TP: méthode d'Euler
Informatique pour tous

Soit $y : [a, b] \longrightarrow \mathbb{R}$ une solution de:

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases}$$

On veut approximer cette solution y.
On rappelle que la méthode d'Euler consiste à subdiviser I en des points $t_0 = a$, t_1, ..., $t_n = b$ régulièrement espacés de h (donc $t_k = t_0 + k \times h$) et à approximer chaque $y(t_k)$ par y_k $(0 \leq k \leq n)$ défini par récurrence:

- $y_0 = y(t_0)$
- $y_{k+1} = y_k + h \times f(t_k, y_k)$

Par exemple, considérons la solution y de l'équation différentielle suivante:

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

Dans ce cas, $t_0 = -2$, $y_0 = -1$ et $f(a, b) = ab^2 - a^2$.
Pour approximer y sur $[-2, 0]$ avec un pas de 0.1, c'est à dire pour $t_0 = -2$, $t_1 = -1.99$, $t_2 = -1.98$, ..., $t_n = 0$, on calcule donc la suite y_k définie par récurrence:

- $y_0 = -1$
- $y_{k+1} = y_k + 0.1 \times (t_k y_k^2 - t_k^2)$

On a alors $y_k \approx y(t_k)$.

1. Retrouver d'où vient l'équation de récurrence $y_{k+1} = y_k + h \times f(t_k, y_k)$.

Dans la suite, on veut approximer sur $[-1, 5]$ une solution y de l'équation différentielle suivante, en utilisant un pas de 0.02:

$$\begin{cases} y'(t) = t^3 - y(t)^3 \\ y(-1) = 4 \end{cases}$$

2. Quels temps d'approximations t_k peut-on utiliser?
3. Écrire l'équation de récurrence vérifiée par les y_k de la méthode d'Euler.
4. Écrire des instructions Python pour stocker dans des listes `t` et `y` ces t_k et y_k. Ainsi, `t[k]` doit contenir t_k et `y[k]` doit contenir y_k.
5. Afficher graphiquement l'approximation obtenue. Pour cela on rappelle qu'il faut écrire:
   - `import matplotlib.pyplot as plt`
   - `plt.plot(t, y)` pour afficher `y` en fonction de `t`
   - `plt.show()` pour afficher la fenêtre graphique
6. Afficher d'autres solutions de la même équation différentielle $y'(t) = t^3 - y(t)^3$ pour d'autres valeurs initiales (par exemple pour $y(-1) = -3$, $y(-1) = 0$, $y(-1) = -2$).
   On peut afficher plusieurs courbes sur le même dessin en écrivant plusieurs `plt.plot`. Par exemple, pour dessiner `y1` en fonction de `t` et `y2` en fonction de `t`, on peut écrire:
   `plt.plot(t, y1)`

```
plt.plot(t, y2)
plt.show()
```

7. Que peut-on conjecturer sur le comportement asymptotique des solutions de $y'(t) = t^3 - y(t)^3$? On pourra aussi dessiner la courbe $y = x$.