Chapitre 2

Équations différentielles ordinaires (EDO)

Ibrahim ALAME

ESTP

23/01/2024

Equations différentielles

Problème de Cauchy

$$(\star)\begin{cases} y'(t) = f(t; y(t)) \text{ dans } I \\ y(t_0) = y_0 \end{cases}$$

- I un intervalle de $\mathbb{R}$
- f une fonction à deux variables de $I \times \mathbb{R}^n$ dans $\mathbb{R}^n$.
- $y : I \to \mathbb{R}^n$ la fonction inconnue.
- $t_0 (= 0)$ est un point de I.

$$\begin{cases} y_1'(t) = f_1(t, y_1(t), y_2(t), ...y_n(t)) \\ y_2'(t) = f_2(t, y_1(t), y_2(t), ...y_n(t)) \\ ..... \\ y_n'(t) = f_n(t, y_1(t), y_2(t), ...y_n(t)) \end{cases}$$

Quelques exemples

- $y' = y + 1$ est une équation différentielle linéaire du premier ordre. En se limitant à l'intervalle $[0; 5]$ et avec la condition $y(0) = 3$, on a un problème de Cauchy.
- $\begin{cases} x' = x + y + t \\ y' = 2x - 3y + t^2 \end{cases}$ est un système différentiel à deux inconnues du premier ordre. On se ramène à la forme générale $Y' = F(t; Y)$ en posant une fonction inconnue vectorielle

$$Y(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

- Pour une équation différentielle du second ordre comme $y'' + y' + y = t$ on peut se ramener à la forme présentée en posant une fonction inconnue vectorielle

$$Y(t) = \begin{pmatrix} y'(t) \\ y(t) \end{pmatrix}$$

Une équation différentielle d'ordre p de la forme

$$y^{(p)} = f(t, y, y', \cdots, y^{(p-1)})$$

où $p > 1$ se ramène à un système équivalent en posant :

$$\begin{cases} z_1'(t) = z_2(t) \\ z_2'(t) = z_3(t) \\ \cdots\cdots \\ z_p'(t) = f(t, z_1(t), z_2(t), \ldots z_{p-1}(t)) \end{cases}$$

autrement dit :

$$z'(t) = F(t; z(t))$$

Exemple

$$y''(t) - ay'(t) - by(t) = f(t)$$

On pose $z_1 = y$ et $z_2 = y'$ d'où le système

$$\begin{cases} z_1'(t) = z_2(t) \\ z_2'(t) = bz_1(t) + az_2(t) + f(t) \end{cases}$$

Theorem (de Cauchy-Lipschitz)

Soit f une application continue :

$$f : (t, z) \in (0, T) \times \mathbb{R}^n \mapsto f(t, z) \in \mathbb{R}^n$$

Si f vérifie la condition de Lipschitz,

$$\forall t \in [0, T], \quad \forall |z_1|, |z_2| \in \mathbb{R}^n, \quad \|f(t, z_1) - f(t, z_2)\| \leq L\|z_1 - z_2\|$$

alors il y a existence et unicité locale de la solution du problème de Cauchy.

Le théorème est admis dans ce cours, il s'obtient via un argument de point fixe pour les applications contractantes dans l'espace de Banach...

Nous considérons maintenant trois exemples typiques :

- $y'(t) = ay(t)$

$$f(t,y) = ay \Longrightarrow |f(t,y_1) - f(t,y_2)| = a|y_1 - y_2|$$

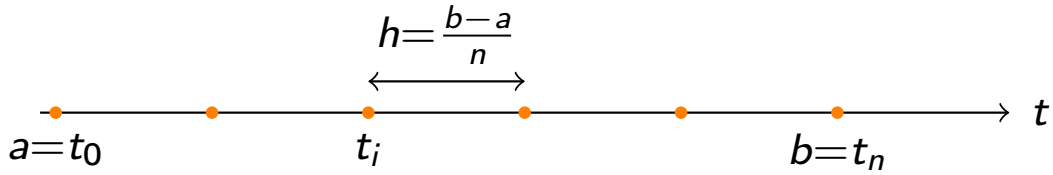
- $y'(t) = [y(t)]^2$.
  Nous vérifions facilement que $f(t,y) = y^2$ est localement lipschitzienne car f est de classe C^1 donc localement lipschitzienne d'après le Théorème des accroissements fnis.
- $y'(t) = \sqrt[3]{y(t)}$.
  La fonction $f(t,y) = \sqrt[3]{y(t)}$ n'est pas lipschitzienne car elle n'est pas dérivable en 0 et donc le théorème ne s'applique pas !

Méthodes numériques

On va s'intéresser à différentes méthodes numériques de calcul d'approximations de solutions d'équations (ou de systèmes) différentielles. Plus précisément de résolution de problèmes de Cauchy (une équation assortie d'une condition initiale garantissant l'unicité de la solution).

Principe

On subdivise l'intervalle $[a; b]$ en n intervalles de longueur h :



- On a obtenu des nœuds t_i, $0 \leq i \leq n$. $(n+1$ nœuds).
- On cherche à calculer des quantités y_i, une valeur approchée de la vraie valeur $y(t_i)$ de la solution du problème de Cauchy, que l'on ne connait pas.

La méthode d'Euler

Problème de Cauchy :

$$(\star)\begin{cases} y'(t) = f(t; y(t)) \text{ dans } I \\ y(t_0) = y_0 \end{cases}$$

On cherche à approcher $y(t_i) \simeq y_i$. Taylor à l'ordre 1 :

$$y(t_i + h) = y(t_i) + hy'(t_i) + o(h) \text{ où } h = t_{i+1} - t_i = \frac{T}{n}$$

or $y'(t_i) = f(t_i, y(t_i))$ d'où "l'idée" de considérer le schéma suivant :

Schéma d'Euler explicite :

$$\begin{cases} y_{i+1} = y_i + hf(t_i, y_i) \\ y_0 = y(0) \end{cases}$$

Un exemple (trop ?) élémentaire

On considère le problème de Cauchy sur $I = [0; 1]$ décrit par l'équation $y' = y$ et $y(0) = 1$. On subdivise I en n intervalles de longueur $h = \frac{1}{n}$.
Ici $f(t; y) = y$. Le schéma d'Euler explicite s'écrit donc

$$y_{i+1} = y_i + hy_i = (1 + h)y_i$$

Pour tenir compte de la condition initiale, on a $y_0 = 1$.
On peut écrire ici $y_k = (1 + h)^k y_0 = (1 + h)^k$.

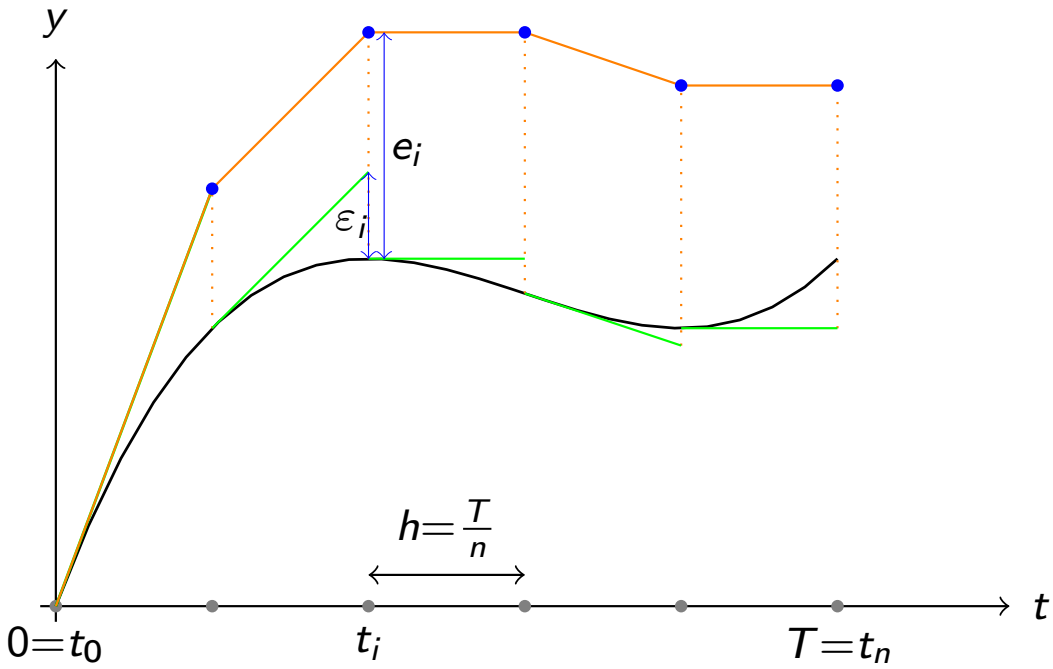
Un exemple (trop ?) élémentaire

On peut aussi résoudre explicitement le problème de Cauchy qui a pour solution $y(t) = e^t$. On peut se demander si y_k approche bien $y(t_k) = e^{kh}$. On est amené à majorer $|y(t_k) - y_k|$ pour k vérifiant $0 \leq k \leq n$. On obtient

$$\max_{0 \leq k \leq n} |y(t_k) - y_k| \leq \frac{e}{2} h$$

On dira, pour cet exemple, que la méthode d'Euler explicite converge (l'erreur tend vers 0 lorsque $n \to \infty$ ou, ce qui est équivalent, lorsque $h \to 0$).

