

Modélisation et simulation de l'excitation cardiaque

1 Éléments pédagogiques

1.1 Objectifs

Pendant le cours de méthodes numériques de base, vous avez étudié de manière théorique différents outils utilisés dans de nombreuses applications. Bien que la connaissance théorique des outils soit indispensable pour résoudre les problèmes, il est crucial de comprendre l'utilisation pratique de ces méthodes.

L'objectif de ce TP est de vous faire manipuler certaines méthodes numériques étudiées en cours à travers une application biomédicale, afin de vous sensibiliser aux difficultés de développement et de déploiement des outils numériques.

Ce TP sera réalisé par binôme uniquement. Vous programmerez dans l'environnement Scilab disponible sur les stations de travail de l'ENSIMAG et en téléchargement libre.

Le TP fera l'objet d'un rapport précis, clair et concis mettant en avant les résultats obtenus ainsi que vos observations sur les méthodes utilisées.

1.2 Livrables

Vous devez rédiger un compte-rendu de TP dans lequel vous répondrez à toutes les questions de l'énoncé, explicitez les méthodes employées, présenterez et commenterez les résultats obtenus. Il n'y a qu'un seul rapport à rendre par binôme.

Notez que ce sujet ne constitue que la base de ce qui vous est demandé: soyez critique par rapport à vos résultats, proposez d'autres idées, solutions ou tests. La mise en oeuvre de techniques de visualisation est fortement encouragée.

La dernière page de votre compte-rendu devra être une sorte de manuel d'utilisation où vous expliquerez comment utiliser vos programmes. Le compte-rendu sera dactylographié. Le rapport devra être au format PDF.

Le compte-rendu ne devra pas excéder 10 pages et ne devra pas contenir de programmes. Les programmes Scilab seront soumis sous Teide. La lisibilité du code et la pertinence des commentaires seront pris en compte dans la note du TP.

La qualité de la rédaction, de la synthèse, de l'analyse des résultats obtenus sont des critères importants pour la note. N'oubliez pas de donner des résultats graphiques.

1.3 Organisation du travail

Une séance d'interactions concernant le TP est prévue. Nous vous invitons à venir préparés à cette séance.

Plus généralement, n'hésitez pas à demander des conseils, des précisions ou à poser vos questions par mail aux enseignants de TP.

Le travail demandé dans ce TP est conséquent. Nous vous encourageons fortement à débiter le travail le plus tôt possible.

1.4 Remise des livrables

Le TP est à rendre pour le lundi 12 Mai 2014 à 20h00 sous Teide. Votre livrable devra contenir

- le compte rendu au format pdf,
- le code Scilab que vous avez utilisé,
- les images et animations mettant en valeur votre travail.

Un livrable soumis hors délai se verra attribuer la note 0.

2 Modélisation et discrétisation

2.1 Description du problème

Dans les stratégies diagnostiques et thérapeutiques en médecine, l'outil informatique est devenu indispensable. Il permet d'obtenir et d'analyser précisément l'état d'un patient à un instant donné afin de lui fournir les soins les plus adaptés. D'un autre côté, la simulation numérique reste encore fortement inexploitée pour la prédiction de l'évolution des maladies. Le coût prohibitif de la simulation, le manque de précision sur les caractéristiques physiques des tissus pour un patient et la complexité des modèles utilisés sont des freins à la démocratisation.

Il est cependant indispensable de comprendre les phénomènes avant de proposer les outils de simulation.

Afin de réduire les coûts liés au calcul, il est souvent nécessaire de faire des hypothèses simplificatrices.

Dans le travail proposé, l'étude portera sur la modélisation des courants transmembranaires des cellules cardiaques produisant les potentiels d'actions, c'est-à-dire la dépolarisation des membranes.

Pour aller plus loin

Le projet européen *Virtual Physiological Human* ambitionne de virtualiser l'intégralité du corps humain afin par exemple de simuler l'effet des traitements sur un patient et de détecter les conséquences potentiellement néfastes. Cet ambitieux projet combine les compétences des médecins, des informaticiens et des mathématiciens.

2.2 Un modèle 0D : le modèle cellulaire

Le premier modèle considéré permet de comprendre au niveau cellulaire les mécanismes des courants électriques cardiaque. Nous allons donc tout d'abord considérer un modèle générique de milieu excitable. Ce modèle permet, par exemple, de simuler le comportement des cellules cardiaques ou des neurones qui sont très similaires dans leur fonctionnement.

