Equations Différentielles II

STEP, MINES ParisTech*

5 décembre 2019 (`#05c9041`)

Table des matières

Introduction

Ce chapitre est consacré à la résolution numérique d'équations différentielles

$$\dot{x} = f(t,x) \qquad , \qquad x(t_0) = x_0 \ .$$

La nécessité de développer des méthodes d'intégration numériques vient du constat que seule une infime partie des équations différentielles sont résolvables exactement. Or, on a parfois besoin de connaître le plus précisément possible le comportement futur d'un système dynamique :

— soit en temps fini, par exemple pour déterminer la trajectoire d'une fusée pour la mise en orbite d'un satellite;
— soit en temps *long*, par exemple pour déterminer un cycle limite asymptotique (dynamique de population) ou bien se prononcer sur la stabilité de notre système solaire.

La méthode la plus connue est la *méthode d'Euler* datant de 1768, qui consiste à implémenter

$$x^{j+1} = x^j + \Delta t\, f(t_j, x^j) \qquad x^0 = x_0$$

pour un pas de temps Δt suffisamment petit. Cette méthode appartient à la famille des méthodes *explicites*, c'est-à-dire que x^{j+1} est directement et explicitement défini en fonction de x^j. En 1824, Cauchy montre la convergence de cette méthode lorsque le pas de temps Δt tend vers 0, et prouve ainsi l'existence et l'unicité des solutions (en fait, il utilise plutôt la version *implicite* de la méthode d'Euler).

Même si la méthode d'Euler suffit dans les cas simples, elle exige parfois de recourir à des pas très faibles pour obtenir une précision acceptable sur des temps longs (voir Systèmes raides plus bas). Parfois, le compromis entre précision à

chaque itération et accumulation des erreurs d'arrondis devient même impossible. De plus, cette méthode n'est pas adaptée à la simulation de certains systèmes dont certaines proprétés cruciales (comme la conservation de l'énergie) ne sont pas préservées (voir Systèmes Hamiltoniens plus bas). Au cours des derniers siècles, les scientifiques ont donc progressivement développé des méthodes de plus en plus complexes et performantes : schémas multi-pas d'ordre supérieur, méthodes implicites, variation du pas, schémas symplectiques etc.

En fait, dans l'histoire des équations différentielles, c'est souvent la mécanique céleste qui a été motrice des plus grandes avancées. Au milieu du XIX^e siècle, les astronomes Adams et Le Verrier prédisent mathématiquement l'existence et la position de la planète Neptune et l'on entend parler pour la première fois de méthodes multi-pas. Ensuite, les progrès se sont enchaînés au rythme des modèles physiques. La première tendance a été de rechercher des schémas permettant toujours plus de précision à pas plus grand. Parmi les dates clés, on peut citer la publication en 1895 de la première méthode de Runge-Kutta par Runge, puis en 1901, de la populaire méthode de Runge-Kutta d'ordre 4 par Kutta, et ensuite en 1910, de l'*extrapolation de Richardson* permettant la montée en ordre et donc le recours à des pas plus grand pour une même précision. Mais au milieu du XX^e siècle, on découvre des systèmes, dits *raides* (Hirschfelder, 1952), pour lesquels cette montée en ordre ne suffit pas et pour lesquels il faut repenser de nouveaux schémas (Dalquist, 1968). Enfin, à partir des années 80, les scientifiques développent l'intégration numérique *géométrique*, c'est-à-dire qui préservent les propriétés structurelles du système (symmétrie, conservation d'énergie etc.), utile en particulier pour la simulation des systèmes hamiltoniens.

Objectifs du cours

Ce cours a pour but de sensibiliser aux problèmes apparaissant lors de la simulation numérique des solutions d'équations différentielles, et de donner les bases d'analyse d'erreur numérique. Pour un exposé plus approfondi, on pourra par exemple se referrer à (Demailly 2006).

En première lecture :

- comprendre les limites d'un schéma d'Euler
- comprendre qu'un schéma numérique à un pas consiste à discrétiser une intégrale, en connaître quelques-uns autres que le schéma d'Euler
- comprendre les notions de consistance/convergence d'un schéma et leur ordre. Savoir montrer que le schéma d'Euler explicite est convergent d'ordre 1.

En deuxième lecture :

- comprendre comment fonctionne un schéma implicite et comment l'implémenter

— comprendre que la convergence est la combinaison de deux concepts : la consistance et la stabilité
— savoir montrer la convergence de schémas de base, tels que ceux donnés en exercice
— comprendre l'apport de schémas symplectiques pour les systèmes hamiltoniens.

Limites du schéma d'Euler

La première limite du schéma d'Euler est qu'il est d'ordre 1, c'est-à-dire qu'il produit une erreur en Δt^2 à chaque pas. Nous verrons dans la suite d'autres algorithmes d'ordre supérieur qui permettent d'utiliser un pas plus grand pour une précision donnée. Mais au delà de cette problématique, il existe des systèmes pour lesquels de telles méthodes (même d'ordre supérieur) échouent. En voici deux exemples célèbres.

Systèmes raides

La dénomination *systèmes raides* a été introduite en 1952 par Hirschfelder pour désigner des systèmes comprenant des dynamiques aux constantes de temps très différentes. Dans ce cas, le pas nécessaire pour simuler avec précision les dynamiques très rapides est si petit, qu'il est alors impossible de simuler assez longtemps pour observer les parties lentes. La particularité de ces systèmes est que cette décroissance du pas apparaît alors que la solution est parfaitement régulière, et non pas proche de singularités. C'est le cas des systèmes linéaires

$$\dot{x} = Ax + b$$

avec A Hurwitz quand le rapport entre les parties réelles maximales et minimales des valeurs propres devient très grand. Ce phénomène peut notamment apparaître dans un simple système masse/ressort

$$m\ddot{y} = -\rho\dot{y} - ky$$

qui se met sous la forme précédente avec $x = (y, \dot{y}) \in \mathbb{R}^2$ et $A = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\rho}{m} \end{pmatrix}$. Lorsque les valeurs propres sont réelles (i.e. $\rho > 2\sqrt{mk}$), leur rapport est donné par

$$\frac{1 + \sqrt{1 - 4\frac{mk}{\rho^2}}}{1 - \sqrt{1 - 4\frac{mk}{\rho^2}}}$$

qui explose lorsque $\frac{mk}{\rho^2}$ tend vers 0. Par exemple, lorsque les frottements sont très grands par rapport à la raideur du ressort, ou bien lorsque ρ et k sont du même ordre de grandeur et très grands.

