Université Paris-Dauphine
Département MIDO
DE MI2E deuxième année

Méthodes numériques
Année 2018/2019

Projet : Réactions chimiques oscillantes

1 Contexte

Les équations de réactions chimiques s'écrivent en général sous la forme

$$A + B \longrightarrow C + D$$

Les espèces A et B sont appelées les réactifs et elles réagissent pour donner naissance aux espèces C et D, appelées les produits de la réaction. L'étude de processus chimiques complexes fait intervenir très souvent des réactions qui s'enchaînent. Il peut arriver que certaines espèces jouant le rôle de réactifs à une certaine étape soient aussi le produit dans une étape ultérieure. Ce phénomène, appelé auto-catalyse, donne lieu à des changements périodiques dans la concentration de ces espèces. Quelques exemples de ce type de réactions oscillantes, pour lesquelles on pourra cliquer pour voir des vidéos, sont :

- Réaction de Briggs-Rauscher
- Réaction de Beloussov-Jabotinski
- Coeur battant de mercure

Dans ce projet, nous proposons d'illustrer ce phénomène à travers le modèle simple du Brusselator, où les concentrations x et y de deux espèces chimiques évoluent au cours du temps $t \in \mathbb{R}_+$ suivant le système d'équations différentielles

$$\begin{cases} \dot{x}(t) &= A + x^2(t)y(t) - (B+1)x(t) \\ \dot{y}(t) &= Bx(t) - x^2(t)y(t), \end{cases} \tag{1.1}$$

où A et B sont deux réels et $\dot{x}$ et $\dot{y}$ sont les dérivées en temps de x et y. Le système part d'un état initial

$$(x(0), y(0)) = (x_0, y_0) \in \mathbb{R}^2$$

donné. Quand $B \leq 1 + A^2$, le système tend vers une position d'équilibre

$$\lim_{t\to\infty} (x(t), y(t)) = (A, B/A).$$

En revanche, le système est instable pour $B > 1 + A^2$ et donne lieu à des oscillations perpétuelles dans la concentration des deux espèces autour du point $(A, B/A)$.

1. En notant $z = (x, y)$ et $\dot{z} = (\dot{x}, \dot{y})$, montrer que pour $t \in [0, T]$, le système (1.1) s'écrit de façon équivalente comme

$$\begin{cases} \dot{z}(t) &= f(z(t)), \quad \forall t \in [0, T] \\ z(0) &= z_0 = (x_0, y_0) \text{ donné} \end{cases} \tag{1.2}$$

avec $f : \mathbb{R}^2 \to \mathbb{R}^2$ une fonction que l'on explicitera.

2 Résolution avec le schéma Euler explicite

Pour $T > 0$ donné, nous cherchons une approximation numérique de $z(t)$ solution de (1.2) pour tout instant $t \in [0, T]$. Notons $\tilde{z}(t)$ une fonction qui approche $z(t)$ "au mieux possible" en $P + 1$ instants de temps $t_p \in [0, T]$. On considère le cas où les instants sont uniformément répartis sur $[0, T]$ de sorte que, en posant

$$h \coloneqq T/P,$$

on a

$$\begin{aligned} t_0 &= 0 \\ t_{p+1} &= t_p + h, \quad 0 \leq p \leq P-1, \\ t_P &= T. \end{aligned}$$

Le point de départ pour approcher $z(t_p)$ pour $p = 0, \ldots, P$ part du développement de Taylor de $z(t_{p+1})$ autour de t_p qui est

$$z(t_{p+1}) = z(t_p + h) = z(t_p) + h\,\dot{z}(t_p) + \mathcal{O}(h^2) = z(t_p) + h\,f(z(t_p)) + \mathcal{O}(h^2), \quad 0 \leq p \leq P-1,$$

En négligeant les termes d'ordre supérieur à h, la fonction

$$\widetilde{z}(t_{p+1}) := \widetilde{z}(t_p) + hf(\widetilde{z}(t_p)), \quad 0 \leq p \leq P-1,$$

est une approximation de $z(t_{p+1})$ à l'ordre h près. L'algorithme d'Euler explicite consiste à calculer ces approximations $\widetilde{z}(t_{p+1})$ en partant de l'état initial $\widetilde{z}(0) = z(0)$.

Questions : Appliquer la méthode d'Euler explicite pour donner une solution approchée de $z(t)$. Pour cela, on pourra :

1. Implémenter une fonction `f(z)` qui, étant donné le vecteur z, retourne le vecteur $f(z)$. Afin de ne pas introduire les constantes A et B dans l'appel à la fonction `f`, on pourra structurer cette partie du code comme suit :

```
def brusselator(A, B):
    def f(z):
        # Ecrire ici la fonction f
        return ...
    return f
```

   On pourra ensuite évaluer la fonction comme suit dans le reste du code :

```
(A, B) = (·, ·)
f = brusselator(A, B)
z = ...
f(z)
```

   Nous conseillons vivement d'exprimer les vecteurs sous la forme `array` de la librairie `numpy` et pas sous forme de listes.

