Résolution numérique d'équations différentielles et sensibilisation aux erreurs numériques.

1 Positionnement du problème

Dans ce TD, nous étudions des méthodes numériques de résolution approchée des équations différentielles du premier ordre. *Les équations différentielles ne sont pas nécessairement linéaires.*

On écrit l'équation différentielle (E) sous la forme

$$\forall t \geq 0, \quad y'(t) = F\big(y(t), t\big) \qquad \text{avec} \qquad y(0) = y_0$$

On suppose que l'équation différentielle régit l'évolution d'un système physique au cours du temps. y_0 correspond à l'état initial du système au temps $t = 0$ et on cherche à connaître l'état du système sur l'intervalle $[0, t_{\max}]$.

F est une fonction de deux variables, que l'on suppose *suffisamment régulière* pour que notre équation ait une unique solution.

Quelques exemples :

- avec l'équation linéaire $y' = ay + b$, la fonction F est définie par $F : (y,t) \mapsto ay + b$.
- avec coefficient continu : $y'(t) = -2t \times y(t)$, on pose $F : (y,t) \mapsto -2t \times y$.
- avec une équation non linéaire : $F : (y,t) \mapsto 2\sin\big(t \times y\big)$

Principe de discrétisation :

On discrétise l'intervalle $[0, t_{\max}]$ avec un pas de temps constant[1] Δt.

On note $(t_n)_{n\in\mathbb{N}}$ les instants successifs avec $t_0 = 0$ et pour $n \geq 0$, $t_{n+1} = t_n + \Delta t$.

Si on discrétise l'intervalle $[0, t_{\max}]$ avec p pas de temps, on aura $\forall n \in [\![0, p-1]\!]$, $t_n = n\dfrac{t_{\max}}{p-1} = n\Delta t$

2 Méthode d'Euler et instabilité numérique

On approxime la dérivée par le taux de variation entre deux instants successifs.

$$y'(t_n) = \frac{y(t_{n+1}) - y(t_n)}{t_{n+1} - t_n} = \frac{y(t_{n+1}) - y(t_n)}{\Delta t}$$

L'équation différentielle s'écrit alors

$$\frac{y(t_{n+1}) - y(t_n)}{\Delta t} = F\big(y(t_n), t_n\big) \iff y(t_{n+1}) = y(t_n) + \Delta t \times F\big(y(t_n), t_n\big)$$

1) Étude numérique

(a) Programmer une fonction **edl**(`a,b,y0,tmax,p`) qui trace la solution approchée de l'équation différentielle $y' = ay + b$ $(a \neq 0)$ par la méthode d'Euler, avec l'intervalle $[0, \texttt{tmax}]$ découpé en p points équidistants.

(b) Tester cette fonction avec des équations différentielles linéaires simples. Tracer la solution exacte sur le même graphique.

(c) On considère l'équation différentielle $y' = -y$ avec $y_0 = 1$ sur $[0, 30]$. Tracer les graphes avec différentes valeurs de p.

Remarquer que l'on observe trois situations différentes de convergence (ou divergence) en fonction de Δt.

(d) Pour $a = -1$, conjecturer les valeurs limites de Δt entre ces trois situations.

2) Étude mathématique

On s'intéresse à l'équation $y' = ay$ avec $a < 0$ (on pourra poser $\lambda = -a > 0$ pour simplifier les calculs).

(a) Trouver l'expression de $y_n = y(t_n)$ en fonction de y_0, de a, n et de Δt.

[1] Ce n'est pas nécessaire, mais plus simple pour nous.

(b) Démontrer ses conjectures sur la convergence de la méthode en fonction de Δt. Interpréter en terme de *temps caractéristique*.

3) **Étude numérique de l'erreur**

(a) Pour l'équation différentielle $y' = ay$, tracer la valeur de l'erreur $|y_n - z(t_n)|$ en fonction du temps où y désigne la solution approchée et z la solution exacte.

(b) Pour cette même équation, tracer l'évolution du maximum de l'erreur en fonction du pas de temps Δt.

3 Méthode d'Euler implicite

La méthode implicite utilise la valeur en $n+1$ pour calculer la pente (et non plus la valeur en n) :

$$y'(t_{n+1}) = \frac{y(t_{n+1}) - y(t_n)}{t_{n+1} - t_n} = \frac{y(t_{n+1}) - y(t_n)}{\Delta t}$$

L'équation différentielle s'écrit alors

$$\frac{y(t_{n+1}) - y(t_n)}{\Delta t} = F\left(y(t_{n+1}), t_{n+1}\right)$$

L'obtention de $y_{n+1} = y(t_{n+1})$ requiert donc la résolution d'un équation à chaque étape.

1) Programmer une fonction **edlImplicite**(`a,b,y0,tmax,p`) qui trace la solution approchée de l'équation différentielle $y' = ay + b$ par la méthode d'Euler implicite, avec l'intervalle $[0, \texttt{tmax}]$ découpé en p points équidistants.

2) Tester cet algorithme sur les mêmes exemples que la méthode d'Euler. Que conjecture-t-on sur la stabilité numérique de la méthode ?

3) **Étude mathématique :** Démontrer mathématiquement sa conjecture pour $a < 0$.

4) **Étude numérique de l'erreur :** Comme pour la méthode d'Euler, tracer le maximum de l'erreur en fonction de Δt, et comparer avec la méthode d'Euler.

4 Application aux équations non linéaires

1) Programmer une fonction **euler**(`fct, y0, tmax, p`) qui trace la solution approchée de l'équation différentielle définie par la fonction `fct`, entre 0 et `tmax`, avec p points équidistants en utilisant la méthode d'Euler.

2) **Méthode de Runge-Kutta d'ordre 2**

Sauf cas particulier, la méthode implicite devient très lourde car elle nécessite la résolution d'une équation à chaque pas de temps. Nous proposons donc une méthode intermédiaire par prédiction/correction.

On commence par donner une valeur prédictive de y_{n+1} avec la méthode d'Euler

$$\widetilde{y_{n+1}} = y_n + \Delta t F\left(y_n, t_n\right)$$

On calcule ensuite une valeur corrigée de y_{n+1} en prenant la moyenne des dérivées en y_n et en $\widetilde{y_{n+1}}$:

$$y_{n+1} = y_n + \frac{\Delta t}{2}\left(F\left(y_n, t_n\right) + F\left(\widetilde{y_{n+1}}, t_{n+1}\right)\right)$$

(a) Programmer une fonction **eulerRK**(`fct, y0, tmax, p`) qui trace la solution approchée de l'équation différentielle $y' = \texttt{fct}(y, t)$ par la méthode d'Euler modifiée, avec l'intervalle $[0, \texttt{tmax}]$ découpé en p points équidistants.

(b) Comparer avec les méthodes précédentes sur l'équation $y' = -y$

(c) Trouver mathématiquement pour quel pas de temps la méthode converge (avec l'équation $y' = -y$).

(d) Tester avec la méthode d'Euler sur les équations définies par $F : (y, t) \mapsto 2ty$ et $F : (y, t) \mapsto 2\sin(ty)$

(e) Comparer les erreurs entre la méthode d'Euler et la méthode de Runge-Kutta sur l'équation $y' = -2ty$ de solution $y' = \mathrm{e}^{-t^2}$.

Tracer un graphe des erreurs en fonction de Δt : observer que pour Δt suffisamment petit, la méthode de Runge-Kutta est très sensiblement meilleure.

Cela peut être démontré analytiquement, mais nous manquons de quelques outils pour cela. Il existe des méthodes de Runge-Kutta d'ordre supérieur, dont l'ordre 4 qui est davantage utilisé.