Plus généralement, la coexistence de dynamiques très lentes à très rapides apparaît en cinétique chimique ou en biologie. La réaction de Robertson (1966)

$$\begin{aligned} A &\xrightarrow{0.04} B \quad \text{(lente)} \\ B + B &\xrightarrow{3\times10^7} B + C \quad \text{(très rapide)} \\ B + C &\xrightarrow{10^4} A + C \quad \text{(rapide)} \end{aligned}$$

modélisée par

$$\begin{aligned} \dot{x}_a &= -0.04x_b + 10^4 x_b x_c \\ \dot{x}_b &= 0.04x_a - 10^4 x_b x_c - 3 \times 10^7 x_b^2 \\ \dot{x}_c &= 3 \times 10^7 x_b^2 \end{aligned}$$

en est un exemple classique, souvent utilisée pour tester les schémas numériques. Il s'avère que pour ces systèmes, des schémas dits *implicites* performent beaucoup mieux car ils autorisent l'utilisation de pas plus grands pour une même précision et plus de stabilité (voir l'exercice *Explicite ou Implicite?*). Pour plus de détails voir (Hairer and Wanner 1996).

Systèmes hamiltoniens

La mécanique hamiltonienne permet typiquement de modéliser le comportement de systèmes dont une certaine énergie est conservée au cours du temps. Il peut s'agir par exemple de planètes en interaction gravitationnelle, de particules en interaction électromagnétique, etc.

Par exemple, dans un problème à N corps en interaction gravitationnelle, l'hamiltonien s'écrit [1]

$$H(q,p) = \sum_{i=1}^{N} \frac{1}{2m_i} p_i^\top p_i - \sum_{1\le i<k\le N} G\frac{m_i m_j}{\|q_i - q_k\|}$$

où $q_i \in \mathbb{R}^3$ désigne la position de chaque corps, m_i sa masse, et $p_i = m_i\dot{q}_i \in \mathbb{R}^3$ sa quantité de mouvement. Le comportement de chaque corps est alors régi par

1. Pour obtenir l'hamiltonien, on commence par définir le lagrangien $L(t,q,\dot{q})$, puis la quantité de mouvement $p = \nabla_{\dot{q}} L(t,q,\dot{q})$, et enfin l'hamiltonien $H(t,q,p)$ est obtenu par transformée de Legendre. Notons que dans ce cas général où H peut dépendre explicitement du temps (par exemple si de l'énergie est injectée ou prélevée par une action extérieure au système), on a $\frac{d}{dt}H(t,q(t),p(t)) = \nabla_t H(t,q(t),p(t))$, donc l'hamiltonien varie selon cet effet extérieur, et n'est plus constant.

la dynamique hamiltonienne [2]

$$\dot{q}_i = \nabla_{p_i} H(q,p) = \frac{1}{m_i} p_i$$
$$\dot{p}_i = -\nabla_{q_i} H(q,p) = -G \sum_{k \neq i} \frac{m_i m_j}{\|q_i - q_k\|^3} (q_i - q_k)$$

On a alors le long des trajectoires

$$\frac{d}{dt} H(q(t),p(t)) = \langle \nabla_q H(t,q(t),p(t)), \dot{q} \rangle + \langle \nabla_p H(t,q(t),p(t)), \dot{p} \rangle = 0$$

et l'énergie $H(q,p)$ est donc conservée.

Or, lorsqu'on essaye de simuler le système solaire avec un schéma d'Euler (explicite), l'énergie augmente peu à peu à chaque révolution et les trajectoires sont des spirales divergentes. Avec un schéma d'Euler implicite, Jupiter et Saturne s'effondre vers le soleil et sont éjectées du système solaire ! Même des schémas d'ordre supérieur ne permettent pas de simuler correctement ce système sur des temps "courts" sur l'échelle de temps astronomique (à moins de prendre des pas déraisonnablement petits). En fait, le problème c'est que ces méthodes d'intégration ne préservent pas les propriétés structurelles des solutions telles que la conservation de l'énergie. Il faut donc développer des schémas particuliers, appelés *symplectiques*, comme illustré sur un simple oscillateur dans l'exercice *Schéma symplectique*. Pour aller plus loin sur ces méthodes, voir (Hairer, Lubich, and Wanner 2010).

FIGURE

Méthodes à un pas

Principe

Pour approximer les solutions d'une équation différentielle sur un intervalle $[0, \bar{t}]$, les méthodes numériques à un pas se basent sur la représentation intégrale

$$x(t) = x_0 + \int_{t_0}^{t} f(s, x(s)) ds = x_0 + \sum_{j=0}^{J-1} \int_{t_j}^{t_{j+1}} f(s, x(s)) ds$$

où $t_0 < t_1 < \ldots < t_J$ avec $t_J = \bar{t}$. L'idée est d'approximer les intégrales $\int_{t_j}^{t_{j+1}} f(s, x(s))$ sur des intervalles $[t_j, t_{j+1}]$ suffisamment petits.

2. L'application des lois de Newton donnerait directement

$$m_i a_i = m_i \ddot{q}_i = \sum_{k \neq i} F_k = -G \sum_{k \neq i} \frac{m_i m_j}{\|q_i - q_k\|^2} \frac{(q_i - q_k)}{\|q_i - q_k\|}$$

où F_k sont les forces de gravitation exercées par chaque corps k sur le corps i.

Dans la suite, on note x^j l'approximation au temps t_j de la valeur exacte $x(t_j)$ et $\Delta t_j = t_{j+1} - t_j$ le $j$ème pas de temps. L'idée est de calculer récursivement

$$x^{j+1} = x^j + \Delta t_j \Phi_{\Delta t_j}(t_j, x^j)$$

où $\Phi_{\Delta t_j}(t_j, x^j)$ doit donc approximer

$$\frac{1}{t_{j+1} - t_j} \int_{t_j}^{t_{j+1}} f(s, x(s)) \ .$$

Les différentes méthodes de quadrature, i.e. d'approximation de l'intégrale, peuvent donc être mises à profit. La difficulté ici est que seule la valeur initiale $f(t_j, x(t_j))$ de f est connue (ou du moins estimée) à l'itération j, par $f(t_j, x^j)$. On distingue donc les méthodes *explicites* où $\Phi_{\Delta t_j}(t_j, x^j)$ est écrite directement explicitement en fonction de la valeur initiale x^j, et les méthodes *implicites* où cette expression n'est connue qu'implicitement et des étapes intermédiaires de calcul sont nécessaires.