2. Implémenter une méthode `stepEuler(f, zt, h)` qui, partant de l'approximation `zt` à un certain instant t, renvoie l'approximation `zt+h f(zt)` à l'instant $t+h$ suivant. Nous recommandons ici aussi d'exprimer les vecteurs `zt` sous la forme `array` de la librairie `numpy` et pas avec des listes.
3. En partant de `z0` comme condition initiale, faire une boucle sur tous les temps qui permet d'obtenir l'approximation $\widetilde{z}(t_p)$ à tous les instants $p = 0, \ldots, P$. Stocker la trajectoire (i.e. l'ensemble des approximations `zt` pour tous les instants t_p) dans une liste.
4. Utiliser le code pour obtenir la trajectoire de 0 à T pour

$$T = 18, \quad A = 1,\ B = 3, \quad z_0 = (0,1)^T, \quad P = 1000.$$

   Au vu des valeurs de A et B, prédire le comportement du système.
5. Même question pour $A = 1$ et $B = 1.5$.
6. Visualisation des résultats :
   (a) Construire une fonction `plot-concentration` qui représente graphiquement l'évolution des concentrations $x(t)$ et $y(t)$ en fonction du temps pour $t = t_p,\ p = 0, \ldots, P$. On pourra sauver le graphique avec la commande `plt.savefig('evol-concentration.pdf')`.
   (b) Construire une fonction `plot-trajectoire` qui représente graphiquement la trajectoire (x, y) des concentrations pour tout $t = t_p,\ p = 0, \ldots, P$. Sauver la figure avec le nom `'trajectoire.pdf'`.

   Utiliser ces fonctions pour représenter graphiquement l'évolution des questions 4 et 5.

3 Méthode Runge-Kutta d'ordre 4

Si la fonction $t \to z(t)$ est suffisamment régulière, l'erreur d'approximation avec la méthode d'Euler explicite au temps final $t = T$ est de l'ordre du pas temps h, i.e.,

$$\exists\, C > 0 \text{ s.t. } \|z(T) - \widetilde{z}(T)\| \leq Ch. \tag{3.1}$$

Donc, lorsque $h \to 0$, la fonction $\widetilde{z}(T)$ converge vers $z(T)$. Cependant, dans la pratique la constante C de l'inégalité est souvent très grande et augmente avec T. Pour cette raison, il est en général nécessaire de prendre des pas de temps h extrêmement petits pour arriver à une précision convenable. Cela augmente considérablement le nombre d'instants intermédiaires et rallonge les temps de calcul et motive la recherche d'approximations dont l'erreur est de la forme

$$\exists\, C > 0 \text{ s.t. } \|z(T) - \widetilde{z}(T)\| \leq Ch^r,$$

avec $r > 1$. On parle de schémas d'ordre supérieur. Pour les construire, une première possibilité serait de prendre des développements de Taylor d'ordre supérieur mais cela est en général très coûteux voire parfois même impossible en fonction du problème.

Une façon plus simple de construire des schémas d'ordre supérieur est le schéma Runge-Kutta. Dans cette approche, f est évaluée en un certain nombre de points et l'approximation se construit comme suit. Pour tout $0 \leq p \leq P-1$, on pose

$$\widetilde{z}(t_{p+1}) := \widetilde{z}(t_p) + h \sum_{j=1}^{m} \gamma_j K_j(t_p)$$

avec

$$K_j(t_p) = \begin{cases} f\left(\widetilde{z}(t_p)\right) \text{ si } j = 1 \\ f\left(\widetilde{z}(t_p) + h\sum_{l=1}^{j-1} \beta_{j,l} K_l(t_p)\right) \text{ si } 2 \leq j \leq m. \end{cases} \tag{3.2}$$

Les coefficients $\beta_{j,l}$ et γ_l sont usuellement donnés dans un tableau

α_j	β_{jl}
	γ_l

appelé tableau de Butcher. Les indices j sont des indices de ligne et les indices ℓ des indices de colonne. Le tableau comporte des coefficients α_j dont on n'aura pas besoin dans ce problème. Dans ce type d'approche, étant donné un ordre d'approximation r souhaité, la question est de trouver de bons coefficients $\beta_{j,l}$ et γ_l qui satisfont cet ordre d'approximation r. Afin de réduire le coût des calculs, il faut aussi que m soit le plus petit possible. Les questions tournant autour de la recherche des coefficients $\beta_{j,l}$ et γ_l donnant un certain ordre r et avec le plus petit degré possible m est une question difficile et nous nous contenterons de considérer le tableau suivant

0						
1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	−7200/2197	7296/2197			
1	439/216	−8	3680/513	−845/4104		
1/2	−8/27	2	−3544/2565	1859/4104	−11/40	
	16/135	0	6656/12825	28561/56430	−9/50	2/55

qui donne un schéma d'ordre $r = 4$, que nous appellerons RK4. Notons qu'ici $m = 6$.

