Résolution numérique des équations différentielles

Jean-Pierre Becirspahic
Lycée Louis-Le-Grand

Présentation du problème

Nous nous intéressons aux équations différentielles de la forme :

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

Une solution est une fonction x de classe $\mathscr{C}^1$ définie sur I vérifiant :

$$\forall t \in I, \quad x'(t) = f(x(t),t).$$

Présentation du problème

Nous nous intéressons aux équations différentielles de la forme :

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

Une solution est une fonction x de classe $\mathscr{C}^1$ définie sur I vérifiant :

$$\forall t \in I, \quad x'(t) = f(x(t),t).$$

On cherche à résoudre un *problème de* CAUCHY :

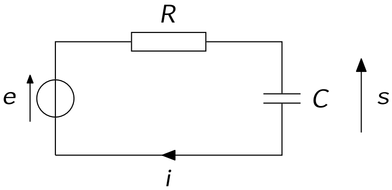
$$\begin{cases} x' = f(x,t) \\ x(t_0) = x_0 \end{cases}$$

Sous certaines conditions ce problème admet une unique solution que nous allons chercher à déterminer numériquement.

Présentation du problème

Exemple

Considérons le circuit électrique suivant :

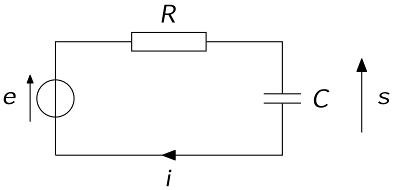


Il est régi par les relations $s = e - Ri$ et $i = C\dfrac{\mathrm{d}s}{\mathrm{d}t}$ qui conduisent à l'équation différentielle : $RC\dfrac{\mathrm{d}s}{\mathrm{d}t} + s = e$.

Présentation du problème

Exemple

Considérons le circuit électrique suivant :



Il est régi par les relations $s = e - Ri$ et $i = C\dfrac{\mathrm{d}s}{\mathrm{d}t}$ qui conduisent à l'équation différentielle : $RC\dfrac{\mathrm{d}s}{\mathrm{d}t} + s = e$.

En posant $\tau = RC$ on est amené à résoudre le problème de CAUCHY :

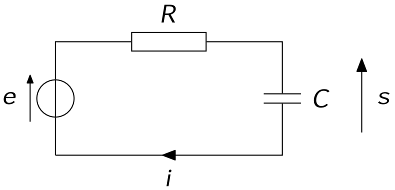
$$\begin{cases} s' = \dfrac{1}{\tau}(e(t) - s) \\ s(0) = \dfrac{q_0}{C} \end{cases}$$

où q_0 désigne la charge initiale du condensateur.

Présentation du problème

Exemple

Considérons le circuit électrique suivant :

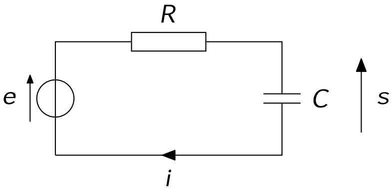


- lorsque $e(t) = 0$ on obtient : $s(t) = \dfrac{q_0}{C}\,\mathrm{e}^{-t/\tau}$.

Présentation du problème

Exemple

Considérons le circuit électrique suivant :



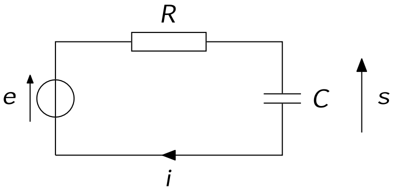
- lorsque $e(t) = 0$ on obtient : $s(t) = \dfrac{q_0}{C}\,\mathrm{e}^{-t/\tau}$.
- lorsque $e(t) = E\cos(\omega t)$ on obtient :

$$s(t) = \mathrm{e}^{-t/\tau}\left(\frac{q_0}{C} - \frac{E}{1+\omega^2\tau^2}\right) + \frac{E}{1+\omega^2\tau^2}\Big(\cos(\omega t) + \omega\tau\sin(\omega t)\Big)$$

Présentation du problème

Exemple

Considérons le circuit électrique suivant :



- lorsque $e(t) = 0$ on obtient : $s(t) = \dfrac{q_0}{C}\,\mathrm{e}^{-t/\tau}$.
- lorsque $e(t) = E\cos(\omega t)$ on obtient :

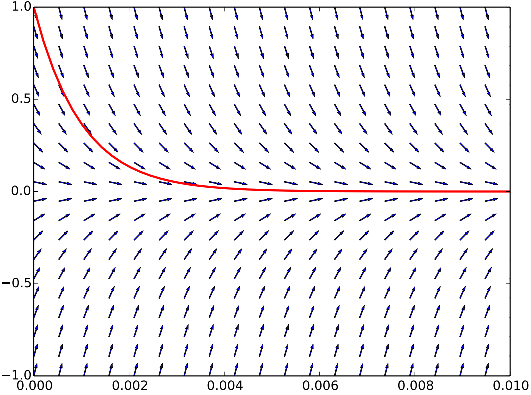
$$s(t) = \mathrm{e}^{-t/\tau}\Big(\frac{q_0}{C} - \frac{E}{1+\omega^2\tau^2}\Big) + \frac{E}{1+\omega^2\tau^2}\Big(\cos(\omega t) + \omega\tau\sin(\omega t)\Big)$$

- dans d'autres cas la solution mathématique n'est pas toujours satisfaisante $\longrightarrow$ on se tourne vers les méthodes de résolution numérique.

Champ de vecteurs

On associe à l'équation différentielle $x' = f(x,t)$ le champ de vecteurs défini par : $\vec{v}(x,t) = \vec{\imath} + f(x,t)\vec{\jmath}$.

Exemple : champ de vecteurs associé à l'équation différentielle $s' = -s/\tau$.



Chaque vecteur du champ est tangent à la solution qui passe en ce point.

