

Résolution numérique des équations différentielles

1. Méthodes d'EULER

1.1 Présentation du problème

Nous allons dans un premier temps nous intéresser aux équations différentielles que l'on peut mettre sous la forme :

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

où f est une fonction définie sur une partie \mathcal{U} de \mathbb{R}^2 , à valeurs dans \mathbb{R} .

Une *solution* de cette équation différentielle est une fonction x de classe \mathcal{C}^1 définie sur un certain intervalle I de \mathbb{R} et à valeurs dans \mathbb{R} vérifiant :

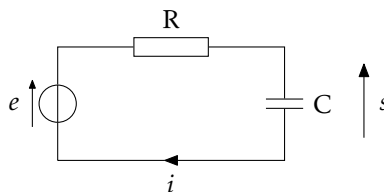
- (i) $\forall t \in I, (x(t), t) \in \mathcal{U}$;
- (ii) $\forall t \in I, x'(t) = f(x(t), t)$.

Nous allons adjoindre à cette équation différentielle une *condition initiale* sous la forme d'un couple $(x_0, t_0) \in \mathcal{U}$ et chercher à résoudre le *problème de CAUCHY* suivant :

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

Sous certaines conditions sur f que nous ne détaillerons pas, ce problème admet une unique solution¹, que nous allons chercher à déterminer numériquement.

Exemple. Considérons le circuit électrique suivant :



Il est régi par les relations $s = e - Ri$ et $i = C \frac{ds}{dt}$ qui conduisent à l'équation différentielle : $RC \frac{ds}{dt} + s = e$. En posant $\tau = RC$ on est amené à résoudre le problème de CAUCHY :

$$\begin{cases} \frac{ds}{dt} = \frac{1}{\tau}(e(t) - s) \\ s(0) = \frac{q_0}{C} \end{cases}$$

où q_0 désigne la charge initiale du condensateur.

Bien sur, il est possible de résoudre analytiquement cette équation lorsque la fonction d'entrée $e(t)$ est connue. Par exemple, dans le cas d'un signal d'entrée nul $e(t) = 0$ (décharge du condensateur) on obtient :

$$s(t) = \frac{q_0}{C} e^{-t/\tau}$$

Dans le cas d'un signal d'entrée sinusoïdal $e(t) = E \cos(\omega t)$ la résolution exacte reste possible mais plus délicate ; néanmoins on peut obtenir :

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

Dans d'autres cas (par exemple la réponse du système à des impulsions périodiques) l'expression formelle qu'on peut obtenir n'est plus forcément suffisante pour en permettre une analyse qualitative. On se tourne dans ce cas vers les méthodes de résolution numérique.

1. Plus précisément, une unique solution *maximale*, c'est-à-dire ne pouvant être prolongée sur un intervalle plus grand.

• Champ de vecteurs

On appelle *champ de vecteurs* une application qui à tout point (x, t) de \mathcal{U} associe un vecteur. Lorsque $x(t)$ est une solution de l'équation différentielle, le vecteur $\vec{v}(t) = \vec{i} + x'(t)\vec{j} = \vec{i} + f(x(t), t)\vec{j}$ est un vecteur tangent de cette solution au point de paramètre t . Ceci permet d'associer naturellement à l'équation différentielle $x' = f(x, t)$ le champ de vecteurs défini sur \mathcal{U} par : $\vec{v}(x, t) = \vec{i} + f(x, t)\vec{j}$.

On représente en général un champ de vecteurs en des points régulièrement espacés de \mathcal{U} et en normalisant les vecteurs. La fonction quiver du module `matplotlib.pyplot` permet de tracer un champ de vecteurs. Utilisons-la pour obtenir celui associé à l'équation différentielle $s' = -s/\tau$ (décharge d'un condensateur). La courbe rouge est la solution déterminée analytiquement.

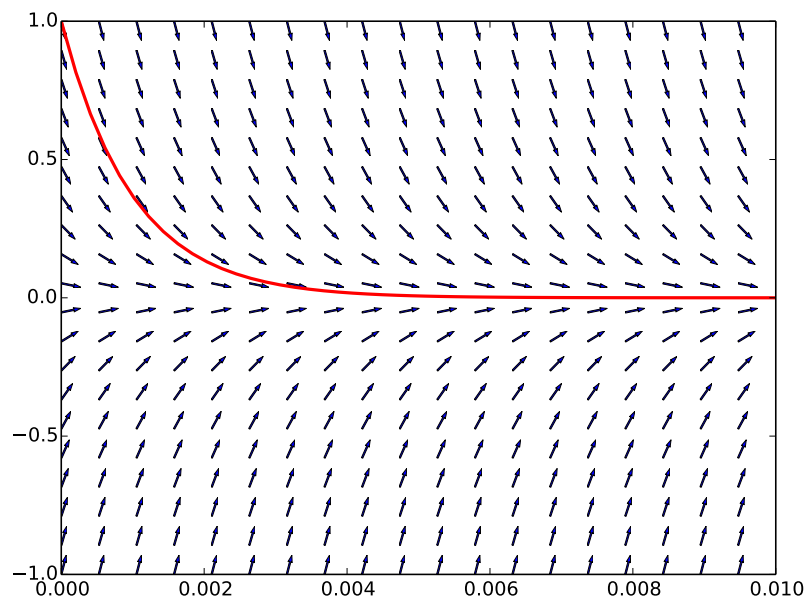


FIGURE 1 – Décharge du condensateur.