Le modèle est relativement simple: il utilise deux variables d'états et nécessite la résolution d'un système d'équations différentielles non linéaires. Ce modèle permet de simuler des comportements numériques complexes. Ces comportements peuvent s'interpréter comme de la fibrillation ventriculaire.

Le modèle mathématique cellulaire que nous allons utiliser s'écrit

$$\begin{cases} \frac{\partial e}{\partial t} = F(e, r) = -ke(e - \alpha_1)(e - 1) - er \\ \frac{\partial r}{\partial t} = G(e, r) = \left(\varepsilon + \mu_1 \frac{r}{e + \mu_2} \right) (-r - ke(e - \alpha_2 - 1)) \end{cases} \quad (1)$$

où e s'appelle le potentiel transmembranaire, r représente la conductance du courant repolarisant et t représente le temps. Les autres grandeurs, k , α_1 , α_2 , ε , μ_1 et μ_2 sont des paramètres du modèle qui n'ont pas d'interprétation physique à proprement parlé, mais qui permettent de simuler des effets de seuil.

Remarque 1

Les grandeurs du modèle (1) sont sans dimension. Afin de retrouver les grandeurs physiques correspondantes et de les comparer aux grandeurs médicales, nous devons procéder à leur redimensionnement. Le vrai potentiel transmembranaire E et le temps T sont donnés par

$$E = 100e - 80 \quad \text{et} \quad T = 12.9t$$

Avant de réaliser la simulation de ce modèle, nous allons poser les bases de la résolution numérique d'une équation différentielle.

Pour aller plus loin

Il existe de très nombreux modèles permettant de simuler l'excitation cardiaque. Les plus simples utilisent seulement deux variables, les plus complexes peuvent utiliser jusqu'à une cinquantaine de variables pour prendre en compte les différents aspects physiologiques. Une liste non exhaustive des modèles possibles pour les cellules cardiaques est disponible dans les références présentes sur le kiosk.

2.2.1 Schéma temporel

Afin de résoudre le système différentiel, nous allons mettre en place une méthode de résolution temporelle.

La première méthode que nous allons utiliser repose sur la méthode d'Euler explicite appliquée à chacune des équations.

La méthode d'Euler explicite est une méthode numérique permettant de résoudre une équation différentielle ordinaire de façon explicite.

On considère l'équation différentielle ordinaire suivante

$$\begin{cases} \frac{du(t)}{dt} &= f(t, u(t)) \\ u(t=0) &= u_0 \end{cases} \quad (2)$$

où f est une fonction continue sur un ouvert de $[0, T] \times \mathbb{R}$ à valeur dans \mathbb{R} avec $T > 0$ et $u_0 \in \mathbb{R}$.

La première étape consiste à diviser l'intervalle $I = [0, T]$ en un nombre fini d'intervalles n_t . Cela permet d'obtenir une grille régulière du temps. Les noeuds de la grille temporelle ont pour coordonnées (t_k) avec $t_k = kdt$, $dt = T/(n_t + 1)$ et $0 \leq k \leq n_t + 1$.

En utilisant une formule de Taylor à l'ordre 1, on exprime la dérivée temporelle en t_k

$$\frac{du(t_k)}{dt} = \frac{1}{dt}(u(t_{k+1}) - u(t_k)) + \mathcal{O}(dt)$$

En substituant dans l'équation différentielle, on obtient

$$\frac{u(t_{k+1}) - u(t_k)}{dt} = f(t_k, u(t_k)) + \mathcal{O}(dt)$$

On remplace donc l'équation différentielle ordinaire par l'équation approchée

$$u^{k+1} = u^k + dt f(t_k, u^k)$$

où u^k est l'approximation de $u(t_k)$ au point t_k . En pratique, on ne choisit pas n_t , mais on choisit dt tel que la méthode soit stable.

Nous allons maintenant mettre en oeuvre cette première méthode.

Question 1

Ecrire une fonction `euler1` qui prend comme argument d'entrée la condition initiale u_n , l'instant t_n , le pas de temps dt , et une fonction f représentant le second membre. Cette fonction implémentera le calcul d'un pas de temps de la méthode d'Euler explicite.

Attention, il faut faire en sorte que la fonction `euler1` puisse être appliquée pour u un vecteur et f une fonction manipulant des vecteurs.

Question 2

Afin de vérifier que `euler1` résout correctement les équations différentielles ordinaire du premier ordre, appliquer

euler1 au problème

$$\begin{cases} \frac{du(t)}{dt} = -0.1tu(t), \forall t < 5.0 \\ u_0 = 1 \end{cases}$$

Pour cela, écrire un script qui fera appel à la fonction `euler1`.

