

TD MATH/INFO/PHY

– OSCILLATIONS ET EULER –

En physique vous avez vu que pour **modéliser des oscillations**, on devait résoudre une équation différentielle d'ordre 2. Dans les cas de « petites » oscillations les équations différentielles étaient assimilables à des équations linéaires, et la théorie déjà étudiée permettait de résoudre algébriquement ces équations pour trouver les solutions. En revanche, si les oscillations sont plus importantes les équations différentielles ne sont plus linéaires et il faut recourir à des **méthodes numériques** comme la **méthode d'Euler** pour la résoudre. C'est l'objet de ce travail...

1 La méthode d'Euler pour des équations différentielles d'ordre 1

La méthode d'Euler permet de construire de proche en proche une solution d'une équation différentielle du type

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

En particulier, cela permet de résoudre des équations du type :

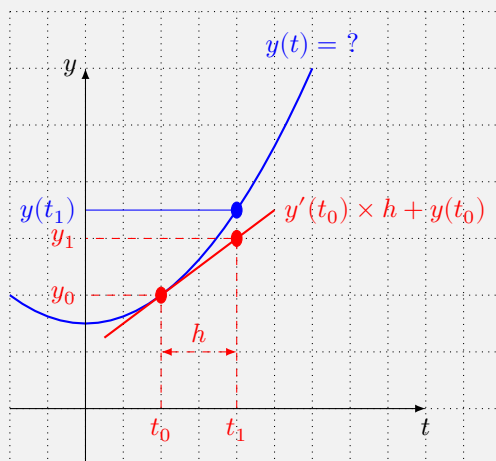
$$\begin{cases} y'(t) + a(t)y(t) = b(t) \\ y(t_0) = y_0 \end{cases}$$

cela correspond à la première situation avec $f(x, t) = -a(t)x + b(t)$.

Point clé de la méthode d'Euler

L'idée dans la méthode d'Euler est que si l'on connaît $y(t_0)$ et $y'(t_0)$ alors en utilisant « **la meilleure approximation affine de la fonction y en t_0** » on peut estimer la valeur de $y(t_0 + h)$ lorsque h est assez petit.

Voyons cela sur un graphique :



Au voisinage du point (t_0, y_0) , on approxime la courbe représentative (bleue) de la fonction y par sa tangente (rouge)

$$y(t_0 + h) \approx y'(t_0) \times h + y(t_0)$$

- On connaît la valeur de y à l'instant t_0 , c'est y_0 .
- Si y est une solution de l'équation différentielle alors $y'(t_0) = f(y(t_0), t_0) = f(y_0, t_0)$.
On connaît donc la valeur de $y'(t_0)$.
- En utilisant « la meilleure approximation affine de y en t_0 » on peut trouver une approximation de y à l'instant $t_1 = t_0 + h$ (où h est un petit pas de temps) que l'on notera y_1 : $y(t_1) \approx y_1 = y'(t_0) \times h + y_0$.
- Maintenant que l'on connaît y_1 à l'instant t_1 , on recommence pour trouver une approximation de y à l'instant $t_2 = t_1 + h$...

2 Mise en pratique via python sur des exemples

2.1 Exemple 1 : $y'(t) = f(t)$

On va dans cet exemple obtenir une liste de valeurs pour représenter graphiquement la solution de l'équation différentielle :

$$(E_1) \begin{cases} y'(t) = \frac{1}{1+t^2} \\ y(0) = 0 \end{cases}$$

Q1 Écrire une fonction python `f(t : float)->float` recevant un paramètre t et renvoyant la valeur de $\frac{1}{1+t^2}$.