1.2 Méthode d'EULER explicite

Les méthodes que nous allons étudier consistent à subdiviser l'intervalle de temps $[t_0, t_0 + T]$ en $n + 1$ points $t_0 < t_1 < \dots < t_n = t_0 + T$ puis à approcher la relation :

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

La *méthode d'EULER explicite* consiste à approcher cette intégrale par la méthode du rectangle gauche, autrement dit à approcher $\int_{t_k}^{t_{k+1}} f(x(t), t) dt$ par $(t_{k+1} - t_k)f(x(t_k), t_k)$.

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

$$\forall k \in \llbracket 0, n-1 \rrbracket, \quad \boxed{x_{k+1} = x_k + h_k f(x_k, t_k)}$$

On observera qu'en général, seul le premier point x_0 de cette méthode est une valeur exacte ; les autres points sont calculés à partir de l'approximation précédente, ce qui peut conduire la valeur calculée x_k à s'écartier de plus en plus de la valeur exacte $x(t_k)$. Ce phénomène peut être particulièrement marqué pour la méthode d'EULER ; on observera ce phénomène figure 2 avec la résolution du problème de CAUCHY $x' = x$, $x(0) = 1$ (dont la solution exacte est $x(t) = e^t$).

On se convaincra aisément que lorsque la subdivision s'affine la précision en général s'améliore (sans pour autant faire disparaître ce phénomène de divergence). D'autres méthodes décrites plus loin permettent de

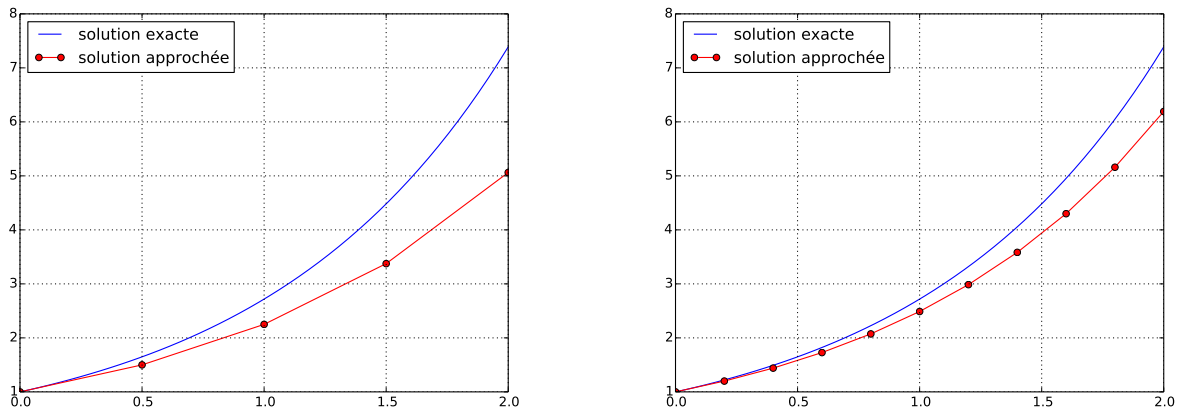


FIGURE 2 – Méthode d’Euler explicite avec $n = 4$ puis avec $n = 10$.

notablement atténuer la divergence de la solution numérique de la solution analytique. Cependant, dans de nombreux cas la méthode d’EULER procure des résultats acceptables (au moins qualitativement). On peut en observer un exemple avec la réponse du circuit RC à un signal d’entrée sinusoïdal figure 3.