Exemples

1. Méthodes explicites:

— Euler explicite: l'intégrale est approximée par l'aire d'un rectangle déterminé par la valeur initiale de f à *gauche de l'intervalle*, i.e.

$$x^{j+1} = x^j + \Delta t_j \, f(t_j, x^j) \ .$$

— méthode de Heun : l'intégrale est approximée par l'aire d'un trapèze déterminé par la valeur initiale de f et une approximation de sa valeur finale, i.e.,

$$x^{j+1} = x^j + \frac{\Delta t_j}{2} \Big(f(t_j, x^j) + f\big(t_{j+1}, x^j + \Delta t_j f(t_j, x^j)\big) \Big) \ .$$

— schéma de Runge–Kutta d'ordre 4:

$$\begin{cases} F_1 = f(t_j, x^j) \\ F_2 = f\left(t_j + \dfrac{\Delta t_j}{2}, x^j + \dfrac{\Delta t_j}{2} F_1\right) \\ F_3 = f\left(t_j + \dfrac{\Delta t_j}{2}, x^j + \dfrac{\Delta t_j}{2} F_2\right) \\ F_4 = f(t_j + \Delta t_j, x^j + \Delta t_j F_3), \end{cases}$$

et on pose

$$x^{j+1} = x^j + \Delta t \, \frac{F_1 + 2F_2 + 2F_3 + F_4}{6} \ .$$

2. Méthodes implicites:

— Euler implicite : l'intégrale est approximée par l'aire d'un rectangle déterminé par la valeur finale de f à *droite de l'intervalle*, i.e.

$$x^{j+1} = x^j + \Delta t_j \, f(t_{j+1}, x^{j+1}) \ .$$

— méthode des trapèzes (ou Crank–Nicolson) : l'intégrale est approximée par l'aire du trapèze déterminé par les valeurs initiales et finales de f, i.e.

$$x^{j+1} = x^j + \frac{\Delta t_j}{2} \Big(f(t_j, x^j) + f(t_{j+1}, x^{j+1}) \Big) \ .$$

— méthode du point milieu : l'intégrale est approximée par l'aire d'un rectangle déterminé par une approximation de la valeur de f au milieu de l'intervale, i.e.

$$x^{j+1} = x^j + \Delta t_j f \left(\frac{t_j + t_{j+1}}{2}, \frac{x^j + x^{j+1}}{2} \right) \ .$$

On peut bien sûr construire des méthodes plus compliquées et plus précises pour des méthodes de Runge–Kutta d'ordre supérieur (explicites ou implicites).

Définition implicite de Φ

Dans les schémas implicites, l'application $\Phi_{\Delta t}$ est définie de manière implicite. Par exemple, pour le schéma d'Euler, on a :

$$\Phi_{\Delta t_j}(t_j, x^j) = f\Big(t_j + \Delta t_j, x^j + \Delta t_j \Phi_{\Delta t_j}(t_j, x^j)\Big).$$

Il faut donc s'assurer que Φ est bien définie, c'est-à-dire qu'il existe bien x^{j+1} tel que

$$x^{j+1} = x^j + \Delta t_j \, f(t_{j+1}, x^{j+1}) \ .$$

Pour cela, nous pouvons voir x^{j+1} comme le point fixe de l'application F_j définie par

$$F_j(x) = x^j + \Delta t_j \, f(t_{j+1}, x) \ .$$

à x^j, Δt_j, t_{j+1} fixés. L'existence (et l'unicité) de ce point fixe peut alors être démontrée par le théorème de point fixe de Banach. Si $x \mapsto f(t_{j+1}, x)$ est Lipschitzienne, c'est-à-dire s'il existe L_j tel que

$$|f(t_{j+1}, x_a) - f(t_{j+1}, x_b)| \leq L_j |x_a - x_b| \qquad \forall (x_a, x_b) \in \mathbb{R}^n \times \mathbb{R}^n \ ,$$

alors $F_j : \mathbb{R}^n \to \mathbb{R}^n$ est contractante pour un pas de temps Δt_j suffisamment petit puisque

$$|F_j(x_a) - F_j(x_b)| \leq \Delta t_j L_j |x_a - x_b| \ .$$

Puisque $\mathbb{R}^n$ est complet, on déduit par le théorème du point fixe que x^{j+1} existe bien.

En pratique, on peut utiliser la méthode itérative de construction de ce point fixe donnée par la preuve du théorème pour approcher x^{j+1}. Une stratégie est de partir de la valeur donnée par le schéma d'Euler explicite

$$x^{j,0} = x^j + \Delta t_j f(t_j, x^j)$$

et affiner ensuite par l'algorithme du point fixe en itérant

$$x^{j,k+1} = F(x^{j,k})$$

jusqu'à ce que l'évolution relative

$$\frac{x^{j,k+1} - x^{j,k}}{x^{j,0}}$$

devienne inférieure à un seuil choisi par l'utilisateur. Puisque la suite $(x^{j,k})_{k\in\mathbb{N}}$ est de Cauchy, on sait que cette algorithme s'arrête en un nombre fini d'itérations.

Un tel schéma est plus lourd en terme de calculs qu'un algorithme explicite mais il apporte en général plus de stabilité et permet souvent d'utiliser un pas plus grand. C'est en particulier utile pour les systèmes raides, comme illustré dans l'exercice *Explicite ou Implicite?*.

Analyse d'erreur

L'objectif de l'analyse d'erreur *a priori* est de donner une estimation de l'erreur commise par la méthode numérique en fonction des paramètres du problème (temps d'intégration, pas de temps, propriétés de f). L'idée générale est de remarquer qu'à chaque pas de temps, on commet une erreur d'intégration locale (erreur de troncature dans la discrétisation de l'intégrale, à laquelle s'ajoutent souvent des erreurs d'arrondi), et que ces erreurs locales s'accumulent. Le contrôle de cette accumulation demande l'introduction d'une notion de stabilité adéquate, alors que les erreurs locales sont liées à une notion de consistance. L'alliance de stabilité et de consistante donne une propriété de convergence qui est souhaitée lors de l'implémentation de méthodes numériques.

