Energétique I **Méthodes Numériques II**
Sup'Galilée **Année 25-26**

TRAVAUX PRATIQUES - E.D.O.

Durée : 8h00

Travail individuel et personnel

Table des matières

1 Tests Algorithmique et Matlab

Une archive compressée au format **zip**, nommée CodesFournis_Mosaiques.zip,

`https://www.math.univ-paris13.fr/~cuvelier/docs/Enseignements/Energetique/MethNumII/25-26/CodesFournis_Mosaiques.zip`

ou au format **tar.gz**, nommée CodesFournis_Mosaiques.tar.gz,

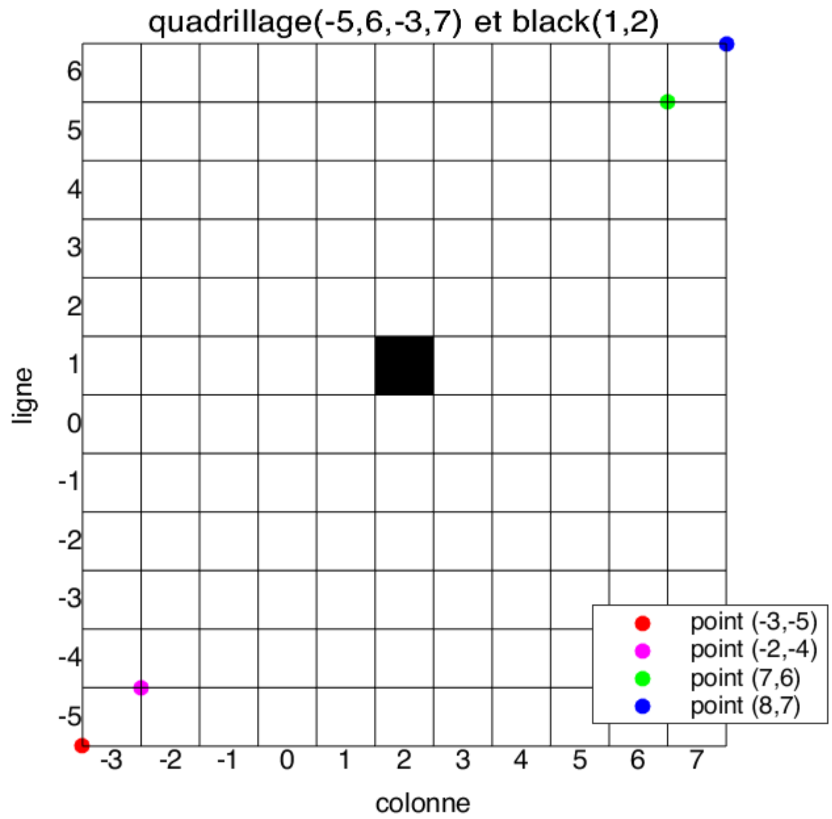
`https://www.math.univ-paris13.fr/~cuvelier/docs/Enseignements/Energetique/MethNumII/25-26/CodesFournis_Mosaiques.tar.gz`

est disponible directement avec un des liens précédents ou sur la page web du cours.

`https://www.math.univ-paris13.fr/~cuvelier/Enseign/ENER1-2526.html`

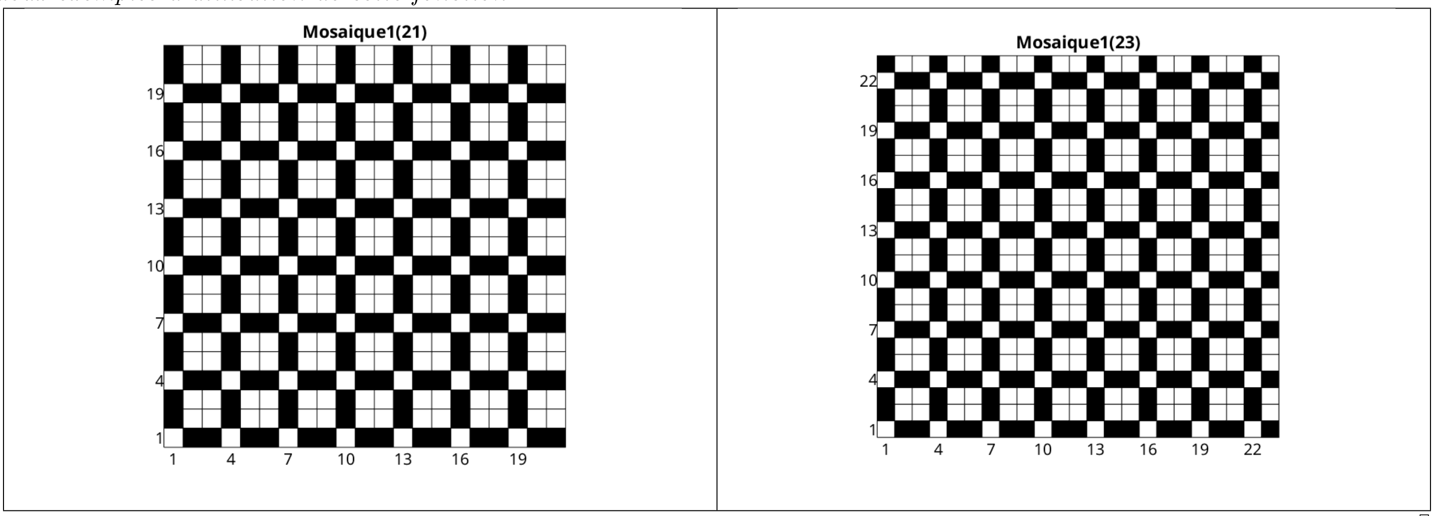
Il faut télécharger l'archive et la décompresser dans un répertoire.
Cette archive contient, entre autres, la fonction `black` et le programme `Quadrillagefigure`. Dans le programme `Quadrillagefigure` l'appel à la fonction `Quadrillage` manquante a été mis en commentaire. Ce programme servira uniquement à débuguer/tester/valider la fonction `Quadrillage` que vous allez écrire.