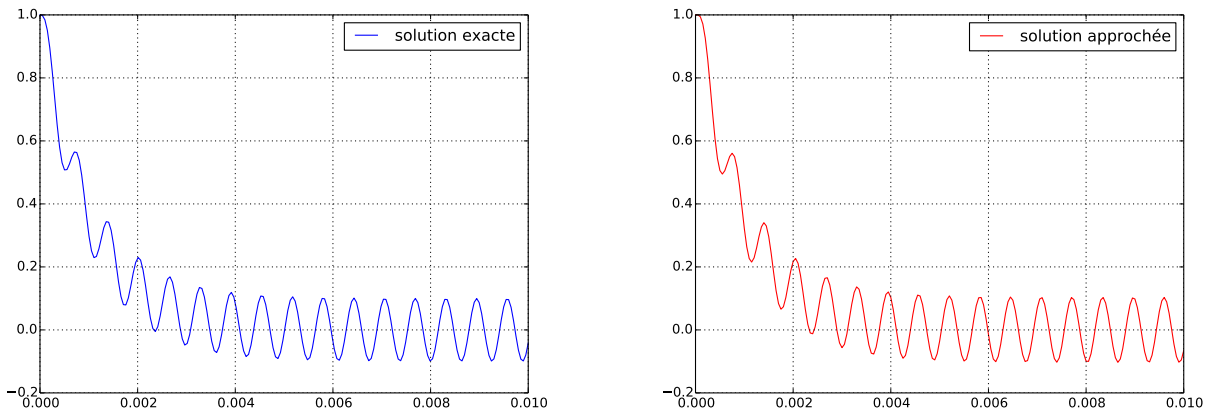


FIGURE 3 – Réponse d’un circuit RC à une sollicitation sinusoïdale par la méthode d’Euler avec $n = 200$.

Remarque. Les exemples présentés utilisent tous une subdivision de pas régulier ; cependant de meilleurs résultats sont obtenus en adaptant le pas à la fonction f : 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. Dans ce cas, on parle de méthode à *pas adaptatif*, mais nous n’irons pas plus loin sur ce sujet.

1.3 Méthode d’EULER implicite

La *méthode d’EULER implicite* consiste à approcher l’intégrale $\int_{t_k}^{t_{k+1}} f(x(t), t) dt$ par la méthode du rectangle droit, ce qui conduit à définir la suite (x_0, x_1, \dots, x_n) par les relations :

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

On observe que cette relation ne procure pas une relation explicite de x_{k+1} puisque ce terme est aussi présent dans le second membre. Pour calculer ce terme il est souvent nécessaire de coupler cette méthode à une méthode de résolution numérique des équations telle la méthode de NEWTON-RAPHSON (voir le chapitre 8). Pour cette raison, elle se révèle plus coûteuse à mettre en œuvre.

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. On

trouvera figure 4 une comparaison des deux méthodes d'Euler qui met en évidence l'instabilité de la méthode explicite sur le long terme. La solution de l'équation $x' = t \sin(x)$ avec la condition initiale $x(0) = 1$ est une fonction croissante qui tend vers π lorsque t tend vers $+\infty$.

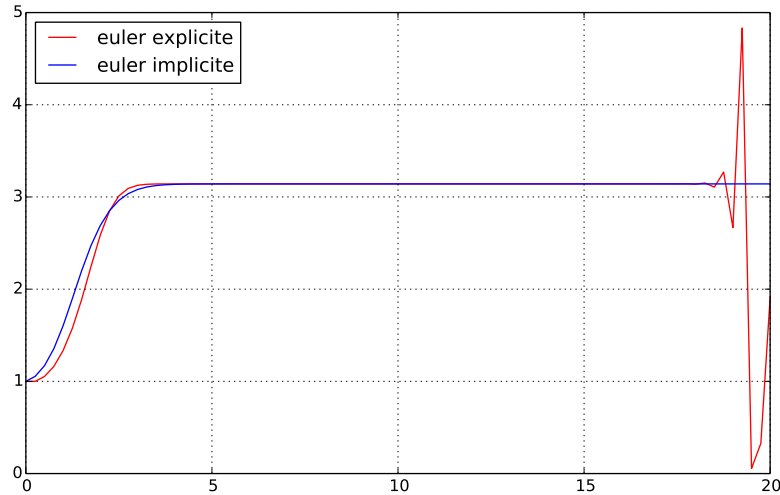


FIGURE 4 – Comparaison des méthodes d'Euler explicite et implicite.

Sur un intervalle de temps par trop important les deux méthodes donnent des résultats comparables. Cependant, à partir d'un certain rang l'instabilité sur le long terme de la méthode explicite se révèle. En effet, la relation d'EULER explicite s'écrit :

$$x_{k+1} = x_k + ht_k \sin(x_k)$$

où h (fixé) est le pas de la subdivision choisi.

Puisque x_k est voisin de π le terme $\sin(x_k)$ est proche de 0 mais non nul. Dans le même temps, le terme t_k ne cesse de croître et prend progressivement le pas sur le terme $\sin(x_k)$. Dès lors, le comportement de la suite x devient chaotique : lorsque la valeur calculée x_k s'écarte un peu trop de π , le comportement de la suite devient imprévisible.

En revanche, l'expérience montre que même sur un grand intervalle de temps la solution fournie par la méthode d'EULER implicite continue de fournir un résultat cohérent.

