

Projet 6

*Résolution approchée d'équations différentielles /
Modélisation de systèmes dynamiques*

Groupe 1 - Equipe 1 - 12408

Responsable: ndo001

Secrétaire: adsilva008

Codeurs: helfani, mbrassart001, cdieudonn001

Résumé : Le but de ce projet est de maîtriser les différentes méthodes de résolution d'équations différentielles ordinaires (EDO). L'intérêt de ces équations réside dans le fait qu'elles permettent de modéliser des systèmes complexes relativement facilement.

1 Méthodes numériques de résolution d'équations différentielles

Cette section traite de différentes méthodes numériques qui permettent de résoudre des équations différentielles. Elles constituent un socle sur lequel s'appuyer pour résoudre des problèmes complexes abordés dans les sections suivantes.

1.1 Représentation d'un problème de Cauchy

Tout d'abord, on s'est mis d'accord pour que le problème de Cauchy soit représenté de deux façons différentes. Les conditions initiales en dimension 1 sont représentés en Equation (1), tandis qu'en dimension $n > 1$ elles sont représentées par un tableau de valeurs comme représenté en Equation (2).

$$y(t_0) = y_0(t_0), \quad (1)$$

$$y(t_0) = [y_0(t_0), y_1(t_0), \dots, y_n(t_0)]. \quad (2)$$

Quant aux équations différentielles, elles sont représentées par un tableau de fonctions en dimension $n > 1$ et par une simple fonction en dimension $n = 1$.

On a ensuite implémenté les quatre méthodes de résolution des problèmes de Cauchy qui nous ont été proposées, à savoir la méthode d'Euler, du point milieu, de Heun et de Runge Kutta 4. On a pu comparer ces méthodes entre-elles et leur convergence vers une fonction connue. Par exemple, en dimension 1, on a pu comparer les différentes méthodes avec la Figure 1 et en dimension 2 avec la Figure 2.

On peut voir que la convergence vers la fonction réelle est meilleure pour les méthodes de Runge Kutta 4 et la méthode du point milieu, mais moins bonne pour Heun et Euler. Plus le nombre de subdivisions N utilisés pour évaluer un point est grand, plus l'erreur sur cette évaluation est faible. De plus, plus l'ordre de la méthode est élevé, moins il faut de subdivisions pour avoir une évaluation correcte. (Figure 3 et 4)

On a implémenté une fonction `tangential_field_dim2` qui prend en paramètre une équation différentielle de dimension 2 et les limites des axes et qui dessine le champ de tangentes de cette dernière. Le champ de tangentes de l'équation différentielle de la Figure 2 est visible dans le code.

On a donc pu mettre en oeuvre les méthodes implémentées en les utilisant dans des cas concrets.

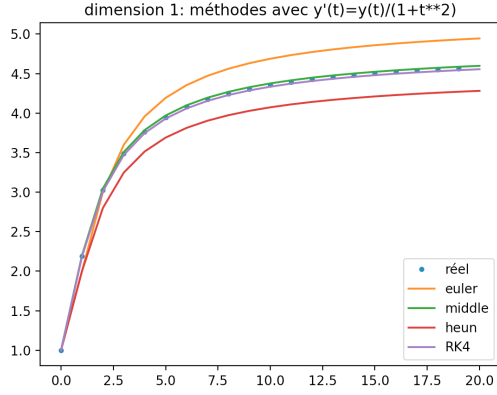


Figure 1: Représentation des approximations de fonction avec les différentes méthodes en dimension 1 pour 20 pas de taille 1.

