TP n°3 – Résolution numérique d'EDO par Méthodes Itératives

Durée : 3 séances de 2h.

Objectif du TP :

Construire un solveur fiable de systèmes d'Equations Différentielles Ordinaires (EDO) d'ordre n : $y^{(n)} = f(t, y, y', y", \dots, y^{(n-1)})$ avec Conditions Initiales (CI).

Par modification du problème initial, on se ramène souvent d'une EDO d'ordre n à une **EDO d'ordre 1 : $\boldsymbol{Y' = F\big(t, Y(t)\big)}$.**

Dans ce TP, il est demandé de **programmer** des méthodes de résolution numérique d'ODE d'ordre 1, les **tester** pour des EDO simples en **comparant** graphiquement les solutions numériques aux solutions analytiques. Ensuite, tester la **stabilité** et la **convergence** des différentes méthodes numériques.

Séance 1 (2 heures) :

1- Euler Explicite (2h)

Vous allez programmer la méthode d'Euler pour la résolution de l'EDO suivante :

$$\text{EDO n°1:} \quad \begin{cases} y'(t) = 2t.y(t) \\ y(0) = 1 \end{cases}$$

a. Exprimer la **solution analytique** de cette EDO.

b. Rappeler la **méthode numérique d'Euler Explicite** : écrire l'algorithme correspondant.

c. Retrouvez le fichier « def_fonc_ode.sci » envoyé par mail, qui contient la fonction $f_1\big(t, y(t)\big) = 2t.y(t)$. Ce fichier prend pour argument un réel t et un vecteur y, et renvoie un vecteur f_1.
Retrouvez également le fichier « EulerExpl.sci », dans lequel vous programmerez la méthode d'Euler Explicite. Il contient déjà un début de programme pour vous aider à démarrer, et surtout une syntaxe d'appel que vous devez respecter. Ce fichier prend pour arguments la fonction f_1, l'intervalle de temps voulu pour la solution $Range_t = [a, b]$, le pas de temps numérique h et la condition initiale y_0, puis renvoie deux vecteurs, le vecteur temps, t, et le vecteur solution, y.

d. **Programmer** la méthode d'Euler explicite.
Attention : y_0 doit être vectoriel, votre solveur doit fonctionner pour un **système** d'équations, donc traiter données et résultats vectoriellement : vecteur de conditions initiales, vecteur solution à chaque pas de temps.

e. **Tester** votre programme pour l'EDO n°1 pour un pas de temps $h = 0.1$ et sur l'intervalle de temps $[0,1]$: tracer dans le même graphique la solution numérique et la solution analytique. Que remarquez-vous ?
Faites varier le pas de temps, que remarquez-vous ?

f. **Scilab** possède déjà des solveurs de systèmes d'EDO : tapez "help EDO".
Tester la fonction « edo » de Scilab sur l'EDO n°1.

g. **Tracer** sur le même graphique, pour $t \in [0,1]$, les 3 solutions : analytique, par méthode d'Euler Explicite et par la fonction « edo » de Scilab. Que remarquez-vous ?

h. Ecrire un nouveau **fichier "principal"**, « Calcul_Y.sci », dans lequel vous intégrez les instructions d'appels à la méthode d'Euler, à la fonction ode de Scilab, qui calcule la solution analytique et trace tous les graphiques. Que peut-on définir comme arguments d'entrée de ce fichier ?
Tester ce fichier.

A rendre – 1 :

- Dans le corps du mail : copiez le **contenu de vos programmes** clairs et commentés : « def_fonc_ode.sci », « EuleExpl.sci », « Calcul_Y.sci » (ce dernier contiendra l'appel à la fonction EulerExplicite et les instructions de plot). Pas de fichier attaché !

- Pour **deux valeurs différentes du pas de temps** h (0.1 et 0.04) : deux **figures** dans lesquelles sont tracées : en trait plein bleu la solution analytique, en pointillés rouges la solution par Euler Explicite, et utilisez une autre couleur pour la solution calculée par le solveur de Scilab.