Erreur de troncature locale

L'erreur de troncature locale à l'itération j est l'erreur résiduelle que l'on obtiendrait si l'on appliquait le schéma numérique à la solution exacte $x(t_j)$. En d'autres termes, c'est l'erreur due à l'approximation de l'intégrale. Elle est ainsi définie comme

$$\eta^{j+1} := \frac{x(t_{j+1}) - x(t_j) - \Delta t_j \Phi_{\Delta t_j}(t_j, x(t_j))}{\Delta t_j}.$$

Consistance

On note $\Delta t = \max_{0\leq j\leq J-1} \Delta t_j$ le pas de temps maximal. On dit qu'une méthode numérique est *consistante* si

$$\lim_{\Delta t\to 0} \left(\max_{1\leq j\leq J} |\eta^j| \right) = 0 \ ,$$

et qu'elle est *consistante d'ordre* p s'il existe une constante c_s telle que, pour tout $0 \leq j \leq J-1$,

$$|\eta^{j+1}| \leq c_s \, (\Delta t_j)^p \ .$$

Condition suffisante

Si $(\Delta t, t, x) \mapsto \Phi_{\Delta t}(t,x)$ est continue et telle que

$$\Phi_0(t,x) = f(t,x) \qquad \forall (t,x) \in \left[0,\bar{t}\right] \times \mathbb{R}^n$$

alors le schéma est consistant.

Démonstration Soit C un ensemble compact tel que $x(t) \in C$ pour tout $t \in \left[0,\bar{t}\right]$. On note toujours $\Delta t = \max_{0\leq j\leq J-1} \Delta t_j$. Par la représentation intégrale des solutions,

$$x(t_{j+1}) = x(t_j) + \int_{t_j}^{t_{j+1}} f(x(s),s)ds$$

l'erreur de troncature locale s'écrit

$$\eta^{j+1} = \frac{1}{\Delta t_j} \int_{t_j}^{t_{j+1}} \Big(f(s,x(s)) - \Phi_{\Delta t_j}(t_j, x(t_j)) \Big) ds \ .$$

On doit montrer que ces erreurs tendent vers 0 lorsque Δt tend vers 0 (uniformément en $j = 1, \ldots, N$). Soit $\varepsilon > 0$. Par la continuité de Φ et f et puisque $\Phi_0 = f$, il existe $\Delta_1 > 0$ tel que si $\Delta t \leq \Delta_1$, alors

$$|\Phi_{\Delta t_j}(t,x) - f(t,x)| \leq \varepsilon \qquad \forall (t,x) \in \left[0,\bar{t}\right] \times C \quad \forall j = 1, \ldots, N \ .$$

Donc

$$|\eta^{j+1}| \leq \varepsilon + \frac{1}{\Delta t_j} \int_{t_j}^{t_{j+1}} \big| f(s,x(s)) - f(t_j, x(t_j)) \big| ds \ .$$

Puisque $s \mapsto f(s,x(s))$ est continue sur le compact $\left[0,\bar{t}\right]$, elle y est uniformément continue, donc il existe $\Delta_2 > 0$ tel que si $\Delta t \leq \Delta_2$,

$$\big| f(s,x(s)) - f(t_j, x(t_j)) \big| \leq \varepsilon \qquad \forall s \in [t_j, t_{j+1}] \quad \forall j = 1, \ldots, N$$

et donc $|\eta^{j+1}| \leq 2\varepsilon$ pour tout j. Le schéma est donc bien consistant. ■

Exemples

Reprenons les exemples donnés plus haut.

- Euler explicite : $\Phi_{\Delta t_j}(t,x) = f(t,x)$ indépendamment de Δt_j donc la condition est trivialement satisfaite.
- Méthode de Heun : $\Phi_{\Delta t_j}(t,x) = \frac{f(t,x)+f(t,x+\Delta t f(t,x))}{2}$ donne bien $f(t,x)$ si $\Delta t = 0$.
- Runge Kutta d'ordre 4 : lorsque $\Delta t = 0$, $F_1 = F_2 = F_3 = F_4 = f(t,x)$ donc $\Phi_{\Delta t_j}(t,x) = \frac{F_1+2F_2+2F_3+F_4}{6} = f(t,x)$.

De même, la consistance des méthodes implicites s'obtiennent en remarquant que $x^{j+1} = x^j$ lorsque $\Delta t = 0$.

Cette condition suffisante permet donc de prouver facilement le caractère consistant d'un schéma. Cependant, en pratique, on s'intéresse surtout à son ordre de consistance. Pour cela, l'erreur de consistance se calcule souvent par des développements de Taylor des solutions lorsque celles-ci sont suffisamment régulières, et la constante c_s s'exprime alors comme une borne sur les dérivées des solutions. En fait, on remarque que lorsque f est continue, la solution est C^1 (par définition de nos solutions). Mais puisque $\dot{x}(t) = f(t,x(t))$, $\dot{x}$ hérite de la régularité de f: si f est C^k alors les solutions x sont C^{k+1}. Le calcul de l'ordre de consistance dans le cas dus schéma d'Euler explicite est donné ci-dessous. Pour les autres schémas, voir l'exercice *Consistance de schémas*

Ordre de consistance du schéma d'Euler explicite

L'erreur de troncature s'écrit

$$\eta^{j+1} = \frac{x(t_j+\Delta t_j) - \Big(x(t_j) + \Delta t_j\, f(t_j, x(t_j))\Big)}{\Delta t_j}.$$

Or, si f est C^1, alors x est C^2 et par application la formule de Taylor avec reste intégral, on a

$$x(t_j+\Delta t_j) = x(t_j) + \Delta t f(t_j, x(t_j)) + \Delta t_j^2 \int_0^1 \ddot{x}(t_j + s\Delta t_j)(1-s)ds,$$

en utilisant $\dot{x}(t_j) = f(t_j, x(t_j))$. Ceci donne donc

$$|\eta^{j+1}| \leq \Delta t_j \int_0^1 \ddot{x}(t_j+s\Delta t_j)(1-s)ds \leq \frac{\Delta t_j}{2} \max_{t\in[t_j,t_{j+1}]} \|\ddot{x}(t)\| \leq \frac{\Delta t_j}{2} \max_{t\in[0,T]} \|\ddot{x}(t)\|.$$

Le schéma d'Euler explicite est donc consistant d'ordre 1 avec

$$c_s = \frac{\max_{t\in[0,T]} \|\ddot{x}(t)\|}{2}.$$

Notons qu'en utilisant $\dot{x}(t) = f(t, x(t))$,

$$\ddot{x}(t) = \partial_t f(t, x(t)) + \partial_x f(t, x(t)) \cdot f(t, x(t)),$$

et on peut exprimer c_s en fonction de bornes sur x et sur les dérivées de f.