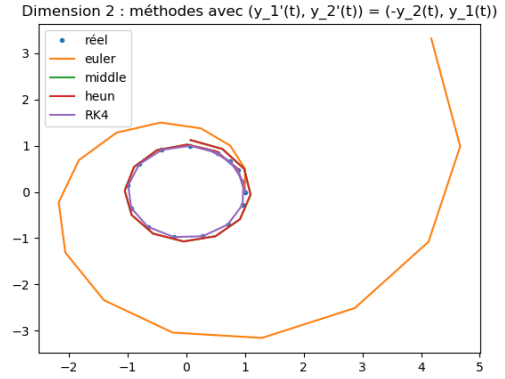


Figure 2: Représentation des approximations de fonction avec les différentes méthodes en dimension 2 pour 15 pas de taille 0.5

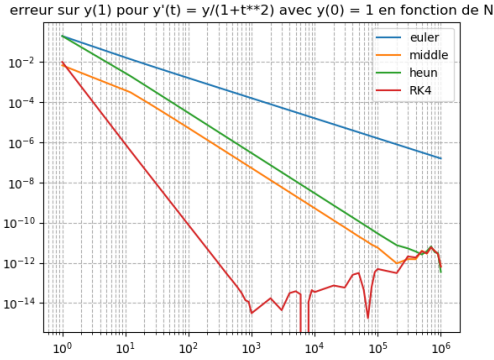


Figure 3: Erreur sur l'évaluation de $e^{\arctan(1)}$

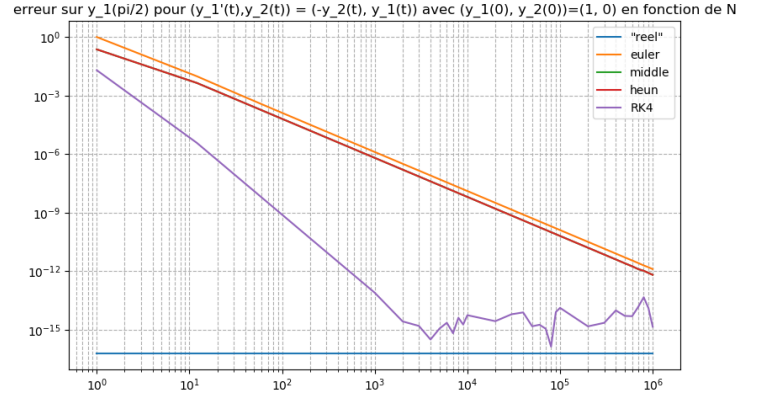


Figure 4: Erreur sur l'évaluation de $\cos(\frac{\pi}{2})$

2 Système proie-prédateur de Lokta-Volterra

On s'intéresse dans cette partie à l'évolution d'une population en essayant de la modéliser. Les premiers modèles d'évolution des populations étaient de la forme de l'équation (3).

$$\frac{dN(t)}{dt} = b \times N(t) - d \times N(t) = \gamma \times N(t). \quad (3)$$

Ainsi, γ représente la différence entre le taux de natalité et le taux de mortalité de la population: Si γ est négatif, alors la population devient de plus en plus réduite tandis que si γ est positif, la population croît.

Bien que ce modèle s'applique à la croissance d'une espèce qui ne possède pas de grand prédateur comme les hommes, elle n'est pas représentative des espèces qui en possèdent. Ainsi, Verhulst a

proposé une amélioration de ce modèle en ajoutant dans l'équation (4) un facteur d'auto-régulation κ appelé capacité d'accueil.

$$\frac{dN(t)}{dt} = \gamma \times N(t) \times \left(1 - \frac{N(t)}{\kappa}\right) \quad \kappa, \gamma > 0. \quad (4)$$

On peut s'apercevoir en étudiant l'équation 4 que:

- $y = \kappa$ est une solution particulière de l'équation différentielle non linéaire
- si $y < \kappa$ alors la population croît car $1 - y/\kappa > 0$ donc la dérivée est positive.
- si $y > \kappa$ alors la population décroît car $1 - y/\kappa < 0$ donc la dérivée est négative.

La représentation graphique des solutions de ces deux modèles peut être générée avec notre code en exécutant `proiePredateur.py`. On va néanmoins se concentrer sur le modèle suivant.