Question 3

Comment choisir dt afin que la méthode soit stable?

2.2.2 Résolution du modèle cellulaire

Nous allons maintenant appliquer le schéma d'Euler explicite au modèle cellulaire. Le schéma d'Euler explicite permettant de résoudre le problème approché s'écrit

$$\begin{cases} \frac{e^{k+1} - e^k}{dt} = F(e^k, r^k) \\ \frac{r^{k+1} - r^k}{dt} = G(e^k, r^k) \end{cases}$$

Pour un coeur normal, les valeurs des paramètres du modèle sont $k = 8$, $\alpha_1 = \alpha_2 = 0.1$, $\varepsilon = 0.01$, $\mu_1 = 0.1$ et $\mu_2 = 0.1$. Pour les conditions initiales, on choisira $e_0 = 1.0$ et $r_0 = 0.0$, c'est-à-dire que l'on soumet la cellule à une impulsion électrique.

Question 4

En utilisant les fonctions écrites précédemment, écrire un script Scilab permettant de résoudre le problème approché.

Pour aller plus loin

- Il est possible d'ajouter une sollicitation extérieure à l'équation différentielle de l'excitation. Ce terme source permet de simuler le rythme cardiaque sous la forme d'un train d'impulsion.
- Pour une équation différentielle, les isoclines sont les courbes telles que la dérivée soit constante. Elles permettent de comprendre le comportement des solutions dans différentes situations. Pour les systèmes d'équations, on s'intéresse notamment à l'isocline 0.

3 Simulation en dimension supérieure

Maintenant que nous avons un modèle fiable pour la simulation d'une cellule nous allons considérer son extension à un problème en dimension deux.

Afin de limiter le temps de développement, et toujours avec le souci de vérifier la validité du modèle, nous choisissons de travailler sur une région carrée du coeur $\Omega = [0, 1] \times [0, 1]$. Cette simplification du domaine ne permettra pas de prendre en compte toute la complexité du coeur, mais suffira à mettre en exergue certains phénomènes physiologiques.

3.1 Premier modèle

La première étape consiste à diviser chacune des directions de l'espace en un nombre fini d'intervalles. Cela permet d'obtenir une grille régulière de l'espace. Les noeuds de la grille ont pour coordonnées (x_i, y_j) avec $x_i = i * h$, $y_j = j * h$, $h = 1/(n - 1)$, $0 \leq i, j \leq n - 1$, où n un entier.

On notera $e_{i,j}$ (respectivement $r_{i,j}$) l'approximation de $e(x_i, y_j)$ (respectivement $r(x_i, y_j)$) au point (x_i, y_j) . Chaque noeud de cette grille peut-être vu comme une cellule cardiaque régit par le modèle (1).

Nous allons appliquer ce modèle sur les vecteurs e et r de taille n^2 .

Question 5

Ecrire un script Scilab qui génère une grille sur Ω , initialise les vecteurs e et r et applique le modèle (1) sur ces variables.

Les solutions pour chaque cellule sont rigoureusement identiques. Cette solution est naturellement inexacte car le modèle (1) ne prend pas en compte la dépendance spatiale entre les cellules.

3.2 Modèle complet

Cette propagation des courants d'une cellule à une autre est réalisé à l'aide de l'opérateur

$$\nabla \cdot (D \nabla) = \sum_{i,j=1}^3 \frac{\partial}{\partial x_i} d_{ij} \frac{\partial}{\partial x_j}$$

où $D \in \mathbb{R}^{3 \times 3}$ est un tenseur représentant les caractéristiques du milieu qui nous intéresse. Dans le cas général, le milieu est anisotrope, c'est-à-dire que D dépend de la direction des tissus.

L'opérateur de diffusion anisotrope s'applique sur le potentiel transmembranaire seulement. Le nouveau modèle mathématique s'écrit donc

$$\begin{cases} \frac{\partial e}{\partial t} = \sum_{i,j=1}^3 \frac{\partial}{\partial x_i} d_{ij} \frac{\partial e}{\partial x_j} - ke(e - \alpha_1)(e - 1) - er \\ \frac{\partial r}{\partial t} = \left(\varepsilon + \mu_1 \frac{r}{e + \mu_2} \right) (-r - ke(e - \alpha_2 - 1)) \end{cases} \quad (3)$$

où e est le potentiel d'action transmembranaire, r est la repolarisation et d_{ij} est le tenseur de conductivité prenant en compte l'orientation des fibres du coeur.