Q. 1 *Ecrire la fonction Matlab* `Quadrillage(imin,imax,jmin,jmax)` *permettant de générer un quadrillage pour les lignes* `imin` *à* `imax` *et les colonnes* `jmin` *à* `jmax`*. Voici un exemple avec la commande* `Quadrillage(−5,6,−3,7)` *représentant uniquement les traits noirs sur la figure :*



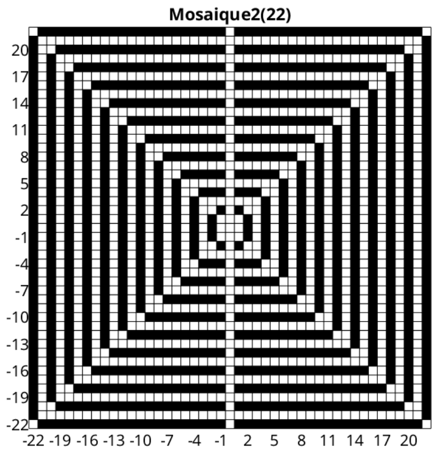
On peut noter que les coordonnées des points sont exprimées dans le plan classique xOy. On peut tester cette fonction avec le programme `Quadrillagefigure` *fourni pour obtenir la figure précédente.*
Le carré noir en ligne 1 et colonne 2 a été représenté à l'aide de la commande `black(1,2)`*, la fonction* `black` *étant fournie.*
On rappelle que pour représenter un segment entre les points $A_1 = (x1, y1)$ et $A_2 = (x2, y2)$, on peut utiliser sous Matlab, la commande **plot**`([x1 x2],[y1 y2])`. □

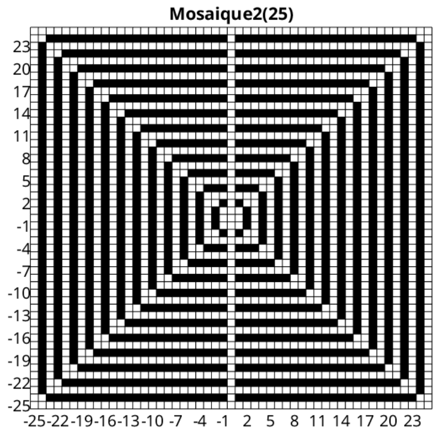
Q. 2 *Ecrire la fonction* `Mosaique1(n)` *permettant de créer une mosaïque sur le quadrillage* `Quadrillage(1,n,1,n)`*. Voici deux exemples d'utilisation de cette fonction :*



□

Q. 3 *Ecrire la fonction* `Mosaique2(n)` *permettant de créer une mosaïque sur le quadrillage* `Quadrillage(-n,n,-n,n)`*. Voici deux exemples d'utilisation de cette fonction :*





A faire en 2h30 (temps indicatif)

- ◇ Créer une archive compressée nommée `<NOM>-TP1-Q1a3` contenant les fichiers `Quadrillage.m`, `black.m`, `Mosaique1.m`, `Mosaique2.m` et tout autre fichier permettant l'éxecution des fonctions `Mosaique1.m` et `Mosaique2.m`. Ici `<NOM>` correspond évidemment à votre nom.
- ◇ Envoyer un mail à `cuvelier@math.univ-paris13.fr` ayant pour **sujet** `"<NOM> TP1 Q1a3"` et en fichier joint l'archive compressée créée précédement.

