Université Paris Cité

2023/2024

Programmation et traitement numérique

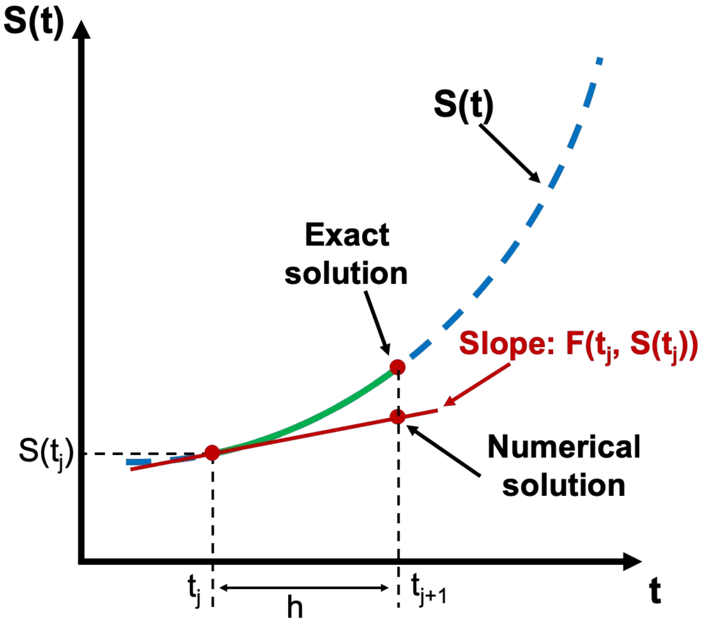
Python 3 : pour la physique

Équation différentiel ordinaire

FRERE Maxime

Approximer les solutions des EDO

- Les E.D.O. sont dénués de solution analytique (hors linéaire)
- Toutes les simulations complexes sont des résolutions d'équations différentielles
- Lors de ce type de résolution, toujours vérifier :
    - La convergence
    - La consistance
- Les solutions d'approximations
    - Euler
    - Runge-Kutta
    - Newmart
    - Différences finies
    - éléments finis



Les équations différentielles

La fonction est inconnue, on ne connaît que sa ou ses dérivés.

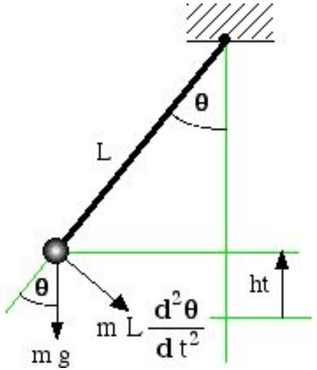
$$\begin{cases} x'(t_n) & = & f(t_n, x(t_n)), \\ x(t_0) & = & x_0 \ \text{donné.} \end{cases}$$

On doit résoudre l'équation sur papier pour exprimer la forme de la fonction (f(t,x)) et le ramener au premier ordre.

From a force balance we obtain:

$$m\,g\,\sin(\theta) + m\,L\,\frac{d^2\theta}{dt^2} = 0$$

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\sin(\theta)$$



On doit lui donner les conditions initiales.

Puis, on résout pas à pas. (ou en multi pas)

$$\begin{cases} x_{n+1} & = & x_n + hf(t_n, x_n), \ \ \text{pour} \ \ n = 0, ..., N-1, \\ x(t_0) & = & x_0. \end{cases}$$