La méthode d'Euler



- $e_i = y(t_i) - y_i$ l'erreur commise au point t_i
- $\varepsilon_{i+1} = y(t_{i+1}) - y(t_i) - hf(t_i, y(t_i))$ l'erreur de consistance, c'est l'erreur systématique commise sur y_{i+1} : c'est l'erreur de passage de la valeur à l'instant t_i supposée exacte à la valeur approchée en t_{i+1}.

Etude de l'erreur

On a

$$\begin{aligned}
|e_{i+1}| &= |y(t_{i+1}) - y_{i+1}| \\
&= |y(t_{i+1}) - y(t_i) - hf(t_i, y(t_i)) \\
&\quad + y(t_i) + hf(t_i, y(t_i)) \\
&\quad - y_i + y_i \\
&\quad + hf(t_i, y_i) - hf(t_i, y_i) \\
&\quad - y_{i+1}| \\
&\leq |\varepsilon_{i+1}| + |e_i| + h\,|f(t_i, y(t_i)) - f(t_i, y_i)|
\end{aligned}$$

Si f est lipschitzienne en y :

$$\boxed{|e_{i+1}| \leq |\varepsilon_{i+1}| + (1 + Lh)\,|e_i|}$$

d'autre part

$$\begin{aligned} |\varepsilon_{i+1}| &= |y(t_{i+1}) - y(t_i) - hf(t_i, y(t_i))| \\ &= \left| \int_{t_i}^{t_{i+1}} y'(s)\mathrm{d}s - \int_{t_i}^{t_{i+1}} f(t_i, y(t_i))\mathrm{d}s \right| \\ &= \left| \int_{t_i}^{t_{i+1}} (y'(s) - y'(t_i))\mathrm{d}s \right| \\ &\leq M \cdot \frac{h^2}{2} \end{aligned}$$

où $M = \sup_{[0,T]} |f''|$. Finalement

$$\boxed{|e_{i+1}| \leq (1 + Lh)\,|e_i| + M \cdot \frac{h^2}{2}}$$

Lemme

Soit $(\theta_i)_i$ une suite de nombres positifs satisfaisant :

$$\theta_{i+1} \leq (1 + A)\theta_i + B$$

où A et B sont des constantes strictement positives alors :

$$\theta_i \leq \exp(iA)\theta_0 + \frac{\exp(iA) - 1}{A} B$$

Démonstration par récurrence. Grâce au lemme, on a :

$$|e_i| \leq \exp(iLh)|e_0| + \frac{\exp(iLh) - 1}{Lh} \cdot M\frac{h^2}{2}$$

or $e_0 = 0$ d'où le théorème :

Estimation de l'erreur (Euler explicite)

Théorème (convergence et erreur pour Euler explicite)

Supposons que les conditions du théorème de Cauchy-Lipschitz soient satisfaites (f est L-lipschitzienne) , avec y une solution de classe $\mathscr{C}^2$.
On note $e_k = y(t_k) - y - k$ et $M = \sup_{t\in I} |y''(t)|$.
On a la majoration pour $0 \leq k \leq n$:

$$|e_k| \leq \frac{e^{khL} - 1}{2L} Mh$$

Donc

$$\max_{0\leq i\leq N} |e_i| \leq \frac{\exp(LT) - 1}{2L} Mh$$

Et

$$\lim_{n\to\infty} \max_{0\leq k\leq n} |e_k| = 0$$

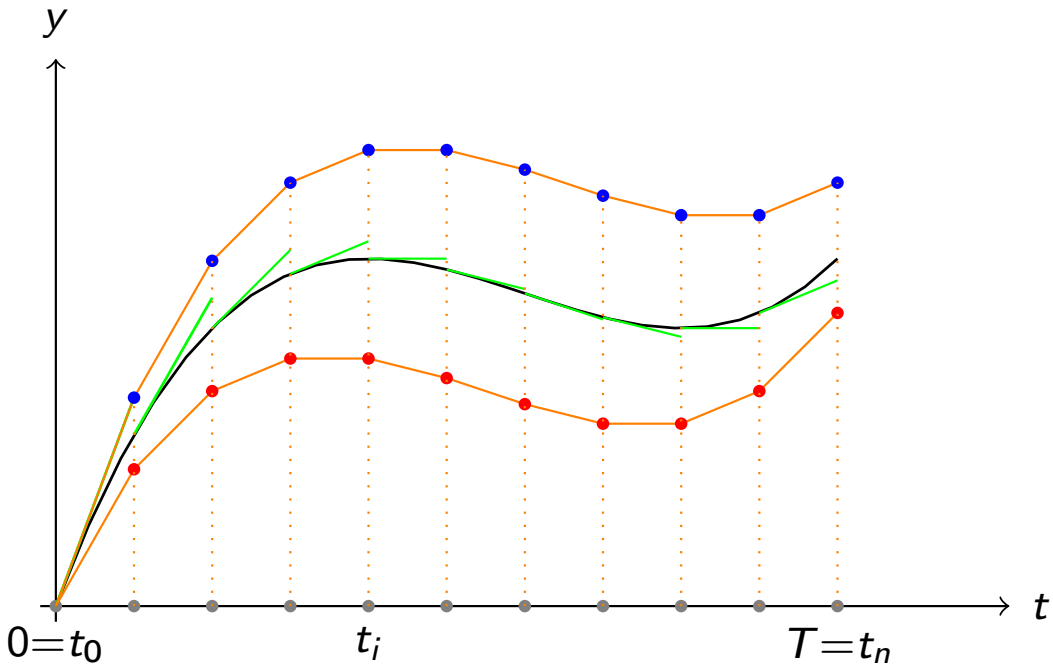
à partir de ce théorème on peut démontrer le résultat suivant :

Proposition

Si y_h est la fonction affine par morceaux telle que $y_h(t_i) = y_i$ alors

$$\|y - y_h\|_\infty \longrightarrow 0 \text{ quand } h \longrightarrow 0$$

La méthode d'Euler implicite



$$y(t_i) = y(t_{i+1} - h) = y(t_{i+1}) - hy'(t_{i+1}) + o(h)$$

Donc :

$$y(t_{i+1}) = y(t_i) + hf(t_{i+1}, y(t_{i+1}) + o(h)$$

d'où "l'idée" de considérer le schéma implicite :

$$\begin{cases} y_{i+1} = y_i + hf(t_{i+1}, y_{i+1}) \\ y_0 = y(0) \end{cases}$$

La méthode d'Euler implicite

À noter que cette formule est implicite, au sens où pour calculer y_{i+1} à partir de y_n, il faut résoudre une équation d'inconnue y_{i+1}. En général, cette équation ne peut être résolue que par des procédés numériques (par exemple, méthode de dichotomie, point fixe, Newton...). Ce qui rend la méthode souvent très coûteuse à mettre en œuvre.
Dans le cas de l'exemple étudié plus haut, la résolution est aisée. Le schéma

$$\begin{cases} y_{i+1} = y_i + hy_{i+1} \\ y_0 = 1 \end{cases}$$

donne

$$\begin{cases} y_{i+1} = \frac{1}{1-h} y_i \\ y_0 = 1 \end{cases}$$

et on parvient à expliciter $y_k = (1-h)^{-k}$.

