Programmes Python, qui utilisent les fonctionnalités SageMath

Attention !

- Deux types d'entiers : `ZZ` (SageMath) et `int` (Python)
  - Entiers par défaut : `ZZ`
  - mais `range(12)` renvoie des `int`
  - Conversion : `ZZ(n)`
- Beaucoup de types de flottants : `RR`, … (SageMath) et `float` (Python)
  - Flottants par défaut : `RR`
  - Moins de risque qu'avec les entiers

Ressource

Disponible gratuitement et légalement en ligne : http://sagebook.gforge.inria.fr/

Jupyter : *notebook* interactif

Feuille de calcul

- *Cellules* de code ou de texte, mélangées
- *Noyau* pour exécuter du code

→ Sorte d'IDE adapté au calcul *interactif*

Logiciel Jupyter

- **Libre !**
- Exécution dans le navigateur (mais hors ligne)
- Noyau appelle les logiciels de la machine (SageMath pour nous)

Démo