



🔗 Retour à IS104 – Algorithmique Numérique (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=159)

Projet 6 : Résolution approchée d'équations différentielles / Modélisation de systèmes dynamiques

en raison de la pandémie de Coronavirus, vous trouverez ci-dessous les supports additionnels pour le projet.

- Le chapitre 10 de « Methodes Numeriques, Algorithmes, analyse et applications ». Le livre est disponible sur mon compte à l'enseirb [~mfaverge/public_html/IS104/Methodes-Numeriques-Algorithmes-analyse-et-applications.pdf](http://mfaverge/public_html/IS104/Methodes-Numeriques-Algorithmes-analyse-et-applications.pdf)
- Les slides de présentation de ce chapitre sont disponibles ici (<http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/wp-content/cours/IS104/files/slides-chapter6.pdf>). Ils sont également disponibles sur mon compte à l'enseir au même endroit que le livre.
- Quelques slides additionnels pour présenter les méthodes du point milieu (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/wp-content/cours/IS104/files/is104_EDO_shemas_pt_milieu.pdf), de Heun (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/wp-content/cours/IS104/files/is104_EDO_shemas_heun.pdf), et de Runge-Kutta 4 (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/wp-content/cours/IS104/files/is104_EDO_shemas_RK4.pdf).

Pour cet ultime projet, nous allons nous intéresser aux 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. Ce qui ne signifie pas forcément que leur résolution ne pose pas de problèmes. Ce projet comporte deux parties :

- Dans un premier temps, il s'agit de programmer les méthodes de résolution numérique à un pas, adaptées à des équations différentielles ordinaires en dimension finie, et de les tester dans un cadre simple.
- La deuxième partie est ensuite constituée d'applications de ces méthodes à un certain nombre de systèmes dynamiques :
 - le modèle de biologie des populations dit de Lotka-Volterra,
 - le pendule à N maillons,
 - le traitement d'image avec la méthode de Perona et Malik,
 - les réactions chimiques oscillantes, comme celles de Belousov-Zhabotinsky,
 - le problème à trois corps.

Ce projet laisse une part importante à l'expérimentation et à la recherche de paramètres mettant en évidence des comportements particuliers des équations différentielles rencontrées. Il est donc plus que conseillé de faire preuve d'initiative et de tester plusieurs exemples par soi-même.

Il est demandé de réaliser au moins deux applications pour mettre en valeur les problèmes numériques liés à la

résolution d'équations différentielles.

Méthodes numériques de résolution d'équations différentielles

Dans cette partie, on programme les différentes méthodes numériques de résolution des équations différentielles que nous testerons par la suite. Ces algorithmes sont tous des méthodes à un pas, de la forme suivante :

$$y_{n+1} = y_n + h_n \Phi(y_n, t_n, h_n)$$

Les algorithmes reprenant tous le même principe, il est possible de répondre à une grande partie des questions sans programmer toutes les méthodes demandées, pour y revenir plus tard lorsqu'elles auront été vues en cours.

1. Définissez la façon dont vous allez représenter un problème de Cauchy dans votre code. Il s'agit de fixer la façon dont vous allez représenter la condition initiale et la fonction différentielle représentant l'équation, afin d'unifier le code écrit dans chacune des parties suivantes.

$$\begin{cases} y(t_0) = y_0 \\ y'(t) = f(y(t), t) \end{cases}$$

Votre représentation doit pouvoir prendre en compte des équations différentielles en dimension arbitraire.

1. Programmez les méthodes de résolution suivantes :

-
- la méthode d'Euler $\Phi(y_n, t_n, h_n) = f(y_n, t_n)$,
 - la méthode du point milieu,
 - la méthode de Heun,
 - et la méthode de Runge-Kutta d'ordre 4.
-

Remarque : Pour chacune de ces méthodes on programmera, comme pour les méthodes d'intégration, de la manière la plus générique possible :

- pour chaque méthode de résolution, une fonction `step_<name>(y, t, h, f)` qui calcule un seul pas de la méthode,
- un premier algorithme `meth_n_step(y0, t0, N, h, f, meth)` calculant un nombre `N` de pas de taille constante `h`,
- puis un second algorithme `meth_epsilon(y0, t0, tf, eps, f, meth)` calculant une solution approchée avec un paramètre d'erreur `epsilon`.