Étude générale des méthodes à un pas

Définition

Les méthodes à un pas sont des méthodes de la forme :

$$\begin{cases} y_{i+1} = y_i + h\,\Phi(t, y_i, h) \text{ pour } i = 0, 1, \cdots, n-1 \\ y_0 = y_{0,h} \end{cases}$$

Où Φ est une fonction continue sur $[0, T] \times \mathbb{R}^p \times [0, H]$, H désignant un pas de discrétisation maximal. C'est cette fonction qui caractérise le schéma.

Remarque : si $\Phi(t; y; h) = f(t; y)$, on obtient la méthode d'Euler explicite.

Consistance d'un schéma à un pas :

Définition

La méthode est dite consistante si $\Sigma_h \longrightarrow 0$ quand $h \longrightarrow 0$

$$\Sigma_h = \sum_{i=0}^{N-1} |y(t_{i+1}) - y(t_i) - h\Phi(t_i, y(t_i), h)|$$

Théorème (à admettre)

Une condition nécessaire et suffisante pour qu'une méthode à un pas soit consistante est que pour tout $t \in [0, T]$ et $y \in \mathbb{R}^p$:

$$\phi(t, y, 0) = f(t, y)$$

Consistance d'un schéma à un pas :

$$\begin{aligned}
\varepsilon_{i+1} &= y(t_{i+1}) - y(t_i) - h\Phi(t_i, y(t_i), h) \\
&= \int_{t_i}^{t_{i+1}} y'(s)\mathrm{d}s - \int_{t_i}^{t_{i+1}} \Phi(t_i, y(t_i), h)\mathrm{d}s \\
&= \int_{t_i}^{t_{i+1}} (f(s, y(s)) - \Phi(t_i, y(t_i), h))\mathrm{d}s
\end{aligned}$$

$$\begin{aligned}
|f(s, y(s)) - \Phi(t_i, y(t_i), h)| \leq\ & |f(s, y(s)) - f(t_i, y(t_i))| \\
& + |f(t_i, y(t_i)) - \Phi(t_i, y(t_i), 0)| \\
& + |\Phi(t_i, y(t_i), 0) - \Phi(t_i, y(t_i), h)|
\end{aligned}$$

Un argument de continuité uniforme de f,y et Φ permet de conclure.

Stabilité

On perturbe le schéma de la façon suivante

$$\begin{cases} \tilde{y}_{i+1} = \tilde{y}_i + h\Phi(t, \tilde{y}_i, h) + \varepsilon_i \text{ pour } i = 0, 1, \cdots, N-1 \\ \tilde{y}_0 = \tilde{y}_{0,h} \end{cases}$$

La méthode est dite stable s'il existe deux constantes S_1 et S_2 telles que :

$$\max_i \leq S_1 \left| y_{0,h} - \tilde{y}_{0,h} \right| + S_2 \sum_{i=0}^{N-1} |\varepsilon_i|$$

Theorem (Condition suffisante de stabilité)

La méthode à un pas est stable si Φ est L-lipschitzienne par rapport à la deuxième variable y avec L indépendant de deux autres variables t et h.

On note θ_i la différence entre la solution y_i et sa perturbation $\tilde{y}_i$. A l'aide de l'inégalité triangulaire on obtient

$$\theta_{i+1} \leq (1 + Lh)\theta_i + |\varepsilon_i|$$

$$\frac{\theta_{i+1}}{(1+Lh)^{i+1}} \leq \frac{\theta_i}{(1+Lh)^i} + \frac{|\varepsilon_i|}{(1+Lh)^{i+1}}$$

Par sommation (télescopique) :

$$\frac{\theta_i}{(1+Lh)^i} \leq \frac{\theta_0}{(1+Lh)^0} + \sum_{k=0}^{i-1} \frac{|\varepsilon_k|}{(1+Lh)^{k+1}}$$

$$\theta_i \leq (1+Lh)^i \theta_0 + \sum_{k=0}^{i-1} (1+Lh)^{i-k} |\varepsilon_i|$$

$$\max_i \theta_i \leq \exp(LT)\theta_0 + \exp(LT) \sum_{k=0}^{N-1} |\varepsilon_i|$$

D'où le résultat avec $S_1 = S_2 = \exp(LT)$.

Theorem

Tout méthode stable et consistante converge.

Définition : La méthode est d'ordre $p \geq 1$ s'il existe $C > 0$ telle que

$$|\varepsilon_i| = |y(t_{i+1}) - y(t_i) - h\Phi(t_i, y(t_i), h)| \leq Ch^{p+1}$$

Théorème : La méthode est d'ordre p si pour tout t, z et $k \leq p-1$:

$$\begin{aligned}
\Phi(t,z,0) &= f(t,z) \\
\frac{\partial \Phi}{\partial h}(t,z,0) &= \frac{1}{2} f^{[1]}(t,z) \\
&\vdots \\
\frac{\partial^k \Phi}{\partial h^k}(t,z,0) &= \frac{1}{k+1} f^{[k]}(t,z)
\end{aligned}$$

Dans ce cas

$$|\varepsilon_i| = \frac{1}{i!} h^{p+1} \left(\frac{1}{p+1} f^{[p]}(t_i, y(t_i)) - \frac{\partial^p \Phi}{\partial h^p}(t_i, y(t_i), 0) \right) + o(h^{p+1})$$

où $f^{[k]}$ est défini par $f^{[0]} = f$ et $f^{[k+1]}(t,z) = \frac{\partial f^{[k]}(t,z)}{\partial t} + \frac{\partial f^{[k]}(t,z)}{\partial z} \times f(t,z)$

exercise

Soit la méthode à un pas définie par :

$$\Phi(t,z,h) = \alpha_1 f(t,z) + \alpha_2 f(t+\beta_1 h, z+\beta_2 h f(t,z))$$

Déterminer les conditions sur les coefficients pour que la méthode soit d'ordre le plus élevé possible.

- La méthode est consistante d'ordre 1 si : $\Phi(t,z,0) = f(t,z)$ cela implique $\alpha_1 + \alpha_2 = 1$, car

$$\Phi(t,z,0) = \alpha_1 f(t,z) + \alpha_2 f(t+0, z+0)$$

- La méthode est d'ordre 2 si

$$\frac{\partial \Phi}{\partial h}(t,z,0) = \frac{1}{2} f^{[1]}(t,z)$$

où $f^{[1]}(t,z) = \frac{\partial f(t,z)}{\partial t} + \frac{\partial f(t,z)}{\partial z} \times f(t,z)$.

On a

$$\begin{aligned}\frac{\partial \Phi}{\partial h}(t,z,h) &= \alpha_2\beta_1 \frac{\partial f}{\partial t}(t+\beta_1 h, z+\beta_2 h f(t,z)) \\ &\quad +\alpha_2\beta_2 \frac{\partial f}{\partial z}(t+\beta_1 h, z+\beta_2 h f(t,z)) \cdot f(t,z)\end{aligned}$$

Nous avons l'égalité en $h=0$ si $\alpha_2\beta_1 = \alpha_2\beta_2 = \frac{1}{2}$. On vérifie facilement que l'on ne peut pas aller plus loin. Finalement, si on pose $\alpha_2 = \alpha$:

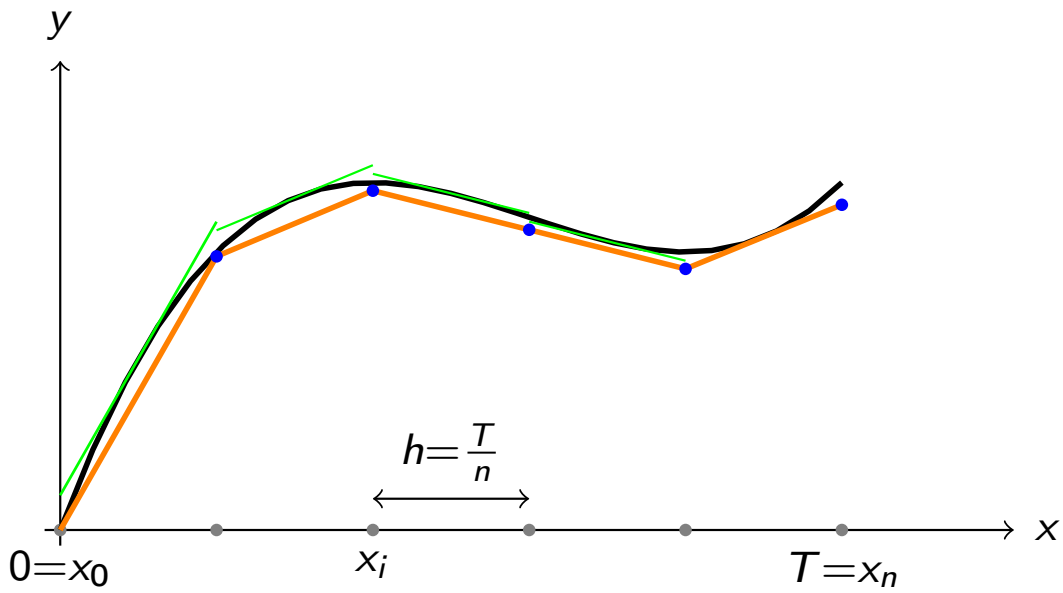
$$\boxed{\Phi(t,z,h) = (1-\alpha)f(t,z) + \alpha f(t+\frac{h}{2\alpha}, z+\frac{h}{2\alpha}f(t,z))}$$