Q2 Compléter le code de la fonction python `euler(f,a,b,y0,h=0.1)->list` ci-dessous. Celle-ci reçoit la fonction f de l'équation $y'(t) = f(t)$, les valeurs a et b (avec $a < b$) déterminant l'intervalle sur lequel on résout l'équation et $y_0 = y(a)$ donnant la condition initiale. Le paramètre h est facultatif et représente le pas de résolution. Elle doit renvoyer deux listes : la première étant la liste des valeurs de t et la deuxième la liste des valeurs de $y(t)$ correspondantes, calculées par la méthode d'Euler :

```

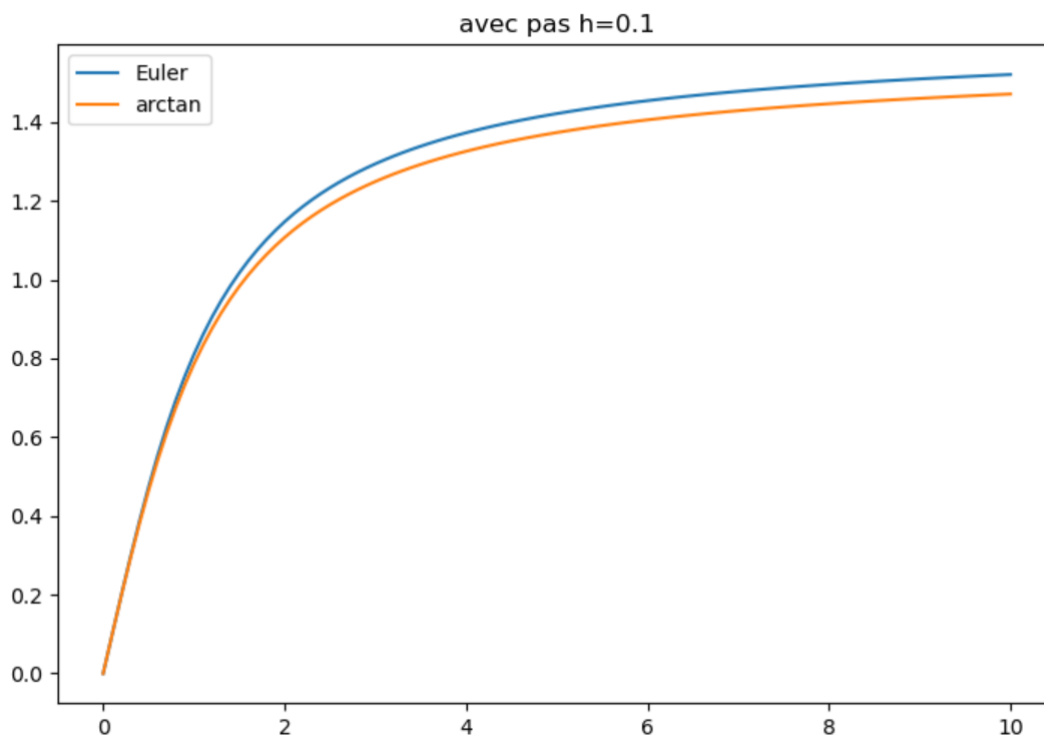
1 def euler(f,a,b,y0,h=0.1):
2     """ Résout l'équation différentielle  $y'=f(t)$  sur l'intervalle  $[a,b]$ 
3     avec la condition initiale  $y(a)=y_0$  et le pas  $h$ .
4     Renvoie les listes des valeurs de  $t$  et de  $y(t)$  correspondantes.
5     """
6     t, y = a, y0
7     T = [t]
8     Y = [y]
9     while t<b-h :
10         
11         Y.append(y)
12         t += h
13         T.append(t)
14     return T,Y

```

Q3 Compléter le programme python pour tracer la courbe de y en fonction de t à l'aide de la bibliothèque `matplotlib`.

Q 4 La solution exacte de cette équation est $y(t) = \arctan(t)$. Ajouter les instructions nécessaire pour tracer la courbe de la fonction $t \mapsto \arctan(t)$ sur le même graphique. Que constate-t-on ?