Afin de simplifier la réalisation numérique du modèle, nous nous plaçons en dimension deux et nous considérons que la conductivité est constante dans le domaine. Le tenseur D devient diagonal. Le modèle s'écrit donc

$$\begin{cases} \frac{\partial e}{\partial t} = D \Delta e - ke(e - \alpha_1)(e - 1) - er \\ \frac{\partial r}{\partial t} = \left(\varepsilon + \mu_1 \frac{r}{e + \mu_2} \right) (-r - ke(e - \alpha_2 - 1)) \end{cases} \quad (4)$$

Avant de considérer le problème complet, l'étude de l'opérateur de diffusion est indispensable.

Pour aller plus loin

La détermination des propriétés physiques des organes, par exemple le tenseur de conductance transmembranaire, est réalisée à l'aide de techniques d'imagerie médicales. Ces techniques portent le nom d'élastographie. Elles mettent en oeuvre des méthodes de corrélation des images, de reconstruction 3D des objets ainsi que de l'identification de forme.

3.3 Approximation par différences finies

La méthode des différences finies est une méthode d'approximation d'équations. L'équation aux dérivées partielles à résoudre est approchée par un système d'équations plus facile à mettre en oeuvre numériquement. C'est le processus de discrétisation.

Nous considérons donc à nouveau la grille discrète de la question précédente.

Soit $u \in \mathcal{C}^2(\Omega)$. En utilisant la formule de Taylor à l'ordre 2, on exprime les dérivées à l'ordre 2 de l'opérateur de diffusion sur chacun des points de la grille

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2}(x_i, y_j) &= \frac{1}{h^2} (u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)) + \mathcal{O}(h^2) \\ \frac{\partial^2 u}{\partial y^2}(x_i, y_j) &= \frac{1}{h^2} (u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})) + \mathcal{O}(h^2)\end{aligned}$$

En substituant dans la définition de l'opérateur de diffusion les expressions des dérivées en (x_i, y_j) , on obtient

$$\begin{aligned}\Delta u &= \frac{1}{h^2} (u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)) \\ &\quad + \frac{1}{h^2} (u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})) + \mathcal{O}(h^2).\end{aligned}$$

Ce développement permet de définir le laplacien approché à l'ordre deux

$$\Delta_h u = \frac{1}{h^2} (u_{i+1,j} + u_{i,j+1} - 4u_{i,j} + u_{i-1,j} + u_{i,j-1})$$

On définit maintenant le vecteur U par

$$U = (u_{1,1}, u_{2,1}, \dots, u_{n,n})$$

Cela permet d'écrire le laplacien discret sous forme matricielle

$$\Delta_h U = AU \tag{5}$$

avec $A \in \mathcal{M}^{n^2 \times n^2}(\mathbb{R})$ la matrice de différences finies donnée par

$$A = \begin{pmatrix} T_1 & \frac{I}{h^2} & 0 & \cdots & 0 \\ \frac{I}{h^2} & T_2 & \frac{I}{h^2} & 0 & \vdots \\ 0 & \frac{I}{h^2} & \ddots & \ddots & 0 \\ \vdots & 0 & \frac{I}{h^2} & T_{n-1} & \frac{I}{h^2} \\ 0 & \cdots & 0 & \frac{I}{h^2} & T_n \end{pmatrix} \tag{6}$$

et $T_k \in \mathcal{M}^{n \times n}(\mathbb{R})$

$$T_1 = T_n = \frac{1}{h^2} \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -3 & 1 & 0 & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & 0 & 1 & -3 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix} \text{ et } T_k = \frac{1}{h^2} \begin{pmatrix} -3 & 1 & 0 & \cdots & 0 \\ 1 & -4 & 1 & 0 & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & 0 & 1 & -4 & 1 \\ 0 & \cdots & 0 & 1 & -3 \end{pmatrix}, \quad 2 \leq k \leq n-1. \quad (7)$$

Afin d'alléger les notations, les opérations entre les vecteurs doivent être comprises comme des opérations éléments à éléments.

Pour aller plus loin

La méthode des différences finies n'est pas le seul outils permettant la discrétisation des équations aux dérivées partielles. Elle est la plus simple et la plus rapide à mettre en oeuvre, mais elle présente l'inconvénient de ne pas s'adapter facilement aux formes compliquées.

3.4 Etude du laplacien discret

Question 6

Ecrire une fonction `laplacien` qui prend comme argument d'entrée le nombre de point de discrétisation n dans une direction et le coefficient de conductivité D et qui construit l'opérateur linéaire L .