Stabilité

La notion de stabilité quantifie la robustesse de l'approximation numérique par rapport à l'accumulation des erreurs locales et perturbations.

Définition

On dit qu'une méthode numérique est *stable* s'il existe une constante $S(T) > 0$ (indépendente des Δt_j) telle que, pour toutes suites $x = \{x^j\}_{1 \leq j \leq J}$ et $z = \{z^j\}_{1 \leq j \leq J}$ vérifiant

$$\begin{cases} x^{j+1} = x^j + \Delta t_j \Phi_{\Delta t_j}(t_j, x^j), \\ z^{j+1} = z^j + \Delta t_j \Phi_{\Delta t_j}(t_j, z^j) + \delta^{j+1}, \end{cases}$$

on ait

$$\max_{1 \leq j \leq J} |x^j - z^j| \leq S(T) \left(|x^0 - z^0| + \sum_{j=1}^{J} |\delta^j| \right).$$

Condition suffisante

Si les $\Phi_{\Delta t_j}$ sont Lipschitziennes en x, c'est-à-dire il existe $L > 0$ tel que pour tout $0 \leq j \leq J$,

$$|\Phi_{\Delta t_j}(t_j, x_a) - \Phi_{\Delta t_j}(t_j, x_b)| \leq L|x_a - x_b| \qquad \forall (x_a, x_b) \in \mathbb{R}^n \times \mathbb{R}^n$$

alors le schéma est stable avec $S(T) = e^{LT}$.

Démonstration On a alors

$$|x^{j+1} - z^{j+1}| \leq |\delta^{j+1}| + (1 + \Delta t_j L)|x^j - z^j| \leq |\delta^{j+1}| + e^{\Delta t_j L}|x^j - z^j|$$

puisque $1 + x \leq e^x$ pour tout $x \in \mathbb{R}$. Par récurrence, on montre alors que pour tout $1 \leq j \leq J$,

$$|x^j - z^j| \leq e^{(t_j - t_0)L}|x^0 - z^0| + \sum_{k=1}^{j} e^{(t_j - t_k)L}|\delta^k| \ .$$

Il s'ensuit que

$$|x^j - z^j| \leq e^{TL} \left(|x^0 - z^0| + \sum_{k=1}^{j} |\delta^k| \right) ,$$

ce qui donne le résultat. ■

Convergence

La combinaison de consistance et de stabilité donne une propriété dite de *convergence* qui dit que l'erreur commise par le schéma par rapport à la vraie solution converge vers 0 lorsque le pas de temps converge vers 0. C'est une propriété cruciale pour un schéma numérique.

Définition

Soit $\Delta t = \max_{0 \leq j \leq J-1} \Delta t_j$. Un schéma numérique est *convergent* si

$$\lim_{\Delta t \to 0} \max_{1 \leq j \leq J} |x^j - x(t_j)| = 0$$

lorsque $x^0 = x(t_0)$. S'il existe $p \in \mathbb{N}_{>0}$ et $c_v > 0$ (indépendent de Δt) tel que

$$\max_{1 \leq j \leq J} |x^j - x(t_j)| \leq c_v (\Delta t)^p$$

on dit que le schéma est *convergent à l'ordre p*.

Théorème de Lax

Une méthode stable et consistante (à l'ordre p) est convergente (à l'ordre p).

Démonstration Notons $z^j = x(t_j)$. On remarque que

$$z^{j+1} = z^j + \Delta t_j \Phi_{\Delta t_j}(t_j, z^j) + \Delta t_j \, \eta^{j+1},$$

où η est l'erreur de consistance. D'après la propriété de stabilité, on a donc

$$|x^j - x(t_j)| \leq S(T) \sum_{j=1}^{J} \Delta t_{j-1} \, |\eta^j| \, ,$$

et par consistance

$$|x^j - x(t_j)| \leq S(T) \, c_s \sum_{j=1}^{J} \Delta t_{j-1} \, (\Delta t_{j-1})^p \leq c_s \, S(T) \, T \, (\Delta t)^p \, .$$

■

Condition suffisante de convergence

L'inconvénient du théorème de Lax est qu'il faut prouver la stabilité pour obtenir la convergence. Or la seule condition suffisante dont nous disposions à cet effet, est le caractère globalement Lipschitzien de $x \mapsto \Phi_{\Delta t}(t,x)$. Mais il s'agit d'une condition très forte. En fait, il est possible de prouver la convergence sous la condition plus faible que $x \mapsto \Phi_{\Delta t}(t,x)$ est "localement Lipschitzienne" :

Si

1. le schéma est consistant d'ordre p,
2. pour tout boule fermée B de $\mathbb{R}^n$, il existe $L > 0$, $\Delta t_m > 0$ tels que pour tout $t \in [0,T]$ et pour tout $\Delta t \in [0, \Delta t_m]$,

$$|\Phi_{\Delta t_j}(t_j, x_a) - \Phi_{\Delta t_j}(t_j, x_b)| \leq L|x_a - x_b| \qquad \forall (x_a, x_b) \in B \times B$$

Alors il existe un pas de temps maximal $\Delta t_{\max} > 0$ tel que le schéma est convergent d'ordre p.

L'hypothèse 2. est en particulier vérifiée si $x \mapsto \Phi_{\Delta t}(t,x)$ est C^1 d'après une version un peu plus générale du théorème des accroissement finis.