Méthode d'Euler explicite

On subdivise l'intervalle $[t_0, t_0 + T]$ en $n+1$ points $t_0 < t_1 < \cdots < t_n = t_0 + T$ et on approche la relation :

$$x(t_{k+1}) - x(t_k) = \int_{t_k}^{t_{k+1}} x'(t)\,\mathrm{d}t = \int_{t_k}^{t_{k+1}} f(x(t), t)\,\mathrm{d}t.$$

Méthode d'Euler explicite

On subdivise l'intervalle $[t_0, t_0 + T]$ en $n+1$ points $t_0 < t_1 < \cdots < t_n = t_0 + T$ et on approche la relation :

$$x(t_{k+1}) - x(t_k) = \int_{t_k}^{t_{k+1}} x'(t)\,\mathrm{d}t = \int_{t_k}^{t_{k+1}} f(x(t), t)\,\mathrm{d}t.$$

La méthode d'Euler explicite approche cette intégrale par la méthode du rectangle gauche : $\displaystyle\int_{t_k}^{t_{k+1}} f(x(t), t)\,\mathrm{d}t \approx (t_{k+1} - t_k) f(x(t_k), t_k)$.

Méthode d'EULER explicite

On subdivise l'intervalle $[t_0, t_0+T]$ en $n+1$ points $t_0 < t_1 < \cdots < t_n = t_0+T$ et on approche la relation :

$$x(t_{k+1}) - x(t_k) = \int_{t_k}^{t_{k+1}} x'(t)\,\mathrm{d}t = \int_{t_k}^{t_{k+1}} f(x(t),t)\,\mathrm{d}t.$$

La méthode d'EULER explicite approche cette intégrale par la méthode du rectangle gauche : $\displaystyle\int_{t_k}^{t_{k+1}} f(x(t),t)\,\mathrm{d}t \approx (t_{k+1}-t_k)f(x(t_k),t_k)$.

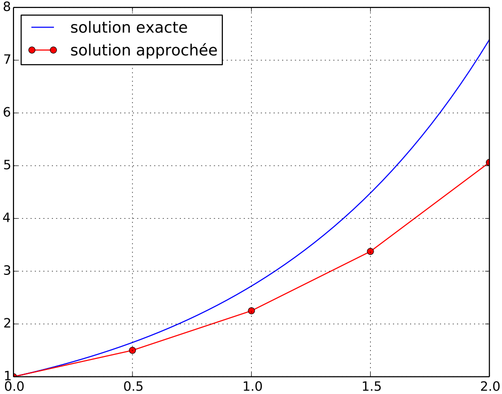
En posant $h_k = t_{k+1} - t_k$, on définit une suite de valeurs $x_0, x_1, \ldots, x_n$ à partir de la condition initiale x_0 et de la relation de récurrence :

$$\forall k \in [\![0, n-1]\!], \quad x_{k+1} = x_k + h_k f(x_k, t_k)$$

Méthode d'Euler explicite

$$x_{k+1} = x_k + h_k f(x_k, t_k)$$

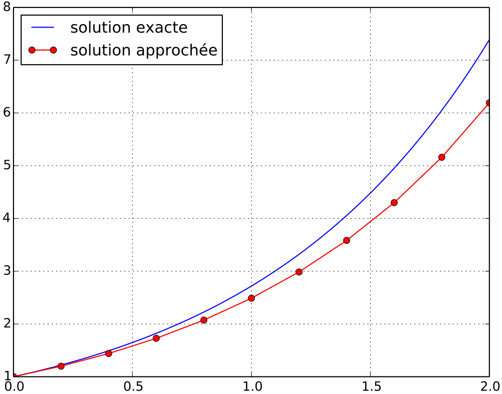
Seul le premier point x_0 est une valeur exacte : les autres sont calculés à partir de l'approximation précédente, ce qui peut conduire à accumuler des erreurs.



Méthode d'EULER explicite

$$x_{k+1} = x_k + h_k f(x_k, t_k)$$

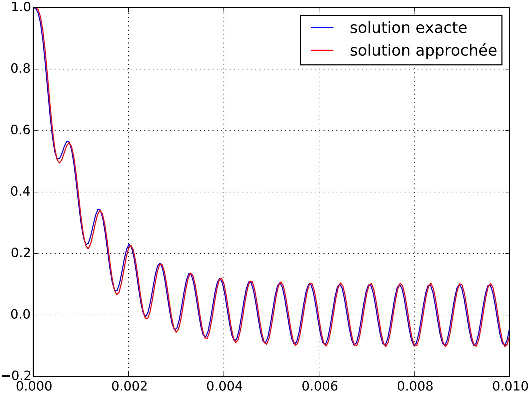
Lorsque la subdivision s'affine la précision en général s'améliore (sans pour autant faire disparaître le phénomène de divergence).



Méthode d'EULER explicite

$$x_{k+1} = x_k + h_k f(x_k, t_k)$$

Dans de nombreux cas la méthode d'EULER procure des résultats acceptables (au moins qualitativement).



Méthode d'Euler explicite

$$x_{k+1} = x_k + h_k f(x_k, t_k)$$

Dans de nombreux cas la méthode d'Euler procure des résultats acceptables (au moins qualitativement).

Les exemples présentés utilisent une subdivision de pas régulier ; cependant de meilleurs résultats sont obtenus en adaptant le pas à la fonction f (méthode à pas adaptatif) : si $f(x_k, t_k)$ est faible alors x varie peu et on peut utiliser un pas plus grand. On contraire, on réduira le pas lorsque la valeur de $f(x_k, t_k)$ augmente.

Méthode d'Euler implicite

Elle consiste à approcher l'intégrale $\int_{t_k}^{t_{k+1}} f(x(t),t)\,\mathrm{d}t$ par la méthode du rectangle droit, ce qui conduit à définir la suite $(x_0, x_1, \dots, x_n)$ par :

$$x_{k+1} = x_k + h_k f(x_{k+1}, t_{k+1}).$$

Il est nécessaire de coupler cette méthode à une méthode de résolution numérique des équations.

Méthode d'EULER implicite

Elle consiste à approcher l'intégrale $\int_{t_k}^{t_{k+1}} f(x(t),t)\,\mathrm{d}t$ par la méthode du rectangle droit, ce qui conduit à définir la suite $(x_0, x_1, \dots, x_n)$ par :

$$x_{k+1} = x_k + h_k f(x_{k+1}, t_{k+1}).$$

Il est nécessaire de coupler cette méthode à une méthode de résolution numérique des équations.