Il est en effet possible de modéliser des écosystèmes un peu plus riches en ajoutant les interactions de deux populations proies/prédateurs. C'est notamment le cas du modèle Lotka-Volterra qui est défini par l'équation (5):

$$\begin{cases} \frac{dN(t)}{dt} = N(t) \times (a - b \times P(t)), & a; b; c; d > 0 \\ \frac{dP(t)}{dt} = P(t) \times (c \times P(t) - d). \end{cases} \quad (5)$$

où $N(t)$ représente la population de proies, $P(t)$ la population de prédateurs, a le taux de reproduction des proies en l'absence de prédateurs, b le taux de mortalité des proies dû aux prédateurs rencontrés, c le taux de reproduction des prédateurs en fonctions des proies mangées et d le taux de mortalité des prédateurs en l'absence de nourriture.

Ainsi, ce modèle présente d'une part la variation du nombre de proies par sa propre croissance moins le taux de prédation. D'autre part, il présente la variation de la population de prédateurs comme étant la croissance des prédateurs moins le nombre de morts naturelles dû à l'absence de nourriture (proies).

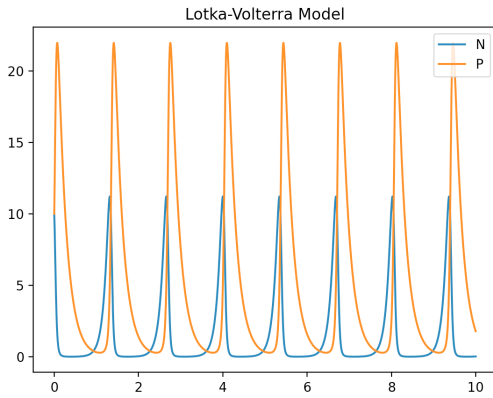


Figure 5: Solutions du modèle de lotka-Volterra

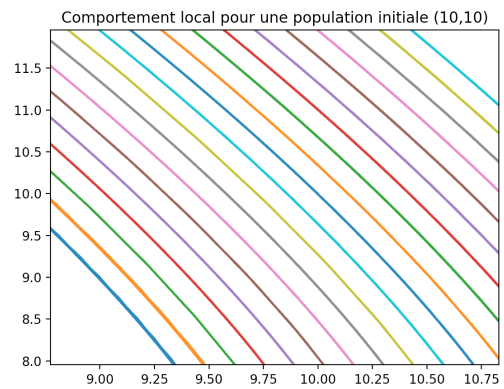


Figure 6: Comportement local de l'évolution des proies en fonction des prédateurs

Les solutions de ce système d'équations sont périodiques comme on peut le voir en Figure 5. Enfin, le comportement local du modèle de Lotka-Volterra représenté en Figure 6 nous permet de voir qu'en partant de conditions initiales proches de y_0 , les courbes ont un comportement similaire à celle de y_0 et ne se croisent pas, montrant une unicité locale de la solution de l'équation différentielle.

Enfin, on peut s'apercevoir que les dérivées s'annulent lorsque $(N(t), P(t)) \in \{(0, 0), (\frac{d}{c}, \frac{a}{b})\}$. Ce sont les points singuliers de l'équation différentielle (5).

3 Pendule à N maillons

Dans cette partie, on modélise un pendule constitué de un ou deux maillons. Un maillon est représenté dans ce modèle par une tige sans masse et de longueur l avec à son extrémité, une masse m . De plus pour cette modélisation, les frottements de l'air ont été négligés.

3.1 Pendule simple

Afin de modéliser ce système on a utilisé l'équation (6):

$$\ddot{\theta} + \frac{g}{l} \times \sin(\theta) = 0. \quad (6)$$

On peut retrouver le résultat de la modélisation avec un graphe représentant l'angle et la vitesse de la masse au cours du temps. Cela a permis d'obtenir la fréquence en fonction de l'angle initial, ce qui permet de vérifier que pour de faibles angles, la fréquence d'oscillation du pendule est égale à $\sqrt{\frac{g}{l}}$. Par contre pour de plus grands angles cette relation n'est plus vérifiée comme on peut le voir sur la figure 7.