Comparer l'utilité de l'implémentation générique par rapport au projet précédent.

3. Écrire une fonction permettant de dessiner le champ des tangentes de l'équation différentielle (on ne s'intéressera qu'au cas de la dimension 2, et on utilisera la fonction `quiver`)
3. Testez vos implémentation sur des équations différentielles connues, et comparez-les aux résultats exacts. En particulier, on tracera les courbes correspondant aux deux équations différentielles suivantes :

- En dimension 1 : $\begin{cases} y(0) = 1 \\ y'(t) = \frac{y(t)}{1+t^2} \end{cases}$

▪ En dimension 2 :

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}$$

$$\begin{cases} y(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ y'(t) = \begin{bmatrix} -y_2(t) \\ y_1(t) \end{bmatrix} \end{cases}$$

Quels sont les types d'erreurs que vous rencontrez ?

Système proie-prédateur de Lotka-Volterra

Dans cette partie, on s'intéresse à la modélisation de l'évolution de la population d'une ou plusieurs espèces dans un milieu naturel. Les équations différentielles décrites ici sont des modèles de population. Considérons la fonction

$N(t)$ décrivant la variation de la population d'un ensemble d'individus au cours du temps. Les premiers modèles de population (dits malthusiens, car dûs à Malthus à la fin du XVIIIème siècle) décrivaient ainsi les variations de cette fonction :

$$\begin{aligned} \frac{dN(t)}{dt} &= \text{naissances} - \text{morts} \\ &= bN(t) - dN(t) = \gamma N(t), \quad b, d > 0 \end{aligned}$$

Néanmoins, cette approche n'est pas vraiment réaliste (même si la croissance de la population humaine ne contredit pas effectivement le modèle), et Verhulst a proposé un autre modèle de population auto-limitant :

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

1. Expliquer en quoi ces deux équations différentielles peuvent modéliser les variations d'une population, et en particulier à quoi correspondent les constantes apparaissant à l'intérieur.
1. Résoudre ces équations différentielles en utilisant une méthode de votre choix, et en choisissant différents points de départ crédibles.

Ces modèles peuvent être modifiés à loisir, mais lorsque l'on veut décrire un écosystème de manière un peu plus riche, il est possible de modéliser les interactions de deux populations proies/prédateurs : $N(t)$ est une population de proies et $P(t)$ est une population de prédateurs. Le modèle de Lotka-Volterra est le suivant :

$$\begin{cases} \frac{dN(t)}{dt} = N(t)(a - bP(t)) \\ \frac{dP(t)}{dt} = P(t)(cN(t) - d) \end{cases}, \quad a, b, c, d > 0$$

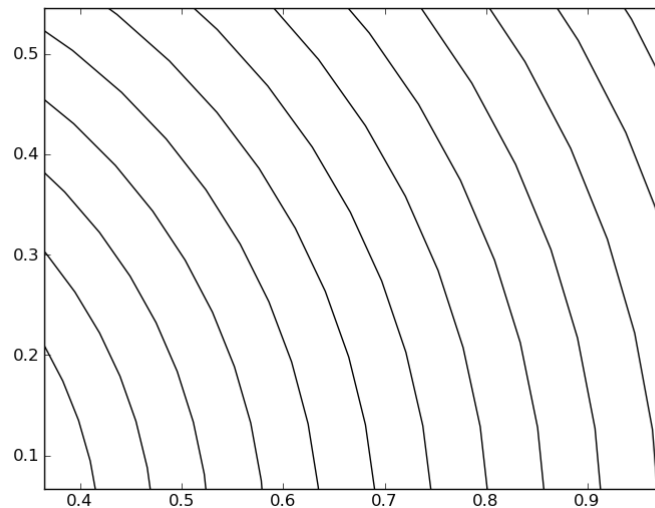
3. Expliquer en quoi ces deux équations peuvent-elles modéliser un écosystème proie/prédateur.
3. Résoudre numériquement ce système d'équations différentielles en utilisant une méthode de votre choix, et en choisissant différents points de départ crédibles. Tracer les variations des deux populations au cours du temps.
3. Tracer sur une courbe les variations du couple $(N(t), P(t))$ au cours du temps. Quels types de solutions fait-on apparaître ? Quelles sont les solutions constantes ?
3. Mettre en place un algorithme permettant de calculer la période de ces solutions (de manière approchée).