Dans la pratique, la méthode d'EULER implicite se révèle souvent plus stable que la méthode explicite : elle est moins précise à court terme, mais diverge moins rapidement de la solution exacte que la méthode explicite.

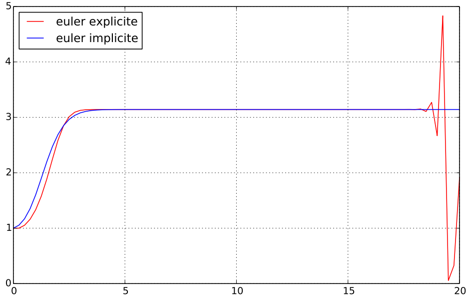
Méthode d'EULER implicite

Elle consiste à approcher l'intégrale $\int_{t_k}^{t_{k+1}} f(x(t),t)\,\mathrm{d}t$ par la méthode du rectangle droit, ce qui conduit à définir la suite $(x_0, x_1, \ldots, x_n)$ par :

$$x_{k+1} = x_k + h_k f(x_{k+1}, t_{k+1}).$$

Il est nécessaire de coupler cette méthode à une méthode de résolution numérique des équations.

Exemple : on résout $x' = t\sin(x)$ avec $x(0) = 1$.



Méthode d'Euler implicite

Elle consiste à approcher l'intégrale $\int_{t_k}^{t_{k+1}} f(x(t),t)\,\mathrm{d}t$ par la méthode du rectangle droit, ce qui conduit à définir la suite $(x_0, x_1, \dots, x_n)$ par :

$$x_{k+1} = x_k + h_k f(x_{k+1}, t_{k+1}).$$

Il est nécessaire de coupler cette méthode à une méthode de résolution numérique des équations.
Exemple : on résout $x' = t\sin(x)$ avec $x(0) = 1$.

Euler explicite : $x_{k+1} = x_k + ht_k \sin(x_k)$.

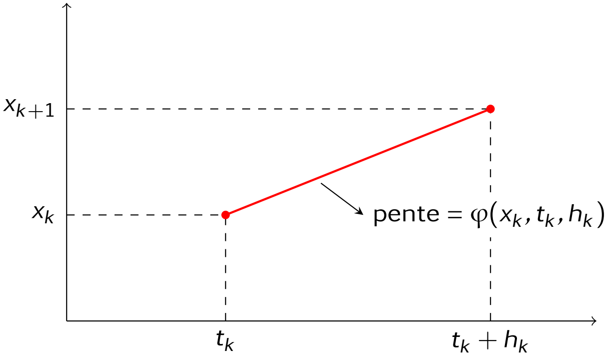
$\sin(x_k) \approx 0$ et $t_k \to +\infty$. Progressivement t_k prend le pas sur $\sin(x_k)$. Dès lors, le comportement de la suite x devient chaotique.

Schéma numérique

On appelle schéma numérique explicite à un pas toute équation de récurrence de la forme :

$$\forall k \in [\![0, n-1]\!], \quad x_{k+1} = x_k + h_k \varphi(x_k, t_k, h_k)$$

$\varphi(x, t, h)$ est la valeur qui approche la dérivée entre t et $t + h$:



Exemple : la méthode d'EULER correspond au schéma $\varphi(x, t, h) = f(x, t)$.

Schéma numérique

Autre schémas

Un schéma numérique implicite à un pas prend la forme :

$$x_{k+1} = x_k + h_k \varphi(x_k, x_{k+1}, t_k, h_k)$$

c'est le cas par exemple de la méthode d'Euler implicite.

Un schéma numérique explicite à p pas prend la forme :

$$x_{k+1} = x_k + h_k \varphi(x_k, x_{k-1}, \dots, x_{k-p+1}, t_k, h_k)$$

Le calcul de x_k n'est possible qu'à partir de l'indice p et la méthode doit être complétée par un calcul initial des p premières valeurs, par exemple par une méthode à un pas.

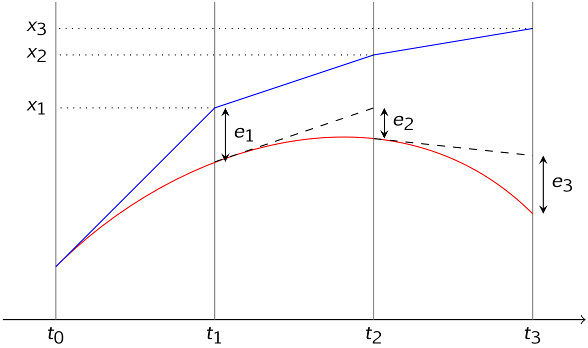
Aucun de ces deux schémas ne sera abordé dans la suite de ce cours.

Schéma numérique

Erreur de consistance

L'erreur de consistance est définie par :

$$e_{k+1} = x(t_{k+1}) - x(t_k) - h_k \varphi(x(t_k), t_k, h_k).$$

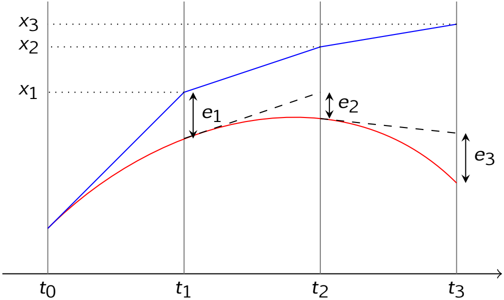


Un schéma est dit d'ordre p lorsque $e_{k+1} = \mathrm{O}(h_k^{p+1})$.

Une méthode numérique est dite consistante lorsque : $\lim\limits_{n \to +\infty} \sum\limits_{k=1}^{n} e_k = 0$.

Schéma numérique

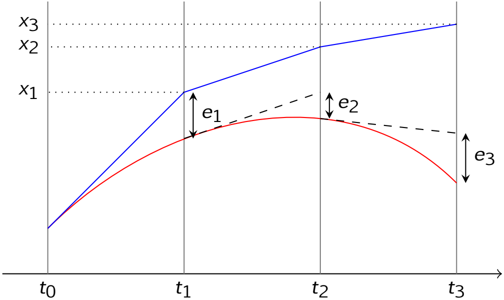
Erreur de consistance



À la date t_1, $x(t_1) = x_1 + e_1$.