Séance 2 (2 heures) :

2- Euler Explicite pour EDO du 2nd ordre (1h30)

On fait la même chose que pour l'EDO n°1 pour l'EDO suivante :

$$\text{EDO n°2 :} \quad \begin{cases} y''(t) = y(t) \\ y(0) = 1 \\ y'(0) = 0 \end{cases}$$

a. Exprimer la solution analytique de cette EDO.

b. Réécrire l'EDO n°2 d'ordre 2 sous la forme d'une **EDO du 1er ordre** : $Y'(t) = f_2(t, Y)$, avec $\boldsymbol{Y}$ et $\boldsymbol{f_2}$ deux vecteurs **colonnes**.

c. Dans le même fichier « def_fonc_ode.sci », définir la fonction f_2.

d. Tester la méthode d'Euler Explicite pour cette EDO.
Tester la fonction « edo » de Scilab.

e. Tracer sur le même graphique les 3 solutions : analytique, par méthode d'Euler Explicite et par la fonction « edo » de Scilab. (Utiliser le fichier Calcul_Y.sce). Que remarquez-vous ?

f. Faites varier le pas de temps. Que remarquez-vous ?

3- Euler Prédicteur-Correcteur (ou Euler Modifiée) (30min)

a. Rappeler la méthode numérique d'**Euler Implicite**.

b. Rappeler la méthode d'**Euler Prédicteur-Correcteur** nommée aussi **Euler Modifiée**.

c. **Programmez** la méthode d'**Euler Modifiée** en modifiant "un peu" le fichier EulerExpl.sci. Nommez le nouveau fichier EulerMod.sci.

d. **Testez**-la sur les deux EDO précédentes (f_1 et f_2). Comparez-la graphiquement à Euler Explicite et edo de Scilab.

e. **Conclusions** ?

A rendre – 2 :

- Dans le corps du mail : **les fichiers** « def_fonc_ode.sce » avec f1 et f2, « EulerExpl.sci », « EulerMod.sci » et « Calcul_Y.sci ». Pas de fichier attaché !

- **Les graphiques** pour l'ODE n°2 avec **deux pas de temps différents**, pour les 3 méthodes (Euler explicite, Euler modifié et « edo » de scilab) et la solution analytique.

Séance 3 (2 heures) :

4- Etude de stabilité (1h30) :

On considère l'EDO suivante :

EDO n°3 : $\begin{cases} y'(t) = -20y(t) \\ y(0) = 1 \end{cases}$

a. Exprimer la **solution analytique** de cette EDO.

b. Exprimer les **conditions de stabilité** des méthodes d'**Euler Explicite** et **Euler Modifiée** en fonction du pas de temps h.

c. Dans le fichier « def_fonc_ode.sce », définir la fonction $f_3\big(t, y(t)\big) = -20y(t)$.

d. **Tracer** les différents graphiques de résolution de l'EDO n°3. Essayez différents intervalles de temps.

e. Testez la **stabilité** de la méthode d'**Euler explicite**, en faisant varier le pas de temps h de part et d'autre de la limite de stabilité.

f. Testez la **stabilité** de la méthode d'**Euler modifiée**, en faisant varier le pas de temps h de part et d'autre de la limite de stabilité.

g. Les zones de stabilité pour ces deux méthodes donnent-elles des résultats "satisfaisants" ?

5- Euler Implicite (30min) :

Pour l'EDO n°3, on peut programmer la méthode d'Euler Implicite.

a. Quelle est la **condition de stabilité** de la méthode d'**Euler implicite** ?

b. **Implémentez** la méthode d'Euler Implicite.

c. **Testez**-la dans les mêmes zones d'étude de stabilité des méthodes d'Euler Explicite et Euler Modifiée.

d. **Conclusions ?**

A rendre – 3 :

Dans le corps du mail :

- **Les fichiers** « def_fonc_ode.sce » avec f1, f2 et f3, « EulerMod.sce », « EulerImpl.sce » et « Calcul_Y.sce ».

- **Pour au moins trois pas de temps différents** (en-dessous, égal et au-dessus de la limite de stabilité), les **graphiques** pour l'ODE n°3 pour les quatre méthodes numériques (Euler Explicite, Implicite et Modifiée, et « ode » de Scilab) ainsi que la solution analytique.

Pas de fichier attaché.

Bon courage.