S3



2.2 Exemple 2 : $y'(t) = f(y(t), t)$

Dans cet exemple on souhaite obtenir une liste de valeurs pour représenter graphiquement la solution de l'équation différentielle :

$$(E_2) \begin{cases} y'(t) + \frac{1}{2} \sin(y(t)) = t \\ y(0) = 1 \end{cases}$$

Q 5 Écrire cette équation sous la forme $y'(t) = f(y(t), t)$ en précisant l'expression de la fonction f ainsi que la fonction python associée.

S4

Q 6 Qu'est-ce qui change dans le code de la fonction `euler` par rapport à l'exemple 1, pour tracer la solution de l'équation (E_2) .

3 Application en physique sur un pendule

Lorsque l'on modélise par exemple la position d'un pendule, l'équation du mouvement est donnée par une équation différentielle de la forme :

$$(E_3) \begin{cases} \theta''(t) + \omega_0^2 \sin(\theta(t)) = 0 \\ \theta(0) = \theta_0 \\ \theta'(0) = 0 \end{cases}$$

⇒ Voir les rappels et les détails en cours de physique

L'équation n'étant pas linéaire (sauf si θ est « petit », car dans ce cas on peut approximer $\sin(\theta)$ par θ), on ne sait pas la résoudre algébriquement. On va envisager encore une résolution numérique.

Ce qui change ici par rapport à la première partie c'est que l'équation est d'ordre 2. Voyons comment adapter ce que l'on a déjà fait. On va envisager deux méthodes différentes : la méthode d'**Euler explicite** et la méthode d'**Euler semi-implicite**.

3.1 Méthode d'Euler explicite

On va commencer par s'inspirer de la technique d'Euler vue dans la partie précédente. On procédera ici en **deux étapes** puisque il y a une dérivée seconde dans l'équation.

On rappelle le principe basé sur la « meilleure approximation affine » d'une fonction au voisinage d'un point :

$$f(t_0 + h) \approx f'(t_0) \times h + f(t_0)$$

Méthode d'Euler explicite pour une équation d'ordre 2

L'idée est la même que pour les équations d'ordre 1, mais il faut faire deux étapes pour trouver une approximation de $\theta(t_0 + h)$ et de $\theta'(t_0 + h)$ à partir de $\theta(t_0)$ et $\theta'(t_0)$.

On part de la relation liant θ'' à θ :

$$\theta''(t) = -\omega_0^2 \sin(\theta(t))$$

Puisque l'on connaît $\theta(t_0)$ et $\theta'(t_0)$, on connaît aussi la valeur de $\theta''(t_0)$. On peut alors trouver par la méthode d'Euler :

- une approximation de $\theta(t_0 + h)$ à partir de $\theta'(t_0)$ et $\theta(t_0)$:

$$\theta(t_0 + h) \approx \theta'(t_0) \times h + \theta(t_0)$$

- une approximation de $\theta'(t_0 + h)$ à partir de $\theta''(t_0)$ et $\theta'(t_0)$:

$$\theta'(t_0 + h) \approx \theta''(t_0) \times h + \theta'(t_0) \approx -\omega_0^2 \sin(\theta(t_0)) \times h + \theta'(t_0)$$

Ensuite on se retrouve dans la même situation que pour t_0 et on peut recommencer pour trouver une approximation de $\theta(t_0 + 2h)$ et $\theta'(t_0 + 2h)$ à partir de $\theta(t_0 + h)$ et $\theta'(t_0 + h)$...