Pour aller plus loin

La matrice du laplacien peut-être associée à un graphe possédant n^2 noeuds. Pour le traitement numérique des images, l'application d'une matrice du laplacien permet de flouter les contours.

L'opérateur L est une matrice contenant énormément de 0. D'un point de vue mémoire, le stockage de l'opérateur L demande donc n^4 **double**, alors que nous avons besoin de stocker approximativement $5n^2$ **double**. Pour remédier à cette perte de mémoire, nous allons utiliser ce que l'on appelle un stockage creux qui permet de conserver uniquement les éléments non nuls de la matrice.

Question 7

Ecrire une fonction `slaplacien` qui prend comme argument d'entrée le nombre de point de discrétisation n dans une direction et le coefficient de conductivité D et qui construit l'opérateur linéaire L_s en utilisant le stockage creux implémenté dans la fonction `sparse` de Scilab.

Nous allons maintenant comparer les performances des deux implémentations.

Question 8

Construire le vecteur b de taille n^2 ne contenant que des 1. Comparer les temps d'exécution des produits $L \times b$ et $L_s \times b$ pour différentes valeurs de n . Donner une explication des résultats obtenus.

Pour aller plus loin

Le stockage creux utilisé par Scilab s'appelle le stockage ligne compressé (CSR). Il est l'un des stockages les plus efficaces pour les opérations d'algèbres linéaires.

3.5 Simulation du problème complet

Afin de simplifier les notations et de pouvoir adapter les méthodes numériques, nous allons identifier les parties linéaires et non linéaires dans chacune des équations. Soient les fonctions $F : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n$, $G : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n$ et $L : \mathbb{R}^n \mapsto \mathbb{R}^n$ des opérateurs définies par

$$\begin{cases} L(e) &= D A e \\ F(e, r) &= -k e (e - \alpha_1)(e - 1) - e r \\ G(e, r) &= \left(\varepsilon + \mu_1 \frac{r}{e + \mu_2} \right) (-r - k e (e - \alpha_2 - 1)) \end{cases} \quad (8)$$

On définit les vecteur E et R dépendant du temps par

$$\begin{cases} E(t) = (e_{1,1}(t), e_{2,1}(t), \dots, e_{n,n}(t)) \\ R(t) = (r_{1,1}(t), r_{2,1}(t), \dots, r_{n,n}(t)) \end{cases}$$

Le problème (4) se réécrit

$$\begin{cases} \frac{dE(t)}{dt} &= L(E(t)) + F(E(t), R(t)) \\ \frac{dR(t)}{dt} &= G(E(t), R(t)) \end{cases} \quad (9)$$

avec

$$\begin{cases} L(E(t)) &= A E(t) \\ F(E(t), R(t)) &= -k e_{i,j}(t) (e_{i,j}(t) - \alpha_1) (e_{i,j}(t) - 1) - e_{i,j}(t) r_{i,j}(t) \\ G(E(t), R(t)) &= \left(\varepsilon + \mu_1 \frac{r_{i,j}(t)}{e_{i,j}(t) + \mu_2} \right) (-r_{i,j}(t) - k e_{i,j}(t) (e_{i,j}(t) - \alpha_2 - 1)) \end{cases}$$

Nous avons donc un système d'équations différentielles du premier ordre non linéaire couplées à résoudre. En appliquant le schéma d'Euler explicite à (9), nous obtenons

$$\begin{cases} \frac{e_{i,j}^{k+1} - e_{i,j}^k}{dt} &= L(e_{i,j}^k) + F(e_{i,j}^k, r_{i,j}^k) \\ \frac{r_{i,j}^{k+1} - r_{i,j}^k}{dt} &= G(e_{i,j}^k, r_{i,j}^k) \end{cases} \quad (10)$$

Question 9

En utilisant les fonctions écrites précédemment, écrire un script Scilab permettant de résoudre le problème approché.

Avant de réaliser la simulation à proprement dit, il est nécessaire de vérifier que le script fonctionne correctement.

Pour cela, nous allons créer un problème artificiel pour lequel nous connaissons la solution analytique. Cette technique s'appelle la méthode des solutions manufacturées.

Par exemple, on souhaite que $u_a(x, y, t) = \cos(\pi x) \cos(\pi y) e^{-t}$ et $v_a = \cos(2\pi x) \cos(2\pi y) e^{-t}$ soient solutions du problème initial. Il est donc nécessaire d'ajouter pour chaque équation un terme correctif qui permette d'obtenir ces solutions.