Schéma numérique

Erreur de consistance

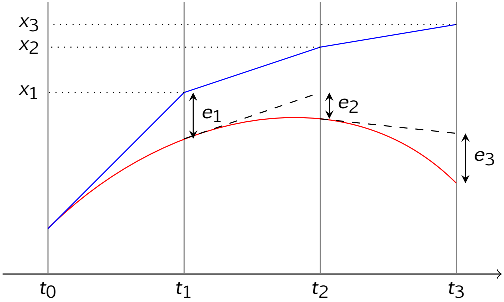


À la date t_1, $x(t_1) = x_1 + e_1$.

À la date t_2, $x(t_2) = x_2 + (e_1 + e_2) + h_1\Big(\varphi(x(t_1), t_1, h_1) - \varphi(x_1, t_1, h_1)\Big)$

Schéma numérique

Erreur de consistance



À la date t_1, $x(t_1) = x_1 + e_1$.

À la date t_2, $x(t_2) = x_2 + (e_1 + e_2) + h_1\Big(\varphi(x(t_1), t_1, h_1) - \varphi(x_1, t_1, h_1)\Big)$

À la date t_3,

$$x(t_3) = x_3 + (e_1 + e_2 + e_3) + h_1\Big(\varphi(x(t_1), t_1, h_1) - \varphi(x_1, t_1, h_1)\Big) + h_2\Big(\varphi(x(t_2), t_2, h_2) - \varphi(x_2, t_2, h_2)\Big)$$

Schéma numérique

Erreur de consistance

Lorsque f est de classe $\mathscr{C}^1$ la méthode d'EULER est une méthode consistante d'ordre 1.

Schéma numérique

Erreur de consistance

Lorsque f est de classe $\mathscr{C}^1$ la méthode d'EULER est une méthode consistante d'ordre 1.

Si f est $\mathscr{C}^1$ alors x est $\mathscr{C}^2$ et :

$$x(t_{k+1}) = x(t_k + h_k) = x(t_k) + h_k x'(t_k) + \mathrm{O}(h_k^2) = x(t_k) + h_k f(x(t_k), t_k) + \mathrm{O}(h_k^2)$$

Schéma numérique

Erreur de consistance

Lorsque f est de classe $\mathscr{C}^1$ la méthode d'EULER est une méthode consistante d'ordre 1.

Si f est $\mathscr{C}^1$ alors x est $\mathscr{C}^2$ et :

$$x(t_{k+1}) = x(t_k + h_k) = x(t_k) + h_k x'(t_k) + \mathrm{O}(h_k^2) = x(t_k) + h_k f(x(t_k), t_k) + \mathrm{O}(h_k^2)$$

Ainsi, $x(t_{k+1}) - x(t_k) - h_k f(x(t_k), t_k) = \mathrm{O}(h_k^2)$, soit $e_{k+1} = \mathrm{O}(h_k^2)$. La méthode d'EULER est d'ordre 1.

Schéma numérique

Erreur de consistance

Lorsque f est de classe $\mathscr{C}^1$ la méthode d'EULER est une méthode consistante d'ordre 1.

Si f est $\mathscr{C}^1$ alors x est $\mathscr{C}^2$ et :

$$x(t_{k+1}) = x(t_k + h_k) = x(t_k) + h_k x'(t_k) + \mathrm{O}(h_k^2) = x(t_k) + h_k f(x(t_k), t_k) + \mathrm{O}(h_k^2)$$

Ainsi, $x(t_{k+1}) - x(t_k) - h_k f(x(t_k), t_k) = \mathrm{O}(h_k^2)$, soit $e_{k+1} = \mathrm{O}(h_k^2)$. La méthode d'EULER est d'ordre 1.
On utilise maintenant l'égalité de TAYLOR-LAGRANGE : il existe $\tau_k \in]t_k, t_{k+1}[$ tq :

$$x(t_{k+1}) = x(t_k) + h_k x'(t_k) + \frac{h_k^2}{2} x''(\tau_k) = x(t_k) + h_k f(x(t_k), t_k) + \frac{h_k^2}{2} x''(\tau_k)$$

Schéma numérique

Erreur de consistance

Lorsque f est de classe $\mathscr{C}^1$ la méthode d'EULER est une méthode consistante d'ordre 1.

Si f est $\mathscr{C}^1$ alors x est $\mathscr{C}^2$ et :

$$x(t_{k+1}) = x(t_k + h_k) = x(t_k) + h_k x'(t_k) + \mathrm{O}(h_k^2) = x(t_k) + h_k f(x(t_k), t_k) + \mathrm{O}(h_k^2)$$

Ainsi, $x(t_{k+1}) - x(t_k) - h_k f(x(t_k), t_k) = \mathrm{O}(h_k^2)$, soit $e_{k+1} = \mathrm{O}(h_k^2)$. La méthode d'EULER est d'ordre 1.
On utilise maintenant l'égalité de TAYLOR-LAGRANGE : il existe $\tau_k \in]t_k, t_{k+1}[$ tq :

$$x(t_{k+1}) = x(t_k) + h_k x'(t_k) + \frac{h_k^2}{2} x''(\tau_k) = x(t_k) + h_k f(x(t_k), t_k) + \frac{h_k^2}{2} x''(\tau_k)$$

Autrement dit, $|e_{k+1}| = \dfrac{h_k^2}{2} |x''(\tau_k)| \leqslant \dfrac{M_2}{2} h_k^2$.

Schéma numérique

Erreur de consistance

Lorsque f est de classe $\mathscr{C}^1$ la méthode d'EULER est une méthode consistante d'ordre 1.