Lors de l'étude d'un système d'équations différentielles, on s'intéresse au comportement global des solutions, mais aussi à leur comportement local. En particulier, il est intéressant de regarder les variations entre des solutions partant de conditions initiales très proches.

7. Écrire un algorithme qui permette de tracer les solutions autour d'un point de départ donné. En

fonction d'une condition initiale y_0 , il s'agit de tracer la courbe solution en partant de y_0 , puis les courbes solutions en partant de conditions initiales qui sont proches de y_0 (il est possible de les choisir dans un petit disque/rectangle centré sur y_0).

Le résultat pourra ressembler à :



(On rappelle qu'il s'agit bien d'expliquer le comportement local de l'équation différentielle, et donc que la fenêtre considérée doit être petite. De plus, il est important de ne considérer que les EDO en dimension 2, afin de montrer des courbes solutions qui ne se croisent pas.)

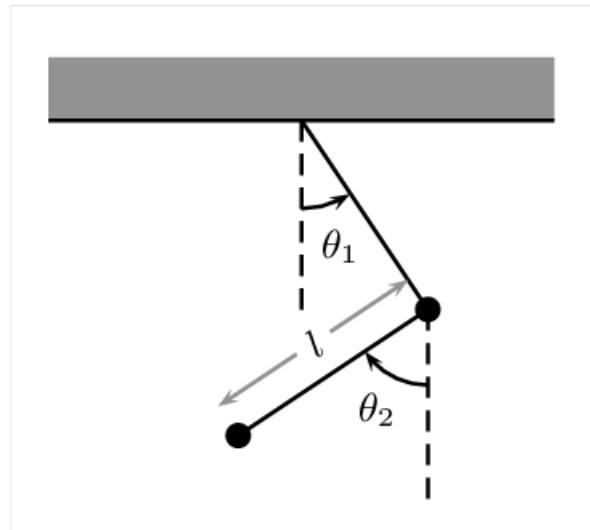
7. Un point singulier de l'équation différentielle est un point où toutes les dérivées s'annulent. Quels sont les points singuliers du système de Lotka-Volterra ?
7. Quelle est la forme générale des diagrammes obtenus à la question 7 (en partant d'une condition initiale différente d'un point singulier) ? Quelle est la forme de ces diagrammes lorsque l'on part d'un des points singuliers de l'équation ?

(Difficile) Si l'on se replace dans le contexte d'une équation différentielle générale, à quels types de diagrammes peut-on s'attendre ?

Pour tester le comportement du système proie-prédateur, on pourra tester le programme NetLogo (<http://www.netlogoweb.org/launch#http://www.netlogoweb.org/assets/modelslib/Sample%20Models/Biology/Wolf%20Sheep%20Predation.nlogo>).

Pendule à N maillons

Un système dynamique très simple peut être construit en considérant un pendule constitué d'un ensemble de N solides de masse m reliés entre eux par des tiges sans masse de même longueur L , les tiges étant supposées en rotation les unes par rapport aux autres (le système est supposé confiné à un plan). Par exemple, pour un pendule à deux maillons, le système est modélisé par le dessin suivant :



Ici, les angles θ_1 et θ_2 permettent de spécifier entièrement le système. On se propose alors d'examiner les évolutions de ce système par rapport au temps, à partir d'une position à vitesse nulle.

1. Modéliser le pendule à un unique maillon. Implémenter une fonction permettant de déterminer la fréquence d'oscillation du système, en fonction de l'angle initial θ . En traçant cette fréquence en fonction de θ , montrer que cette fréquence pour de petites oscillations est égale à $\sqrt{\frac{g}{l}}$.
1. Modéliser le pendule à deux maillons. On pourra s'inspirer de la page (http://www.mypysicslab.com/dbl_pendulum.html) suivante pour les équations.