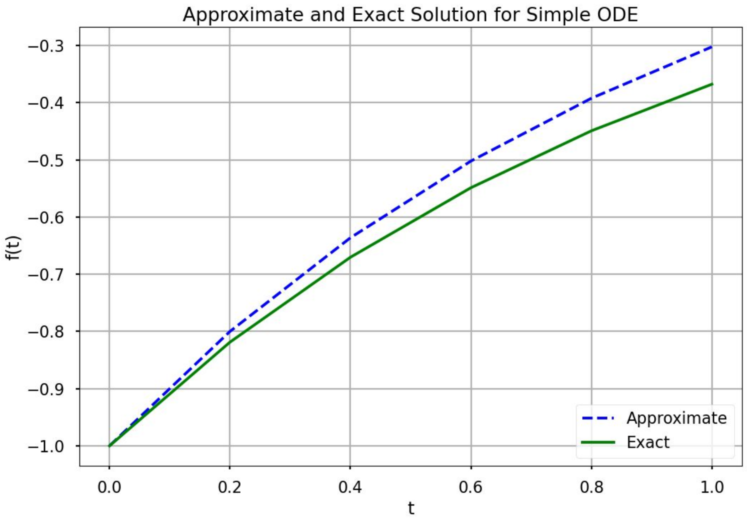
Euler Explicit

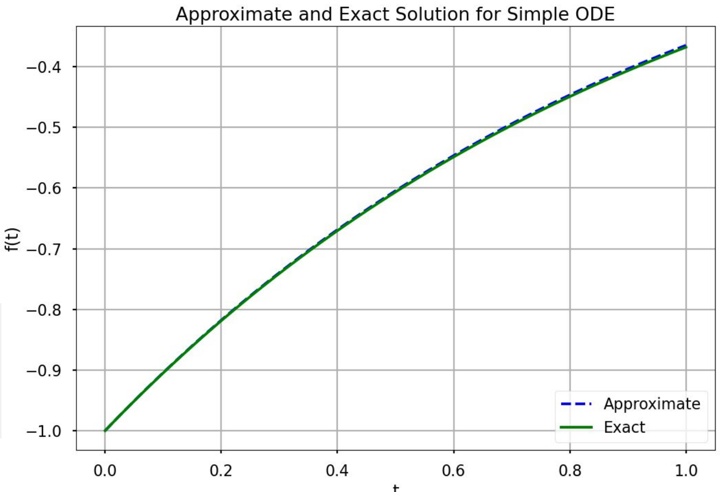
$$\frac{df(t)}{dt} = e^{-t}$$

$$f_0 = -1$$

$$\begin{cases} x'(t_n) & = & f(t_n, x(t_n)), \\ x(t_0) & = & x_0 \ \text{donné.} \end{cases}$$

$$\begin{cases} x_{n+1} & = & x_n + hf(t_n, x_n), \ \text{pour} \ n = 0, ..., N-1, \\ x(t_0) & = & x_0. \end{cases}$$

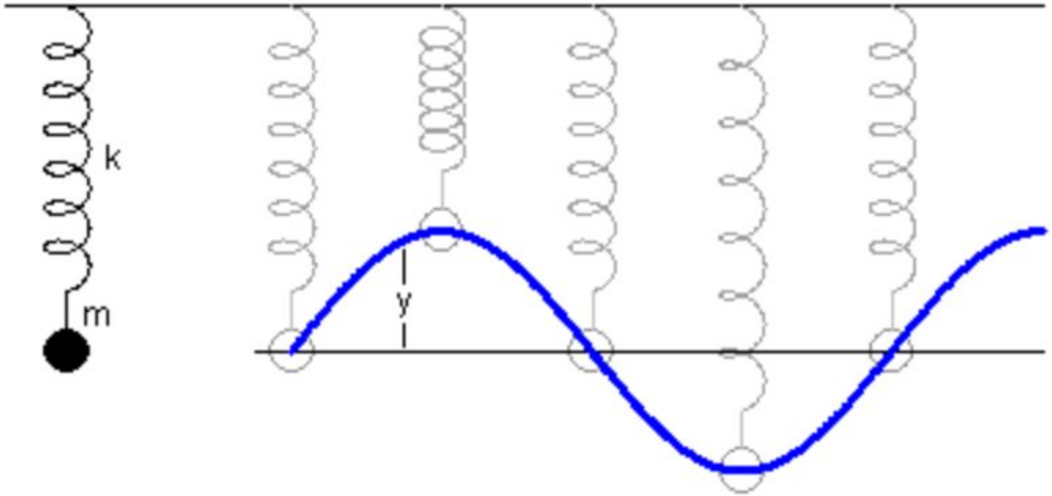




Cas concret : l'oscillateur harmonique

$$\frac{d^2x}{dt^2} + \omega_0^2\, x(t) = 0, \text{ avec } \omega_0 = \sqrt{\frac{k}{m}}.$$

