

Résolution numérique d'EDO et d'EDP.

1- EDO

Soit l'équation différentielle

$$\begin{cases} x' = -y \\ y' = x \end{cases}$$

On appelle f_1 la fonction vectorielle telle que $f_1 : (x, y) \rightarrow (-y, x)$. L'équation différentielle peut donc s'écrire sous la forme vectorielle $(x', y') = f_1(x, y)$.

- ▷ Dans le cas continu, quelle est la solution analytique $(x(t), y(t))$ étant donnée la condition initiale $(x(0), y(0)) = (x_0, y_0)$.

1.1 Fonction du système

Considérez le répertoire `edo/partie_01`.

Le fichier `f1.m` est prévu pour définir la fonction $f_1 : (x, y) \rightarrow (-y, x)$.

- ▷ Complétez cette fonction.
- ▷ Créez dans le même répertoire un fichier `pgm.m` qui fait contenant le code suivant:

```
clear;
clf;

[fx,fy] = f1(1,2);
```

- ▷ Assurez vous que les variables f_x et f_y valent bien respectivement -2 et 1 .

1.2 Champs de vecteur

Considérez le répertoire `edo/partie_02`.

- ▷ Copiez votre fichier `f1.m` dans ce répertoire.

Le code du fichier `pgm.m` permet d'échantillonner la fonction f_1 pour différents points du plan et affiche le résultat vectoriel de celle-ci en tant que champ de vecteurs. Notez l'utilisation de la fonction `quiver` permettant l'affichage d'un champ de vecteurs.

```
clear;
clf;

%Nombre d echantillons
N = 20;

%Vecteur unidimensionnelles suivant x et y
ux = linspace(-1,1,N);
uy = linspace(-1,1,N);

%Tableaux 2D de coordonnées (x,y)
[x,y] = meshgrid(ux,uy);

%Calcul de (fx,fy) pour toutes les valeurs de (x,y)
[fx,fy] = f1(x,y);

%Affichage de fx et fy sous forme de champ de vecteurs
quiver(x,y,fx,fy);
```

- ▷ Observez l'influence sur la visualisation lorsque vous modifiez le nombre d'échantillons.
- ▷ Dans le cas général, il peut être intéressant de visualiser la direction du champ de vecteur indépendamment de son amplitude. Pour cela, faites en sorte de normaliser l'ensemble des vecteurs des variables f_x et f_y . Observez le résultat obtenu. Assurez-vous de réaliser cette opération de manière vectorielle sous Matlab/Octave et non pas en faisant des boucles.

1.3 Intégration numérique, Euler explicite

- ▷ Créez un fichier `euler.m`.
- ▷ Ce fichier doit contenir une fonction permettant d'appliquer une étape d'intégration numérique suivant la méthode d'Euler explicite. Pour cela, la signature de la fonction sera la suivante

```
function [x_out,y_out] = euler(x_in,y_in,F,dt)
%
% x_in: Paramètre x d'entrée
% y_in: Paramètre y d'entrée
% F    : Fonction correspondant au système différentiel
% dt   : Pas d'intégration temporel
%
```

- ▷ Appelez une étape de la fonction Euler depuis votre programme `pgm.m` en vous servant de la syntaxe suivante

```

%Pas d'integration temporel
dt = 0.1;
%"Pointeur" de fonction sur f1
F=@f1;

%Conditions initiales
x0 = 1; y0 = 2;

%Appel de l'integration numérique
[x,y] = euler(x0,y0,F,dt);

```

- ▷ Vérifiez que x, y valent respectivement 0.8, 2.1.
- ▷ Créez un fichier `integration.m` qui contiendra la fonction suivante

```

function [trajectory_x,trajectory_y] = integration(x0,y0,F,integrator,dt,N_step)
%
% x0 : position initiale en x
% y0 : position initiale en y
% F : Fonction a integrer
% integrator : Méthode d'integration
% dt : Pas d'integration temporel
% N_step : Nombre d'étapes d'integrations
%
% trajectory_x : Ensemble des N_step positions en x suivant l'integration
% trajectory_y : Ensemble des N_step positions en y suivant l'integration
%

```

Cette fonction consiste à itérer N_{step} fois l'opération

$$(x(t + \Delta t), y(t + \Delta t)) = \text{integrator}(x(t), y(t)) ,$$

et à stocker chaque valeur $(x(t), y(t))$ intermédiaire dans les deux tableaux `trajectory`. En d'autres termes, cela revient à stocker une approximation des lignes intégrales définies par

$$(x(t), y(t)) = \int_{u=0}^{u=t} f_1(x(u), y(u)) \, du .$$

- ▷ Codez la fonction `integration`.
- ▷ Considérez le répertoire `edo/partie_03` et copiez-y vos fichiers `f1.m`, `euler.m`, `integration.m`.
- ▷ Le programme `pgm` appelle la fonction `integration` et affiche le résultat en tant que courbe sur le champ de vecteur.
- ▷ Votre résultat devrait avoir la forme d'une spirale. Est-ce le résultat attendu ? La spirale est elle convergente ou divergente ? Interprétez pourquoi vous obtenez ce résultat.