Il se trouve que le pendule à deux maillons est un système dynamique chaotique, ce qui peut se voir à travers la sensibilité des trajectoires aux conditions initiales.

3. Dessiner la trajectoire de l'extrémité du pendule à deux maillons en fonction du temps (il est possible d'obtenir des séquences du style de cette vidéo (http://www.youtube.com/watch?v=QXf95_EKS6E))

En considérant plusieurs trajectoires avec des conditions initiales différentes mais très proches, mettre en exergue la dépendance aux conditions initiales de ce système.

3. Une autre manière de mettre en valeur les propriétés du système consiste à déterminer le temps de premier retournement. Ecrire une fonction permettant de calculer ce temps, afin de tracer une carte du type de celle dessinées sur cette page (http://en.wikipedia.org/wiki/Double_pendulum#Chaotic_motion).

Filtrage de Perona et Malik

Une image peut être modélisée comme une fonction de classe \mathcal{C}^1 de \mathbb{R}^2 à valeur dans \mathbb{R} . Une image numérique est alors une version discrétisée de cette fonction sur un certain rectangle. Le filtrage par équation de la chaleur consiste à faire évoluer l'image pendant une durée déterminée suivant l'EDP de la chaleur. On suppose donc que l'image est une fonction qui varie au cours du temps, notée alors $u(x, y, t)$. L'équation de la chaleur se présente sous la forme :

$$\begin{cases} \frac{\partial u}{\partial t}(x, y, t) - \Delta u(x, y, t) = 0, \text{ pour } t > 0 \\ u(0, x, y) = u_0(x, y) \end{cases}$$

... pour u_0 une image initiale donnée de taille $M \times N$. t représente le temps. La solution de cette équation en un temps donné correspond à la convolution avec une gaussienne.

Pour résoudre effectivement ce problème, on considère que l'image d'entrée est une version discrétisée de la fonction U au temps $t = 0$. Le laplacien Δ est calculé, à temps t fixé, via des opérateurs discrets de divergence et de gradient : $\Delta u = \text{div}(\nabla u)$. À temps t fixé, le gradient est un champ de vecteur de taille $M \times N$, et chaque vecteur a deux dimensions.

$$\nabla u_{k,l} = \begin{pmatrix} \begin{cases} u_{k+1,l,1} - u_{k,l,1} & \text{si } k < N, \\ 0 & \text{si } k = N, \end{cases} \\ \begin{cases} u_{k,l+1,1} - u_{k,l,1} & \text{si } l < M, \\ 0 & \text{si } l = M. \end{cases} \end{pmatrix}$$

La divergence est alors l'opérateur conjugué. Il prend un champ de vecteur à deux dimensions en entrée (le résultat du gradient par exemple), et est défini comme suit :

$$\text{div}(p)_{k,l} = \begin{cases} p_{k,l}^1 - p_{k-1,l}^1 & \text{si } 1 < k < N, \\ p_{k,l}^1 & \text{si } k = 1, \\ -p_{k-1,l}^1 & \text{si } k = N, \end{cases} + \begin{cases} p_{k,l}^2 - p_{k,l-1}^2 & \text{si } 1 < l < M, \\ p_{k,l}^2 & \text{si } l = 1, \\ -p_{k,l-1}^2 & \text{si } l = M, \end{cases}$$

avec $p = (p_{k,l}^1, p_{k,l}^2)_{k,l}$ le champ dont on cherche la divergence. $p_{k,l}^1$ (respectivement $p_{k,l}^2$) représente la première (resp. deuxième) coordonnée du champ de vecteur au point de coordonnées (k, l) .