Si f est $\mathscr{C}^1$ alors x est $\mathscr{C}^2$ et :

$$x(t_{k+1}) = x(t_k + h_k) = x(t_k) + h_k x'(t_k) + \mathrm{O}(h_k^2) = x(t_k) + h_k f(x(t_k), t_k) + \mathrm{O}(h_k^2)$$

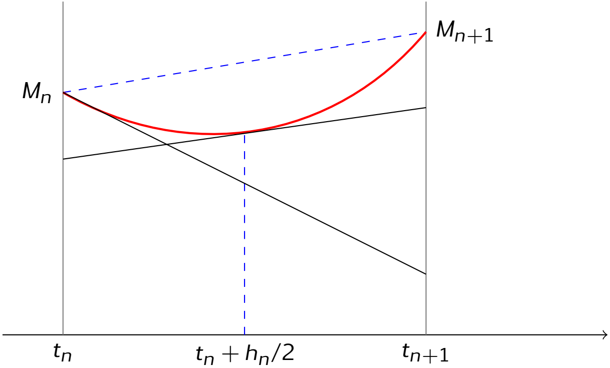
Ainsi, $x(t_{k+1}) - x(t_k) - h_k f(x(t_k), t_k) = \mathrm{O}(h_k^2)$, soit $e_{k+1} = \mathrm{O}(h_k^2)$. La méthode d'EULER est d'ordre 1.
On utilise maintenant l'égalité de TAYLOR-LAGRANGE : il existe $\tau_k \in]t_k, t_{k+1}[$ tq :

$$x(t_{k+1}) = x(t_k) + h_k x'(t_k) + \frac{h_k^2}{2} x''(\tau_k) = x(t_k) + h_k f(x(t_k), t_k) + \frac{h_k^2}{2} x''(\tau_k)$$

Autrement dit, $|e_{k+1}| = \dfrac{h_k^2}{2} |x''(\tau_k)| \leqslant \dfrac{M_2}{2} h_k^2$.
Lorsque $h_k = \dfrac{T}{n}$, $|e_{k+1}| \leqslant \dfrac{M_2 T^2}{2n^2}$ et : $\left| \displaystyle\sum_{k=1}^{n} e_k \right| \leqslant \displaystyle\sum_{k=1}^{n} |e_k| \leqslant \dfrac{M_2 T}{2n} \underset{n \to +\infty}{\longrightarrow} 0.$

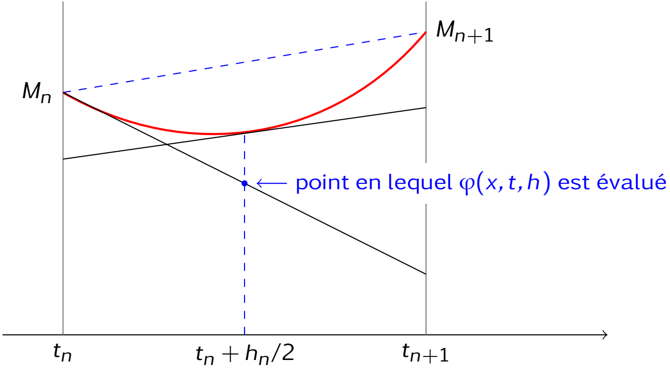
Méthode du point milieu

En général, la pente de $[M_n, M_{n+1}]$ est plus proche de $x'(t_n + h_n/2)$ (pente de la tangente au point milieu) que de $x'(t_n)$.



Méthode du point milieu

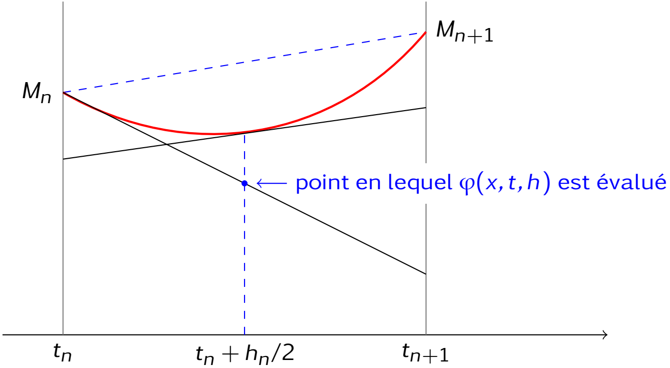
En général, la pente de $[M_n, M_{n+1}]$ est plus proche de $x'(t_n + h_n/2)$ (pente de la tangente au point milieu) que de $x'(t_n)$.



La méthode du point milieu consiste à évaluer la dérivée non pas en (t_n, x_n) (Euler) mais en $\Big(t_n + \dfrac{h_n}{2}, x_n + \dfrac{h_n}{2} f(x_n, t_n)\Big)$.

Méthode du point milieu

En général, la pente de $[M_n, M_{n+1}]$ est plus proche de $x'(t_n + h_n/2)$ (pente de la tangente au point milieu) que de $x'(t_n)$.

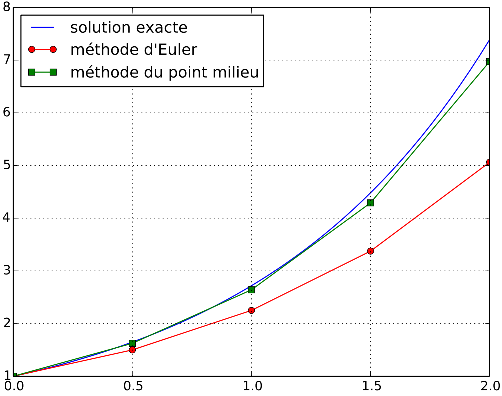


On aboutit aux relations :

$$\boxed{x_{n+1/2} = x_n + \frac{h_n}{2} f(x_n, t_n) \qquad \text{puis} \qquad x_{n+1} = x_n + h_n f(x_{n+1/2}, t_n + h_n/2)}$$

Méthode du point milieu

Nous admettrons que sous des hypothèses suffisantes la méthode du point milieu est une méthode consistante d'ordre 2.



Résolution de $x' = x$, $x(0) = 1$ sur $[0, 2]$ avec $n = 4$.

Méthodes de Runge-Kutta

Une méthode RK_s est définie par le schéma :

$$
\begin{aligned}
k_1 &= f(x,t) \\
k_2 &= f(x+(a_{21}k_1)h, t+c_2h) \\
&\vdots \\
k_s &= f\Big(x+\Big(\sum_{i=1}^{s-1} a_{si}k_i\Big)h, t+c_sh\Big)
\end{aligned}
\qquad
\varphi(x,t,h) = \sum_{i=1}^{s} b_i k_i
$$

où c_i, a_{ij} et b_i sont des constantes qui définissent précisément le schéma.