Le problème initial est modifié de la manière suivante

$$\begin{cases} \frac{\partial e}{\partial t} = D\Delta e - ke(e - \alpha_1)(e - 1) - er + c_1(x, y, t) \\ \frac{\partial r}{\partial t} = \left(\varepsilon + \mu_1 \frac{r}{e + \mu_2} \right) (-r - ke(e - \alpha_2 - 1)) + c_2(x, y, t) \end{cases} \quad (11)$$

avec c_1 et c_2 des fonctions de $[0, 1]^2 \times [0, T]$.

Question 10

Calculer c_1 et c_2 de manière à ce que u_a et v_a soient solution du problème corrigé. Appliquer votre script en prenant en compte les fonctions de corrections.

3.6 Simulation et exploitation des résultats

Maintenant que nous nous sommes assurés de la validité du script de résolution, nous allons l'appliquer sur un cas physique. On considère la condition initiale suivante

$$u(x, y) = \begin{cases} 0 & \text{si } y < 0.5, x \in [0, 1] \\ 1 & \text{si } y \geq 0.5, x \in [0, 1] \end{cases} \quad v(x, y) = \begin{cases} 0 & \text{si } x < 0.5, y \in [0, 1] \\ 1 & \text{si } x \geq 0.5, y \in [0, 1] \end{cases}$$

Question 11

Appliquer votre script de simulation jusqu'à $T = 1500$. Attention, les calculs peuvent être assez long selon les valeurs de n que vous allez utiliser. Observer et commenter les résultats.

L'analyse de stabilité du système nécessaire pour déterminer le pas de temps optimal est relativement complexe. Celle ci donne la contrainte suivante

$$dt < \min\left(\frac{h^2}{4d + k\frac{h^2}{4}((\alpha_1 + 1)^2 + 1)}, \frac{1}{\varepsilon + \frac{\mu_1}{\mu_2}\frac{h^2}{4}((\alpha_1 + 1)^2)}\right)$$

Pour aller plus loin

Le modèle proposé à un comportement chaotique. Il est donc très sensible aux conditions initiales. Pour observer cette sensibilité, l'état initial du système peut-être pris comme étant l'état final d'une autre simulation plus courte.

4 Une méthode d'ordre supérieur

La condition sur le pas de temps est le facteur limitant pour traiter des problèmes de plus grande taille. Le but ici est de relaxer cette contrainte. Pour cela, nous avons besoin de mettre en place de nouveaux outils.

4.1 Splitting de Strang

Le splitting permet de séparer les équations aux dérivées partielles ayant une dépendance au temps en plusieurs équations aux dérivées partielles plus simple. Cette séparation permet d'améliorer la précision et la stabilité de la résolution, mais augmente le coût de la résolution. La méthode que nous allons utiliser est le splitting de Strang d'ordre deux. Pour cela, il faut introduire un temps intermédiaire $t^{k+1/2}$ dans l'intervalle $[t^k, t^{k+1}]$.

Pour le problème (4), le splitting de Strang s'écrit en trois étapes

- Dans l'étape 1, on résout le système d'équations différentielles non linéaires

$$\begin{cases} \frac{e_{i,j}^{k+1/2} - e_{i,j}^k}{dt} = F(e_{i,j}^k, r_{i,j}^k) \\ \frac{r_{i,j}^{k+1/2} - r_{i,j}^k}{dt} = G(e_{i,j}^k, r_{i,j}^k) \end{cases} \quad (12)$$

- Dans l'étape 2, on résout un système d'équation différentielles couplées par l'opérateur de diffusion

$$\frac{e_{i,j}^* - e_{i,j}^{k+1/2}}{dt} = L(e_{i,j}^{k+1/2}) \quad (13)$$

Le * ne correspond pas à un instant intermédiaire, mais à une variable temporaire que nous allons réutiliser.

- Dans la dernière étape, on résout de nouveau le système d'équations différentielles non linéaires, mais avec des conditions initiales différentes

$$\begin{cases} \frac{e_{i,j}^{k+1} - e_{i,j}^*}{dt} = F(e_{i,j}^*, r_{i,j}^{k+1/2}) \\ \frac{r_{i,j}^{k+1} - r_{i,j}^{k+1/2}}{dt} = G(e_{i,j}^*, r_{i,j}^{k+1/2}) \end{cases} \quad (14)$$

Question 12

Modifier le script de la question 10 afin de réaliser le splitting de Strang à l'ordre 2.