2 Résolution numérique d'équations différentielles ordinaires

Pour une explication détaillée voir le polycopié fourni `MethNumII_21janvier2025.pdf`

2.1 Schémas numériques pour la résolution d'un problème de Cauchy

Definition 1 (problème de Cauchy)**.** *Soit* $\boldsymbol{f}$ *l'application continue définie par*

$$\begin{array}{rcll} \boldsymbol{f} & : & [t^0, t^0+T] \times \mathbb{R}^d & \longrightarrow & \mathbb{R}^d \\ & & (t, \boldsymbol{y}) & \longmapsto & \boldsymbol{f}(t, \boldsymbol{y}) \end{array}$$

avec $t^0 \in \mathbb{R}$ *et* $T \in]0, +\infty]$. *Le* ***problème de Cauchy*** *revient à chercher une fonction* $\boldsymbol{y}$ *définie par*

$$\begin{array}{rcll} \boldsymbol{y} & : & [t^0, t^0+T] & \longrightarrow & \mathbb{R}^d \\ & & t & \longmapsto & \boldsymbol{y}(t) \end{array}$$

continue et dérivable, telle que

$$\boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \quad \forall t \in [t^0, t^0+T] \tag{2.1}$$

$$\boldsymbol{y}(t^0) = \boldsymbol{y}^{[0]} \in \mathbb{R}^d. \tag{2.2}$$

Dans tous les schémas qui suivent, on note t^n, $n \in [\![0, N]\!]$, une discrétisation régulière de $[t^0, t^0+T]$, $\boldsymbol{y}^{[n]} \approx \boldsymbol{y}(t^n)$ et $\boldsymbol{f}^{[n]} = \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]})$.

2.1.1 Schéma d'Euler progressif (ordre 1)

$$\left\{ \begin{array}{lcl} \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + h\boldsymbol{f}^{[n]}, \ \forall n \in [\![0, N-1]\!] \\ \boldsymbol{y}^{[0]} & = & \boldsymbol{y}(t^0) \end{array} \right. \tag{2.3}$$

2.1.2 Schéma de la tangente améliorée (ordre 2)

$$\left\{ \begin{array}{lcl} \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + h\boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{f}^{[n]}), \ \forall n \in [\![0, N-1]\!] \\ \boldsymbol{y}^{[0]} & = & \boldsymbol{y}(t^0) \end{array} \right. \tag{2.4}$$

2.1.3 Schéma de Heun (ordre 2)

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{2}\left[\boldsymbol{f}^{[n]} + \boldsymbol{f}\left(t^n + h, \boldsymbol{y}^{[n]} + h\boldsymbol{f}^{[n]}\right)\right]. \tag{2.5}$$

2.1.4 Schémas de Runge-Kutta

- ordre 2 (version 1)

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + h\boldsymbol{k}_2^{[n]}. \end{array} \tag{2.6}$$

- ordre 2 (version 2)

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + h, \boldsymbol{y}^{[n]} + h\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{2}(\boldsymbol{k}_1^{[n]} + \boldsymbol{k}_2^{[n]}). \end{array} \tag{2.7}$$

- ordre 2 (version 3)

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{2h}{3}, \boldsymbol{y}^{[n]} + \frac{2h}{3}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{4}(\boldsymbol{k}_1^{[n]} + 3\boldsymbol{k}_2^{[n]}). \end{array} \tag{2.8}$$

- ordre 3 (version 1)

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{3}, \boldsymbol{y}^{[n]} + \frac{h}{3}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{k}_3^{[n]} & = & \boldsymbol{f}(t^n + \frac{2h}{3}, \boldsymbol{y}^{[n]} + \frac{2h}{3}\boldsymbol{k}_2^{[n]}) \\ \hline \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{4}(\boldsymbol{k}_1^{[n]} + 3\boldsymbol{k}_3^{[n]})). \end{array} \tag{2.9}$$

- ordre 3 (version 2)

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{2h}{3}, \boldsymbol{y}^{[n]} + \frac{2h}{3}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{k}_3^{[n]} & = & \boldsymbol{f}(t^n + \frac{2h}{3}, \boldsymbol{y}^{[n]} + \frac{2h}{3}\boldsymbol{k}_2^{[n]}) \\ \hline \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{8}(2\boldsymbol{k}_1^{[n]} + 3\boldsymbol{k}_2^{[n]} + 3\boldsymbol{k}_3^{[n]})). \end{array} \tag{2.10}$$

- ordre 3 (version 3)

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{2h}{3}, \boldsymbol{y}^{[n]} + \frac{2h}{3}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{k}_3^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]} - h\boldsymbol{k}_1^{[n]} + h\boldsymbol{k}_2^{[n]}) \\ \hline \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{4}(3\boldsymbol{k}_2^{[n]} + \boldsymbol{k}_3^{[n]})). \end{array} \tag{2.11}$$

- ordre 4 (version 1)

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{k}_3^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{k}_2^{[n]}) \\ \boldsymbol{k}_4^{[n]} & = & \boldsymbol{f}(t^n + h, \boldsymbol{y}^{[n]} + h\boldsymbol{k}_3^{[n]}) \\ \hline \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{6}\big(\boldsymbol{k}_1^{[n]} + 2\boldsymbol{k}_2^{[n]} + 2\boldsymbol{k}_3^{[n]} + \boldsymbol{k}_4^{[n]}\big). \end{array} \tag{2.12}$$

- ordre 4 (version 2)