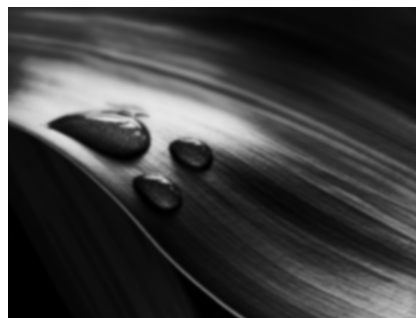
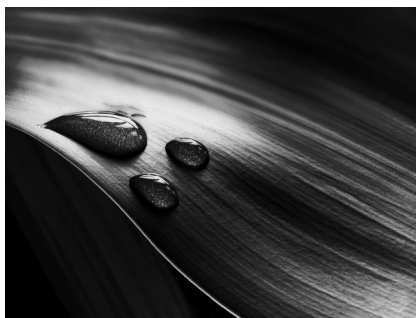


Image originale

Équation de la chaleur

Filtre de Perona Malik

La méthode de Perona et Malik évite de flouter l'image en préservant les contours. Celle-ci repose sur l'EDP suivante :

$$\begin{cases} \frac{\partial u}{\partial t}(x, y, t) = \text{div}(f(x, y) \cdot \nabla u(x, y, t)), & \text{pour } t > 0 \\ u(0, x, y) = u_0(x, y) \end{cases}$$

avec f une fonction de \mathbb{R}^2 à valeur dans $[0, 1]$ à valeur faible près des contours, c'est à dire avec une valeur faible près des zones de l'image présentant un gradient élevé.

La fonction f se calcule à partir de la fonction initiale et permet de limiter l'action du filtrage (floutage) au niveau

des zones de contours. En pratique, c'est un tableau de coefficients de la taille de l'image. L'équation de la chaleur correspond à celle de Perona-Malik où $f \equiv 1$.

Afin de ne pas flouter uniformément l'image, on réduit l'action de l'opérateur laplacien au niveau des zones de contours en multipliant le gradient par une fonction ayant une valeur proche de 1 aux endroits où il n'y a pas de contour (norme du gradient faible) et ayant une valeur proche de 0 aux endroits où il y a des contours (norme du gradient élevée). Une telle fonction peut être choisie de la forme $f(x, y) = \exp(-\|\nabla F(x, y)\|^2)$ avec F une convolution de l'image initiale avec une gaussienne. Cette convolution vise à élargir la zone de coefficients faibles afin de mieux préserver les contours.

1. Implémenter les gradients et divergence de l'image.

1. Implémenter le schéma d'Euler sur l'équation de la chaleur ($U_{n+1} = U_n + \delta_t \cdot \Delta U_n$), prendre un pas de temps ($\delta_t = 0.1$)

1. Implémenter la convolution de l'image avec une gaussienne en prenant soin de considérer que les bords de l'image soient symétriques (utiliser `scipy.signal.convolve2d`). Construire la fonction f et la visualiser.

1. Implémenter le schéma d'Euler correspondant à l'équation de Perona et Malik avec un pas de temps de 0.2 et 200 itérations.

Réactions chimiques oscillantes

Usuellement, les réactions chimiques ont un comportement simple d'un point de vue thermodynamique : la réaction arrive à un équilibre stable à partir duquel on peut considérer qu'il n'y a plus d'interaction entre les réactifs. Néanmoins, il existe des réactions chimiques dites oscillantes, qui ont un comportement périodique. La plus connue de ces réactions est la réaction de Belousov-Zhabotinsky, ou encore celle de Briggs-Schauer.

Le système auquel nous nous intéressons ici est un système oscillant simplifié, à trois dimensions, modélisé par les équations suivantes :

$$\begin{cases} \frac{dX(t)}{dt} = a + bZ - X - XY^2 \\ \frac{dY(t)}{dt} = c(X + XY^2 - Y) \\ \frac{dZ(t)}{dt} = d(Y - Z) \end{cases}, \quad a; b; c; d > 0$$

Pour les conditions de la réaction, les valeurs intéressantes se situent pour $a = 10$, $c = 200$, $d = 50$, et $b \in [0.1, 0.2]$. Les valeurs initiales sont fixées à $[1; 0; 0]$. Dans cette partie, on s'intéresse au comportement de la réaction en fonction du paramètre b .

1. Résoudre les équations précédentes pour différentes valeurs du paramètre b .

Remarque : Les solutions obtenues doivent être périodiques, ou ultimement périodiques. Aucune solution ne diverge vers l'infini. Veillez à vérifier que vos calculs ne divergent pas, en prenant des pas de calculs suffisamment faibles.