On retrouve

- La méthode de la tangente améliorée si $\alpha = 1$.
- La méthode d'Euler modifiée si $\alpha = \frac{1}{2}$.
- La méthode de Heun si $\alpha = \frac{3}{4}$.

La méthode d'Euler améliorée ($\alpha = 1$)

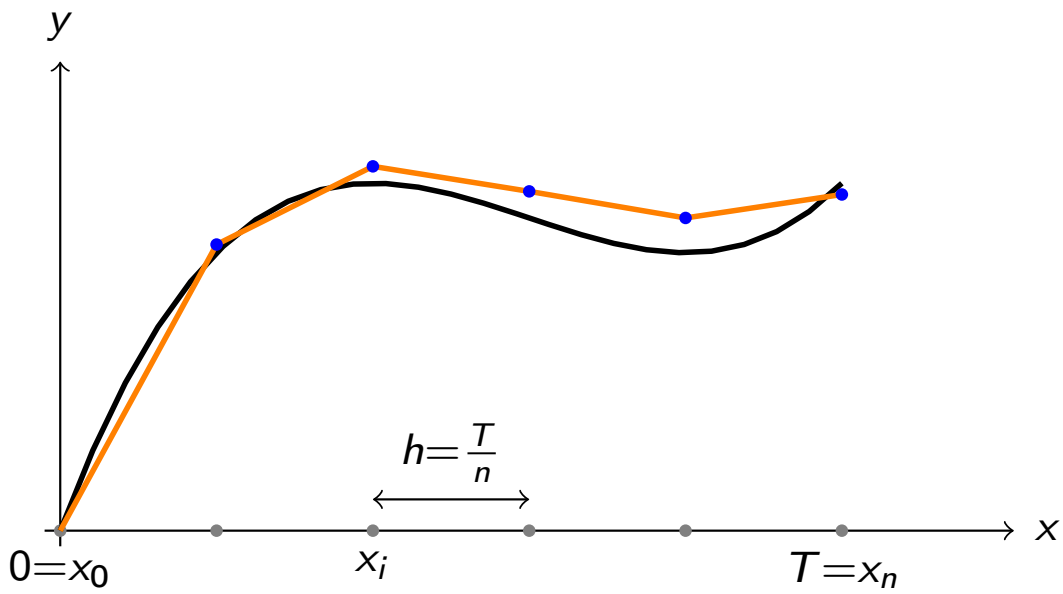
$$\Phi(t, z, h) = f\left(t + \frac{h}{2}, z + \frac{h}{2} f(t, z)\right)$$



$$\begin{cases} y_{i+1} = y_i + hf\left(t_i + \frac{h}{2}, y_i + \frac{h}{2} f(t_i, y_i)\right) \\ y_0 = y(0) \end{cases}$$

La méthode d'Euler modifiée ($\alpha = 1/2$)

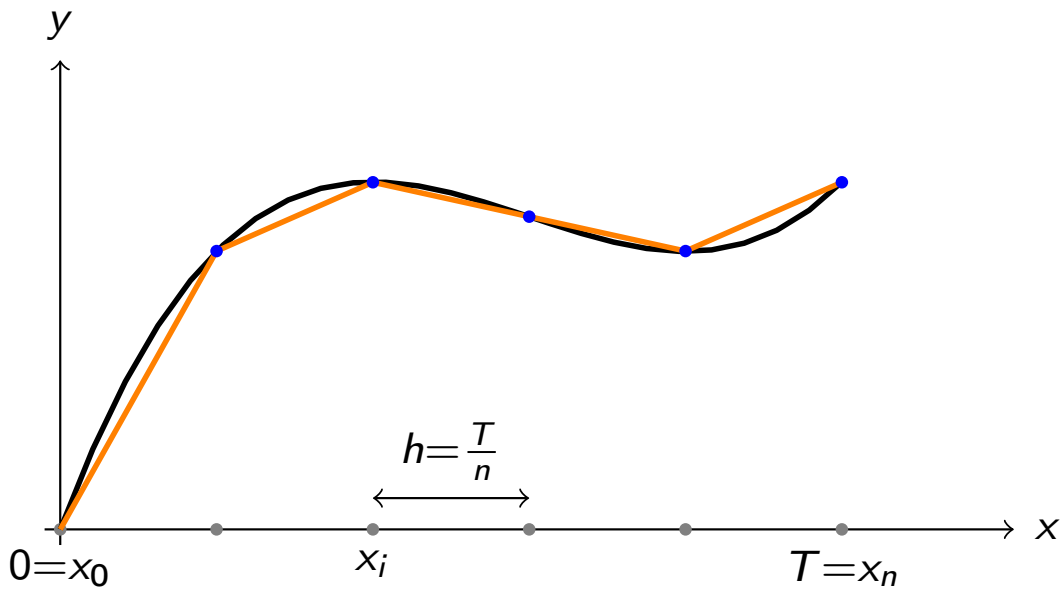
$$\Phi(t, z, h) = \frac{1}{2}\left[f(t, z) + f(t + h, z + hf(t, z))\right]$$



$$\begin{cases} y_{i+1} = y_i + \frac{h}{2}\left[f(t_i, y_i) + f(t_i + h, y_i + hf(t_i, y_i))\right] \\ y_0 = y(0) \end{cases}$$

La méthode de Heun si $(\alpha = \frac{3}{4})$.

$$\Phi(t,z,h) = \frac{1}{4}\left[f(t,z) + 3f(t+\frac{2}{3}h, z+\frac{2h}{3}f(t,z))\right]$$



$$\begin{cases} y_{i+1} = y_i + \frac{h}{4}\left[f(t_i,y_i) + 3f(t_i+\frac{2h}{3}, y_i+\frac{2h}{3}f(t_i,y_i))\right] \\ y_0 = y(0) \end{cases}$$

Méthode de Runge-Kutta

L'idée commence dans l'écriture intégrale de l'équation différentielle :

$$y'(s) = f(s, y(s)) \Longrightarrow \int_{t_i}^{t_{i+1}} y'(s)\mathrm{d}s = \int_{t_i}^{t_{i+1}} f(s, y(s))\mathrm{d}s$$

$$y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(s, y(s))\mathrm{d}s$$

Puis on utilise une formule de quadrature pour approcher l'intégrale

$$\boxed{y(t_{i+1}) = y(t_i) + \sum_{j=0}^{q} b_j f(t_{i,j}, y(t_{i,j}))}$$

Les $t_{i,j}$ sont des points intermédiaire entre t_i et $t_i + h$.

Méthode de Runge-Kutta

On écrit $t_{i,j} = t_i + c_j h$. Une nouvelle formule de quadrature à j points permet d'approcher $y_{i,j}$ avec des coefficients dependant de j :

$$\begin{cases} t_{i,j} & = & t_i + c_j h \\ y(t_{i,j}) & = & y(t_i) + \sum_{k=0}^{j-1} a_{i,k} f(t_{i,k}, y(t_{i,k})) \end{cases}$$

En résumé, on calcul les valeurs de y aux points intermédiaires par des formules de quadrature faisant intervenir les points précédents et les valeurs déjà calculées.