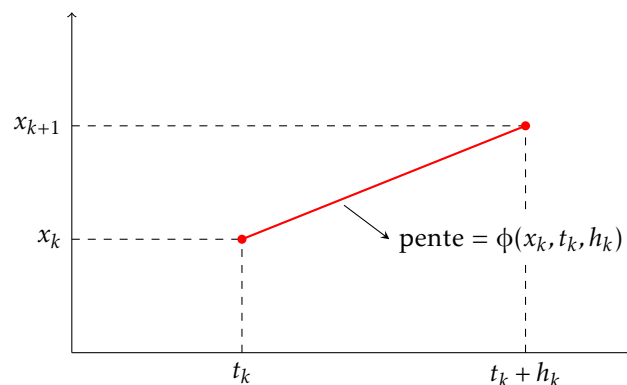
1.4 Schéma numérique

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

$$\forall k \in \llbracket 0, n-1 \rrbracket, \quad x_{k+1} = x_k + h_k \phi(x_k, t_k, h_k)$$

Par exemple, la méthode d'EULER explicite correspond au schéma $\phi(x, t, h) = f(x, t)$.

Autrement dit, $\phi(x, t, h)$ correspond à la valeur que l'on choisit pour approcher la dérivée entre t et $t+h$:



Remarque. Un schéma numérique implicite à un pas prend la forme $x_{k+1} = x_k + h_k \phi(x_k, x_{k+1}, t_k, h_k)$; c'est le cas par exemple de la méthode d'EULER implicite.

Un schéma numérique explicite à p pas prend la forme $x_{k+1} = x_k + h_k \phi(x_k, x_{k-1}, \dots, x_{k-p+1}, t_k, h_k)$. 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.

DÉFINITION. — On appelle erreur de consistance la suite $(e_k)_{1 \leq k \leq n}$ définie par $e_{k+1} = x(t_{k+1}) - x(t_k) - h_k \phi(x(t_k), t_k, h_k)$, où x désigne la solution exacte du problème de CAUCHY.

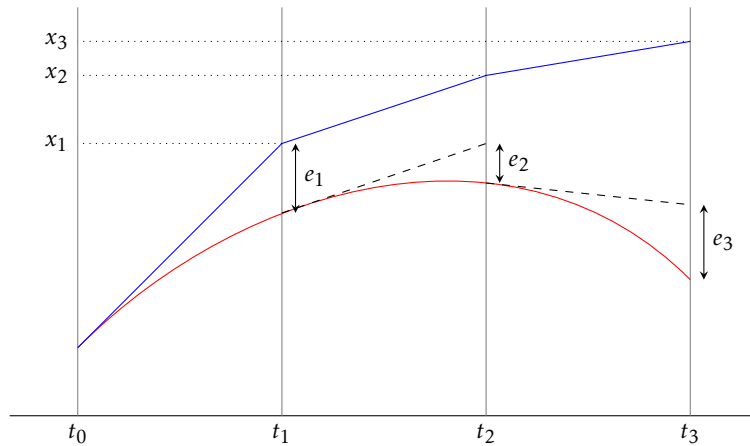


FIGURE 5 – L'erreur de consistance de la méthode d'EULER.

Un schéma numérique de pas régulier est dit d'ordre p lorsque $e_{k+1} = O(h_k^{p+1})$ lorsque h_k tend vers 0.

Une méthode numérique est dite *consistante* lorsque : $\lim_{n \rightarrow +\infty} \sum_{k=1}^n e_k = 0$.

Attention. La consistance d'une méthode est une condition nécessaire *mais pas suffisante* pour que la solution approchée converge uniformément vers la solution exacte sur le segment $[t_0, t_0 + T]$.

Considérons l'illustration de la figure 5. À la date t_1 , e_1 représente effectivement l'écart entre la solution exacte et la solution approchée :

$$x(t_1) = x_1 + e_1$$

Mais dès la date t_2 , à l'erreur de consistance s'ajoute le fait que le calcul de la pente n'a pas été fait pour $x = x(t_1)$ mais pour $x = x_1$. Ainsi, on a :

$$\begin{cases} x_2 = x_1 + h_1 \phi(x_1, t_1, h_1) \\ x(t_2) = x(t_1) + h_1 \phi(x(t_1), t_1, h_1) + e_2 \end{cases} \quad \text{donc} \quad x(t_2) = x_2 + (e_1 + e_2) + h_1 (\phi(x(t_1), t_1, h_1) - \phi(x_1, t_1, h_1))$$

De même, on obtient :

$$x(t_3) = x_3 + \underbrace{(e_1 + e_2 + e_3)}_{\text{erreurs de consistance}} + \underbrace{h_1 (\phi(x(t_1), t_1, h_1) - \phi(x_1, t_1, h_1)) + h_2 (\phi(x(t_2), t_2, h_2) - \phi(x_2, t_2, h_2))}_{\text{erreurs dans l'évaluation de } \phi}$$

L'évaluation de l'erreur totale commise sur l'intervalle $[t_0, t_0 + T]$ est un problème très délicat que nous n'aborderons pas ; nous nous contenterons du résultat suivant :

THÉORÈME. — Lorsque la fonction f est de classe \mathcal{C}^1 la méthode d'EULER est une méthode consistante d'ordre 1.