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{3}, \boldsymbol{y}^{[n]} + \frac{h}{3}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{k}_3^{[n]} & = & \boldsymbol{f}(t^n + \frac{2h}{3}, \boldsymbol{y}^{[n]} - \frac{h}{3}\boldsymbol{k}_1^{[n]} + h\boldsymbol{k}_2^{[n]}) \\ \boldsymbol{k}_4^{[n]} & = & \boldsymbol{f}(t^n + h, \boldsymbol{y}^{[n]} + h\boldsymbol{k}_1^{[n]} - h\boldsymbol{k}_2^{[n]} + h\boldsymbol{k}_3^{[n]}) \\ \hline \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{8}\big(\boldsymbol{k}_1^{[n]} + 3\boldsymbol{k}_2^{[n]} + 3\boldsymbol{k}_3^{[n]} + \boldsymbol{k}_4^{[n]}\big). \end{array} \tag{2.13}$$

- ordre 4 (version 3)

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{4}, \boldsymbol{y}^{[n]} + \frac{h}{4}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{k}_3^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{k}_2^{[n]}) \\ \boldsymbol{k}_4^{[n]} & = & \boldsymbol{f}(t^n + h, \boldsymbol{y}^{[n]} + h\boldsymbol{k}_1^{[n]} - 2h\boldsymbol{k}_2^{[n]} + 2h\boldsymbol{k}_3^{[n]}) \\ \hline \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{6}\big(\boldsymbol{k}_1^{[n]} + 4\boldsymbol{k}_3^{[n]} + \boldsymbol{k}_4^{[n]}\big). \end{array} \tag{2.14}$$

2.1.5 Méthodes d'Adams-Bashforth

On note $\boldsymbol{f}^{[n]} = \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]})$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{2}\left(3\boldsymbol{f}^{[n]} - \boldsymbol{f}^{[n-1]}\right). \tag{2.15}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{12}\left(23\boldsymbol{f}^{[n]} - 16\boldsymbol{f}^{[n-1]} + 5\boldsymbol{f}^{[n-2]}\right). \tag{2.16}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{24}\left(55\boldsymbol{f}^{[n]} - 59\boldsymbol{f}^{[n-1]} + 37\boldsymbol{f}^{[n-2]} - 9\boldsymbol{f}^{[n-3]}\right). \tag{2.17}$$

Ces 3 schémas sont **explicites** et leur ordre correspond au nombre de pas.

2.1.6 Méthodes d'Adams-Moulton

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{2}\left(\boldsymbol{f}^{[n+1]} + \boldsymbol{f}^{[n]}\right). \tag{2.18}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{12}\left(5\boldsymbol{f}^{[n+1]} + 8\boldsymbol{f}^{[n]} - \boldsymbol{f}^{[n-1]}\right). \tag{2.19}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{24}\left(9\boldsymbol{f}^{[n+1]} + 19\boldsymbol{f}^{[n]} - 5\boldsymbol{f}^{[n-1]} + \boldsymbol{f}^{[n-2]}\right). \tag{2.20}$$

Ces 3 schémas sont **implicites** et leur ordre correspond au nombre de pas plus un.

2.1.7 Schéma de Nyström (ordre 3)

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n-1]} + \frac{h}{3}\left(7\boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) - 2\boldsymbol{f}(t^{n-1}, \boldsymbol{y}^{[n-1]}) + \boldsymbol{f}(t^{n-2}, \boldsymbol{y}^{[n-2]})\right) \tag{2.21}$$

2.1.8 Schémas BDF (Backward-Difference Formulas)

- ordre 1

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + h\boldsymbol{f}^{[n+1]} \tag{2.22}$$

- ordre 2

$$\boldsymbol{y}^{[n+1]} = \frac{4}{3}\boldsymbol{y}^{[n]} - \frac{1}{3}\boldsymbol{y}^{[n-1]} + \frac{2h}{3}\boldsymbol{f}^{[n+1]} \tag{2.23}$$

- ordre 3

$$\boldsymbol{y}^{[n+1]} = \frac{1}{11}\left(18\boldsymbol{y}^{[n]} - 9\boldsymbol{y}^{[n-1]} + 2\boldsymbol{y}^{[n-2]} + 6h\boldsymbol{f}^{[n+1]}\right) \tag{2.24}$$

- ordre 4

$$\boldsymbol{y}^{[n+1]} = \frac{1}{25}\left(48\boldsymbol{y}^{[n]} - 36\boldsymbol{y}^{[n-1]} + 16\boldsymbol{y}^{[n-2]} - 3\boldsymbol{y}^{[n-3]} + 12h\boldsymbol{f}^{[n+1]}\right) \tag{2.25}$$

2.1.9 Schéma de Milne (ordre 4)

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n-3]} + \frac{4h}{3}\left(2\boldsymbol{f}^{[n]} - \boldsymbol{f}^{[n-1]} + 2\boldsymbol{f}^{[n-2]}\right) \tag{2.26}$$

2.1.10 Schémas de Milne-Simpson

- ordre 1

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n-1]} + 2h\boldsymbol{f}^{[n+1]} \tag{2.27}$$