On supposera $c_1 = 0$, $c_i = \sum_{j=1}^{i-1} a_{ij}$ et $\sum_{i=1}^{s} b_i = 1$.

Méthodes de Runge-Kutta

Une méthode RK_s est définie par le schéma :

$$\begin{aligned} k_1 &= f(x,t) \\ k_2 &= f(x+(a_{21}k_1)h, t+c_2h) \\ &\vdots \\ k_s &= f\Big(x+\Big(\sum_{i=1}^{s-1} a_{si}k_i\Big)h, t+c_sh\Big) \end{aligned} \qquad \varphi(x,t,h) = \sum_{i=1}^{s} b_i k_i$$

où c_i, a_{ij} et b_i sont des constantes qui définissent précisément le schéma.

On supposera $c_1 = 0$, $c_i = \sum_{j=1}^{i-1} a_{ij}$ et $\sum_{i=1}^{s} b_i = 1$.

La méthode de du point milieu est de type RK_2 car définie par :

$$\begin{aligned} k_1 &= f(x,t) \\ k_2 &= f(x+hk_1/2, t+h/2) \end{aligned} \qquad \varphi(x,t,h) = k_2$$

Méthodes de Runge-Kutta

Une méthode RK_s est définie par le schéma :

$$
\begin{aligned}
k_1 &= f(x,t) \\
k_2 &= f(x+(a_{21}k_1)h, t+c_2h) \\
&\vdots \\
k_s &= f\Big(x+\Big(\sum_{i=1}^{s-1} a_{si}k_i\Big)h, t+c_sh\Big)
\end{aligned}
\qquad
\varphi(x,t,h) = \sum_{i=1}^{s} b_i k_i
$$

On représente une méthode RK_s à l'aide de son tableau de BUTCHER :

$$
\begin{array}{c|ccccc}
c_1 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
$$

Méthodes de Runge-Kutta

En général une méthode RK_s est d'ordre s.

Méthodes de Runge-Kutta

En général une méthode RK_s est d'ordre s.

Méthode d'Euler (ordre 1) :

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

$$\varphi(x,t,h) = f(x,t)$$

Méthodes de Runge-Kutta

En général une méthode RK_s est d'ordre s.

Méthode du point milieu (ordre 2) :

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

$$\begin{aligned} k_1 &= f(x,t) \\ k_2 &= f(x + hk_1/2, t + h/2) \\ \varphi(x,t,h) &= k_2 \end{aligned}$$

Méthodes de Runge-Kutta

En général une méthode RK_s est d'ordre s.

Méthode de HEUN (ordre 2) :

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

$$\begin{aligned} k_1 &= f(x,t) \\ k_2 &= f(x+hk_1, t+h) \\ \varphi(x,t) &= k_1/2 + k_2/2 \end{aligned}$$

Méthodes de Runge-Kutta

En général une méthode RK_s est d'ordre s.

Méthode RK_4 classique (ordre 4) :

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

Méthodes de Runge-Kutta

En général une méthode RK_s est d'ordre s.

Méthode RK_4 classique (ordre 4) :

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

$$\begin{aligned}
k_1 &= f(x,t) \\
k_2 &= f(x + hk_1/2, t + h/2) \\
k_3 &= f(x + hk_2/2, t + h/2) \\
k_4 &= f(x + hk_3, t + h) \\
\varphi(x,t) &= k_1/6 + k_2/3 + k_3/3 + k_4/6
\end{aligned}$$

Systèmes différentiels

On considère un système de p équations différentielles liant p fonctions numériques $x_1, \dots, x_p$:

$$\begin{cases} x_1' = f_1(x_1, x_2, \dots, x_n, t) \\ x_2' = f_2(x_1, x_2, \dots, x_n, t) \\ \cdots = \cdots\cdots\cdots\cdots\cdots \\ x_p' = f_p(x_1, x_2, \dots, x_n, t) \end{cases}$$

Systèmes différentiels

On considère un système de p équations différentielles liant p fonctions numériques $x_1, \dots, x_p$:

$$\begin{cases} x_1' = f_1(x_1, x_2, \dots, x_n, t) \\ x_2' = f_2(x_1, x_2, \dots, x_n, t) \\ \cdots = \cdots\cdots\cdots\cdots\cdots \\ x_p' = f_p(x_1, x_2, \dots, x_n, t) \end{cases}$$

Ces p fonctions numériques constituent les p composantes de la fonction vectorielle X à valeurs dans $\mathbb{R}^p$ et définie par : $X = (x_1, x_2, \dots, x_p)$.

Systèmes différentiels

On considère un système de p équations différentielles liant p fonctions numériques $x_1, \dots, x_p$:

$$\begin{cases} x_1' = f_1(x_1, x_2, \dots, x_n, t) \\ x_2' = f_2(x_1, x_2, \dots, x_n, t) \\ \dots = \dots\dots\dots\dots\dots\dots \\ x_p' = f_p(x_1, x_2, \dots, x_n, t) \end{cases}$$

Ces p fonctions numériques constituent les p composantes de la fonction vectorielle X à valeurs dans $\mathbb{R}^p$ et définie par : $X = (x_1, x_2, \dots, x_p)$.
On considère la fonction vectorielle F dont les fonctions $f_1, \dots, f_p$ sont les composantes :

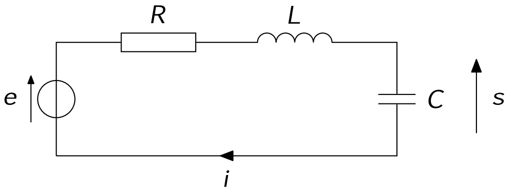
$$F(x_1, \dots, x_p, t) = \Big(f_1(x_1, \dots, x_p, t), f_2(x_1, \dots, x_p, t), \dots, f_p(x_1, \dots, x_p, t)\Big).$$

