Lycée Louis-Le-Grand, Paris 2017/2018
MPSI 4 – Informatique pour tous
N. Carré, A. Troesch

TP n° 11 : Équations différentielles

Exercice 1 – (Schéma d'intégration implicite de Crank-Nicholson)
Résoudre numériquement une équation différentielle $y'(t) = f(y,t)$, où f est une fonction de deux variables à valeur dans $\mathbb{R}$, revient à déterminer un ensemble de réels (y_k) approximant de façon aussi précise que possible les valeurs de la fonction $y(t)$, solution de l'équation différentielle, en un ensemble d'instants t_k.
Les schémas d'intégration étudiés jusqu'à présent étaient des des schémas d'intégration dit *explicites*, où la valeur de y_{k+1} s'exprime simplement en fonction de y_k ($y_{k+1} = y_k + f(y_k,t_k) \times (t_{k+1} - t_k)$ pour la méthode d'Euler, $y_{k+1} = y_k + f(y_k + f(y_k,t_k) \times \frac{t_{k+1}-t_k}{2}, \frac{t_{k+1}+t_k}{2}) \times (t_{k+1} - t_k)$ pour Runge-Kutta à l'ordre 2, etc.)
Certains schémas d'intégration sont dits *implicites*, car la détermination de y_{k+1} repose sur la résolution d'une équation. C'est le cas, par exemple, du schéma d'intégration de *Crank-Nicholson*, où la relation entre y_k et y_{k+1} est gouvernée par la relation

$$y_{k+1} = y_k + \frac{f(y_k,t_k) + f(y_{k+1},t_{k+1})}{2} \times (t_{k+1} - t_k)$$

Le terme recherché y_{k+1} apparaissant également dans le membre de droite de l'expression précédente, il n'est pas toujours aisé de calculer l'ensemble des y_k pour une fonction f quelconque.
On considère l'équation différentielle suivante :

$$y'(t) = f(y,t) \quad \text{avec} \quad f(y,t) = y \times \sin(t) \quad \text{et} \quad y(0) = 1$$

1. Montrer que, dans le cas particulier de l'équation différentielle précédente, il est possible de calculer y_{k+1} en fonction de y_k, t_k et t_{k+1}.
2. Tracer et comparer les solutions obtenues pour $t \in [0,20]$
   - avec le schéma d'intégration d'Euler ;
   - avec le schéma d'intégration de Crank-Nicholson ;
   - avec la fonction `odeint` de `scipy.integrate`

   lorsque l'on effectue deux cent subdivisions de l'intervalle $[0,20]$.
3. Vérifier que le schéma d'intégration de Crank-Nicholson donne d'excellents résultats même avec des pas importants.

Exercice 2 – (Schéma d'intégration adaptatif d'Euler-Richardson)
Bien qu'il existe des méthodes plus précises que le schéma d'intégration de Runge-Kutta à l'ordre 4, elles nécessitent généralement tellement de calculs qu'on leur préfère des schémas d'intégration plus simples, mais en réduisant le pas entre chaque valeur.
Cependant, réduire le pas d'intégration occasionne également davantage de calculs, et il n'est pas indispensable de choisir de petits δt partout : certains intervalles nécessitent d'avancer très prudemment, tandis que d'autres s'accomodent de pas plus grands.
Un schéma d'intégration peut automatiquement ajuster son pas en fonction des besoins. On parle de schémas d'intégration *à pas adaptatif*. On ne fixe plus les t_k à priori, on fournit simplement l'intervalle $[0,T]$ sur lequel on cherche une solution, l'algorithme déterminant lui-même les t_k en même temps que les y_k.
Le schéma d'intégration adaptatif d'Euler-Richardson est basé sur les schéma d'intégration d'Euler et de Runge-Kutta ordre 2. Il adapte son pas h à chaque étape, en comparant $p = f(y_k,t_k)$ (la pente en y_k, utilisée par le schéma d'Euler) et $p' = f(y_k + p \times \dfrac{h}{2}, t_k + \dfrac{h}{2})$ (la pente après un demi-pas d'Euler, utilisée par le schéma de Runge-Kutta à l'ordre 2).
Si ces pentes sont trop différentes, alors le pas h est réduit. Si à l'inverse elles sont très proches, le pas h est augmenté.
Pour intégrer l'équation différentielle $y'(t) = f(y,t)$ sur $[0,T]$, on procède de la façon suivante :