On schématise souvent la méthode de Runge et Kutta par le tableau

$$\begin{array}{c|ccccc} 0 & 0 & & & & \\ c_2 & a_{2,1} & & & & \\ c_3 & a_{3,1} & a_{3,2} & & & \\ \vdots & \vdots & & \ddots & & \\ c_q & a_{q,1} & a_{q,2} & \cdots & a_{q,q-1} & \\ \hline & b_1 & b_2 & \cdots & b_{q-1} & b_q \end{array}$$

Méthode de Runge-Kutta

Ce tableau représente le schéma suivant :

$$
\begin{cases}
\begin{cases} t_{i,1} = t_i + 0 \times h = t_i \\ y_{i,1} = y_i + 0 \times f(\cdot,\cdot) = y_i \end{cases} \\
\begin{cases} t_{i,2} = t_i + c_2 h \\ y_{i,2} = y_i + a_{2,1} h f(t_{i,1}, y_{i,1}) \end{cases} \\
\begin{cases} t_{i,3} = t_i + c_3 h \\ y_{i,3} = y_i + a_{3,1} h f(t_{i,1}, y_{i,1}) + a_{3,2} h f(t_{i,2}, y_{i,2}) \end{cases} \\
\vdots \\
\begin{cases} t_{i,q} = t_i + c_q h \\ y_{i,q} = y_i + a_{q,1} h f(t_{i,1}, y_{i,1}) + a_{q,2} h f(t_{i,2}, y_{i,2}) + \cdots + a_{q,q-1} h f(t_{i,q-1}, y \end{cases} \\
\\
y_{i+1} = y_i + h\left[b_1 f(t_{i,1}, y_{i,1}) + b_2 f(t_{i,2}, y_{i,2}) + \cdots + b_{q-1} f(t_{i,q-1}, y_{i,q-1}) + \right.
\end{cases}
$$

La méthode de Runge-Kutta classique $q = 1$ (RK1) :

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

$$\begin{cases} \begin{cases} t_{i,1} = t_i \\ y_{i,1} = y_i \end{cases} \\ y_{i+1} = y_i + hf(t_{i,1}, y_{i,1}) \end{cases}$$

C'est la méthode d'Euler explicite :

$$y_{i+1} = y_i + hf(t_i, y_i)$$

La méthode de Runge-Kutta classique $q = 2$ (RK2) :

$$\begin{array}{c|cc} 0 & 0 & \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$$

$$\begin{cases} \begin{cases} t_{i,1} = t_i \\ y_{i,1} = y_i \end{cases} \\ \begin{cases} t_{i,2} = t_i + \frac{1}{2}h \\ y_{i,2} = y_i + \frac{1}{2}hf(t_{i,1}, y_{i,1}) \end{cases} \\ y_{i+1} = y_i + h\left[0 \times f(t_{i,1}, y_{i,1}) + 1 \times f(t_{i,2}, y_{i,2})\right] \end{cases}$$

C'est le schéma du point milieu :

$$\begin{cases} \overline{y}_i = y_i + \frac{h}{2}f(t_i, y_i) \\ y_{i+1} = y_i + h\, f(t_i + \frac{h}{2}, \overline{y}_i) \end{cases}$$

La méthode de Runge-Kutta classique $q = 2$ (RK2) :

$$\begin{array}{c|cc} 0 & 0 & \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$\begin{cases} \begin{cases} t_{i,1} = t_i \\ y_{i,1} = y_i \end{cases} \\ \begin{cases} t_{i,2} = t_i + h \\ y_{i,2} = y_i + hf(t_{i,1}, y_{i,1}) \end{cases} \\ y_{i+1} = y_i + h\left[\frac{1}{2} \times f(t_{i,1}, y_{i,1}) + \frac{1}{2} \times f(t_{i,2}, y_{i,2})\right] \end{cases}$$

C'est le schéma de Heun. Ce dernier schéma est également appelé parfois le schéma de Runge-Kutta 2, même si le schéma du point milieu est lui aussi un schéma d'ordre 2 de type Runge-Kutta.

$$\begin{cases} \overline{y}_i = y_i + hf(t_i, y_i) \\ y_{i+1} = y_i + \frac{h}{2}\left[f(t_i, y_i) + f(t_i + h, \overline{y}_i)\right] \end{cases}$$

La méthode de Runge-Kutta classique $q = 4$ (RK4) :

$$
\begin{array}{c|cccc}
0 & 0 & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\hline
 & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6}
\end{array}
$$

$$
\begin{cases}
\begin{cases} t_{i,1} = t_i \\ y_{i,1} = y_i \end{cases} \\
\begin{cases} t_{i,2} = t_i + \frac{1}{2}h \\ y_{i,2} = y_i + \frac{1}{2}hf(t_{i,1}, y_{i,1}) \end{cases} \\
\begin{cases} t_{i,3} = t_i + \frac{1}{2}h \\ y_{i,3} = y_i + \frac{1}{2}hf(t_{i,2}, y_{i,2}) \end{cases} \\
\begin{cases} t_{i,4} = t_i + h \\ y_{i,4} = y_i + hf(t_{i,3}, y_{i,3}) \end{cases} \\
y_{i+1} = y_i + h\left[\frac{1}{6}f(t_{i,1}, y_{i,1}) + \frac{2}{6}f(t_{i,2}, y_{i,2}) + \frac{2}{6}f(t_{i,3}, y_{i,3}) + \frac{1}{6}f(t_{i,4}, y_{i,4})\right]
\end{cases}
$$

Nous admettons que l'ordre de cette méthode est 4.

La méthode de Runge-Kutta classique d'ordre 4 (RK4) :

Algorithme

Cette méthode s'écrit aussi plus couramment :

$$\begin{cases} k_1 = f(t_i, y_i) \\ k_2 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2}k_1) \\ k_3 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2}k_2) \\ k_4 = f(t_i + h, y_i + hk_3) \\ y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{cases}$$

Cette méthode converge en $\frac{1}{n^4}$.

exercise

Estimer l'erreur en $t = 1$ commise en appliquant la méthode de Runge et Kutta à 4 points (RK4) à l'équation différentielle

$$\begin{cases} y'(t) = -y(t) & t \in [0,1] \\ y(0) = 1 \end{cases}$$

Nous avons

$$\begin{cases} y_{i,1} = y_i \\ y_{i,2} = y_i + \frac{h}{2} f(t_{i,1}, y_{i,1}) = (1 - \frac{h}{2}) y_i \\ y_{i,3} = y_i + \frac{1}{2} f(t_{i,2}, y_{i,2}) = y_i - \frac{h}{2} y_{i,2} = (1 - \frac{h}{2} + \frac{h^2}{4}) y_i \\ y_{i,4} = y_i + f(t_{i,3}, y_{i,3}) = y_i - \frac{h}{2} y_{i,3} = (1 - h + \frac{h^2}{2} - \frac{h^3}{4}) y_i \end{cases}$$

$$\begin{cases} y_{i+1} &= y_i + \frac{1}{6} f(t_{i,1}, y_{i,1}) + \frac{2}{6} f(t_{i,2}, y_{i,2}) + \frac{2}{6} f(t_{i,3}, y_{i,3}) + \frac{1}{6} f(t_{i,4}, y_{i,4}) \\ &= y_i - \frac{1}{6} (y_{i,1} + 2y_{i,2} + 2y_{i,3} + y_{i,4}) \\ &= \left(1 - h + \frac{h^2}{2} - \frac{h^3}{6} + \frac{h^4}{24}\right) y_i \end{cases}$$

Le schéma se réduit à

$$\begin{cases} y_{i+1} = \left(1 - h + \frac{h^2}{2} - \frac{h^3}{6} + \frac{h^4}{24}\right) y_i \\ y_0 = 1 \end{cases}$$

D'où $y_n = \left(1 - h + \frac{h^2}{2} - \frac{h^3}{6} + \frac{h^4}{24}\right)^n$ or $h = \frac{1}{n}$ donc

$$y_n = \left(1 - \frac{1}{n} + \frac{1}{2n^2} - \frac{1}{6n^3} + \frac{1}{24n^4}\right)^n = \mathrm{e}^{-1} \left(1 - \frac{1}{120n^4} + o(\frac{1}{n^4})\right)$$

d'où l'erreur

$$e = y(t_n) - y_n \sim -\frac{1}{120\,\mathrm{e}\,n^4}$$

Pour $n = 10$, nous avons $y_{10} = 0,3678798$, la solution exacte étant $y(t) = \mathrm{e}^{-t}$ et $\mathrm{e}^{-1} = 0,3678794$. y_{10} est donc une approximation de e^{-1} à mieux que 10^{-6} près.