La fonction vectorielle X vérifie l'équation différentielle suivante :

$$X' = F(X, t)$$

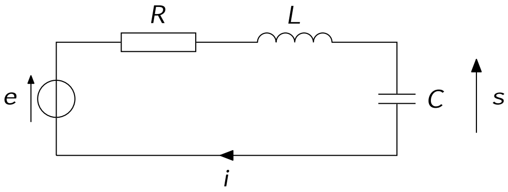
Systèmes différentiels

Exemple



Systèmes différentiels

Exemple

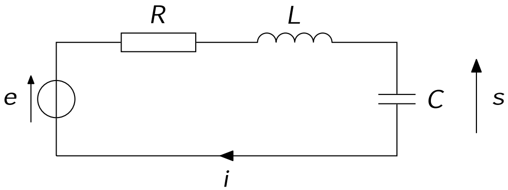


Il est régi par les relations $s = e - Ri - L\dfrac{\mathrm{d}i}{\mathrm{d}t}$ et $i = C\dfrac{\mathrm{d}s}{\mathrm{d}t}$ qui conduisent au système différentiel :

$$\begin{cases} \dfrac{\mathrm{d}i}{\mathrm{d}t} = \dfrac{1}{L}(-Ri - s + e) \\ \dfrac{\mathrm{d}s}{\mathrm{d}t} = \dfrac{i}{C} \end{cases}$$

Systèmes différentiels

Exemple



Il est régi par les relations $s = e - Ri - L\dfrac{\mathrm{d}i}{\mathrm{d}t}$ et $i = C\dfrac{\mathrm{d}s}{\mathrm{d}t}$ qui conduisent au système différentiel :

$$\begin{cases} \dfrac{\mathrm{d}i}{\mathrm{d}t} = \dfrac{1}{L}(-Ri - s + e) \\ \dfrac{\mathrm{d}s}{\mathrm{d}t} = \dfrac{i}{C} \end{cases}$$

Le vecteur $X = (i, s)$ est solution de l'équation différentielle $X' = F(X, t)$ avec $F\big((i,s),t\big) = \left(\dfrac{1}{L}\big(-Ri - s + e(t)\big), \dfrac{i}{C}\right)$.

Systèmes différentiels

Exemple

$$\begin{cases} \dfrac{\mathrm{d}i}{\mathrm{d}t} = \dfrac{1}{L}(-Ri - s + e) \\ \dfrac{\mathrm{d}s}{\mathrm{d}t} = \dfrac{i}{C} \end{cases}$$

Dans le cas vectoriel, la méthode d'EULER calcule la suite de vecteurs $(X_k)_{1 \leqslant k \leqslant n}$ à l'aide des formules :

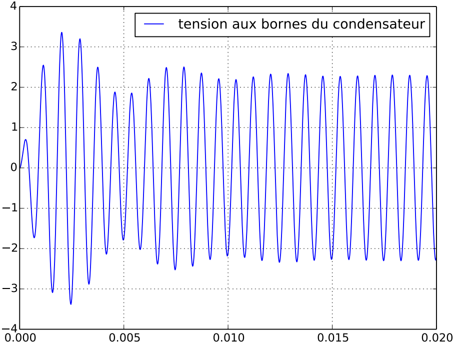
$$X_{k+1} = X_k + h_k F(X_k, t_k) \qquad \text{avec} \quad h_k = t_{k+1} - t_k$$

$$\begin{cases} i_{k+1} = i_k + \dfrac{h_k}{L}\Big(-Ri_k - s_k + e(t_k)\Big) \\ s_{k+1} = s_k + h_k \dfrac{i_k}{C} \end{cases}$$

Systèmes différentiels

Exemple

$$\begin{cases} \dfrac{\mathrm{d}i}{\mathrm{d}t} = \dfrac{1}{L}(-Ri - s + e) \\ \dfrac{\mathrm{d}s}{\mathrm{d}t} = \dfrac{i}{C} \end{cases}$$



Équations différentielles d'ordre 2

Considérons maintenant une équation différentielle d'ordre 2 :

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

On résout un système différentiel en considérant le vecteur $X = (x, x')$:

$$\begin{cases} x' = y \\ y' = f(x, y, t) \end{cases} \quad \text{soit} \quad X' = F(X, t) \quad \text{avec} \quad F\big((x, y), t\big) = \big(y, f(x, y, t)\big).$$

Équations différentielles d'ordre 2

Considérons maintenant une équation différentielle d'ordre 2 :

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

On résout un système différentiel en considérant le vecteur $X = (x, x')$:

$$\begin{cases} x' = y \\ y' = f(x, y, t) \end{cases} \quad \text{soit} \quad X' = F(X, t) \quad \text{avec} \quad F\big((x, y), t\big) = \big(y, f(x, y, t)\big).$$

Exemple. On résout :

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \omega^2 \sin(\theta) = 0 \qquad \text{avec} \quad \omega^2 = \frac{g}{\ell}.$$

en se ramenant au système différentiel :

$$\begin{cases} \dfrac{\mathrm{d}\theta}{\mathrm{d}t} = \dot{\theta} \\ \dfrac{\mathrm{d}\dot{\theta}}{\mathrm{d}t} = -\omega^2 \sin(\theta) \end{cases}$$

Équations différentielles d'ordre 2

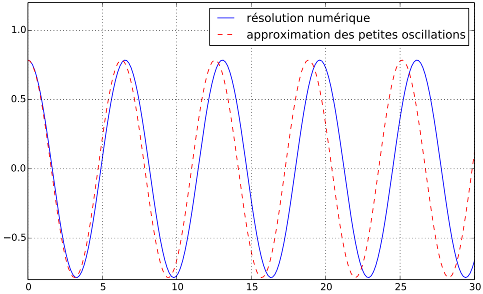
$$\begin{cases} \dfrac{\mathrm{d}\theta}{\mathrm{d}t} = \dot{\theta} \\ \dfrac{\mathrm{d}\dot{\theta}}{\mathrm{d}t} = -\omega^2 \sin(\theta) \end{cases}$$

La méthode d'Euler consiste à itérer les deux suites :

$$\begin{cases} \theta_{k+1} = \theta_k + h_k \dot{\theta}_k \\ \dot{\theta}_{k+1} = \dot{\theta}_k - \omega^2 h_k \sin(\theta_k) \end{cases} \qquad \text{avec} \quad h_k = t_{k+1} - t_k.$$