Q7 Compléter le code ci-dessous pour tracer les solutions :

```

1 # Méthode d'Euler explicite pour résoudre l'équation du pendule

2 import numpy as np
3 import matplotlib.pyplot as plt

4 # Paramètres du système
5 OMEGA0 = 1      # pulsation naturelle du pendule
6 THETA0 = 0.7    # angle initial (en radians)
7 THETAPO = 0     # vitesse angulaire initiale
8 DT = 0.1        # pas de temps
9 T_final = 30    # temps final

10 T = np.arange(0, T_final, DT) # liste des instants de temps

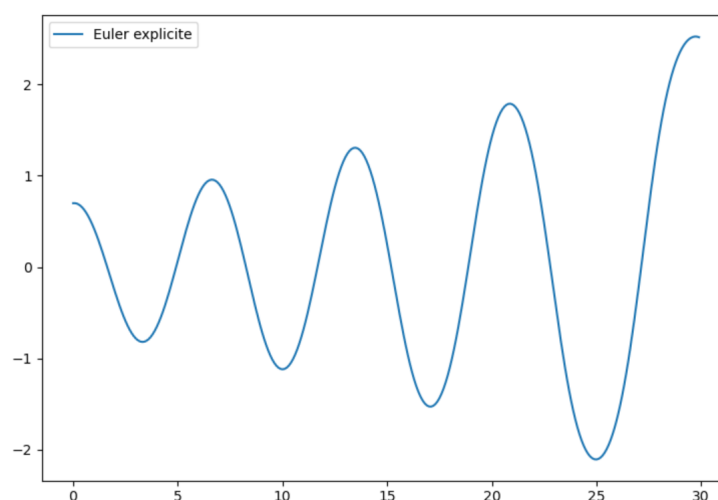
11 def f(theta,t):
12     return -OMEGA0**2*np.sin(theta)

13 # Résolution de l'équation différentielle par la méthode d'Euler explicite
14 def euler_explicite():
15     t = 0
16     theta = THETA0
17     thetap = THETAPO
18     THETA = [theta]
19     THETAp = [thetap]
20     for i in range(len(T)-1):
21         # calcul de theta(t+DT) :
22         
23         # calcul de thetap(t+DT) :
24         
25         THETA.append(theta)
26         THETAp.append(thetap)
27     return THETA

28 # Tracé de la solution
29 Yexpl = euler_explicite()
30 plt.plot(T,Yexpl, label='Euler explicite')
31 plt.legend()
32 plt.show()

```

Q8 Programmer cette méthode et constater qu'il y a un problème...



Pour palier à cet inconvénient de stabilité de l'algorithme, on va utiliser une variante de la méthode d'Euler appelée méthode d'Euler semi-implicite.

3.2 Euler semi-implicite

Dans cette méthode, on va faire les étapes dans un ordre différent : on va d'abord trouver une approximation de $\theta'(t_0 + h)$ à partir de $\theta''(t_0)$ et $\theta'(t_0)$, puis on va trouver une approximation de $\theta(t_0 + h)$ à partir de $\theta'(t_0 + h)$ et $\theta(t_0)$. Ceci permet d'améliorer la stabilité de l'algorithme.

Méthode d'Euler semi-implicite pour une équation d'ordre 2

On part de la relation liant θ'' à θ :

$$\theta''(t) = -\omega_0^2 \sin(\theta(t))$$

Puisque l'on connaît $\theta(t_0)$ et $\theta'(t_0)$, on connaît aussi la valeur de $\theta''(t_0)$. On peut alors trouver par la méthode d'Euler :

- une approximation de $\theta'(t_0 + h)$ à partir de $\theta''(t_0)$ et $\theta'(t_0)$:

$$\theta'(t_0 + h) \approx \theta''(t_0) \times h + \theta'(t_0) \approx -\omega_0^2 \sin(\theta(t_0)) \times h + \theta'(t_0)$$

- une approximation de $\theta(t_0 + h)$ à partir de $\theta'(t_0 + h)$ et $\theta(t_0)$: (C'est là que cela change par rapport à la méthode d'Euler explicite)

