Université Paris Est - Créteil Val de Marne
Faculté des Sciences et Technologie
M1 Mécanique
Année 2023-2024, 1er semestre
Intervenants: Felipe ROCHA et Sara TOUHAMI
Emails: `felipe.figuereredo-rocha@u-pec.fr` et `sara.touhami@u-pec.fr`

Analyse numérique et calcul scientifique - 1

TP0 (3h)

Introduction au Python et erreurs numériques

Objectifs

En termes générales:

- Maîtriser les bases de la programmation en Python.
- Se familiariser avec l'environnement numérique Google Colabs (ou VS-Code).

Tâches à réaliser:

- Découvrir et prendre en main le langage Python
- Librarie scientifiques: Numpy, Scipy, Matplotlib.
- Reproduire les exemples sur l'analyse des erreurs.
- Aucun travail n'est à rendre obligatoirement pour ce TP.

1 Introduction Python

Suivre la démarche suivante:

1. (a) **Option 1 (Recommandé):** Aller sur `https://colab.research.google.com/` et se connecter avec une compte google, par example avec votre adresse gmail, sinon elle peut être crée a partir d'autre email comme `@u-pec.fr`). Ensuite, ouvrir un nouveau fichier notebook Python: `Fichier>Nouveau notebook`.
   (b) **Option 2:** Ouvrir VSCode et créer un fichier `.ipynb`. L'environment Python notebooks doit s'afficher. Sinon, installer les extensions Python et Jupyter.
2. Tester votre environnement avec un calcul simple dans la première cellule de code, e.g., `print("Salut le monde:  (2 + 8)/5 = " , (2 + 8)/5 )`. Les raccourcis `Ctrl+Entrée` (éxecuter la cellule courant) ou `Ctrl+F9` (éxecuter tout le fichier) sont utiles.
3. Importer la librairie mathématique avec `import math` et s'amuser utilisant quelques fonctions, e.g., `math.sin, math.cos, math.pi, math.exp`, etc. En tappant juste `math.` vous verez la liste de toutes les fonctions disponibles.
4. Ouvrir le fichier `intro_python_basique.ipynb` (fourni) qui contient examples de syntaxe les plus usuels pour une référence rapide. Voici quelques points à faire attention:
   (a) Le scope d'une fonction, loop, etc, est défini par son indentation (obligatoire). Il n'y a pas des mots-clés `begin`, `end`, {, }, etc.
   (b) Le premier index d'un liste, vecteur, etc, est `0` (et pas `1`).
   (c) L'utilisation du point-virgule "`;`" n'est pas obligatoire. Il sert juste à écrire plusieurs commandes dans seule ligne.
   (d) Il n'y a pas de type vecteur ou matrice défini par défaut, pour cela il faut utiliser la librarie `Numpy` (expliqué ci-dessous) ou déclarer une liste `[]` qui sert à stocker des objets (même de types hétérogenes), mais pas faire des opérations mathématiques dessus.
5. Ouvrir le fichier `intro_python_scientifique.ipynb` qui contient les commandes basiques des principales libraries pour le calcul scientifique: Numpy (algébre lineaire), Scipy (méthodes numériques), et Matplotlib (graphiques).

Liens utiles:

- `https://personales.unican.es/corcuerp/python/MATLAB%E2%80%93Python%20cheatsheet.html`
- `https://numpy.org/doc/stable/user/numpy-for-matlab-users.html`

2 Exemples

Les exemples suivantes vous permettront d'apprécier les effets des erreurs d'arrondi et de l'ordre des opérations effectués (algorithme) en calcul numérique.

2.1 `NumRepres_Ex1_MachineEpsilon.ipynb`

Objectif : Calculer la précision de la représentation numérique d'un réel en utilisant les formats double et single.

2.2 `NumRepres_Ex2_SingleDouble.ipynb`

Objectif : Calculer l'exponentiel d'une matrice en utilisant deux représentations numériques différentes (formats double et single) et deux algorithmes différents (développement limité et décomposition en valeurs propres).

2.3 `NumRepres_Ex3_RoundOffError.ipynb`

Objectif : Evaluer un polynôme en utilisant deux formules différentes. Les exemples suivantes vous permettront d'apprécier les effets du conditionnement d'un problème d'analyse numérique, c'est-à-dire la sensibilité du résultats par rapport à des perturbations des données d'entrée.

2.4 `Stability_Ex1_Sqrt.ipynb`

Objectif : Calculer une fraction irrationnelle en utilisant différentes formules.

2.5 `Stability_Ex2_Fraction.ipynb`

Objectif : Calculer une fraction en utilisant différentes formules.

2.6 `Condition_Ex1_Roots.ipynb`

Objectif : Calculer les racines d'un polynôme et du même polynôme aprés perturbation d'un coefficient.

2.7 `Condition_Ex2_ODE.ipynb`

Objectif : Calculer la solution d'une équation différentielle ordinaire en perturbant la condition initiale de la solution.

2.8 `Condition_Ex3_Matrix.ipynb`

Objectif : Calculer la solution d'un système linéaire en perturbant successivement les coefficients du système.