Erreurs d'arrondi et pas optimal

A chaque itération, lorsque la machine calcule x^{j+1}, elle commet des erreurs d'arrondi de l'ordre de la précision machine. La solution obtenue est donc donnée par

$$\hat{x}^{j+1} = \hat{x}^j + \Delta t_j \left(\Phi_{\Delta t_j}(t_j, \hat{x}^j) + \rho^{j+1} \right) + \varepsilon^{j+1}$$

au lieu de

$$x^{j+1} = x^j + \Delta t_j \Phi_{\Delta t_j}(t_j, x^j) \ ,$$

où ρ modélise l'erreur commise sur le calcul de $\Phi_{\Delta t_j}$ et ϵ l'erreur sur l'addition finale. La stabilité nous donne alors l'écart

$$\max_{0 \leq j \leq J} |x^j - \hat{x}^j| \leq S(T) \sum_{j=1}^{J} \Delta t_{j-1} |\rho^j| + |\varepsilon^j| \ .$$

En considérant une borne ε des ε^j et ρ des ρ^j, on obtient

$$\max_{0 \leq j \leq J} |x^j - \hat{x}^j| \leq S(T)(T\rho + J\varepsilon) \leq S(T)T \left(\rho + \frac{\varepsilon}{\min \Delta t_j} \right) \ ,$$

et donc finalement, en supposant l'algorithme convergent d'ordre p,

$$\begin{aligned} \max_{0 \leq j \leq J} |x(t_j) - \hat{x}^j| &\leq \max_{0 \leq j \leq J} |x(t_j) - x^j| + |x^j - \hat{x}^j| \\ &\leq c_v (\max_j \Delta t_j)^p + S(T)T \left(\rho + \frac{\varepsilon}{\min_j \Delta t_j} \right) \ . \end{aligned}$$

Les paramètres ε et ρ sont typiquement petits de l'ordre d'un facteur de la précision machine. Cependant, on voit que plus le pas de temps décroit, plus il y a d'itérations et plus les erreurs d'arrondi se propagent. D'un autre côté, plus il augmente, plus les erreurs de quadrature augmentent. En supposant le pas constant, il y a donc un pas "optimal" donné par

$$\Delta t_{opt} = \left(\frac{S(T) T \varepsilon}{c_v p} \right)^{\frac{1}{p+1}} .$$

Projet numérique : adaptation du pas

Jusqu'à présent, on a présenté des schémas dépendant de pas de temps Δt_j, sans jamais dire comment les choisir. Le plus simple est de choisir un pas Δt fixe mais il est difficile de savoir à l'avance quel pas est nécessaire. En particulier, comment savoir si la solution obtenue est suffisamment précise, sans connaître la vraie ?

Une voie empirique est de fixer un pas, lancer la simulation, puis fixer un pas plus petit, relancer la simulation, jusqu'à ce que les resultats *ne semble plus changer* (au sens de ce qui nous intéresse d'observer). Notons que la connaissance des constantes de temps présentes dans le système peut aider à fixer un premier ordre de grandeur du pas. On pourrait aussi directement choisir le pas Δt_{opt} obtenu plus haut en prenant en compte les erreurs d'arrondis. Mais les constantes c_v et $S(T)$ sont souvent mal connues et conservatives.

Consigne Coder une fonction

```
def solve_euler_explicit(f,x0,dt):
  ...
  return t, x
```

prenant en entrée une fonction f, une condition initiale x_0 et un pas de temps dt, et renvoyant le vecteur des temps t^j et de la solution x^j du schéma d'Euler explicite appliqué à $\dot{x} = f(x)$. Tester les performances de votre solver sur une équation différentielle que vous savez résoudre. On pourra par exemple illustrer la convergence du schéma à l'ordre 1.

Bonus Faire de même et comparer la convergence avec un schéma d'ordre 2 de votre choix.

Cette méthode à pas fixe exploite la convergence des schémas, mais

- on ne peut pas prendre un pas de temps arbitrairement petit car on est contraint par le temps de simulation.
- on n'a aucune idée de l'erreur commise et on n'est jamais sûr d'avoir la bonne solution.

— l'utilisation d'un pas très petit peut n'être nécessaire qu'autour de certains points *sensibles* (proches de singularités par exemple) et consomme des ressources inutiles ailleurs.

L'idée serait donc plutôt d'adapter la valeur du pas Δt_j à chaque itération. En d'autres termes, on se fixe une tolérance d'erreur que l'on juge acceptable et on modifie le pas de temps en ligne, selon si l'on estime être au-dessus ou en-dessous du seuil d'erreur. Mais cela suppose d'avoir une idée de l'erreur commise... Il existe justement des moyens de l'estimer.

Tout d'abord, de quelle erreur parle-t-on ?

— erreur *globale* ? L'idéal serait de contrôler $\max_{0\leq j\leq N} |x^j - x(t_j)|$. Or la stabilité nous dit que

$$\max_{0\leq j\leq N} |x^j - x(t_j)| \leq S(T) \sum_{j=1}^{J} \Delta t_{j-1} \, |\eta^j|$$

avec η^j les erreurs de consistances locales. Donc si on se fixe une tolérance sur l'erreur globale $\texttt{Tol}_g$, on a

$$|\eta^j| \leq \frac{\texttt{Tol}_g}{TS(T)} \qquad \Longrightarrow \qquad \max_{0\leq j\leq N} |x^j - x(t_j)| \leq \texttt{Tol}_g \ .$$

En d'autre termes, $\texttt{Tol}_g$ nous fixe une erreur maximale *locale* sur η^j, à chaque itération. Notons cependant que cette borne ne prend pas en compte la propagation des erreurs d'arrondis : plus Δt diminue, plus l'erreur globale risque d'augmenter. Ce phénomène devrait donc en toute rigueur aussi nous donner un pas de temps minimal $\Delta t_{\min}$. Notons que tous ces calculs dépendent des constantes c_v et $S(T)$ qui sont souvent mal connues ou très conservatives.