- ordre 2

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n-1]} + 2h\boldsymbol{f}^{[n]} \tag{2.28}$$

- ordre 4

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n-1]} + \frac{h}{3}\left(\boldsymbol{f}^{[n+1]} + 4\boldsymbol{f}^{[n]} + \boldsymbol{f}^{[n-1]}\right) \tag{2.29}$$

2.2 Schéma prédicteur-correcteur

Il s'agit là d'une des méthodes les plus employées. Une méthode de prédiction-correction procède en deux temps : à l'aide du schéma explicite, on calcule une valeur approchée de la solution au $n^{ième}$ pas (notée $\overline{\boldsymbol{y}}^{(n+1)}$), puis on calcule $\boldsymbol{y}^{[n+1]}$ à l'aide du schéma implicite en substituant, dans l'expression de droite du schéma implicite, $\boldsymbol{y}^{[n+1]}$ par $\overline{\boldsymbol{y}}^{(n+1)}$. On obtient alors une valeur dite *corrigée*.

2.3 Travail à effectuer

2.3.1 Problème de Cauchy scalaire

Le but est de représenter graphiquement les erreurs données par plusieurs schémas et de retrouver numériquement leur ordre. Pour celà, on utilisera un problème de Cauchy **scalaire** dont on connait la solution exacte. Voir l'annexe 4.1 pour plusieurs exemples d'E.D.O. avec solution exacte.

Q. 4 ***a.*** *Ecrire les cinq fonctions Matlab suivantes correspondant à la résolution d'un problème de Cauchy scalaire :*

- `[t,Y]=redEUP(f,to,T,yo,N)` *: schéma d'Euler progressif (fichier* `redEUP.m`*).*
- `[t,Y]=redHEU(f,to,T,yo,N)` *: schéma de Heun (fichier* `redHEU.m`*).*
- `[t,Y]=redRK4(f,to,T,yo,N)` *: schéma de Runge et Kutta d'ordre 4 (version 2)* (2.13) *(fichier* `redRK4.m`*).*
- `[t,Y]=redPM3(f,to,T,yo,N)` *: schéma d'Adams-Bashforth d'ordre* 3 *(fichier* `redPM3.m`*).*
- `[t,Y]=redPC4(f,to,T,yo,N)` *: schéma de type prédiction-correction utilisant les schémas de Milne* (2.26) *d'ordre* 4 *et de Milne-Simpson* (2.29) *d'ordre* 4 *(fichier* `redPC4.m`*).*

Ici les paramètres `f`, `to`, `T`, `yo` *correspondent respectivement aux données* $\boldsymbol{f}$, t^0, T, $\boldsymbol{y}^{[0]}$ *du problème de Cauchy (2.1-2.2). Enfin,* `Y` *est le tableau contenant les* $\boldsymbol{y}^{[n]}$, $n \in \{0, \cdots, \texttt{N}\}$ *et* `t` *est le tableau contenant les* `(N+1)` *réels* t^n, $n \in \{0, \cdots, \texttt{N}\}$

b. *Ecrire le programme principal (fichier* `erreur.m`*) permettant le calcul et le tracé des erreurs. Pour une méthode donnée le tracé de l'erreur correspond au tracé de l'ensemble des points* $(t^n, abs(\boldsymbol{y}^{[n]} - \boldsymbol{y}(t^n)))$, $n \in \{0, \cdots, N\}$.
Voir la figure 1 pour un exemple de tracé. Pour cette figure, la commande Matlab **subplot** *a été utilisée.*

c. *Ecrire le programme principal (fichier* `ordre.m`*) permettant de calculer numériquement l'ordre des 5 schémas et de les représenter en s'inspirant de la figure 2 que l'on essayera de reproduire.*

