Fiche de TP2 : Mise en oeuvre de schémas à un pas implicites : estimation d'ordre et comparaisons

Exercice-1 : Modèles

On se propose dans cette fiche de mettre en oeuvre un schéma à un pas implicite. Sans nuire à la généralité, nous allons considérer les schémas dits θ-schémas.

L'équation différentielle ordinaire que nous considérons s'écrit sous la forme suivante :

$$\begin{cases} X'(t) = F(t, X(t)), \quad t \in]t_0, t_0 + T[. \\ X(t_0) = X^0 \in \mathbb{R}^d. \end{cases} \tag{1}$$

où F est une fonction donnée dans $C^1(\mathbb{R} \times \mathbb{R}^d, \mathbb{R}^d)$, $X^0 \in \mathbb{R}^d$ et où T est strictement positif et $t_0 \in \mathbb{R}$. On suppose que l'on dispose des dérivées partielles de $Y \mapsto F(t, Y)$.

Afin de résoudre numérique ce problème, on se donne $N \in \mathbb{N}^*$ et on pose $h = \frac{T}{N}$ le pas de discrétisation ce qui définit les instants $t_n = t_0 + nh, n = 0, \ldots, N$. On construit alors la suite X_n des valeurs approchées de $X(t_n)$ par le schéma :

$$\begin{cases} X_{n+1} &= X_n + h\big((1-\theta)F(t_n, X_n) + \theta F(t_{n+1}, X_{n+1})\big), \; 0 \leq n \leq N-1 \\ X_0 &= X^0, \end{cases} \tag{2}$$

appelé θ-schéma, où $\theta \in [0, 1]$

Q-1-1 : Montrer qu'à chaque itération n la détermination de X_{n+1} se ramène à la résolution d'un problème non linéaire que l'on peut écrire sous les deux formes suivantes :

- **Forme 1**
$$\mathcal{N}(X_{n+1}, F, X_n, t_n, h, \theta) = 0. \tag{3}$$
- **Forme 2**
$$X_{n+1} = \mathcal{P}(X_{n+1}, F, X_n, t_n, h, \theta). \tag{4}$$

Q-1-2 : Exprimer la matrice jacobienne de l'application $Z \mapsto \mathcal{N}(Z, F, X_n, t_n, h, \theta)$ en fonction des dérivées partielles de F par rapport à sa second variable.

Q-1-3 : Décrire l'algorithme de Newton pour déterminer la solution X_{n+1} de (3). L'algorithme sera initialisée par X_n (solution à l'instant t_n), utilisera un critère d'arrêt défini par un paramètre tol$(ou\ \varepsilon)$ (voir Cours M315) et imposera un nombre maximum `IterMax` d'itérations. On n'oubliera pas de vérifier que les matrices jacobiennes utilisées sont inversibles. (*On pour à ce stade passer à la question* **Q.2.1** *pour sa mise en oeuvre*).

Q-1-4 : Décrire la méthode de point fixe pour déterminer la solution X_{n+1} de (4). L'algorithme sera initialisée par X_n, et utilisera un critère d'arrêt défini par un paramètre tol$(ou\ \varepsilon)$ (identique à celui de Newton) et imposera un nombre maximum `IterMax` d'itérations. (*On pour à ce stade passer à la question* **Q.2.2** *pour sa mise en oeuvre*).

Exercice-2 : Mise en oeuvre

Pour une mise en oeuvre efficace du schéma numérique on adopte les conventions suivantes :

- L'EDO à résoudre est définie à travers une fonction `Matlab` de prototype

```
function [F, DF] = monEDO (t,y)
% F  est  la valeur de la fonction second-membre f  au point (t,y)
% DF est la dériée partielle (matrice jacobienne) par rapport a y de la f au point (t,y) (i.e df/dy )
```

- Un schéma numérique à un pas est défini par la donnée d'une fonction qui explique comment avancer d'un pas . Elle a pour prototype :

```
function [t1, x1, h1] = monSchema(f,t0,x0,h0)
%% ENTREES :
% f    : fonction second-membre f i.e  l' edo à résoudre (voir listing precedent : monEDO)
% t0   : est l'instant présent    (tn)
% x0   : la solution à l'instant present (xn)
% h    : le pas de discrétisation
%% SORTIES :
% t1   : l'instant suivant  (tn+1)
% x1   : la solution à l'instant suivant (xn+1)
% h    : le prochain pas à utiliser
```

Ainsi par exemple, le schéma d'Euler explicite s'écrira :

```
function [t1, x1, h1] = monSchemaEulerExplicite(f,t0,x0,h0)
t1 = t0 + h0;
x1 = x0 + h0 * f(t0,x0);
h1 = h0 ;
end
```

- La résolution effective de l'EDO se fait à travers la fonction suivante, qui fait appel au schéma numérique :

```
function [t,x,h]= solveurEDO(f,t0,tf,x0,N,monSchema)
%% ENTREES :
% f    : fonction second-membre f EDO à résoudre (voir listing precedent : monEDO)
% t0   : est l'instant initial
% tf   : instant final
% x0   : la solution initiale
% N    : le nombre de subdivisions à utiliser
%% SORTIES :
% t    : la suite des instants (tn)
% x    : la suite des solutions (xn) aux instants (tn)
% h    : le pas h de discretisation utilisé
t_temp = linspace(t0,tf,N+1);
h = t_temp(2) - t_temp(1);
t_temp = [];
x = [x0];
t = [t0];
for k = 1:N
%calcul
 [t1,x1,hh] = monSchema(f,t(k),x(:,k),h);
%stockage
  x = [x,x1];
  t = [t,t1];
end
end
```