Preuve. Si f est de classe \mathcal{C}^1 alors x est de classe \mathcal{C}^2 et d'après la formule de TAYLOR-YOUNG,

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

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

Pour montrer que la méthode d'EULER est consistante, reprenons ces calculs mais en appliquant cette fois l'égalité de TAYLOR-LAGRANGE : puisque x est de classe \mathcal{C}^2 il existe $\tau_k \in]t_k, t_{k+1}[$ tel que :

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

Autrement dit, $|e_{k+1}| = \frac{h_k^2}{2} |x''(\tau_k)|$. Puisque x est de classe \mathcal{C}^2 sur le segment $[t_0, t_0 + T]$ sa dérivée seconde est majorée en valeur absolue par une constante M_2 et dans ce cas, $|e_{k+1}| \leq \frac{M_2}{2} h_k^2$.

Dans le cas d'une subdivision de pas régulier nous avons $h_k = \frac{T}{n}$ et $|e_{k+1}| \leq \frac{M_2 T^2}{2n^2}$. Ainsi,

$$\left| \sum_{k=1}^n e_k \right| \leq \sum_{k=1}^n |e_k| \leq \frac{M_2 T}{2n} \quad \text{et} \quad \lim_{n \rightarrow +\infty} \sum_{k=1}^n e_k = 0.$$

□

2. Les méthodes de RUNGE-KUTTA

2.1 La méthode du point milieu

Notons M_n le point de coordonnées $(t_n, x(t_n))$. Comme le montre la figure ci-dessous, le segment $[M_n, M_{n+1}]$ a en général une pente plus proche de $x'(t_n + h_n/2)$ (pente de la tangente au point milieu) que de $x'(t_n)$, pente de la tangente en M_n utilisée dans la méthode d'EULER.

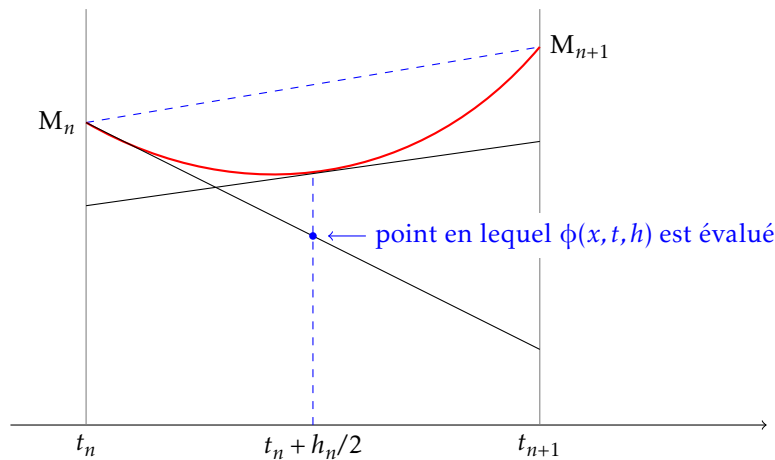


FIGURE 6 – En général, la pente de la tangente au point milieu est une bonne approximation de la pente de la corde.

D'où l'idée de remplacer la relation d'EULER $x_{n+1} = x_n + h_n f(x_n, t_n)$ par : $x_{n+1} = x_n + h_n f(x(t_n + h_n/2), t_n + h_n/2)$. Cependant, comme on ne connaît pas $x(t_n + h_n/2)$, on en cherche une approximation notée $x_{n+1/2}$; le schéma d'EULER nous suggère de prendre $x_{n+1/2} = x_n + \frac{h_n}{2} f(x_n, t_n)$.

On aboutit donc au schéma numérique suivant :

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

qui correspond au schéma $\phi(x, t, h) = f\left(x + \frac{h}{2} f(x, t), t + \frac{h}{2}\right)$.

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

On comparera figure 7 les méthodes d'EULER et du point milieu pour résoudre le problème de CAUCHY $x' = x$, $x(0) = 1$ sur l'intervalle $[0, 2]$ avec $n = 4$.

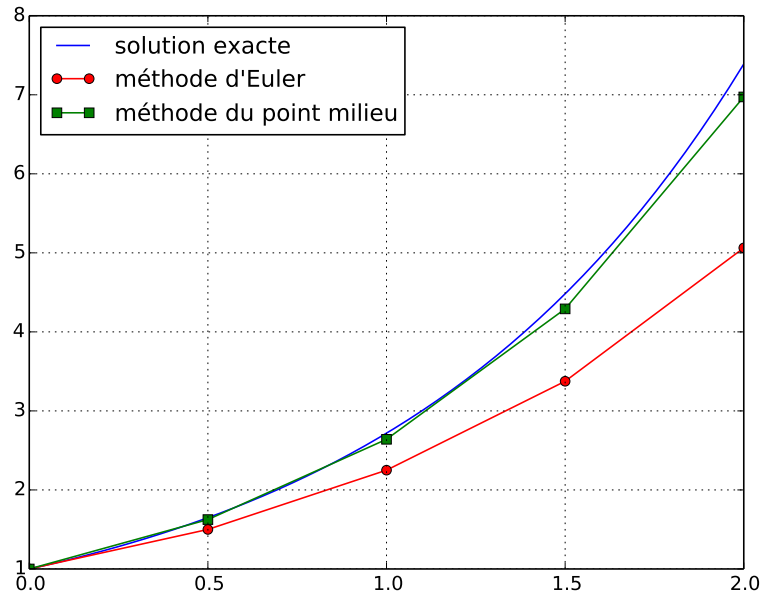


FIGURE 7 – Comparaison entre les méthodes d’Euler et du point milieu.