— erreur (absolue) *locale* ? A chaque itération, une erreur locale est commise dûe à l'approximation de l'intégrale. Cette erreur est donnée par

$$e^{j+1} = \left(x^j + \int_{t_j}^{t_{j+1}} f(s, x(s)) ds \right) - x^{j+1}$$

Notons que si on avait $x^j = x(t_j)$, on aurait exactement $e^{j+1} = \Delta t_j \eta^{j+1}$ l'erreur de constistance. On se donne donc une tolérance d'erreur locale

$$|e^{j+1}| \leq \texttt{Tol}_{abs} \ .$$

— erreur *relative* ? Fixer une erreur absolue est parfois trop contraignant et n'a de sens que si les solutions gardent un certain ordre de grandeur. En effet, l'erreur acceptable quand la solution vaut 1000 n'est peut-être pas la même que lorsqu'elle vaut 1. On peut donc plutôt exiger une certaine erreur relative $\texttt{Tol}_{rel}$, i.e.,

$$\frac{|e^{j+1}|}{|x^j|} \leq \texttt{Tol}_{rel} \ .$$

Mais pour cela nous devons trouver un moyen d'estimer l'erreur locale. C'est souvent fait en utilisant une même méthode à deux pas différents (par exemple Δt_j et $\Delta t_j/2$), ou bien en imbriquant des schémas de Runge-Kutta d'ordres différents.

Consigne Montrer que si f est C^1, on a pour un schéma d'Euler explicite

$$|e^{j+1}| = \Delta t \, \frac{\left|f(t_{j+1}, x(t_{j+1})) - f(t_j, x(t_j))\right|}{2} + O(\Delta t_j^3)$$

On peut donc estimer à chaque itération l'erreur commise e^{j+1} et adapter le pas selon si celle-ci est inférieure ou supérieure au seuil de tolérance.

Consigne Montrer que par ailleurs il existe $c > 0$ telle que

$$|e^{j+1}| \leq c\Delta t_j^2 \ .$$

En déduire qu'une possible stratégie d'adaptation est de prendre

$$\Delta t_{new} = \Delta t \sqrt{\frac{\texttt{Tol}_{abs}}{|e^{j+1}|}}$$

(éventuellement avec une marge de sécurité)

Consigne Coder une fonction

```
def solve_euler_explicit_variable(f,x0,dtmin,dtmax,Tolabs):
  ...
  return t, x
```

prenant en entrée la fonction f, une condition initiale x_0, des bornes $\Delta t_{\min}$, $\Delta t_{\max}$ sur le pas de temps, une tolérance absolue $\texttt{Tol}_{abs}$ et/ou une tolérance relative $\texttt{Tol}_{rel}$, et renvoyant en sortie le vecteur temps (t_j), et la solution approximée (x^j) correspondante. Lorsque le pas nécessaire est inférieur à $\Delta t_{\min}$ le solver s'arrête avec un message d'erreur. Tester ce solveur et comparer ses performances à Euler pas fixe.

Consigne Comparer à la fonction `solve_ivp` de python.

Exercices

Consistance et ordre de schémas

Montrer que :

1. le schéma d'Euler implicite est consistant d'ordre 1
2. le schéma de Heun est consistant d'ordre 2.
3. le schéma du point milieu est consistant d'ordre 2.

4. la méthode des trapèzes est consistante d'ordre 2.

On supposera le pas suffisamment petit pour que les schémas implicites soient définies.

Explicite ou implicite ?

1. Comparer les performances des schémas d'Euler implicites et explicites à pas fixe dans le cas de $\dot{x} = -\lambda x$, $x(0) = 1$, et $\dot{x} = \lambda x$, $x(0) = 1$, sur un horizon de temps T donné.
2. Lorsqu'on modélise des systèmes chimiques ou biologiques, on obtient souvent des réactions aux constantes de temps très différentes. Vaut-il mieux utiliser un schéma d'Euler implicite ou explicite pour simuler

$$\dot{x} = \begin{pmatrix} -1 & 0 \\ 0 & -\mu \end{pmatrix} x$$

avec $\mu >> 1$?

Euler symplectique

Pour $\omega > 0$ donné, considérons le système

$$\dot{x}_1 \;=\; x_2 \quad , \qquad \dot{x}_2(t) \;=\; -\omega^2 x_1$$

de condition initiale $x(0) \;=\; (1, 0)$. On rappelle que pour une suite de la forme $x^{j+1} = Ax^j$ converge vers 0 si les valeurs propres de A sont à l'intérieur du cercle unité et diverge si au moins une valeur propre est à l'extérieur.

1. Montrer que pour n'importe quel pas Δt fixé, un schéma d'Euler explicite donne une solution divergente, et un schéma d'Euler implicite donne une solution qui converge vers 0. Lequel a raison ?

On définit maintenant le schéma suivant qui "mélange" les schémas d'Euler implicites et explicites :

$$\begin{aligned} x_1^{j+1} &= x_1^j + \Delta t\, x_2^j \\ x_2^{j+1} &= x_2^j - \Delta t\, \omega^2 x_1^{j+1} \end{aligned}$$

2. Montrer que la quantité $\omega^2 x_1^2 + x_2^2 + \Delta t\, \omega^2 x_1 x_2$ est conservée. Quelle est alors la forme des solutions obtenues dans le plan de phase si $\omega \Delta t < 2$? En déduire la pertinence de ce schéma. On parle de schéma *symplectique*, car il conserve les volumes.
3. En écrivant le schéma sous la forme $x^{j+1} = Ax^j$, montrer qu'il diverge par contre si $\Delta t\, \omega > 2$.

4. Plus généralement, proposer un schéma pour simuler un système Hamiltonien du type

$$\dot{q} = \nabla_p H(q,p) = \nabla T(p)$$
$$\dot{p} = -\nabla_q H(q,p) = -\nabla V(q)$$

où $(q,p) \in \mathbb{R}^N \times \mathbb{R}^N$ sont les positions généralisées et quantités de mouvement, H est le Hamiltonien que l'on pourra vérifier être conservé le long des trajectoires.

A noter que les conclusions de cet exercice sont les mêmes si on avait utilisé un Euler implicite sur la première composante et un Euler explicite sur la deuxième. Ces deux schémas s'appellent respectivement Euler symplectique A et B.

Corrections

Consistance de schémas

A FAIRE

Explicite ou implicite ?

Prenons d'abord $\dot{x} = -\lambda x$, $x(0) = 1$, dont la solution exacte est $x(t) = e^{-\lambda t}$.