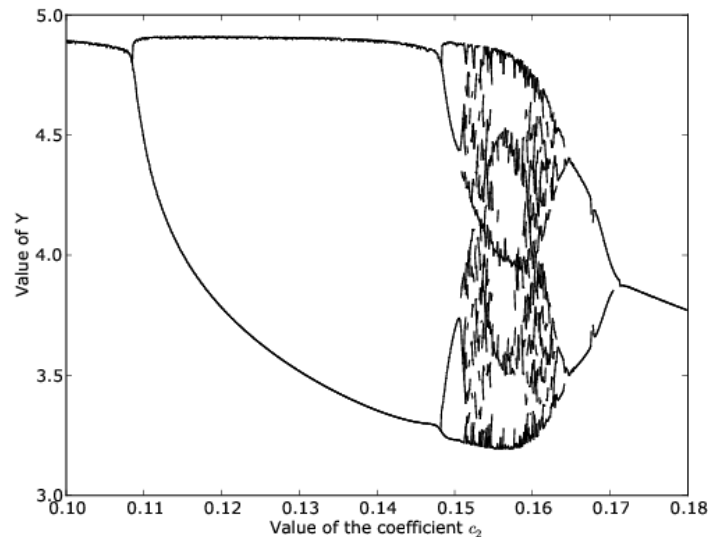
Toutes les solutions ont un comportement ressemblant : premièrement, une phase de démarrage, puis une phase stable dans laquelle la fonction a un comportement périodique, ou quasi-périodique. Pour mesurer la complexité de ce comportement, nous allons définir une notion de période d'une solution : étant donné une solution

$[X(t); Y(t); Z(t)]$, la période de cette solution est égale au nombre de maxima locaux de $Y(t)$ restreinte à sa phase périodique.

2. Comment ne considérer que la phase périodique d'une solution ?
3. Écrire une fonction qui, étant donné une valeur du paramètre b , calcule la période de la solution associée.

Pouvez-vous exhiber des solutions de période 2 ? de période 4 ? de période 5 ? de période 3 ?
Pouvez-vous exhiber des solutions de période infinie ?

4. Tracer le diagramme de bifurcation de ce système dynamique, à savoir une courbe traçant les pics de la solution en fonction du paramètre b .



5. Dans leur article Period 3 implies Chaos, Li et Yorke montrent qu'un système dynamique exhibant des solutions de période 3 devait nécessairement contenir des solutions chaotiques. Est-ce le cas du présent système ? Et sinon, est-ce tout de même un système chaotique ?
6. En quoi la modélisation numérique d'un système dynamique chaotique est elle difficile ?

Modélisation du problème à trois corps

Dans cette partie, on s'intéresse à la modélisation d'un problème apparemment simple de dynamique du solide : le problème à trois corps. Dans un premier temps, on s'attache à modéliser les interactions gravitationnelles classiques entre deux astres, puis on introduit un troisième corps massif pour mettre en évidence les modifications du système.

Considérons un système physique composé de deux corps massifs A , B ponctuels placés dans un plan. Chacun de ces deux corps est soumis uniquement à la force d'interaction gravitationnelle due à la présence de l'autre corps. Les caractéristiques de chaque corps sont donc :

Corps A		Corps B	
Masse :	m_A	Masse :	m_B
Position :	x_A y_A	Position :	x_B y_B
Vitesse :	\dot{x}_A \dot{y}_A	Vitesse :	\dot{x}_B \dot{y}_B

Ces corps sont supposés de masse non nulle, placés initialement à des positions distinctes. Pour rappel, la force

d'interaction gravitationnelle entre deux corps (par exemple la force que A subit sous l'attraction de B) est donnée par :

$$\vec{F}_{B \rightarrow A} = \frac{G m_A m_B}{||AB||^3} \vec{AB}$$

Par la suite, nous supposerons que la masse du corps A est largement supérieure à la masse du corps B . Cela simplifie le problème en impliquant que le corps A peut être supposé immobile, et insensible à l'interaction gravitationnelle de B .