2.2 Les méthodes de RUNGE-KUTTA

Nous allons maintenant généraliser la démarche que nous venons de suivre pour la méthode du point milieu. On appelle *méthode de RUNGE-KUTTA explicite à s étages* (en abrégé RK_s) une méthode définie par le schéma :

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

où les coefficients c_i , a_{ij} et b_i sont des constantes qui définissent précisément le schéma. On supposera toujours dans la suite que $c_1 = 0$, $c_i = \sum_{j=1}^{i-1} a_{ij}$ et $\sum_{i=1}^s b_i = 1$.

Par exemple, la méthode de du point milieu est une méthode de type RK₂ puisqu’elle se définit par les relations :

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

On a ici $c_2 = 1/2$, $a_{21} = 1/2$, $b_1 = 0$, $b_2 = 1$.

On représente usuellement une méthode RK_s à l’aide d’un *tableau de BUTCHER* :

c_1					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
\vdots	\vdots	\vdots	\ddots		
c_s	a_{s1}	a_{s2}	\cdots	$a_{s,s-1}$	
	b_1	b_2	\cdots	b_{s-1}	b_s

Ainsi, les méthodes d'EULER et du point milieu sont des méthodes de types respectifs RK₁ et RK₂ définies par les tableaux :

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

Sous de bonnes hypothèses et en choisissant de manière adéquate les coefficients de ces tableaux, une méthode de RUNGE-KUTTA explicite à *s* étages définit un schéma d'ordre *s*.

Il existe de nombreuses méthode de ce type, les plus fréquentes sont les suivantes :

$\begin{array}{c c} 0 & \\ \hline & 1 \end{array}$ <p>Méthode d'EULER (ordre 1)</p>	$\begin{array}{c cc} 0 & & \\ \hline 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$ <p>Méthode du point milieu (ordre 2)</p>
$\begin{array}{c cc} 0 & & \\ \hline 1 & 1 & \\ \hline & 1/2 & 1/2 \end{array}$ <p>Méthode de HEUN (ordre 2)</p>	$\begin{array}{c ccc} 0 & & & \\ \hline 1/2 & 1/2 & & \\ 1/2 & 0 & 1/2 & \\ \hline 1 & 0 & 0 & 1 \\ \hline & 1/6 & 2/6 & 2/6 & 1/6 \end{array}$ <p>Méthode RK₄ (ordre 4)</p>

FIGURE 8 – Schémas de RUNGE-KUTTA classiques.

Par exemple, le schéma de HEUN est le suivant :

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

soit $\phi(x, t) = \frac{1}{2}(f(x, t) + f(x + hf(x, t), t + h))$. Cette méthode fait intervenir la moyenne des dérivées en *t* et en *t + h* ; lorsque les solutions sont convexes, la première constitue un minorant de la dérivée entre *t* et *t + h*, la seconde un majorant. La moyenne des deux est donc encadrée par ces deux bornes.

Exercice. Rédiger les formules correspondant à la méthode RK₄.

La méthode RK₄ est parmi les méthodes explicites la plus couramment utilisée. Elle fournit en général d'excellents résultats.

3. Systèmes différentiels et équations d'ordre supérieur

3.1 Systèmes différentiels

Considérons maintenant un système de *p* équations différentielles liant *p* fonctions numériques x_1, \dots, x_p et prenant la forme suivante :

$$\begin{cases} x'_1 = f_1(x_1, x_2, \dots, x_n, t) \\ x'_2 = f_2(x_1, x_2, \dots, x_n, t) \\ \dots = \dots \\ x'_p = f_p(x_1, x_2, \dots, x_n, t) \end{cases}$$

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

Considérons maintenant la fonction vectorielle *F* définie sur $\mathbb{R}^p \times \mathbb{R}$ et à valeurs dans \mathbb{R}^p dont les fonctions f_1, \dots, f_p sont les composantes :

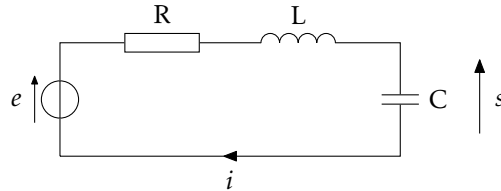
$$F(x_1, x_2, \dots, x_p, t) = (f_1(x_1, x_2, \dots, x_p, t), f_2(x_1, x_2, \dots, x_p, t), \dots, f_p(x_1, x_2, \dots, x_p, t)).$$

La fonction vectorielle X vérifie l'équation différentielle suivante :

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

et dès lors on peut lui appliquer les méthodes de résolution numériques qui ont été présentées dans la section précédente. Autrement dit, plutôt que de voir dans le problème à résoudre un système d'équations différentielles couplées, *il est plus fécond de n'y voir qu'une seule équation* qui concerne une fonction à valeurs vectorielles.