Ainsi, on pourra obtenir la résolution d'un problème par le schéma d'Euler explicite en procédant ainsi :

```
[t,x,h] = solveurEDO(f,t0,tf,x0,N,@monSchemaEulerExplicite)
```

Q-2-1 : Ecrire à présent une fonction `Matlab`

```
function [t1, x1, h1] = monThetaSchemaNewton(f, t0, x0, h0, tol, IterMax, theta)
```

qui avance le θ-schéma d'un pas en utilisant un algorithme de Newton pour déterminer la solution du problème non linéaire en X_{n+1} dans le θ-schéma.

Q-2-2 : Ecrire à présent une fonction Matlab

```
function [t1, x1, h1] = monThetaSchemaPicard(f, t0, x0, h0, tol, IterMax, theta)
```

qui avance le θ-schéma d'un pas en utilisant la méthode de point fixe pour déterminer la solution du problème non linéaire en X_{n+1} dans le θ-schéma.

Exercice-3 : Validations : Evaluation de l'ordre

On considère pour la validation le système d'EDOs suivant :

$$\left\{ \begin{array}{lcl} x' & = & (x^2+y^2)y \\ y' & = & -(x^2+y^2)x. \end{array} \right. \tag{5}$$

Avec $t_0 = 0, T = \pi, (x(0), y(0)) = (1, 0)$

Q-3-1 : Vérifier que la solution exacte du problème est $(x(t), y(t)) = (cos(t), -sin(t))$.

Dans toute la suite on désignera par $e_N^\theta = \|(-1,0) - X_N\|$ l'erreur commise au dernier point de calcul par le θ-schéma, avec N subdivisions.

Q-3-2 : Dans le cas où la méthode de Newton est utilisée avec `IterMax` $= 10$ et $tol = 1e-3$, tracer sur un même graphique la solution exacte et la solution fournie par le θ-Schéma lorsque $N = 50$, et $\theta \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$.

Q-3-3 : Reprendre la question précédente lorsque la méthode du point fixe est utilisée avec les mêmes paramètres, que la méthode de Newton.

Q-3-4 : On considère ici le θ-schéma avec la méthode de Newton où l'on a imposé : `IterMax` $= 10$ et $tol = 1e-3$. Tracer en échelle logarithmique e_N^θ en fonction de h, pour $\theta = 0$, $\theta = 0.5$, $\theta = 1$ et $N = 2^k, 1 \leq k \leq 12$. Que peut-on déduire sur l'ordre de précision des θ-schémas.

Q-3-5 : On considère à présent le θ-schéma avec la méthode du point fixe, où l' on a décidé de n'effectuer qu'une seule itération dans la méthode du point fixe. Tracer en échelle logarithmique e_N^θ en fonction de h, pour $\theta = 0$, $\theta = 0.5$, $\theta = 1$ et $N = 2^k, 1 \leq k \leq 12$. Que constatez-vous ?

Exercice-4 : Validations : Comparaisons

Le problème modèle est encore celui de l'exercice précédent. On suppose cette fois $\theta = 0.5$, dans ce cas le θ-schéma est connu sous le nom de schéma de Crank-Nicolson.

Comparer numériquement les trois schémas suivants ((*on pourra estimer leur ordre de convergence*)) :

1. **Schéma 1** : Le Schéma de Crank-Nicolson dans lequel la méthode de Newton est utilisée pour résoudre les problèmes non-linéaires, avec les paramètres `IterMax = 10`, et `tol = 1e-3`.
2. **Schéma 2** :

$$\left\{ \begin{array}{rcl} X_{n+1}^* & = & X_n + hF(t_n, X_n), \\ X_{n+1} & = & X_n + \frac{h}{2}\left[F(t_n, X_n) + F(t_{n+1}, X_{n+1}^*)\right], \ 0 \leq n \leq N-1 \\ X_0 & = & X^0. \end{array} \right. \tag{6}$$

3. **Schéma 3** :

$$\left\{ \begin{array}{rcl} X_{n+1}^* & = & X_n + \frac{h}{2}\left[F(t_n, X_n) + F(t_{n+1}, X_n)\right], \\ X_{n+1} & = & X_n + \frac{h}{2}\left[F(t_n, X_n) + F(t_{n+1}, X_{n+1}^*)\right], \ 0 \leq n \leq N-1 \\ X_0 & = & X^0. \end{array} \right. \tag{7}$$

4. **Schéma 4** :

$$\left\{ \begin{array}{rcl} X_{n+1}^* & = & X_n + hF(t_n + \frac{h}{2}, X_n + \frac{h}{2}F(t_n, X_n)), \\ X_{n+1} & = & X_n + \frac{h}{2}\left[F(t_n, X_n) + F(t_{n+1}, X_{n+1}^*)\right], \ 0 \leq n \leq N-1 \\ X_0 & = & X^0. \end{array} \right. \tag{8}$$