Équations différentielles d'ordre 2

$$\begin{cases} \dfrac{d\theta}{dt} = \dot{\theta} \\ \dfrac{d\dot{\theta}}{dt} = -\omega^2 \sin(\theta) \end{cases}$$



Utilisation de la fonction odeint

La fonction `odeint` du module `scipy.integrate` est dédiée à la résolution des équations différentielles vectorielles.

```
odeint(func, y0, t)
    Integrate a system of ordinary differential equations.

    Solve a system of ordinary differential equations:

        dy/dt = func(y,t0)

    where y can be a vector.

    Parameters
    --------
    func : callable(y, t0)
        Computes the derivative of y at t0.
    y0 : array
        Initial condition on y (can be a vector).
    t : array
        A sequence of time points for which to solve for y.  The initial
        value point should be the first element of this sequence.

    Returns
    -----
    y : array, shape (len(t), len(y0))
        Array containing the value of y for each desired time in t,
        with the initial value  y0  in the first row.
```

Utilisation de la fonction odeint

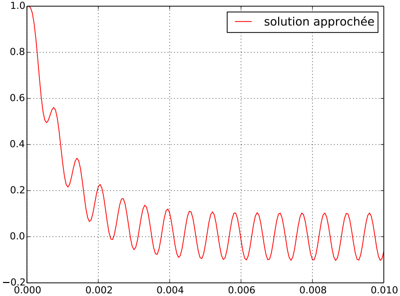
Résolution d'une équation scalaire

$$\frac{\mathrm{d}s}{\mathrm{d}t} = \frac{1}{\tau}(e(t) - s) \quad \text{avec} \quad s(0) = \frac{q_0}{C}.$$

```
def f(s, t):
    return (e(t) - s) / tau

t = np.linspace(0, .01, 200)
s = odeint(f, q0/C, t)

plot (t, s)
```



Utilisation de la fonction odeint

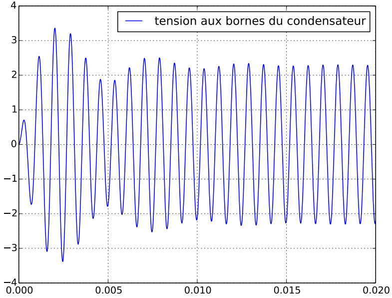
Résolution d'un système différentiel

$$\begin{cases} \dfrac{\mathrm{d}i}{\mathrm{d}t} = \dfrac{1}{L}(-Ri - s - e(t)) \\ \dfrac{\mathrm{d}s}{\mathrm{d}t} = \dfrac{i}{C} \end{cases} \quad \text{avec} \quad i(0) = 0 \text{ et } s(0) = 0.$$

```
def f(x, t):
    [i, s] = x
    return [(-R*i-s-e(t))/L, i/C]

t = np.linspace(0, .02, 200)
x = odeint(f, [0, 0], t)

plot (t, x[:, 1])
```



Le tableau x contient $n+1$ vecteurs de la forme $[i_k, s_k]$:

$$x = \Big[[i_0, s_0], [i_1, s_1], \ldots, [i_k, s_k], \ldots, [i_n, s_n]\Big]$$

Utilisation de la fonction odeint

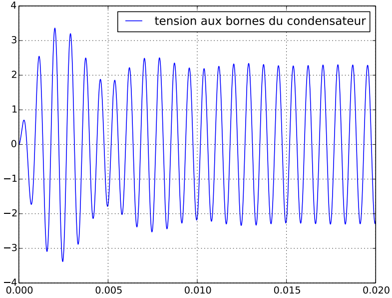
Résolution d'un système différentiel

$$\begin{cases} \dfrac{\mathrm{d}i}{\mathrm{d}t} = \dfrac{1}{L}(-Ri - s - e(t)) \\ \dfrac{\mathrm{d}s}{\mathrm{d}t} = \dfrac{i}{C} \end{cases} \quad \text{avec} \quad i(0) = 0 \text{ et } s(0) = 0.$$

```
def f(x, t):
    [i, s] = x
    return [(-R*i-s-e(t))/L, i/C]

t = np.linspace(0, .02, 200)
x = odeint(f, [0, 0], t)

plot (t, x[:, 1])
```



Le slicing $x[:,1]$ extrait de ce tableau le tableau :

$$x[:,1] = [s_0, s_1, \ldots, s_k, \ldots, s_n]$$

Utilisation de la fonction odeint

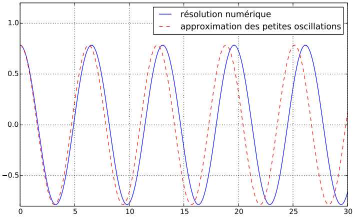
Résolution d'une équation du second ordre

$$\begin{cases} \dfrac{\mathrm{d}\theta}{\mathrm{d}t} = \dot{\theta} \\ \dfrac{\mathrm{d}\dot{\theta}}{\mathrm{d}t} = -\omega^2 \sin(\theta) \end{cases} \quad \text{avec} \quad \theta(0) = \theta_0 \text{ et } \dot{\theta}(0) = 0.$$

```python
def f(x, t):
    [theta, dtheta] = x
    return [dtheta, -omega**2*sin(theta)]

t = np.linspace(0, 30, 200)
x = odeint(f, [theta0, 0], t)

plot (t, x[:, 0])
```



Utilisation de la fonction odeint

Résolution d'une équation du second ordre

$$\begin{cases} \dfrac{\mathrm{d}\theta}{\mathrm{d}t} = \dot{\theta} \\ \dfrac{\mathrm{d}\dot{\theta}}{\mathrm{d}t} = -\omega^2 \sin(\theta) \end{cases} \qquad \text{avec} \qquad \theta(0) = \theta_0 \text{ et } \dot{\theta}(0) = 0.$$

Les valeurs de θ sont dans `x[:, 0]` et celles de $\dot{\theta}$ dans `x[:, 1]`. Il est dès lors facile d'obtenir le diagramme des phases :

```
plot(x[:, 0], x[:, 1])
```