Exemple. Considérons le circuit électrique suivant :



Il est régi par les relations $s = e - Ri - L \frac{di}{dt}$ et $i = C \frac{ds}{dt}$ qui conduisent au système différentiel :

$$\begin{cases} \frac{di}{dt} = \frac{1}{L}(-Ri - s + e) \\ \frac{ds}{dt} = \frac{i}{C} \end{cases}$$

Bien entendu, la résolution exacte est possible en combinant ces deux équations pour obtenir une équation du second ordre vérifiée par la fonction s , mais la démarche numérique consiste à considérer plutôt le vecteur $X = (i, s)$ solution de l'équation différentielle $X' = F(X, t)$ avec $F((i, s), t) = \left(\frac{1}{L}(-Ri - s + e(t)), \frac{i}{C}\right)$.

Dans le cas vectoriel, la méthode d'EULER consiste à discrétiser l'intervalle de temps $[t_0, t_0 + T]$ et à calculer la suite de vecteurs $(X_k)_{1 \leq k \leq n}$ à l'aide des formules :

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

ce qui dans le cas du circuit RLC revient à écrire :

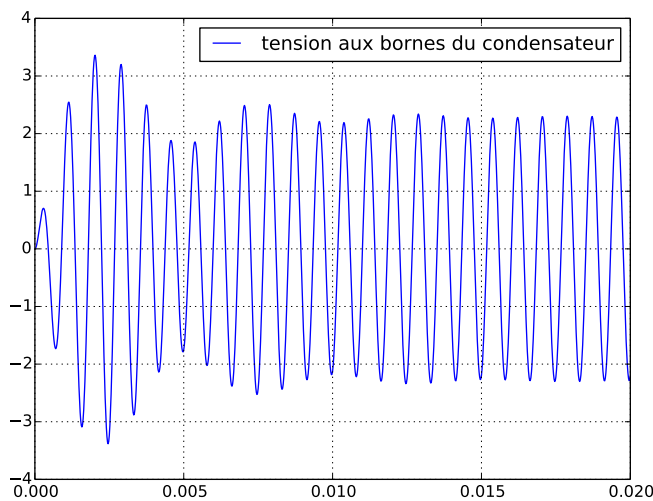
$$\begin{cases} i_{k+1} = i_k + \frac{h_k}{L}(-Ri_k - s_k + e(t_k)) \\ s_{k+1} = s_k + h_k \frac{i_k}{C} \end{cases}$$


FIGURE 9 – Réponse d'un circuit RLC à une sollicitation sinusoïdale.

Bien entendu, toutes les méthodes décrites pour les équations différentielles scalaires s'appliquent encore aux équations différentielles vectorielles.

3.2 Équations différentielles d'ordre 2

Considérons maintenant une équation différentielle scalaire d'ordre 2, c'est-à-dire de la forme :

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

La méthode consiste ici à se ramener à un système différentiel en considérant le vecteur $X = (x, x')$. Ce dernier est en effet solution du système différentiel :

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

Considérons par exemple constitué d'une masse m liée à un point fixe par un fil de longueur ℓ , et oscillant dans le champ de pesanteur terrestre. En notant θ l'angle que fait le fil avec la verticale, on obtient l'équation du mouvement :

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

Pour résoudre numériquement cette équation, on se ramène au système différentiel suivant :

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

Dès lors, on peut appliquer une quelconque des méthodes de résolutions numériques que nous avons étudiées. Par exemple, la méthode d'EULER consiste à discrétiser l'intervalle de temps $[t_0, t_0 + T]$ et à calculer la suite de valeurs $(\theta_k, \dot{\theta}_k)_{1 \leq k \leq n}$ à partir des valeurs initiales $(\theta_0, \dot{\theta}_0)$ et des relations :

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

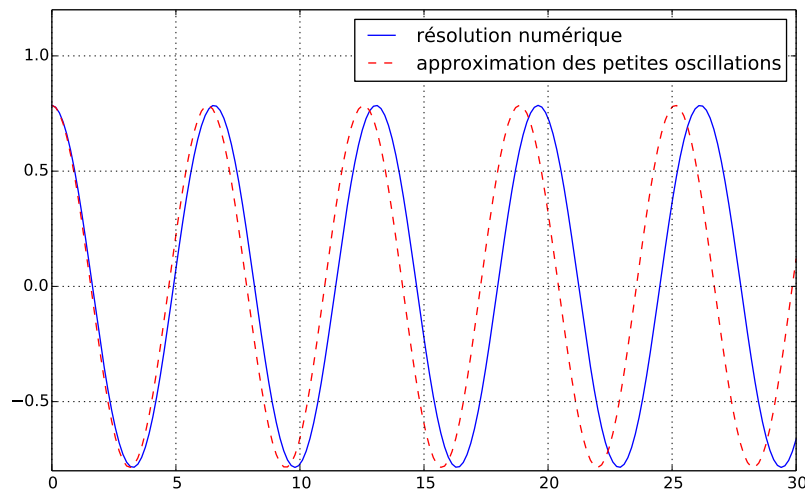


FIGURE 10 – La résolution numérique montre que l'approximation dite « des petites oscillations » sous-estime la valeur de la période.

