Tous les exercices sont précédés de la consigne ci-après : *Cet exercice pourra être fait avec le langage Python (et ses bibliothèques numpy, scipy, matplotlib) ou avec le logiciel Scilab. À chaque question, les instructions ou les fonctions écrites devront être testées.*

Exercice 1 On définit les suites (a_n) et (b_n) par :

$$\begin{cases} a_0 = u \\ b_0 = v \end{cases} \quad \text{et, pour tout } n \in \mathbb{N}, \quad \begin{cases} a_{n+1} = \sqrt{a_n b_n} \\ b_{n+1} = \frac{2}{\frac{1}{a_n} + \frac{1}{b_n}} \end{cases},$$

où u et v sont deux réels strictement positifs.

1. Écrire une fonction `iterer` d'un seul argument p, où p représente le couple (a_n, b_n), et qui renvoie le couple (a_{n+1}, b_{n+1}). Tester cette fonction en calculant `iterer([3,2])`.

2. Créer une fonction *non récurive* `suite` de trois u, v et n et qui renvoie le couple (a_n, b_n).

3. Tester cette fonction en calculant `suite(3,2,2)`, puis `suite(3,2,5)`; que peut-on conjecturer ?

4. On admet que pour deux réels strictement positifs donnés u et v, les suites $(a_n)_{n\in\mathbb{N}}$ et $(b_n)_{n\in\mathbb{N}}$ convergent vers une même limite ℓ qu'elles encadrent (ie, pour $n \geqslant 1$, $b_n \leq \ell \leq a_n$). Créer une fonction `moyenne` de deux arguments u et v qui renvoie une valeur approchée à 10^{-10} près par défaut de cette limite.

5. Créer une fonction *récursive* `suiterec` de trois arguments u, v et n qui renvoie le couple (a_n, b_n). Tester cette fonction par `suiterec(3,2,5)`;

6. Faire tracer `moyenne(1,x)` pour x variant entre 1 et 10 avec un pas de $0,1$.

1. Soit f la fonction définie sur $[-5, 5]$ par $f(x) = \dfrac{1}{1+x^2}$. Tracer la courbe de f.

Il existe différentes méthodes pour approcher une fonction h par un polynôme.
L'une d'entre elles, dite "interpolation de Lagrange", consiste à construire un polynôme de degré $(n-1)$ coïncidant en n points avec la fonction h.
Plus précisément, si h est définie sur un segment contenant les valeurs distinctes $[a_0, a_1, \cdots, a_{n-1}]$ regroupées dans une liste $\mathbf{a}$, alors le polynôme :

$$P_{\mathbf{a},h}(x) = \sum_{i=0}^{n-1} h(a_i) \left(\prod_{k=0, k\neq i}^{n-1} \frac{x - a_k}{a_i - a_k} \right)$$

est le seul polynôme de degré inférieur ou égal à $(n-1)$ tel que

$$\forall i \in \mathbb{N}, \quad 0 \leqslant i < n, \quad P(a_i) = h(a_i).$$

2. Écrire une fonction `interpoler` de trois arguments, une fonction h, un réel x et une liste $\mathbf{a}$ qui renvoie $P_{\mathbf{a},h}(x)$.

3. Écrire une fonction `affiche` d'argument j donnant la représentation graphique de f et de $P_{\mathbf{a},f}$, $\mathbf{a}$ correspondant aux $(j+1)$ valeurs équiréparties sur l'intervalle $[-5, 5]$.

4. Tester la fonction précédente pour $j = 5$, 10, 15, 20. Qu'observe-t-on ?

5. Cette fonction particulière f montre que l'interpolation par des points équirépartis n'est pas toujours adaptée pour obtenir une bonne approximation. Une façon de traiter ce problème consiste à ne plus faire l'interpolation de Lagrange sur des points équirépartis sur le segment, mais sur les abscisses dites "de Thebychev" :

$$a_i = 5\cos\left(\frac{(2i+1)\pi}{2n}\right), \quad i \in \mathbb{N}, \quad 0 \leqslant i < n.$$

Visualiser le résultat pour différentes valeurs de n.

Exercice 2 L'objet de cet exercice est d'étudier numériquement la série $\sum \dfrac{(-1)^{n+1}}{n}$.

1. On admet que cette série est convergente et que $\left|\sum_{k=n}^{+\infty} \frac{(-1)^{k+1}}{k}\right| < \frac{1}{n}$.

   Donner une valeur approchée à 10^{-6} près de la somme de cette série. Vérifier que cette valeur est proche de `log(2)` où `log` désigne la fonction logarithme népérien.
2. Représenter graphiquement les cent premières sommes partielles de cette série.
3. On modifie l'ordre des termes de la série en prenant alternativement 2 termes positifs et 3 termes négatifs comme suit :

$$\left(1+\frac{1}{3}\right)+\left(-\frac{1}{2}-\frac{1}{4}-\frac{1}{6}\right)+\left(\frac{1}{5}+\frac{1}{7}\right)+\left(-\frac{1}{8}-\frac{1}{10}-\frac{1}{12}\right)+\cdots$$

   Écrire une fonction `SDP` d'argument un entier naturel n non nul et qui renvoie la liste des n premières sommes partielles de cette série réordonnée.
4. Représenter graphiquement les 120 premières valeurs de cette série réordonnée. Que peut-on conjecturer sur la somme de cette série ?
5. Examiner ce qui se passe si l'on prend un terme positif, puis un terme négatif, puis deux positifs, puis un seul négatif, puis trois positifs, puis un seul négatif, puis quatre positifs, etc.

Exercice 3

1. Soit le système différentiel avec condition initial suivant :

$$\begin{cases} x'(t) = y(t) \\ y'(t) = (1-x(t)^2)y(t) - x(t) \\ x(0) = 2 \\ y(0) = 0 \end{cases} \tag{1}$$

   Pour un pas de discrétisation $h = \frac{1}{4}$, puis $h = \frac{1}{8}$, représenter, en fonction du temps, les solutions approchées de cette équation obtenues par la méthode d'Euler, pour t dans l'intervalle $[0, 8]$.
2. En utilisant la fonction `odeint` du module `scipy.integrate` de *Python*, résoudre numériquement le problème (1). On prendra garde à définir soigneusement les arguments de cette fonction en lisant attentivement l'aide en ligne.

   Représenter sur la figure précédente la solution numérique trouvée.
3. Représenter les différentes solutions trouvées dans le plan xy.
4. Représenter sur l'intervalle $[0, 8]$ une solution approchée de l'équation $x''(t) = (1-x(t)^2)x'(t) - x(t)$ avec $x(0) = 0$ et $x'(0) = 1$.