Exemple : Méthode d'Euler explicite

Soit l'équadiff

$$\begin{cases} (x+1)y' + 2y = \sin(x)e^{-x/5} \\ y(0) = -1/5 \end{cases}$$

dont la solution exacte est

$$y(x) = \frac{1912/5 - [(325x + 450)\cos x + (65x - 235)\sin x]\,\mathrm{e}^{-\frac{x}{5}}}{338(x+1)^2}$$

```
import numpy as np
import matplotlib.pyplot as plt
def f(x):
    N=1912/5-((325*x+450)*np.cos(x)+(65*x-235)*np.sin(x))*np.exp(-x/5)
    D=338*(x+1)**2
    return N/D
x=np.linspace(0,15,100)
plt.plot(x,f(x))
plt.show()
```

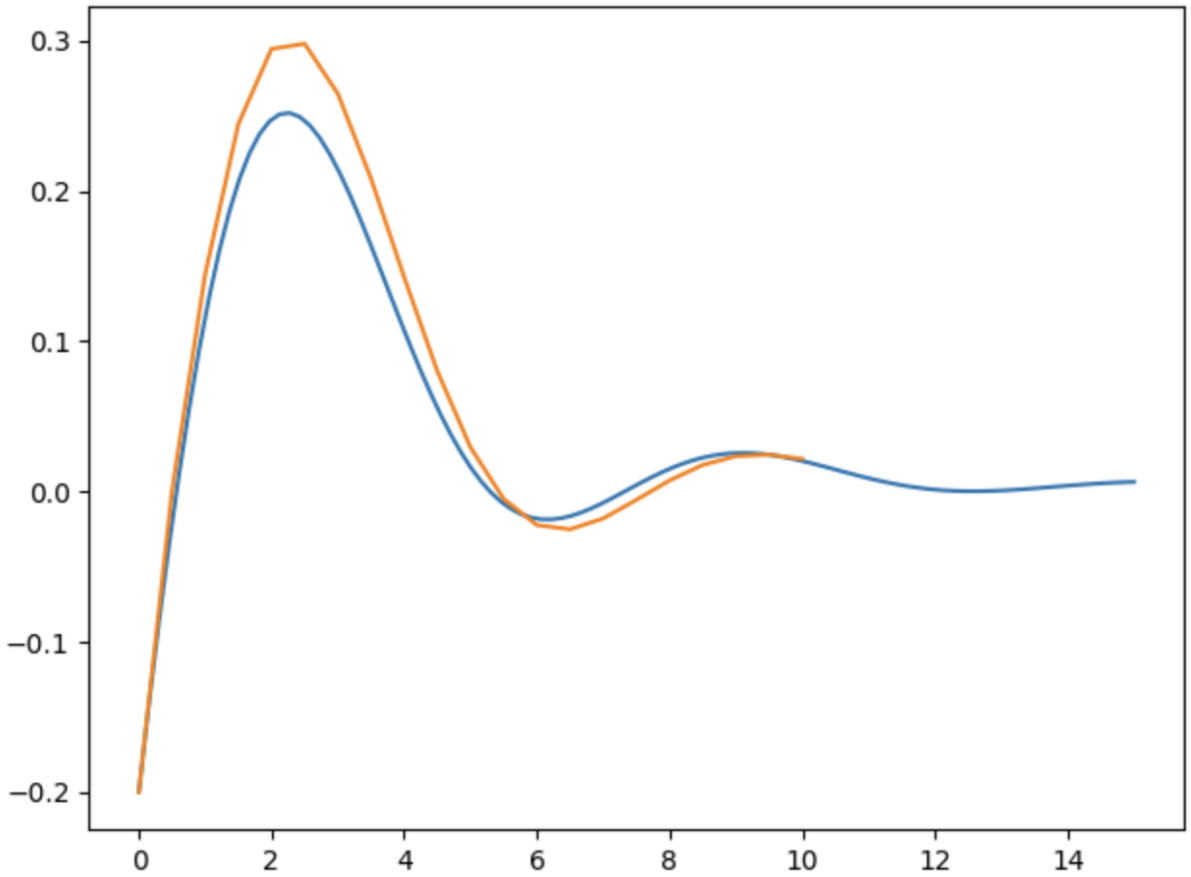
Le problème de Cauchy est

$$\begin{cases} y' = \frac{\sin(t)e^{-t/5}-2y}{t+1} = F(t,y) \\ y(0) = -1/5 \end{cases}$$

```
def F(t,y):
    return (np.sin(t)*np.exp(-t/5)-2*y)/(t+1)

def euler(f,a,b,y0,n):
    h=(b-a)/n
    y=np.zeros(n+1)
    y[0]=y0
    t = np.linspace(a, b, n+1)
    for k in range(n):
        y[k+1]=y[k]+h*f(t[k],y[k])
    return t,y
```

 # Appliquons euler sur l'intervalle $[0, 10]$ *avec* $y_0 = -\frac{1}{5}$ *et* $n = 20$*:*

```
X,Y = euler(F,0,10,-1/5,20)
x=np.linspace(0,15,100)
plt.plot(x,f(x),color='blue') # courbe exacte
plt.plot(X,Y,color='orange')    # solution approchée
plt.show()
```



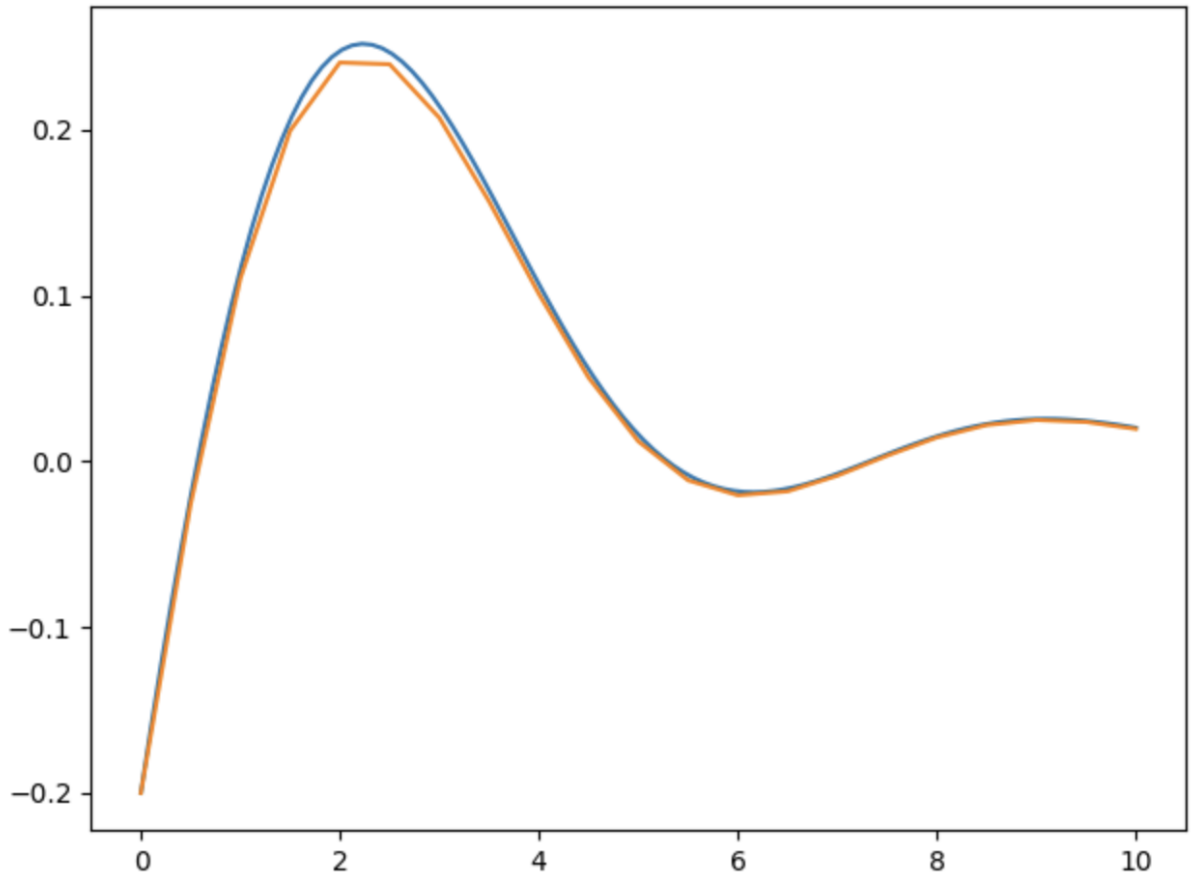
Exemple : Méthode d'Euler améliorée

Examinons le même problème avec la méthode d'Euler améliorée. C'est une méthode à un pas est définie par la fonction :

$$\Phi(t,z,h) = F\left(t+\frac{h}{2}, z+\frac{h}{2}F(t,z)\right)$$

```
def Phi(t,y,h):
    return F(t+h/2,y+h/2*F(t,y))

# Méthode d'Euler améliorée Φ(t,z,h) = F(t + h/2, z + h/2 F(t,z))
def eulerAmelioree(Phi,a,b,y0,n):
    h=(b-a)/n
    y=np.zeros(n+1)
    y[0]=y0
    t = np.linspace(a, b, n+1)
    for k in range(n):
        y[k+1]=y[k]+h*Phi(t[k],y[k],h)
    return t,y
```