□

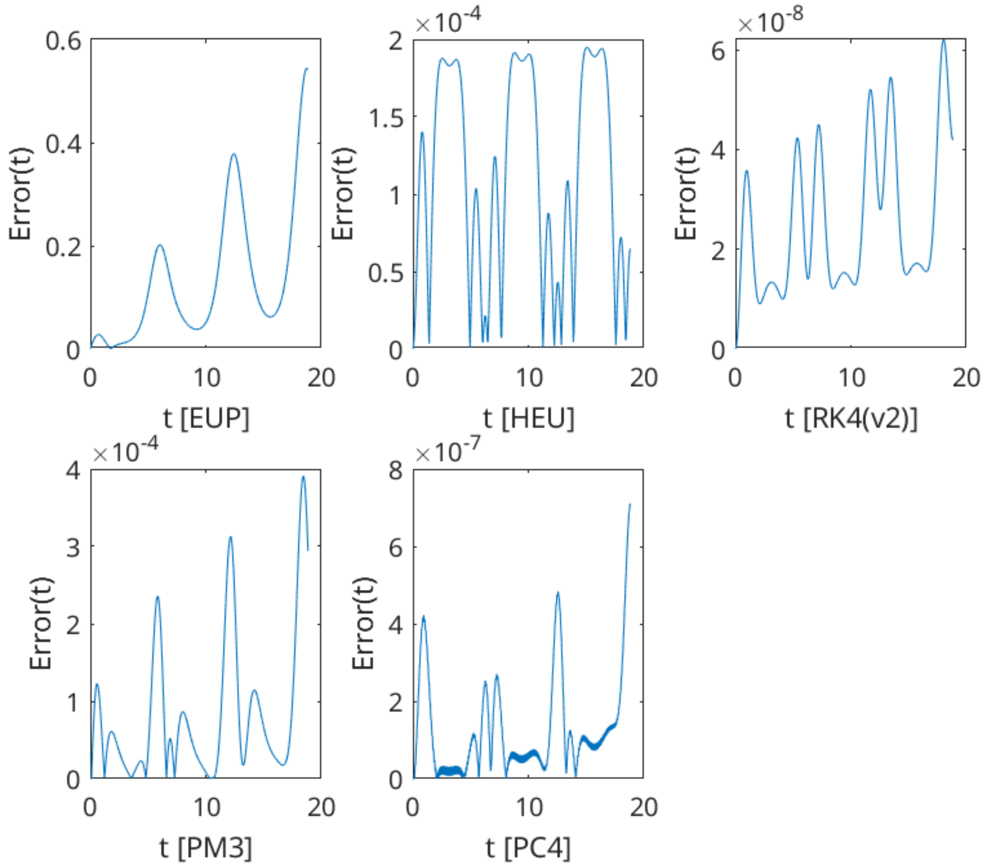


FIGURE 1 – Valeurs absolues des erreurs des 5 schémas représentées dans une unique figure

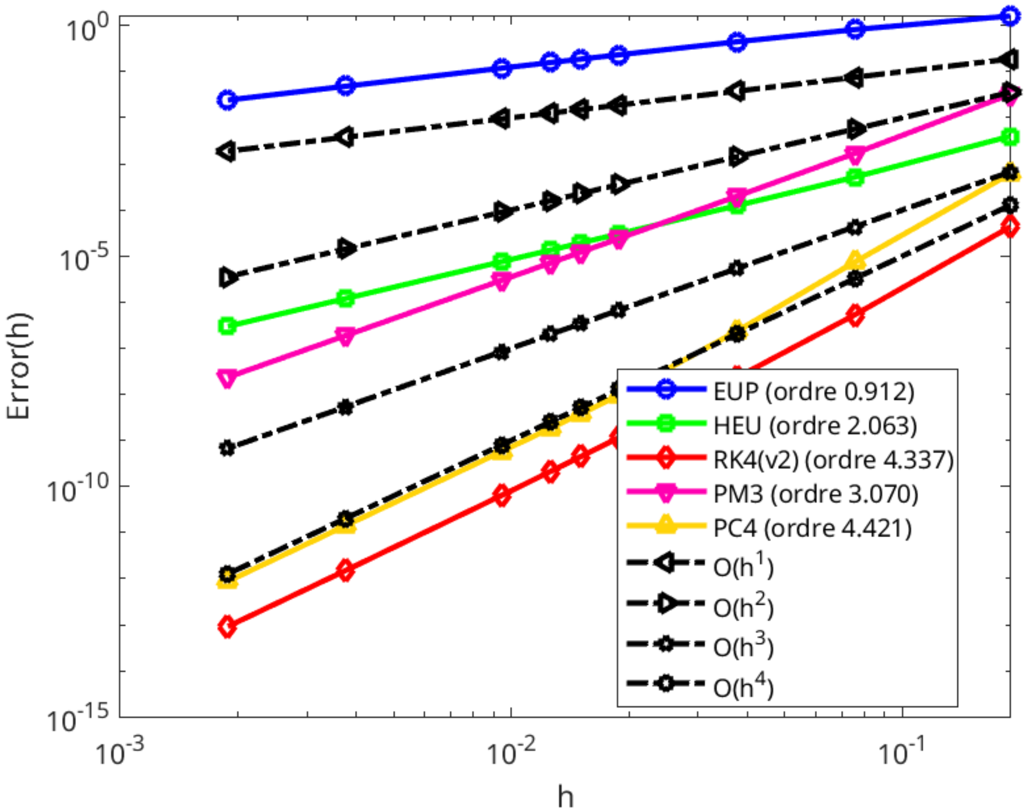


FIGURE 2 – Ordre des 5 schémas

A faire en 3h00 (temps indicatif)

- ◇ Créer une archive compressée nommée `<NOM>-TP1-Q4` contenant les fichiers `redEUP.m`, `redHEU.m`, `redPM3.m`, `redRK4.m`, `redPC4.m`, `erreur.m`, `ordre.m` et toutes autres fonctions nécessaires à l'execution des programmes `erreur.m` et `ordre.m`. Ici `<NOM>` correspond évidemment à votre nom.
- ◇ Envoyer un mail à `cuvelier@math.univ-paris13.fr` ayant pour **sujet** `"<NOM> TP1 Q4"` et en fichier joint l'archive compressée créée précédement.