$$\theta(t_0 + h) \approx \theta'(t_0 + h) \times h + \theta(t_0)$$

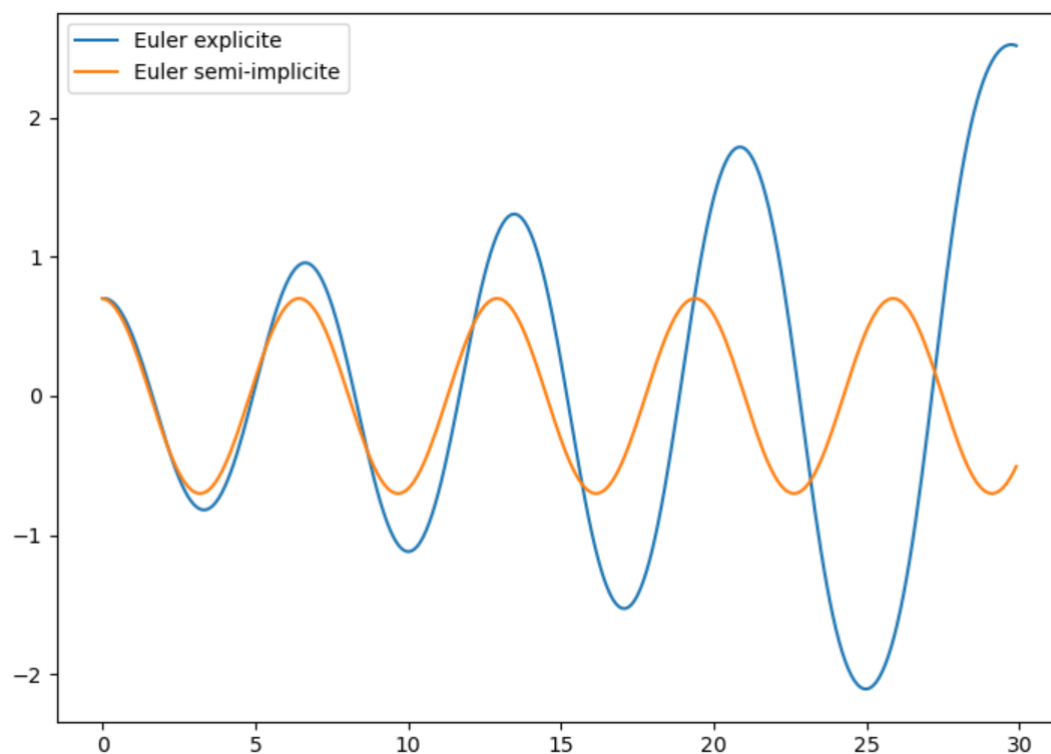
Ensuite on se retrouve dans la même situation que pour t_0 et on peut recommencer pour trouver une approximation de $\theta(t_0 + 2h)$ et $\theta'(t_0 + 2h)$ à partir de $\theta(t_0 + h)$ et $\theta'(t_0 + h)$...

Q9 Compléter le code ci-dessous :

```
1 def euler_semi_implicite():
2     t = 0
3     theta = THETA0
4     thetap = THETAPO
5     THETA = [theta]
6     THETAp = [thetap]
7     for i in range(len(T)-1):
8         # On commence par le calcul de thetap(t+DT) :
9         
10        THETAp.append(thetap)
11        # calcul de theta(t+DT) en fonction de thetap(t+DT) :
12        
13        THETA.append(theta)
14    return THETA
```

Avec le code ci-dessous on peut tracer les deux solutions sur le même graphique pour les comparer :

```
1 # Visualisation des courbes par les deux méthodes
2 Yexpl = euler_explicite()
3 Ysemi = euler_semi_implicite()
4 mpl.plot(T,Yexpl, label='Euler explicite')
5 mpl.plot(T,Ysemi, label='Euler semi-implicite')
6 mpl.legend()
7 mpl.show()
```



4 TD informatique : comparaison avec la solution linéarisée

Dans ce TD informatique, on va programmer la méthode d'Euler semi-implicite et comparer les solutions obtenues pour l'équation du pendule à la solution de l'équation linéarisée du pendule. On essayera de déterminer à partir de quel angle initial θ_0 les solutions de l'équation linéarisée et de l'équation non linéaire commencent à diverger significativement.