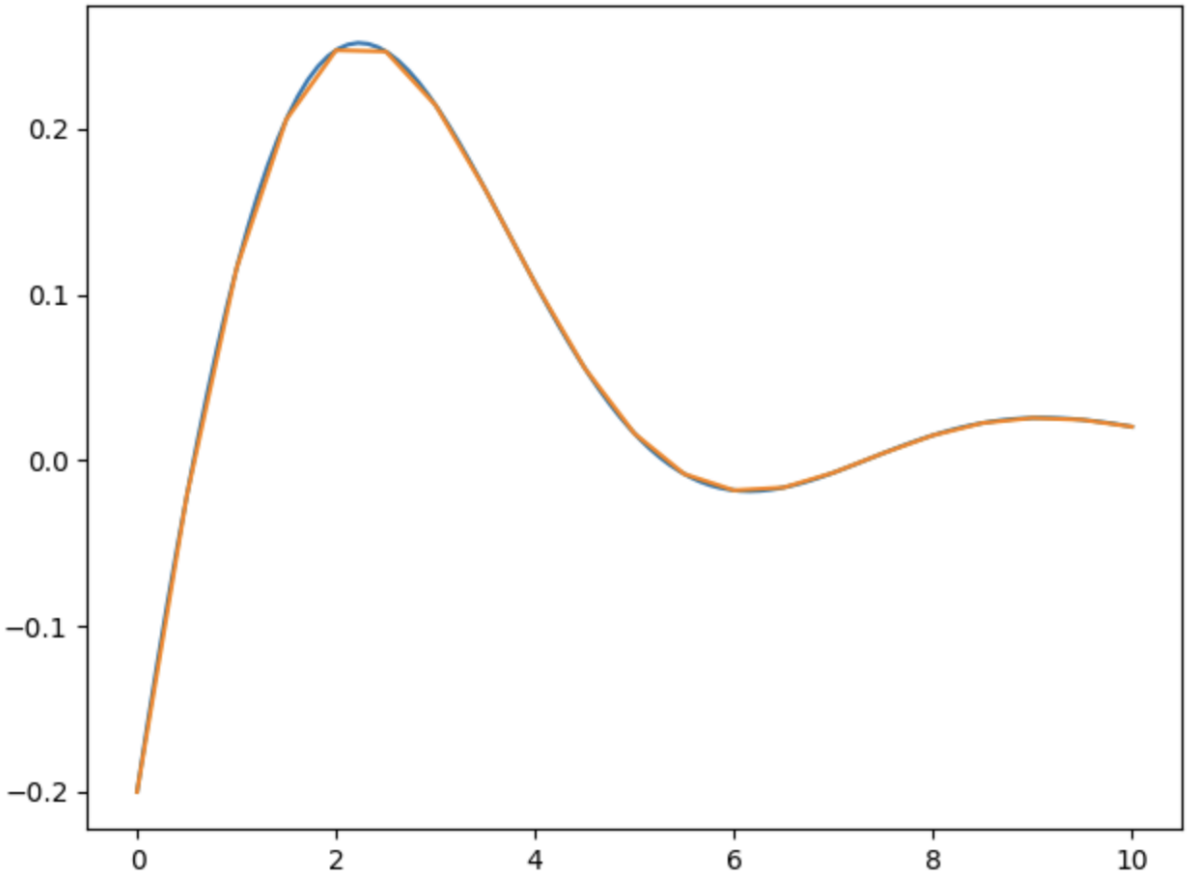
```
X,Y = eulerAmelioree(Phi,0,10,-1/5,20)
x=np.linspace(0,15,100)
y=f(x)
plt.plot(x,f(x),color='blue') # courbe exacte
plt.plot(X,Y,color='orange')    # solution approchée
plt.show()
```



La méthode de Runge-Kutta

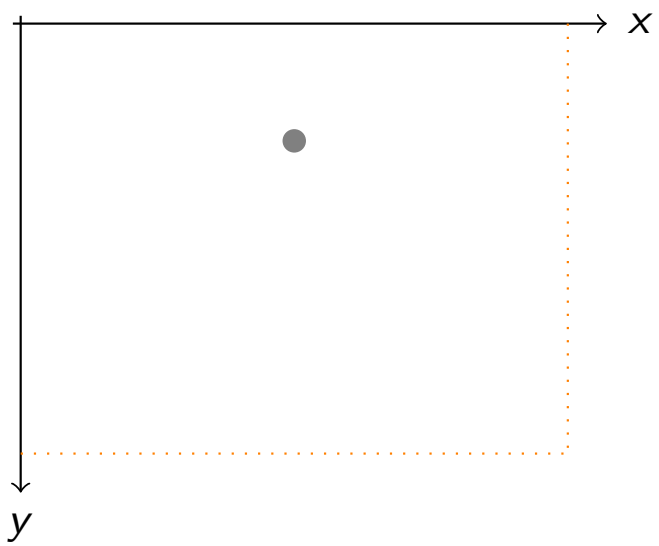
```
def rk4(Phi,a,b,y0,n):
    h=(b-a)/n
    y=np.zeros(n+1)
    y[0]=y0
    t = np.linspace(a, b, n+1)
    for k in range(n):
        K1=F(t[k],y[k])
        K2=F(t[k]+h/2,y[k]+h/2*K1)
        K3 = F(t[k] + h / 2, y[k] + h / 2 * K2)
        K4 = F(t[k] + h , y[k] + h * K3)
        y[k+1]=y[k]+h*(K1+2*K2+2*K3+K4)/6
    return t,y


X,Y = rk4(F,0,10,-1/5,20)
x=np.linspace(0,15,100)
plt.plot(x,f(x),color='blue') # courbe exacte
plt.plot(X,Y,color='orange')    # solution approchée
plt.show()
```

La méthode de Runge-Kutta



Exemple : Chute libre

On lance un projectile de masse m avec une vitesse initiale $\vec{v_0}$ verticale. L'axe vertical $(O; y)$ est orienté vers le bas.