2.3.2 Problème de Cauchy vectoriel

On souhaite maintenant résoudre numériquement un problème de Cauchy **vectoriel**. Voir l'annexe 4.1 pour plusieurs exemples d'E.D.O. avec solution exacte.

Q. 5 ***a.*** *Ecrire la fonction* `[t,Y]=redRK4Vec(f,to,T,yo,N)` *permettant de résoudre un problème de Cauchy* ***vectoriel*** *par le schéma de Runge et Kutta d'ordre 4 (version 2)* (2.13)*.*

b. *Ecrire un programme (fichier* `erreurVec.m`*) permettant le calcul et le tracé de l'erreur commise entre la solution exacte d'un problème de Cauchy* ***vectoriel*** *et la solution numérique approchée donnée par le schéma de Runge et Kutta d'ordre 4 (version 2).*

c. *Ecrire un programme (fichier* `ordreVec.m`*) permettant de calculer numériquement l'ordre du schéma et de le représenter. On utilisera un problème de Cauchy* ***vectoriel*** *dont on connait la solution exacte.*

□

A faire en 0h30 (temps indicatif)

- ◇ Créer une archive compressée nommée `<NOM>-TP1-Q5` contenant les fichiers `redRK4vec.m`, `erreurVec.m`, `ordreVec.m` et toutes autres fonctions nécessaires à l'execution des programmes `erreurVec.m` et `ordreVec.m`. Ici `<NOM>` correspond évidemment à votre nom.
- ◇ Envoyer un mail à `cuvelier@math.univ-paris13.fr` ayant pour **sujet** `"<NOM> TP1 Q5"` et en fichier joint l'archive compressée créée précédement.

3 Le pendule pesant

3.1 Position du problème et équations différentielles

On considère un pendule de masse M, fixé à une tige rigide de longueur L et de masse négligeable, dans un milieu visqueux dont le coefficient de viscosité vaut k. On note θ l'angle formé par le pendule et l'axe verticale : il vérifie l'équation différentielle suivante (principe fondamental de la dynamique) :

$$\theta''(t) = -\frac{g}{L}\sin(\theta(t)) - \frac{k}{ML^2}\theta'(t), \ \forall t \geqslant 0 \tag{3.1}$$

avec les conditions initiales

$$\theta(0) = \theta_0 \text{ et } \theta'(0) = \theta'_0. \tag{3.2}$$

θ_0 est l'angle initial en radian et θ'_0 la vitesse angulaire initiale en radian/seconde.

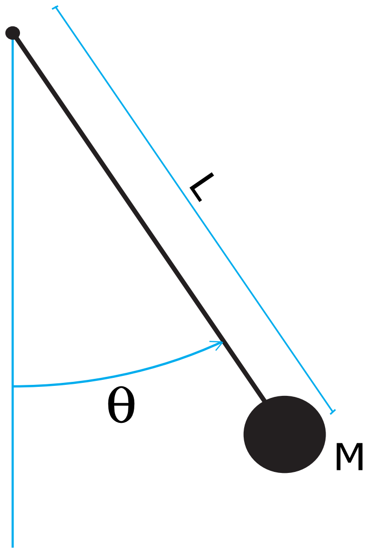


FIGURE 3 – Pendule pesant

On peut prendre, par exemple, $M = 1kg$, $L = 1m$, $g = 9.8m.s^{-2}$ et $k = 0.5USI$.
L'E.D.O. (3.1-3.2) ne peut être résolue de manière exacte. On se propose d'utiliser la méthode de Runge-Kutta d'ordre 4 pour l'étude des différents types de mouvements possibles, suivant les conditions de l'expérience (dans le vide $k = 0$, dans l'air $k = 0.1$, dans l'eau $k = 0.5$,...) et les conditions initiales imposées (pendule lancé, laché, ...). On souhaite ensuite tracer les deux courbes discrètes $\theta(t)$ et $\theta\prime(t)$.

3.2 Résolution numérique

Q. 6 ***a.*** *Ecrire la fonction Matlab `fCauchy` (fichier `fCauchy.m`) correspondant à la fonction f du problème de Cauchy associé à (3.1-3.2). (On pourra utiliser des variables globales pour les paramètres physiques M, k, L et g. Voir l'aide sur `global` de Matlab. Toutefois il y a mieux en utilisant les fonctions anonymes générées avec `@`)*

b. *Ecrire le programme `prg4` (fichier `prg4.m`) permettant de représenter la position et la vitesse du pendule au cours du temps.*

c. *En s'aidant du programme `PenduleMovie` (non réaliste) et de la fonction `PlotPendule` fournis, écrire le programme `PenduleVideo` permettant de réaliser une vidéo représentant le pendule en mouvement au cours du temps.*

□

A faire en 2h00 (temps indicatif)

- ◇ Créer une archive compressée nommée `<NOM>-TP1-Q6` contenant les fichiers `fCauchy.m`, `prg4.m` et `PenduleVideo.m`, ainsi que tout autre code nécessaire à l'exécution de `prg4` et `PenduleVideo`. Ici `<NOM>` correspond évidemment à votre nom.
- ◇ Envoyer un mail à `cuvelier@math.univ-paris13.fr` ayant pour **sujet** `"<NOM> TP1 Q6"` et en fichier joint l'archive compressée créée précédement.

