

### Exercice 1

Implémenter la méthode d'Euler pour résoudre l'équation  $y' + y^2 = \cos(t)$  avec la condition initiale  $y(0) = 0$ . On tracera la courbe de la solution sur plusieurs intervalles en guise de test.

### Exercice 2

Le but est ici d'analyser un fichier csv.

1. Ouvrir le script python `notes_alea.py` et l'exécuter. Il doit avoir créé un fichier csv dans le même dossier. Le séparateur de colonne choisi est le point-virgule.
2. Ouvrir le fichier et stocker les lignes dans une liste. Fermer le fichier. On pourra utiliser :

```
f = open(nomdufichier)
L = f.readlines()
f.close()
```

Créer maintenant la liste `notes_eleves` dont chaque élément est une liste qui contient comme premier élément le nom, et ensuite les notes générées sous forme de flottant.

Indications : si `s` est une chaîne, `s.split` est très utile.

3. Créer une liste dont chaque élément (représente un élève et) est une liste de 3 éléments : nom, plus haute note, plus basse note
4. Trouver : l'élève ayant obtenu la plus haute note, la plus basse.
5. Trouver l'élève ayant obtenu la meilleure moyenne.
6. Calculer la moyenne générale de la classe. Sachant que les notes ont été générées au hasard, à quoi doit-on s'attendre ?

### Exercice 3

On souhaite résoudre le système de Cramer  $AX = B$  de matrice  $A \in GL_n(\mathcal{R})$  et second membre  $B$ . Pour cela on utilise la méthode du pivot de Gauss. On supposera, à chaque étape, que le coefficient sur la diagonale est non nul et peut donc servir de pivot.

Compléter la fonction `resolution` du fichier `pivot_gauss.py`. Pour tester on pourra utiliser

```
A = np.array([[2.0,1], [1,2.0]])
B = np.array([[1.0], [1]]) # on utilise des flottants
# copie des connées, au cas où l'algorithme ne fonctionne pas...
A1 = A.copy(); B1 = B.copy(); resolution(A1, B1)
```

La solution théorique est  $(\frac{1}{3}, \frac{1}{3})$  (remarquer le magnifique vecteur propre!).

### Exercice 4

Ecrire une fonction de tri de liste de votre choix, sans utiliser `list.sort`. Rappeler la complexité des meilleurs algorithmes de tris que vous connaissez et comparer avec la complexité du tri choisi. On prendra comme opérations élémentaires : comparaison entre éléments de la liste ainsi que les affectations dans la liste.

### Exercice 5

Créer une fonction `valeurs_derive(L)` qui prend une liste de nombres  $L$  de longueur  $n$  représentant l'évolution d'une grandeur physique au cours du temps et qui retourne une liste de  $n - 1$  valeurs de la dérivées aux  $n - 1$  derniers instants. Comment modifier pour avoir accès à la dérivée aux  $n - 1$  premiers instants ?

### Exercice 6

Ecrire une fonction `positions(LV)` qui prend en argument une liste de vitesse à des instants successifs et retourne la liste des positions atteintes (on se déplace le long d'un axe, la position d'origine est 0).

### Exercice 7

Traiter tous les sujets d'info de la banque PT :)