1.4 Runge Kutta

- ▷ Afin d'améliorer le résultat précédent, codez dans un fichier `rk2.m` la fonction d'intégration numérique de Runge Kutta d'ordre 2. Les paramètres d'entrées et de sorties devront être similaires aux cas de la fonction `euler`.
- ▷ Observez le résultat de l'intégration en passant par l'approche de Runge Kutta d'ordre 2. Pour cela, vous pouvez modifier la ligne `integrator = @euler`; en `integrator = @rk2`; dans `pgm.m`. Vous devriez obtenir une courbe ressemblant cette fois à un cercle.
- ▷ Calculez l'évolution de la distance des échantillons par rapport à l'origine. Concluez sur le caractère convergent ou divergent de la méthode d'intégration.

1.5 Équation de Lokta Volterra

- ▷ Similairement à la fonction `f1`, créez un nouveau fichier `f2.m` ainsi que la fonction associée décrivant l'équation de Lokta Volterra. On pourra considérer $\alpha = 1, \beta = 0.5, \gamma = 2, \delta = 1$.
- ▷ Modifiez dans le code de `pgm` la fonction utilisée afin de la placer sur `f2`.
- ▷ Observez les trajectoires solutions pour différentes valeurs initiales. Pensez à changer les valeurs (x_0, y_0) initiales et à changer également les intervalles de définition du champ de vecteurs.
- ▷ Pour quelles valeurs, la population de proie et de prédateur reste constante au cours du temps. Quelles valeurs de (x_0, y_0) doit on prendre en fonction de $\alpha, \beta, \gamma, \delta$.

2- EDP

2.1 Equation de diffusion

Soit l'équation aux dérivées partielles

$$\frac{\partial f}{\partial t}(x, t) = \frac{\partial^2 f}{\partial x^2}(x, t) . \quad (1)$$

On considèrera dans le cas de ce TP $x \in [0, 1]$, les conditions limites¹ $f(0, t) = f(1, t) = 0$, et $f(x, 0) = f_0(x)$. On considèrera $f_0(x) = 1$ pour $x \in [a, b] = [0.4, 0.6]$, et $f_0(x) = 0$ sinon.

Cette équation est appelée équation de diffusion. Elle peut correspondre en particulier à l'évolution temporelle de la chaleur le long d'un axe x (ici avec un coefficient de diffusion de 1). Dans le cas présent cela revient étudier l'évolution de la température d'une barre à 0 degrés sauf sur la partie centrale. La température est laissée libre d'évoluer sauf sur les points extrêmes pour lequel la température est fixe et vaut 0 degrés.

¹Des conditions sur la valeur de f fixes sont appelées des conditions de Dirichlet

2.1.1 Intégration explicite

Supposons que l'on discrétise l'équation en temps et en espace de la manière suivante:

$$\frac{\partial f}{\partial t}(x, t) \simeq \frac{f(x, t + \Delta t) - f(x, t)}{\Delta t}$$

$$\frac{\partial^2 f}{\partial x^2}(x, t) \simeq \frac{f(x + \Delta x, t) - 2f(x, t) + f(x - \Delta x, t)}{(\Delta x)^2}$$

- ▷ Donnez la relation permettant d'écrire la relation discrète donnant $f(x, t + \Delta t)$ en fonction des valeurs de f à l'instant t .

Considérez le répertoire `edp/01_diffusion/partie_01/`. Ce répertoire contient le programme `pgm.m` qui initialise la fonction discrétisée suivant un pas d'échantillonnage spatial et temporel donné en tant que variable. Il réalise N étapes d'intégrations temporelles en appelant la fonction `evolve`. Cette fonction que vous devez coder dans le fichier `evolve.m` doit implémenter la formule donnant les valeurs de $f(x, t + \Delta t)$ en fonction des valeurs de f à l'étape courante.

- ▷ Implémentez la fonction `evolve`. Assurez-vous qu'au bout de 10 itérations la courbe résultante soit une version lissée de la courbe d'origine. Prenez soin dans la mesure du possible d'éviter des boucles dans cette fonction, préférez une écriture vectorielle plus rapide.
- ▷ En gardant la valeur du pas d'échantillonnage spatial fixe, augmentez le pas d'échantillonnage temporel. Observez qu'au-dessus d'une certaine valeur, le système d'intégration devient divergent.
- ▷ Quelle est la relation à vérifier entre Δt et Δx pour que la méthode de résolution converge ? Cette contrainte vous semble-t-elle simple à vérifier pour réaliser des calculs avec une précision spatiale importante ?

Considérez désormais le répertoire `edp/01_diffusion/partie_02/`. Copiez-y votre fichier `evolve.m`. Cette fois le programme `pgm.m` affiche les courbes d'évolutions de la température à des intervalles de temps fixe. Le pas de temps est calculé de manière automatique.

- ▷ Vérifiez que ce programme affiche bien différentes étapes du processus de diffusion.
- ▷ Comment se comporte le programme si vous souhaitez obtenir un calcul dix ou cinquante fois plus précis en termes de résolution spatiale ? Expliquez la raison.