L'équation différentielle du mouvement $\frac{\mathrm{d}^2 M}{\mathrm{d}t^2} = \vec{g}$ s'écrit :

$$y'' = g$$

Exemple : Chute libre

On pose $z_0 = y$, $z_1 = y'$. On a alors

$$\begin{cases} z_0' = z_1 \\ z_1' = g \end{cases}$$

On a donc un problème de Cauchy :

$$\begin{cases} Z' = f(t, Z) \\ Z_0 \text{ donné} \end{cases}$$

où $Z = [z_0, z_1]$, $f(t, Z) = [z_1, g]$ et $Z_0 = [y_0, -v_0]$
Le schéma d'Euler s'écrit :

$$\begin{cases} Z_{k+1} = Z_k + h\, f(t_k, Z_k) \\ Z_0 = [600, -100] \end{cases}$$

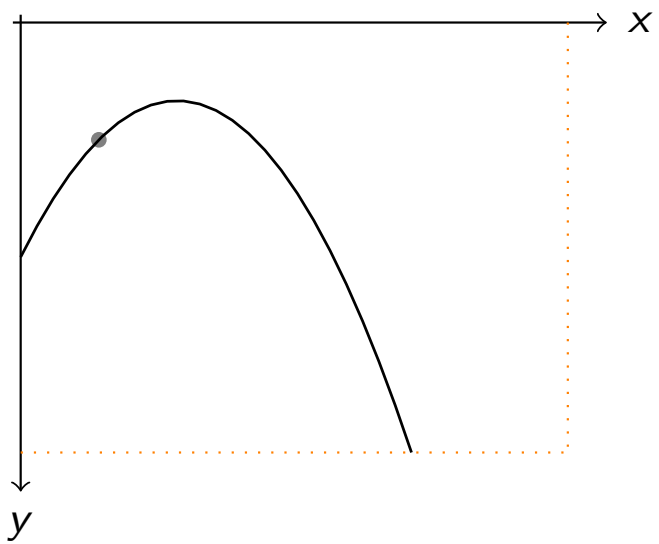
où $h = 0.1s$ et $g = 9.81m/s^2$.

```
from tkinter import *
import numpy as np
# coordonnees initiales (x0, y0)
x0, y0 = 10, 200
# vitesse initiale: angle de tir α et vitesse v0
alpha=np.pi/3 ; V0=50
h=0.1 # pas du temps
z=np.array([x0,y0,V0*np.cos(alpha),-V0*np.sin(alpha)])
def f(z):
    return np.array([z[2],z[3],0,9.81])

def Euler():
    global z
    z=z+h*f(z)
    # deplacement de la balle a la nouvelle position
    can1.coords(balle, z[0], z[1], z[0] + 30, z[1] + 30)
    # La fenetre fen1 est actualisee en executant la
    # fonction Euler toutes les 10 millisecondes
    fen1.after(10, Euler)
```

```
# ========== Programme principal =============
# Creation de la fenetre principale :
fen1 = Tk()
fen1.title("Chute libre")
# creation du canvas :
H=W=750
can1 = Canvas(fen1, bg="dark grey", height=H, width=W)
can1.pack()
# creation de la balle
balle = can1.create_oval(x0, y0, x0 + 30, y0 + 30, width=2, fill="red")
# Lancement de la fonction Euler
Euler()
# demarrage de la boucle principale:
fen1.mainloop()
```

Exemple : Problème de tir

On lance un projectile de masse m avec une vitesse initiale $\vec{v_0}$ faisant un angle α avec l'axe horizontale. Le plan de tir est porté par un système d'axes $(O; x, y)$.



L'équation différentielle du mouvement $\frac{\mathrm{d}^2 M}{\mathrm{d}t^2} = \vec{g}$ s'écrit en coordonnées x, y de M :

$$\begin{cases} x'' = 0 \\ y'' = g \end{cases}$$

Exemple : Problème de tir

On pose $z_0 = x$, $z_1 = y$, $z_2 = x'$,$z_3 = y'$. On a alors

$$\begin{cases} z_0' = z_2 \\ z_1' = z_3 \\ z_2' = 0 \\ z_3' = g \end{cases}$$

On a donc un problème de Cauchy :

$$\begin{cases} Z' = f(t, Z) \\ Z_0 \text{ donné} \end{cases}$$

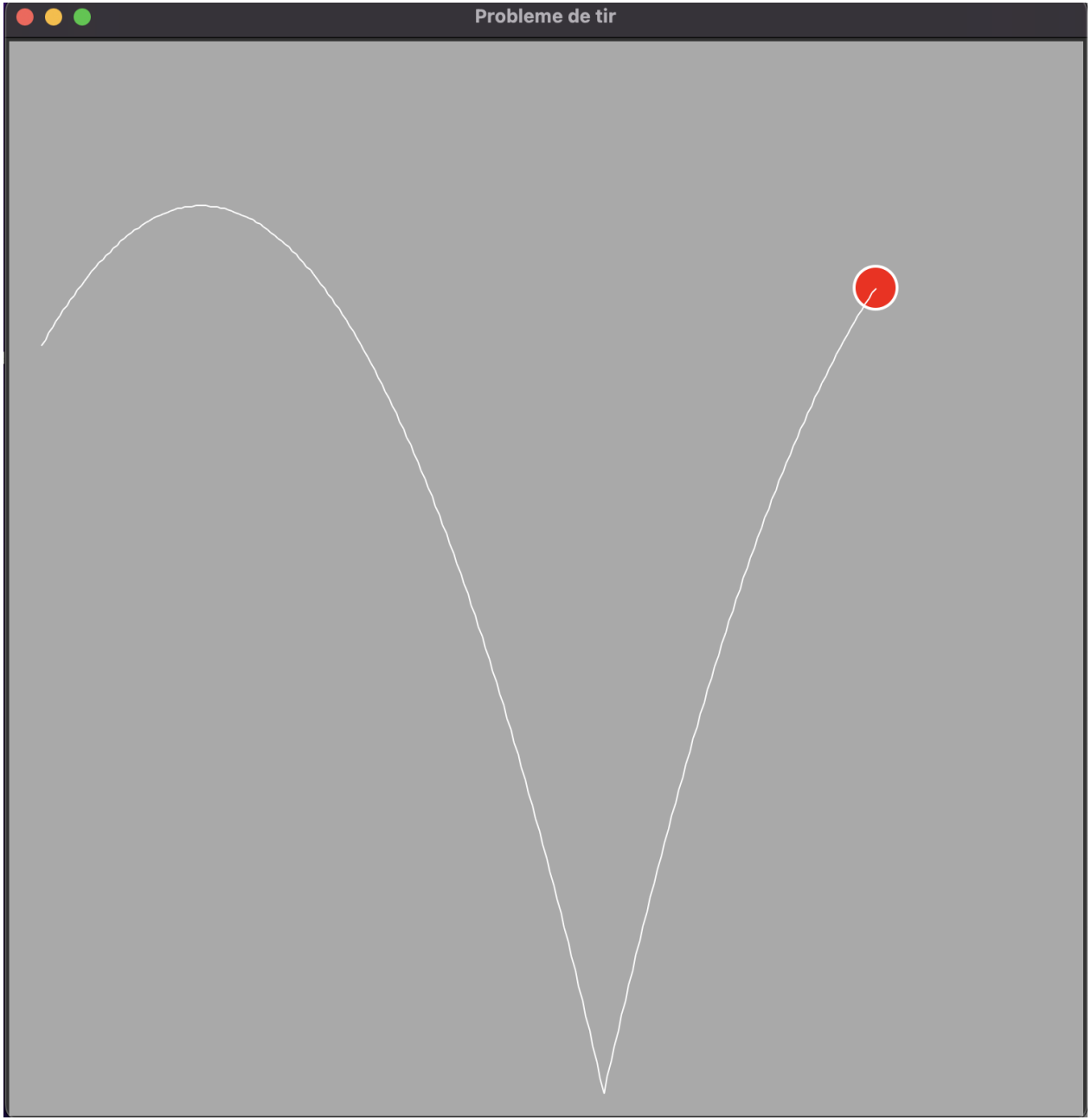
où $Z = [z_0, z_1, z_2, z_3]$, $f(t, Z) = [z_2, z_3, 0, g]$ et
$Z_0 = [x_0, y_0, v_0 \cos(\alpha), -v_0 \sin(\alpha)]$
Le schéma d'Euler s'écrit :

$$\begin{cases} Z_{k+1} = Z_k + h\, f(t_k, Z_k) \\ Z_0 = [10, 200, 50\cos(\pi/3), -50\sin(\pi/3)] \end{cases}$$

où $h = 0.1s$ et $g = 9.81m/s^2$.

```
from tkinter import *
import numpy as np
H=W=750
# coordonnées initiales
x0,y0=W/2,600
# vitesse initiale
V0=100
h=0.05
z=np.array([y0,-V0])
# équadiff: y'=f(t,y)
def f(y):
    return np.array([z[1],9.81])
def Euler():
    global z
    z = z + h * f(z)
    # position de la balle
    can1.coords(balle,x0,z[0],x0+30,z[0]+30)
    fen1.after(10,Euler)
```

```
#========== Programme principal ============
fen1 = Tk()
fen1.title("Problème de tir")
can1 =Canvas(fen1, bg='dark grey',height=H,width=W)
can1.pack()
balle = can1.create_oval(x0,y0,x0+30,y0+30,width=2,fill='red')
Euler()
fen1.mainloop()
```


Probleme de tir