$$x(t) = x_m \cos\left(\omega_0 t + \phi_0\right),$$



$$v = \frac{dx}{dt}$$

$$\frac{dx}{dt} = v$$

$$\frac{dv}{dt} = -\omega^2 x$$

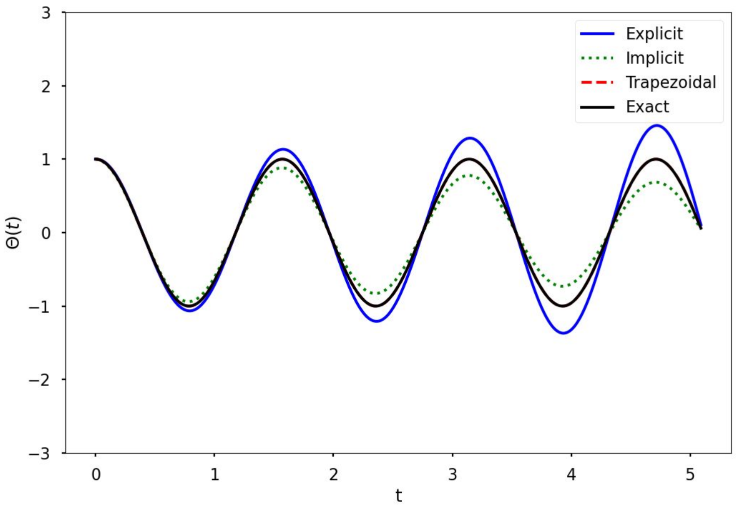
$$Y'(t) = f(Y, t)$$

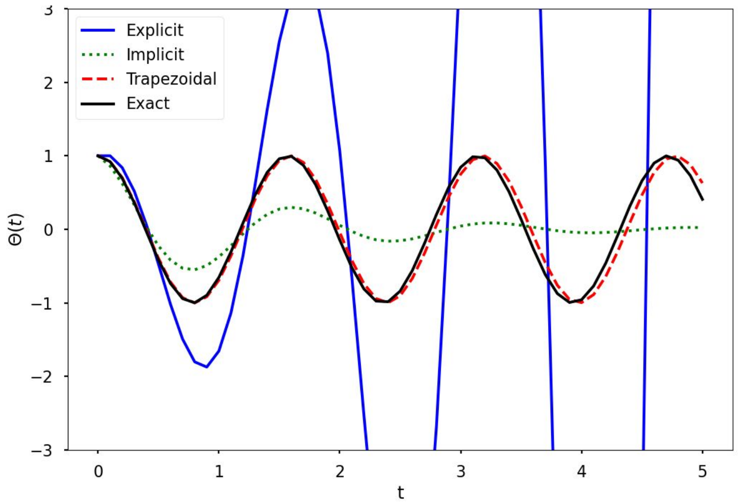
$$Y = \begin{pmatrix} x \\ v \end{pmatrix}$$

$$\frac{dY}{dt} = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix} Y$$

Convergence et stabilité

- Explicite :
  - Haute divergence
  - Basse stabilité
- Implicite :
  - Haute convergence
  - Basse stabilité
- Trapézoïdal :
  - Haute convergence
  - Haute stabilité





Runge-Kutta

$$y_{n+1} = y_n + \frac{1}{6}h\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

$$t_{n+1} = t_n + h$$

for $n = 0, 1, 2, 3, \ldots$, using[3]

$$\begin{aligned} k_1 &= \ f(t_n, y_n), \\ k_2 &= \ f\left(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right), \\ k_3 &= \ f\left(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}\right), \\ k_4 &= \ f\left(t_n + h, y_n + hk_3\right). \end{aligned}$$