2.1.2 Intégration implicite

Considérons désormais le schéma de résolution implicite où l'on approxime l'équation (1) par la relation discrète suivante

$$\frac{f(x, t + \Delta t) - f(x, t)}{\Delta t} = \frac{f(x + \Delta x, t + \Delta t) - 2f(x, t + \Delta t) + f(x - \Delta x, t + \Delta t)}{(\Delta x)^2}.$$

Considérons le vecteur $v(t)$ tel que l'entrée $k \in [0, N - 1]$ soit $v_k(t) = f(x + k\Delta x, t)$, avec N le nombre d'échantillons contenu dans le vecteur.

- ▷ Montrez que la relation précédente peut s'écrire sous la forme d'une relation matricielle

$$M v(t + \Delta t) = v(t) ,$$

avec M une matrice $N \times N$ que l'on précisera.

- ▷ Quelle caractéristique notable possède la matrice M ?

L'utilisation de l'approche implicite consiste donc ici à résoudre à chaque pas de temps un système linéaire de dimension $N \times N$ afin de trouver les valeurs de f aux instants $t + \Delta t$.

On rappelle que pour résoudre un système linéaire sous Matlab de la forme $Mx = y$, il est important de

- Stocker votre matrice M en dans une structure de données de matrice creuse (sparse) si celle-ci contient de nombreux zéros. N'oubliez pas qu'une matrice creuse n'enregistre que les valeurs non nulles, contrairement à une matrice pleine qui stocke l'ensemble des valeurs de la matrice.
 - Résoudre votre système linéaire en appelant la commande $M \backslash y$, et surtout pas en appelant explicitement le calcul de l'inverse de la matrice² $M^{-1} y$ qui reviendrait à calculer de manière explicite la matrice inverse pleine.
- ▷ Modifiez votre programme pour réaliser cette intégration implicite. Pour cela, vous
- Pré-calculez la matrice M avant de commencer les itérations.
 - Calculez itérativement la fonction à partir de la résolution du système linéaire à la place de l'utilisation de la fonction `evolve`.
 - Vérifiez que vous obtenez un résultat quasi identique au cas du calcul par méthode explicite dans les conditions données initialement avec $\Delta x = 0.01$ et $\Delta t = 4 \cdot 10^{-5}$.
 - Sans modifier Δx , multipliez par 10 la valeur de Δt . Comment se comporte désormais votre résolution ? Que se serait-il passé dans le cas d'une résolution explicite.
 - Fixez désormais Δx à 10^{-4} et Δt à $5 \cdot 10^{-4}$. Observez visuellement que votre résultat reste correct. Quelle est la taille de la matrice M dans ce cas ?
 - Comparez le temps de calcul entre l'évaluation de $M \backslash v$, et le calcul de `inv(M)`.

2.2 Équation de propagation

Soit l'équation aux dérivées partielles

$$\frac{\partial^2 f}{\partial t^2}(x, t) - \frac{\partial^2 f}{\partial x^2}(x, t) = 0 . \quad (2)$$

Cette équation, dite de propagation, permet de modéliser par exemple le comportement de propagation d'ondes (ici se déplaçant à une vitesse de 1 m/s) où f représente l'amplitude de l'onde.

²Les problèmes d'analyse numérique réels demandant explicitement l'inversion de matrice sont très rares.

On considérera dans le cadre du TP la condition initiale $\forall t \leq 0, f(x, t) = f_0(x)$ pour $x \in [0, 1]$. Et les conditions aux limites $f(0, t) = f(1, t) = 0$. La solution analytique de cette équation étant donnée les conditions initiales et limites est donnée par $f(x, t) = \frac{1}{2}(f_0(x - t) + f_0(x + t))$. C'est-à-dire deux ondes se propageant vers l'avant et l'arrière à vitesse constante. Cette solution est valide jusqu'à ce que l'onde rencontre le bord du domaine. Une fois que l'onde touche le bord, la condition de Dirichlet impose l'apparition d'une onde symétrique modélisant la réflexion de l'onde sur celui-ci. Dans le cas théorique, les deux ondes opposées se propagent ainsi d'un bord à l'autre en se réfléchissant indéfiniment sur les bords.

- ▷ Discrétisez cette équation suivant un schéma d'Euler explicite. Donnez l'expression de $f(x, t + \Delta t)$ en fonction de $f(x, t)$ et $f(x, t - \Delta t)$.

Le répertoire `edp/02_propagation/` contient un programme permettant de traiter et d'enregistrer les différentes étapes de la propagation. La fonction `condition_initiale` permet d'initialiser la condition initiale sous la forme d'une gaussienne.

- ▷ Quelle est la forme théorique attendue des courbes au cours des itérations.
- ▷ Complétez l'intégration numérique suivant le schéma d'Euler explicite. Assurez-vous de garder $\Delta t / \Delta x < 1$ (condition de Courant-Friedrichs-Lewy).
- ▷ Les courbes sont sauvegardées sous forme d'images dans le répertoire `output/`. Vous pouvez utiliser le script `create_anim.sh` qui lit l'ensemble des images de `output/` et les concatène dans un fichier `.gif` animé permettant de visualiser la propagation sous forme d'animation.