- On choisit un seuil de précision $\epsilon > 0$ et un pas initial $h > 0$.
- On initialise $y_0 = y(0)$ et $t_0 = 0$;
- Puis, pour tout $k \geqslant 0$ tant que $t_k < T$:
  - $*$ on calcule $p = f(y_k, t_k)$ puis $p' = f(y_k + p \times \frac{h}{2}, t_k + \frac{h}{2})$;
  - $*$ on détermine $e = \frac{h}{2}\,|p' - p\,|$;
  - $*$ si $e > \epsilon$, alors on réduit le pas avec $h = 0.95\ h\ \sqrt{\frac{\epsilon}{e}}$, et on revient au calcul de p et p' ;
  - $*$ si $e \leqslant \epsilon$, alors $t_{k+1} = t_k + h$, $y_{k+1} = y_k + p' \times h$, et on ajuste le pas avec $h = 0.95\ h\ \sqrt{\frac{\epsilon}{e}}$.

1. Programmer le schéma d'intégration adaptatif d'Euler-Richardson, l'utiliser pour résoudre l'équation différentielle $y'(t) = f(y, t)$ avec $f(y, t) = y \times \sin(t)$ et $y(0) = 1$ sur l'intervalle $[0, 20]$ et tracer le résultat.
2. Quel ϵ permet d'obtenir un résultat satisfaisant, et combien de subdivisions de l'intervalle $[0, 20]$ ont été nécessaires ?

Exercice 3 – (Ordre et chaos, le papillon de Lorenz)

Dans les années 1960, Edward Lorenz s'intéresse à la météorologie et à la prédiction de l'évolution des conditions atmosphériques. Le système étant trop compliqué à résoudre même numériquement, il utilise des modèles très simplifiés, au nombre desquels un système différentiel approximant le phénomène de convexion de Rayleigh-Bénard avec seulement trois degrés de liberté (intensité de la convexion, différence de température entre les courants ascendants et descendants, et écart du profil de température vertical par rapport à un profil linéare).

Ce système est le suivant, où σ, ρ et β sont trois paramètres réels :

$$\begin{cases} x'(t) = \sigma \times (y(t) - x(t)) \\ y'(t) = \rho \times x(t) - y(t) - x(t) \times z(t) \\ z'(t) = x(t) \times y(t) - \beta \times z(t) \end{cases}$$

Pour certaines valeurs des paramètres, le système a un comportement chaotique. Notamment, lorsque $\sigma = 10$, $\rho = 28$ et $\beta = 8/3$, on voit apparaître un « attracteur étrange » en forme de papillon, qui se manifeste pour pratiquement toutes les conditions initiales (autres que les quelques points fixes du système, tel que l'origine $(0, 0, 0)$).

Bien que visible sur les simulation, l'existence de cet attracteur étrange, conjecturée par E. Lorenz en 1963, n'a pu être démontrée rigoureusement qu'en 2001 par Warwick Tucker.

1. Utiliser la fonction `odeint` pour résoudre le système d'équations différentielles pour $t \in [0, 50]$ (on subdivisera cet intervalle en dix mille) et des conditions initiales x_0, y_0 et z_0 choisies librement dans l'intervalle $[-10, 10]$.
2. Pour tracer le résultat, importer le module `pyplot` ainsi que `Axes3D` du module `mpl_toolkits.mplot3d`, puis utiliser la commande `matplotlib.pyplot.axes(projection='3d').plot(X, Y, Z)` si `X`, `Y` et `Z` sont les listes correspondant aux x_k, y_k et z_k.
3. Choisir deux conditions initiales très proches, et vérifier le caractère chaotique du système, en comparant le résultat obtenu pour chacune des conditions initiales après un temps $T = 50$.