Questions :

1. Implémenter une méthode `stepRK4(f, zt, h)` qui, partant de l'approximation `zt` à un certain instant t, renvoie l'approximation à l'instant $t + h$ avec la méthode Runge-Kutta d'ordre 4.
2. Résoudre les deux exemples donnés en question 4 et 5 de la section 2 par cette méthode.
3. Représenter graphiquement l'évolution des concentrations et la trajectoire obtenues.

4 Pas de temps adaptatif : Schéma RK45

En utilisant deux schémas Runge-Kutta d'ordre différents, il est possible de calculer la différence des solutions pour estimer l'erreur d'approximation à chaque pas de temps. Il est possible de garantir que l'erreur ne dépasse pas une certaine valeur $\varepsilon_{\max} > 0$ en adaptant le pas de temps h.

Pour mettre en oeuvre cette idée, nous utiliserons deux schémas de Runge-Kutta construits de sorte à ne différer que dans la valeur des coefficients γ_ℓ : le schéma RK4 d'ordre 4 de la section précédente et un schéma d'ordre 5 dont les coefficients $\bar{\gamma}_\ell$ sont donnés dans la dernière ligne du

tableau de Butcher étendu :

0						
1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	−7200/2197	7296/2197			
1	439/216	−8	3680/513	−845/4104		
1/2	−8/27	2	−3544/2565	1859/4104	−11/40	
	16/135	0	6656/12825	28561/56430	−9/50	2/55
	25/216	0	1408/2565	2197/4104	−1/5	0

Nous noterons $\bar{\gamma}_\ell$ les valeurs du schéma d'ordre 5. En notant $\delta_\ell = \gamma_\ell - \bar{\gamma}_\ell$, l'estimation de l'erreur entre t et $t+h$ s'écrit

$$e(t+h) = \tilde{y}(t+h) - \bar{y}(t+h) = h\sum_{\ell=1}^{m}(\gamma_\ell - \bar{\gamma}_\ell)K^\ell = h\sum_{\ell=1}^{m}\delta_\ell K^\ell.$$

et on pose

$$\varepsilon(t+h) \coloneqq \|e(t+h)\|_\infty.$$

L'algorithme adaptatif marche de la façon suivante. Supposons que l'on aie calculé la solution à l'instant t. On cherche de façon itérative :

1. le pas de temps h_{next} qui garantisse que $\varepsilon(t+h_{\text{next}}) < \varepsilon_{\max}$. Ce pas de temps définit l'instant suivant $t_{\text{next}} = t + h_{\text{next}}$ où la solution $\tilde{z}(t_{\text{next}})$ est calculée avec RK4.
2. le pas de temps h_{init} qui sera utilisé à l'instant suivant pour initialiser l'algorithme d'actualisation des pas de temps définit comme suit :
   — Pour $k = 0$, on fixe h_0 à la valeur h_{init} trouvée à l'instant précédent.
   — Pour $k \geq 0$, on définit un nouveau pas de temps suivant la règle

$$h_{k+1} = h_k \begin{cases} 0.1, & \text{si } c < 0.1 \\ 5, & \text{si } c > 5, \\ c, & \text{sinon} \end{cases} \qquad \text{avec } c = 0.9\left(\frac{\varepsilon_{\max}}{\varepsilon(t+h_k)}\right)^{\frac{1}{5}}. \tag{4.1}$$

   Si $\varepsilon(t+h_k) > \varepsilon_{\max}$, on passe à l'itération suivante. Si $\varepsilon(t+h_k) \leq \varepsilon_{\max}$, alors on arrête les itérations et on pose $h_{\text{next}} = h_k$, $h_{\text{init}} = h_{k+1}$. On calcule alors la solution à $t_{\text{next}} = t + h_{\text{next}}$, et on passe temps suivant $t_{\text{next}} = t + h_{\text{next}}$. Pour traiter ce nouvel instant, on utilisera la valeur de h_{init} obtenue pour initialiser l'algorithme d'actualisation du pas de temps.

Questions :

1. Implémenter la méthode de Runge-Kutta adaptative `stepRK45(f, zt, hinit)` qui, partant de l'approximation `zt` à un certain instant t et d'une valeur h_{init} pour actualiser le pas de temps, renvoie la solution approchée $\tilde{z}(t+h_{\text{next}})$ avec RK4 et la nouvelle valeur h_{init} pour l'instant suivant. On pourra fixer $\varepsilon_{\max} = 10^{-5}$.
2. Résoudre les deux exemples donnés en question 4 et 5 de la section 2 par cette méthode.
3. Représenter graphiquement l'évolution des concentrations et la trajectoire obtenues.
4. Représenter graphiquement la valeur des pas de temps en fonction du temps.