Le schéma d'Euler explicite donne

$$x^{j+1} = x^j - \lambda \Delta t x^j = (1 - \lambda \Delta t)^j$$

soit

$$x^J = (1 - \lambda \Delta t)^J = (1 - \lambda \Delta t)^{\frac{T}{\Delta t}} = \left((1 - \lambda \Delta t)^{\frac{T}{\lambda \Delta t}}\right)^{\lambda} .$$

On a bien

$$\lim_{\Delta t \to 0} x^J = e^{-\lambda T} .$$

Cependant, il faut $|1 - \lambda \Delta t| < 1$ pour que la solution converge au moins vers 0. Sinon, pour $\lambda \Delta t = 2$, $x^J = (-1)^J$, qui n'a rien à voir avec la solution. Pire, pour $\lambda \Delta t = 2$, l'algorithme diverge. Il faut donc adapter Δt à la constante de temps λ du système. Ceci peut poser problème lorsque l'on simule des systèmes sur des temps longs (par rapport à λ)

De l'autre côté, le schéma d'Euler implicite donne

$$x^{j+1} = x^j - \lambda \Delta t x^{j+1}$$

soit

$$x^J = \frac{1}{(1 + \lambda \Delta t)^J} = \frac{1}{(1 + \lambda \Delta t)^{\frac{T}{\Delta t}}}$$

qui tend vers 0 quelque soit le pas Δt ! On parle de stabilité inconditionnelle. Ceci est très pratique pour des simulations sur temps longs, où la condition $\lambda \Delta t < 1$ est trop contraignante.

Prenons maintenant $\dot{x} = \lambda x$, $x(0) = 1$, dont la solution exacte est $x(t) = e^{\lambda t}$. Cette fois-ci, Euler explicite donne

$$x^J = (1 + \lambda \Delta t)^{\frac{T}{\Delta t}}$$

qui fait maintenant sens même pour des pas grands. Par contre, Euler implicite donne

$$x^J = \frac{1}{(1 - \lambda \Delta t)^{\frac{T}{\Delta t}}}$$

qui n'est pas défini pour $\lambda \Delta t = 1$ et qui explose pour des valeurs proche de 1.

Maintenant, lorsque l'on a deux dynamiques asymptotiquement stables aux constantes de temps très différentes la condition de stabilité de Euler explicite exige de choisir un pas câlé sur la plus petite constante de temps, i.e. il faut $\Delta t < \frac{1}{mu}$. Ceci est très exigeant car il faut attendre un nombre d'itérations de l'ordre de μ pour voir l'évolution du système lent. Par contre, une méthode implicite permet de choisir librement le pas de temps en fonction des performances souhaitées.

Euler symplectique

1. Dans le cas d'Euler explicite, $x^{j+1} = Ax^j$ avec

$$A = \begin{pmatrix} 1 & \Delta t \\ -\Delta t\, \omega^2 & 1 \end{pmatrix}$$

   dont les valeur propres sont $1 \pm i\omega\Delta t$ de norme $\sqrt{1 + \Delta t^2\, \omega^2} > 1$. Donc les solutions divergent.

Dans le cas d'Euler implicite, $x^{j+1} = Ax^j$ avec

$$A = \frac{1}{1 + \Delta t^2\, \omega^2} \begin{pmatrix} 1 & \Delta t \\ -\Delta t\, \omega^2 & 1 \end{pmatrix}$$

dont les valeurs propres sont $1/(1 \pm i\omega\Delta t)$ de norme $1/\sqrt{1 + \Delta t^2\, \omega^2} < 1$. Donc les solutions convergent vers 0.

Or on peut vérifier que le long des vraies solutions, l'énergie $\omega^2 x_1^2 + x_2^2$ est constante donc les trajectoires sont bornées et ne peuvent pas converger vers zéro. Aucun des deux schémas n'approxime les solutions correctement sur le long-terme.

2. On vérifie par le calcul que

$$\omega^2 x_1^2 + x_2^2 + \Delta t\, \omega^2 x_1 x_2 = x^\top \begin{pmatrix} \omega^2 & \frac{\Delta t\, w^2}{2} \\ \frac{\Delta t\, w^2}{2} & 1 \end{pmatrix} x$$

est constante. Pour $\omega^2 - \frac{\Delta t^2 \omega^4}{4} > 0$, soit $\omega \Delta t < 2$, cette matrice est définie positive, donc les solutions restent sur une ellipse. Cette ellipse se rapproche de la vraie solution lorsque Δt tend vers 0. Ce schéma est donc approprié pour simuler les trajectoires sur un temps long.

3. L'algorithme symplectique est décrit par $x^{j+1} = Ax^j$ avec

$$A = \begin{pmatrix} 1 & \Delta t \\ -\Delta t\, \omega^2 & 1 - \Delta t^2\, \omega^2 \end{pmatrix}$$

dont le polynôme caractéristique s'écrit

$$s^2 - (2 - \Delta t^2\, \omega^2)s + 1$$

On a les cas suivants :

— si $(1-\Delta t^2\, \omega^2)^2 - 4 < 0$, i.e., si $\omega \Delta t < 2$, les valeurs propres sont imaginaires conjuguées et de module 1.
— si $\omega \Delta t > 2$, les valeurs propres sont réelles de produit 1, donc l'une est supérieure à 1 est le schéma diverge.
— dans le cas extrême où $\omega \Delta t = 2$, il y a une valeur propre double en -1.

4. Pour un système hamiltonien, on peut donc proposer

$$q^{j+1} = q^j + \Delta t\, \nabla T(p^j)$$
$$p^{j+1} = p^j - \Delta t\, \nabla V(q^{j+1})$$

ou bien

$$q^{j+1} = q^j + \Delta t\, \nabla T(p^{j+1})$$
$$p^{j+1} = p^j - \Delta t\, \nabla V(q^j)$$

pour Δt suffisamment petit.

Références

Demailly, J.-P. 2006. *Analyse Numérique et équations Différentielles*. EDP Sciences. Grenoble Sciences.

Hairer, E., C. Lubich, and G. Wanner. 2010. *Geometric Numerical Integration : Structure-Preserving Algorithms for Ordinary Differential Equations*. Edited by Springer Series in Computational Mathematics. 2nd ed. Springer-Verlag, Berlin.

Hairer, E., and G. Wanner. 1996. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Edited by Springer Series in Computational Mathematics. 2nd ed. Springer-Verlag, Berlin.