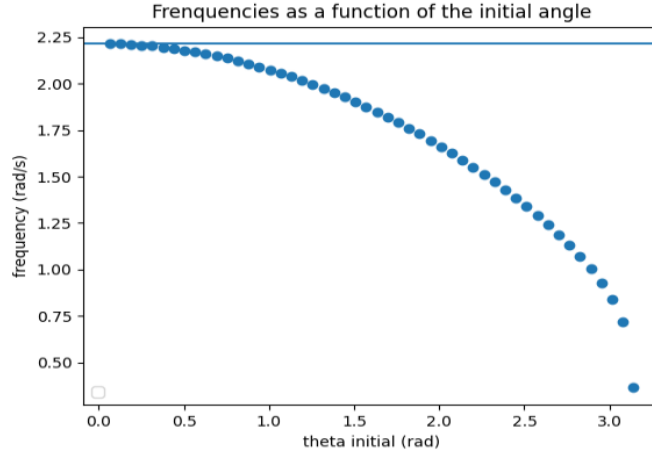


Figure 7: Fréquence en fonction de l'angle initial

3.2 Pendule double

Pour modéliser ce pendule, on a utilisé le système d'équations (7) donné dans le sujet.

$$\begin{cases} \dot{\theta}_1 = \omega_1, \\ \dot{\theta}_2 = \omega_2, \\ \dot{\omega}_1 = \frac{-g \times (2 \times m_1 + m_2) \times \sin \theta_1 - m_2 \times g \times \sin(\theta_1 - 2 \times \theta_2) - 2 \times \sin(\theta_1 - \theta_2) \times m_2 \times (\omega^2 \times l_2 + \omega_1^2 \times l_1 \times \cos(\theta_1 - \theta_2))}{l_1 \times (2 \times m_1 + m_2 \times (1 - \cos(2 \times \theta_1 - 2 \times \theta_2)))}, \\ \dot{\omega}_2 = \frac{2 \times \sin(\theta_1 - \theta_2) \times ((m_1 + m_2) \times (\omega_1^2 \times l_1 + g \times \cos \theta_1) + \omega_2^2 \times l_2 \times m_2 \times \cos(\theta_1 - \theta_2))}{l_2 \times (2 \times m_1 + m_2 \times (1 - \cos(2 \times \theta_1 - 2 \times \theta_2)))}. \end{cases} \quad (7)$$

A l'aide de cela et des équations (8) on a pu tracer les trajectoires de m_2 pour des conditions initiales données. Cela est illustré lors de l'exécution du programme `pendule.py`, de même qu'une animation d'une des deux trajectoires de la masse m_2 . Pour des conditions initiales très proches, les trajectoires obtenues sont très différentes, ce qui est relevant d'un système dynamique chaotique.

$$\begin{cases} x = l_1 \times \sin \theta_1 + l_2 \times \sin \theta_2, \\ y = l_1 \times \cos \theta_1 + l_2 \times \cos \theta_2, \end{cases} \quad (8)$$

Une autre manière de mettre en valeur l'aspect chaotique du système dynamique, est de comparer les temps de premier retournement en fonction des conditions initiales. Ce qui est illustré en figure 8.

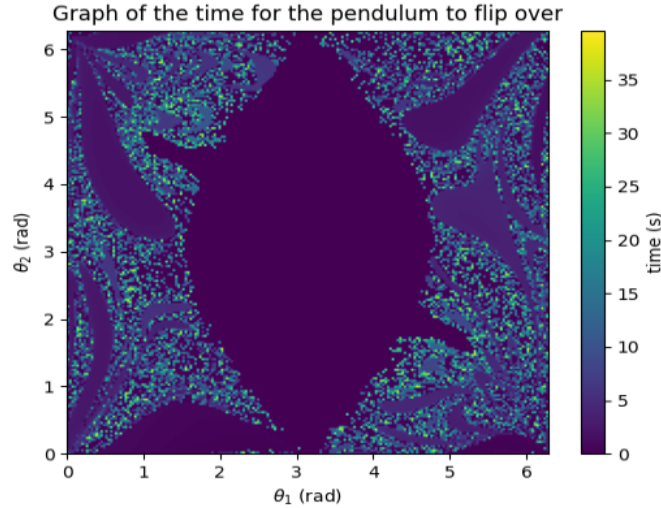


Figure 8: Carte du temps de premier retournement