Le splitting ainsi construit est d'ordre 2, mais la méthode d'Euler est seulement d'ordre 1, ce qui détériore la qualité de la méthode.

4.2 Schémas temporels d'ordre supérieur

Afin d'augmenter l'ordre de la méthode, il est nécessaire d'avoir recours à des schémas temporels d'ordre supérieur. Les équations différentielles à résoudre sont de nature très différentes. Pour développer ces méthodes indépendamment du reste du simulateur 2D, il est préférable de repartir des scripts de la question 2. Dans ce qui suit, on considère l'équation différentielle (2).

4.2.1 Schéma explicite d'ordre 2

Le schéma explicite d'ordre 2 le plus simple s'appelle Runge-Kutta d'ordre 2 (RK2). Les méthodes de Runge-Kutta consiste à évaluer les dérivées de u en des points intermédiaires de l'intervalle $[t_k, t_{k+1}]$. Pour RK2, cela consiste à estimer

la dérivée de u au milieu de l'intervalle d'intégration. Pour l'équation de référence, la méthode de $RK2$ s'écrit

$$\begin{aligned} k_1 &= f(t_k, u^k) \\ k_2 &= f\left(t_k + \frac{dt}{2}, u^k + \frac{dt}{2}k_1\right) \\ u^{k+1} &= u^k + \frac{dt}{2}(k_1 + k_2) \end{aligned}$$

Question 13

Ecrire une fonction `rk2` qui prend comme argument d'entrée la condition initiale u_n , l'instant t_n , le pas de temps dt et une fonction f représentant le second membre. Cette fonction implémentera le calcul d'un pas de temps de la méthode de Runge-Kutta d'ordre 2.

4.2.2 Schéma implicite d'ordre 2

Le schéma de discrétisation temporel d'ordre 2 implicite que nous allons utiliser est le schéma de Crank-Nicholson. Il s'obtient en combinant les schémas d'Euler explicite et implicite. Pour l'équation (2), le schéma de Crank-Nicholson s'écrit

$$u^{k+1} = u^k + \frac{dt}{2}(f(t_k, u^k) + f(t_{k+1}, u^{k+1}))$$

Le schéma est dit implicite car il nécessite de connaître $f(t_{k+1}, u^{k+1})$ pour calculer u^{k+1} . Dans le cas général, il faudrait utiliser une méthode de point fixe, par exemple la méthode de Newton, pour déterminer la solution à l'instant t_{k+1} . Nous allons supposer ici que la fonction f est linéaire en u , c'est à dire que $f(t, u) = -\kappa(t)u$ où $\kappa \in \mathcal{C}^0([0, T])$ et $\forall t \in [0, T], \kappa(t) \geq 0$.

Sous ces conditions, le schéma de Crank-Nicholson se réécrit

$$u^{k+1} = \frac{2 - dt\kappa(t_k)}{2 + dt\kappa(t_{k+1})}u^k$$

Question 14

Ecrire une fonction `cn1` qui prend comme argument d'entrée la condition initiale u_n , l'instant t_n , le pas de temps dt et une fonction f représentant le second membre. Cette fonction implémentera le calcul d'un pas de temps de la méthode de Crank-Nicholson.

4.3 Utilisation des schémas temporels

Les schémas de la section précédente possèdent des propriétés et des contraintes très différentes. Ils ne pourront donc pas s'appliquer sur les mêmes étapes.

Ainsi, pour les étapes 1 et 3 du splitting de Strang, nous allons utiliser la méthode $RK2$, et pour l'étape 2, nous appliquerons la méthode de Crank-Nicholson.

Le nouveau schéma temporel s'écrit

- Dans l'étape 1, on résout le système d'équations différentielles non linéaires

$$\begin{cases} e_{i,j}^{k+1/2} &= e_{i,j}^k + \frac{dt}{2} \left(F(e_{i,j}^k, r_{i,j}^k) + F\left(e_{i,j}^k + \frac{dt}{2}F(e_{i,j}^k, r_{i,j}^k), r_{i,j}^k + \frac{dt}{2}F(e_{i,j}^k, r_{i,j}^k)\right) \right) \\ r_{i,j}^{k+1/2} &= r_{i,j}^k + \frac{dt}{2} \left(G(e_{i,j}^k, r_{i,j}^k) + G\left(e_{i,j}^k + \frac{dt}{2}G(e_{i,j}^k, r_{i,j}^k), r_{i,j}^k + \frac{dt}{2}G(e_{i,j}^k, r_{i,j}^k)\right) \right) \end{cases} \quad (15)$$