1. Ecrire les équations du mouvement associées au corps B en utilisant l'équation fondamentale de la dynamique. De combien d'inconnues avez-vous besoin pour modéliser entièrement votre système ? Il est plus que fortement conseillé de rendre vos équations adimensionnelles, afin de simplifier les calculs. Cela consiste à simplifier les équations en éliminant les constantes inutiles. Ici en particulier, une bonne idée consiste à considérer que la constante $G = 1$ (ce qui réduit proportionnellement la force subie par A) et que les masses et distances sont des valeurs simples.
1. Résoudre ce système d'équations différentielles en utilisant une méthode de votre choix, et en choisissant différents points de départ crédibles. Tracer les évolutions de vos planétoïdes au cours du temps.

Quelles sont les courbes que l'on est censé obtenir ? Quels types de comportement erronés arrivez-vous à obtenir ?

La partie suivante s'intéresse à une modification du modèle précédent, appelée le problème à trois corps restreint. Dans ce nouveau système, on suppose que le système contient un troisième corps C , lui aussi de masse négligeable devant les masses de A et de B . On pourra par exemple s'inspirer des valeurs suivantes :

$$m_A \gg m_B \gg m_C$$

Exemple :

$$\begin{cases} m_A = 1 \\ m_B = 0.01 \\ ||AB|| = 1 \end{cases}$$

Le centre de gravité du système se situe donc à peu près au centre de A , ce qui permet de se placer dans un référentiel placé au centre du soleil, supposé fixe. Le corps B est supposé en rotation circulaire autour du soleil, avec une période de 2π jours terrestres.

3. Modifier votre système d'équations pour modéliser ce nouveau système en 4 dimensions d'espace et une dimension de temps (Indice : on pourra supposer le mouvement de B connu et circulaire autour de A , et ne modéliser que le mouvement de l'astéroïde).

Résoudre ce système d'équations différentielles en utilisant une méthode de votre choix.

Comme le mouvement de B est connu, il est possible d'effectuer un changement de base après la fin du calcul (une rotation), de manière à ce que les corps A et B restent fixes. Il est alors possible de ne s'intéresser qu'aux mouvements du corps C .

4. Transformer les trajectoires obtenues en appliquant à chaque point la rotation faisant que la planète B reste fixe.

4. Que se passe-t'il lorsque l'on choisit deux points de départ pour le corps c qui sont initialement très proches l'un de l'autre ? En particulier, que deviennent les trajectoires au bout d'un certain temps ?
4. Rechercher dans ce système les points particuliers évoqués au cours d'un précédent projet, appelés points de Lagrange, qui correspondent à des points d'équilibre du système physique et des points singuliers du système d'équations différentielles.

Placez le corps c initialement proche du dernier sommet du triangle équilatéral dont la base est formée par le segment Soleil-Jupiter. Que constatez-vous ?

Noter qu'il existe des solutions périodiques au problème général à n corps, dont certaines qu'il est possible de visualiser là (<http://www.maths.manchester.ac.uk/~jm/Choreographies/>).

Dans cette section

IS104 – Algorithmique Numérique (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=159)
Configuration de l'environnement (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=272)
Fonctionnement des projets (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=204)
Présentation de Numpy/Scipy (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=231)
Aide mémoire Numpy/Scipy (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=220)
Syntaxe du langage Python (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=276)
Projet 1 : Méthodes de calcul numérique / Limites de la machine (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=286)
Projet 2 : Méthode du gradient conjugué / Application à l'équation de la chaleur (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=293)
Projet 3 : Compression d'image à travers la factorisation SVD (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=298)
Projet 4 : Non-linear systems of equations / Newton-Raphson method (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=302)
Projet 5 : Interpolation and integration methods / Cubic splines and surface interpolation (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=304)
Projet 6 : Résolution approchée d'équations différentielles / Modélisation de systèmes dynamiques (http://mfaverge.vvv.enseirb-matmeca.fr/wordpress/?page_id=309)

© 2022 Mathieu Faverge.

Construit avec ❤ par Thèmes Graphene (<https://www.graphene-theme.com/>).

