Exercice 1. Méthode de Newton avec estimation de la dérivée

```
def newtonh(f,x,h,n):
   for i in range(n):
      df = (f(x+h) - f(x-h))/(2*h)
      x = x - f(x)/df
   return x
```

Question : Pourquoi utilise-t-on $\frac{f(x+h)-f(x-h)}{2h}$ plutôt que $\frac{f(x+h)-f(x)}{h}$?

Exercice 2. Calcul intégral

Ecrire les fonctions rectanglesd(f,a,b,n), rectanglem(f,a,b,n) et trapeze(f,a,b,n) calculant une approximation de $\int_a^b f(t)\mathrm{d}t$ suivant les méthodes des rectangles à droite, du point médian, des trapèzes.

Exercice 3. Exemple de dichotomie

On considère la fonction suivante.

```
def dicho(f,a,b,epsilon):
   while b-a > epsilon:
      c = (a + b)/2
      if f(a)*f(c) < 0:
         b = c
      else:
         a = c
   return (a+b)/2
```

La commande suivante provoque une boucle infinie. Expliquer pourquoi.

```
dicho(lambda x: x**(1/3)-100,10**5,10**7,10**-12)
```

Exercice 4. Recherche dans une liste triée

La variable *liste* est une liste de nombres entiers classés dans l'ordre croissant. Ecrire une fonction recherche_dicho(*liste*,*n*) qui recherche un nombre *n* dans la liste en effectuant une recherche dichotomique (teste un élément central à chaque étape et divise l'intervalle de recherche par 2). Cette fonction renverra l'index de l'élément s'il est dans la liste ou –1 sinon.

Exercice 5. En s'inspirant de la dichotomie

a. Ecrire une fonction recherche_min(*f*,*a*,*b*,*epsilon*) qui retourne une valeur approchée du minimum d'une fonction strictement décroissante puis strictement croissante sur un intervalle $[a, b]$. A chaque étape, on coupera l'intervalle $[a, b]$ en trois :

$[a, b] = [a, c]\cup[c, d]\cup[d, b]$, et on éliminera un des trois intervalles.

b. Commenter le retour suivant :

```
In [2]: recherche_min(lambda x: x**2-2*x,0,10,10**-12)
Out[2]: 0.9999999925498303
```

c. Ecrire une fonction qui retourne le maximum d'une fonction strictement croissante puis strictement décroissante sur un intervalle $[a, b]$.

Exercice 6. Méthode d'Euler (inspiré du sujet 0 d'informatique CCP PSI)

On étudie la déformation longitudinale d'une tige que l'on assimile à n points alignés repérés par leurs abscisses $x_1(t), x_2(t), \dots, x_n(t)$, pour $t \in [0, T]$ (en secondes). A l'état initial, $x_1(0), x_2(0), \dots, x_n(0)$ forment une subdivision régulière de l'intervalle [0 ; 10] et les vitesses $\dot{x}_i$ sont nulles.
Le principe fondamental de la dynamique appliqué à ce système conduit à l'équation différentielle :

$$\ddot{X}(t) + A(t)\dot{X}(t) + B(t)X(t) = C(t)$$

où $X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$, $\dot{X} = \begin{pmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{pmatrix}$, $\ddot{X} = \begin{pmatrix} \ddot{x}_1 \\ \vdots \\ \ddot{x}_n \end{pmatrix}$, A et B sont des fonctions de [0, 300] dans $M_n(\mathbb{R})$, C est une fonction de [0, 300] dans $\mathbb{R}^n$. Les fonctions A, B, C sont supposées connues et retournent des éléments du type array, de la bibliothèque numpy importée par « import numpy as np ».

La résolution de l'équation différentielle matricielle est obtenue à l'aide d'un schéma d'Euler implicite.

L'intervalle de temps $[0, T]$ est discrétisé en pas de temps de même durée, notés dt. Ainsi, le piquet de temps t_{q+1} est donné par $t_{q+1} = dt + t_q$ soit par récurrence, $t_{q+1} = (q + 1)dt$ avec $t_0 = 0$.

A l'instant t_q, le vecteur déplacement est noté X_q, la vitesse V_q et l'accélération A_q.

a. En appliquant le schéma d'Euler implicite, exprimer V_q en fonction de dt, X_q et X_{q-1}.

b. En déduire, toujours à l'aide du schéma d'Euler implicite, une expression de A_q en fonction de dt, X_q, X_{q-1} et X_{q-2}.

c. Ecrire l'équation différentielle matricielle à l'instant t_q sous la forme $M_qX_q = N_q$ où M_q et N_q sont deux matrices à déterminer.

d. En utilisant np.linalg.solve(M_q,N_q) pour résoudre le système $M_qX_q = N_q$, écrire une fonction calcul(n,A,B,C,T,$npts$) qui renvoie un tableau de dimension $npts \times n$ dont les vecteurs lignes sont $X_0, X_1, \ldots, X_{npts-1}$.

Pour résoudre un système différentiel du type $y' = f(x, y)$, le schéma d'Euler explicite consiste à approcher y'_k par $\frac{y_{k+1} - y_k}{h}$, ce qui donne

$y_{k+1} = y_k + hy'_k = y_k + hf(x_k, y_k)$ avec $y' = f(x, y)$ et $h = x_{k+1} - x_k$.

Le schéma d'Euler implicite consiste à approcher y'_{k+1} par $\frac{y_{k+1} - y_k}{h}$, ce qui donne $y_{k+1} = y_k + hy'_{k+1} = y_k + hf(x_{k+1}, y_{k+1})$.