- Dans l'étape 2, on résout un système d'équation différentielles couplées par l'opérateur de diffusion

$$-\frac{dt}{2}L(e_{i,j}^*) + e_{i,j}^* = e_{i,j}^{k+1/2} + \frac{dt}{2}L(e_{i,j}^{k+1/2}) \quad (16)$$

- Dans la dernière étape, on résout de nouveau le système d'équations différentielles non linéaires, mais avec des conditions initiales différentes

$$\begin{cases} e_{i,j}^{k+1} &= e_{i,j}^* + \frac{dt}{2} \left(F(e_{i,j}^*, r_{i,j}^{k+1/2}) + F\left(e_{i,j}^* + \frac{dt}{2}F(e_{i,j}^*, r_{i,j}^{k+1/2}), r_{i,j}^{k+1/2} + \frac{dt}{2}F(e_{i,j}^*, r_{i,j}^{k+1/2})\right) \right) \\ r_{i,j}^{k+1} &= r_{i,j}^{k+1/2} + \frac{dt}{2} \left(G(e_{i,j}^*, r_{i,j}^{k+1/2}) + G\left(e_{i,j}^* + \frac{dt}{2}G(e_{i,j}^*, r_{i,j}^{k+1/2}), r_{i,j}^{k+1/2} + \frac{dt}{2}G(e_{i,j}^*, r_{i,j}^{k+1/2})\right) \right) \end{cases} \quad (17)$$

En utilisant la définition de L , l'équation (16) se réécrit

$$\left(-\frac{dtD}{2}A + Id\right) e_{i,j}^* = \left(\frac{dtD}{2}A + Id\right) e_{i,j}^{k+1/2} \quad (18)$$

où Id est la matrice identité de taille n^2 .

Pour pouvoir appliquer la méthode de Crank-Nicholson, il faut donc résoudre un système linéaire. Comme la matrice à inverser ne change pas au cours du temps, on peut utiliser la méthode LU pour résoudre le système linéaire.

Question 15

Ecrire une fonction `cn2` qui implémente la méthode de Crank-Nicholson pour un opérateur linéaire.

Pour aller plus loin

La décomposition LU de Scilab est réalisée par la bibliothèque `UMFPACK`. Scilab offre également la possibilité de résoudre les systèmes linéaires en utilisant un des solveurs itératifs vu en cours.

4.4 Simulation de modèle à l'ordre 2

Il s'agit maintenant de combiner les différentes briques afin d'obtenir un simulateur d'ordre plus élevé.

Question 16

En utilisant les fonctions `rk2` et `cn2` écrites précédemment, écrire un script Scilab permettant de résoudre le problème (4) par un splitting de Strang à l'ordre 2.

A ce stade il est important de vérifier que le script résout le système correctement. Pensez à réutiliser les éléments des solutions manufacturées.

Question 17

Appliquer votre script de simulation jusqu'à $T = 1500$. Attention, les calculs peuvent être assez long selon les valeurs de n que vous allez utiliser. Observer et commenter les résultats.

Pour aller plus loin

La non-linéarité des étapes 1 et 3 nous force à utiliser un schéma explicite d'ordre 2. Il est possible d'utiliser des schémas de Runge-Kutta implicites spécialement conçus pour les problèmes raides. L'un d'entre eux est `SDIRK` d'ordre 2. Il implique l'utilisation d'une méthode de point fixe à chaque itération en temps, ce qui peut éventuellement le rendre

coûteux.

4.5 Notion de performance

Dans la réalisation des scripts et le choix des méthodes, nous avons uniquement considéré l'ordre et la stabilité des méthodes. L'un des critères qui a été ignoré est celui de la performance. Pour comparer les performances, il est nécessaire de réaliser une analyse de complexité pour chaque pas de temps et de calculer le nombre de pas de temps nécessaire pour atteindre l'état final.

Question 18

Pour chaque script, réaliser une évaluation du coût algorithmique et comparer les résultats en fonction de n le nombre de cellules dans une direction.

Pour aller plus loin

Dans les cas réels, le nombre de cellules dans chaque direction est de l'ordre de 3000. Pour simuler les modèles, il faut avoir recours au calcul haute performance, soit sur des grappes de calculs ou sur des cartes graphiques.

Vous venez de réaliser votre premier programme de simulation numérique. L'histoire ne s'arrête pas ici, et le sujet vous offre de nombreuses possibilités d'exploration. N'hésitez pas à explorer et à mettre en valeur votre travail.