4. Utilisation de la fonction odeint

Les graphes des figures 9 et 10 n'ont pas été obtenus par la méthode d'EULER (qui donne souvent des résultats médiocres lorsqu'il s'agit de résoudre un système différentiel) mais à l'aide de la fonction `odeint` du module `scipy.integrate`.

Cette fonction est dédiée à la résolution des systèmes numériques, mais s'applique aussi aux équations scalaires (qui somme toute ne sont que des systèmes différentiels d'ordre 1) et aux équations différentielles d'ordre supérieur, à charge pour l'utilisateur de convertir cette dernière sous la forme d'un système différentiel, comme nous venons de l'expliquer.

Examinons une version simplifiée de l'aide en ligne :

```
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.
```

On notera que le premier paramètre de `odeint` est une fonction chargée d'exprimer la dérivée première de y en fonction de y et de t : c'est la fonction f qui représente l'équation différentielle à résoudre $y' = f(y, t)$. Le second paramètre est la condition initiale y_0 (qui suivant les cas sera un scalaire ou un vecteur) et le troisième un tableau contenant les valeurs discrètes choisies dans l'intervalle $[t_0, t_0 + T]$ et dont le premier terme est t_0 .

La valeur retournée par la fonction `odeint` est un tableau de même longueur que `t` et qui contient les valeurs calculées par la méthode de résolution numérique. Attention, chaque case de ce tableau `y` a même dimension que `y0` (c'est un scalaire lorsqu'on résout une équation différentielle scalaire, c'est un vecteur lorsqu'on résout un système différentiel).

Résolution d'une équation scalaire

Le script qui a été utilisé pour obtenir le graphe de la figure 3 ressemble à :

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

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

plot (t, s)
```

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

Résolution d'un système différentiel

Le script qui a été utilisé pour obtenir le graphe de la figure 9 ressemble à :

```
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])
```

$$\begin{cases} \frac{di}{dt} = \frac{1}{L}(-Ri - s - e(t)) \\ \frac{ds}{dt} = \frac{i}{C} \end{cases} \text{ avec } i(0) = 0 \text{ et } s(0) = 0.$$

On notera l'usage qui est fait du slicing pour extraire de x les valeurs de s . En effet, le tableau x contient $n + 1$ vecteurs de la forme $[i_k, s_k]$, un pour chacune des valeurs t_k comprises dans le tableau t :

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

Le slicing $x[:, 1]$ extrait de ce tableau le tableau $[x[k, 1] \mid 0 \leq k \leq n]$ autrement dit :

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

Pour obtenir le tracé de l'évolution de l'intensité i il suffirait d'écrire de même : `plot(t, x[:, 0])`.

Résolution d'une équation différentielle du second ordre

Le script qui a été utilisé pour obtenir le graphe de la figure 10 ressemble à :

```
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])
```

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

Remarque. Sachant que les valeurs de θ sont rangées dans le tableau $x[:, 0]$ et celles de $\dot{\theta}$ dans $x[:, 1]$ il suffit pour obtenir le diagramme des phases (c'est-à-dire la courbe paramétrée par $\theta(t)$ en abscisse et $\dot{\theta}(t)$ en ordonnée) d'écrire :

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

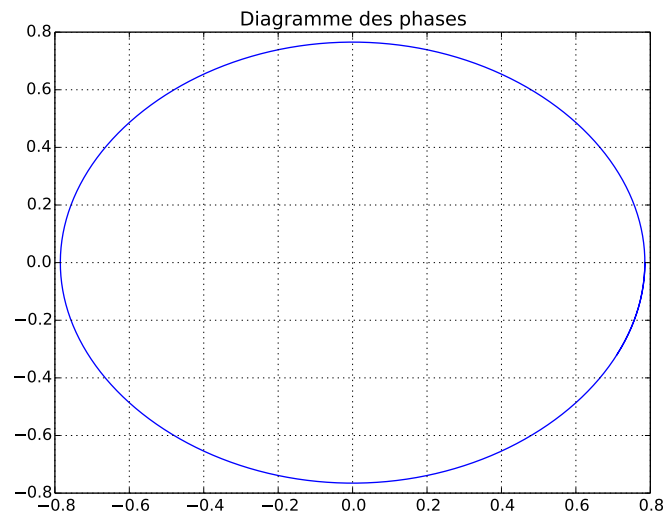


FIGURE 11 – Diagramme des phases du pendule pesant.