4 Annexes

4.1 Quelques E.D.O. avec solution exacte

4.1.1 Exemple 1

Soit $\alpha \in \mathbb{R}$. L'E.D.O.

$$\left\{\begin{array}{rcl} y'(t) & = & \cos(t),\ \forall t \geqslant 0, \\ y(0) & = & \alpha, \end{array}\right.$$

a pour solution $y(t) = \sin(t) + \alpha$.

4.1.2 Exemple 2

Soit $\beta \in \mathbb{R}$. L'E.D.O.

$$\left\{\begin{array}{rcl} y'(t) & = & \sin(t),\ \forall t \geqslant 0, \\ y(0) & = & \beta, \end{array}\right.$$

a pour solution $y(t) = -\cos(t) + 1 + \beta$.

4.1.3 Exemple 3

L'E.D.O.

$$\left\{\begin{array}{rcl} y'(t) & = & -y(t)\sin(t),\ \forall t \geqslant 0, \\ y(0) & = & e = \exp(1), \end{array}\right.$$

a pour solution $y(t) = \exp(\cos(t))$.

4.1.4 Exemple 4

Soit le système de 2 équations différentielles ordinaires à coefficients constants

$$\boldsymbol{y}'(t) = \begin{pmatrix} \frac{2}{3} & -\frac{2}{3} \\ \frac{2}{3} & \frac{7}{3} \end{pmatrix} \boldsymbol{y}(t),\ t \in [0, T]$$

où $\boldsymbol{y} : [0, T] \longrightarrow \mathbb{R}^2$ et $\boldsymbol{y}(0) = (1,\ 2)^t$.
Cette EDO admet comme solution exacte

$$\boldsymbol{y}(t) = \begin{pmatrix} 3\,Ce^{(2\,t)} - 2\,Be^t \\ -6\,Ce^{(2\,t)} + Be^t \end{pmatrix}$$

avec $B = -\frac{4}{3}$ et $C = -\frac{5}{9}$.

4.1.5 Exemple 5

Soit le système de 2 équations différentielles ordinaires à coefficients constants

$$\boldsymbol{y}'(t) = \begin{pmatrix} \frac{977}{802} & -\frac{600}{401} \\ \frac{42}{401} & \frac{113}{401} \end{pmatrix} \boldsymbol{y}(t),\ t \in [0, T]$$

où $\boldsymbol{y} : [0, T] \longrightarrow \mathbb{R}^2$ et $\boldsymbol{y}(0) = (1,\ -1)^t$.
Cette EDO admet comme solution exacte

$$\boldsymbol{y}(t) = \begin{pmatrix} -\frac{5}{4}\,Ce^{\left(\frac{1}{2}\,t\right)} + 8\,Be^t \\ -\frac{3}{5}\,Ce^{\left(\frac{1}{2}\,t\right)} + \frac{7}{6}\,Be^t \end{pmatrix}$$

avec $B = \frac{222}{401}$ et $C = \frac{1100}{401}$.

4.1.6 Exemple 6

Soit le système de 3 équations différentielles ordinaires à coefficients constants

$$\boldsymbol{y}'(t) = \begin{pmatrix} \frac{30788}{31803} & \frac{1264}{10601} & \frac{4355}{31803} \\ -\frac{10865}{63606} & \frac{4833}{10601} & -\frac{3515}{63606} \\ \frac{7495}{21202} & \frac{726}{10601} & \frac{8679}{21202} \end{pmatrix} \boldsymbol{y}(t),\ t \in [0, T]$$

où $\boldsymbol{y} : [0, T] \longrightarrow \mathbb{R}^3$ et $\boldsymbol{y}(0) = (1,\ -1,\ 2)^t$.
Cette EDO admet comme solution exacte

$$\boldsymbol{y}(t) = \begin{pmatrix} \frac{2}{3}\,Ce^{\left(\frac{1}{2}\,t\right)} + \frac{5}{3}\,De^{\left(\frac{1}{3}\,t\right)} + \frac{9}{2}\,Be^t \\ -3\,Ce^{\left(\frac{1}{2}\,t\right)} - \frac{5}{6}\,De^{\left(\frac{1}{3}\,t\right)} - \frac{5}{3}\,Be^t \\ \frac{1}{3}\,Ce^{\left(\frac{1}{2}\,t\right)} - 7\,De^{\left(\frac{1}{3}\,t\right)} + \frac{5}{2}\,Be^t \end{pmatrix}$$

avec $B = \frac{2694}{10601}$, $C = \frac{2577}{10601}$. et $D = -\frac{1944}{10601}$.