

Les programmes sont donnés : il s'agit essentiellement de changer à la marge les paramètres mais aussi de savoir représenter les solutions.

Résolution des équations de Physique typiques (en dimension 1 et d'ordre 2) :

$$y'' = f(t, y, y')$$

Pour résoudre numériquement, on se ramène au système différentiel d'ordre 1 en dimension 2 :

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

La résolution des systèmes $X'(t) = F(t, X(t))$ se résout numériquement avec PYTHON par le pseudo-code :

```
def F(X,t):                # X vecteur
    return np.array([...])  # on renvoie le vecteur F(t, X)
T = np.arange(0,tmax+pas,pas) # discrétisation du temps T = [t0, ..., tmax]
X = integr.odeint(F,X0,T)    # X0 est le vecteur initial (conditions initiales)
```

La résolution renvoie une matrice $X[i, j]$ telle que le vecteur $X[i]$ donne une évaluation de $X(t_i)$.

Autrement dit, $X[:, 0]$ renvoie la première coordonnée de X en fonction du temps.

Exemple : On résout $y'' + y' + \sin(y) = \cos(t)$ et $(y(0), y'(0)) = (1, 0)$

Ce qui correspond au système $\begin{cases} y' = z \\ z' = \cos(t) - z - \sin(y) \end{cases}$

Le code donné sera donc de la forme (cf memento et modules d'analyse numérique) :

```
00. import numpy as np
00. import scipy.optimize as resol
00. import scipy.integrate as integr
00. import matplotlib.pyplot as plt

01. def F(X,t):                # X vecteur représentant (y, z) = (y, y')
02.     return np.array([X[1], np.cos(t)-X[1]-sin(X[0])])
03. T = np.arange(...)
04. X0 = ...
04. X = integr.odeint(F,X0,T)    # X0 condition initiale
05. plt.plot(T,X[:,0])
06. plt.show()
```

Ainsi, pour résoudre l'équation sur $t \in [0, 5]$, on prend

```
03. T = np.arange(0,5.01,0.01)
04. X0 = np.array([1,0])      ### ne pas oublier les crochets et les parenthèses
```

L'algorithme renvoie le tracé de $t \mapsto (t, y(t))$ sur $[0, 5]$, car $y(t)$ est la coordonnée de X d'indice 0.

Remarque : Si on voulait représenter le portrait de phase $t \mapsto (y(t), y'(t))$, on prendrait :

```
05. plt.plot(X[:,0],X[:,1])
```