TPs EDP [a]

Travaux Pratiques N^o 3

•••••••

Algorithmique numérique : E.D.O.

a. Version du 17 septembre 2018

1 Résolution numérique d'équations différentielles ordinaires

1.1 Rappels de quelques schémas numériques usuels

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

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

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

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

continue et dérivable, telle que

$$\boldsymbol{y}'(t) = \boldsymbol{f}(t,\boldsymbol{y}(t)), \quad \forall t \in [t^0, t^0+T] \tag{1}$$

$$\boldsymbol{y}(t^0) = \boldsymbol{y}^{[0]} \in \mathbb{R}^d. \tag{2}$$

■

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

1.1.1 Schéma d'Euler progressif

$$\left\{ \begin{array}{lcl} \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + h\boldsymbol{f}^{[n]},\ \forall n \in [\![0, N-1]\!] \\ \boldsymbol{y}^{[0]} & = & \boldsymbol{y}(t^0) \end{array} \right. \tag{3}$$

Ce schéma est d'ordre 1.

1.1.2 Schéma de la tangente améliorée

$$\left\{ \begin{array}{lcl} \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + h\boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{f}^{[n]}),\ \forall n \in [\![0, N-1]\!] \\ \boldsymbol{y}^{[0]} & = & \boldsymbol{y}(t^0) \end{array} \right. \tag{4}$$

Ce schéma est d'ordre 2.

1.1.3 Schéma de Runge-Kutta d'ordre 4

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{k}_3^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{k}_2^{[n]}) \\ \boldsymbol{k}_4^{[n]} & = & \boldsymbol{f}(t^n + h, \boldsymbol{y}^{[n]} + h\boldsymbol{k}_3^{[n]}) \\ \hline \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{6}(\boldsymbol{k}_1^{[n]} + 2\boldsymbol{k}_2^{[n]} + 2\boldsymbol{k}_3^{[n]} + \boldsymbol{k}_4^{[n]}). \end{array} \tag{5}$$

1.1.4 Méthodes d'Adams-Bashforth

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

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{2}\left(3\boldsymbol{f}^{[n]} - \boldsymbol{f}^{[n-1]}\right). \tag{6}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{12}\left(23\boldsymbol{f}^{[n]} - 16\boldsymbol{f}^{[n-1]} + 5\boldsymbol{f}^{[n-2]}\right). \tag{7}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{24}\left(55\boldsymbol{f}^{[n]} - 59\boldsymbol{f}^{[n-1]} + 37\boldsymbol{f}^{[n-2]} - 9\boldsymbol{f}^{[n-3]}\right). \tag{8}$$

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

1.1.5 Méthodes d'Adams-Moulton

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{2}\left(\boldsymbol{f}^{[n+1]} + \boldsymbol{f}^{[n]}\right). \quad (9)$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{12}\left(5\boldsymbol{f}^{[n+1]} + 8\boldsymbol{f}^{[n]} - \boldsymbol{f}^{[n-1]}\right). \quad (10)$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{24}\left(9\boldsymbol{f}^{[n+1]} + 19\boldsymbol{f}^{[n]} - 5\boldsymbol{f}^{[n-1]} + \boldsymbol{f}^{[n-2]}\right). \quad (11)$$

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

1.2 Schéma prédicteur-correcteur

Une méthode de prédiction-correction procède en deux temps : on fournit explicitement une valeur approchée de la solution au $n^{ième}$ pas (soit $\overline{\boldsymbol{y}}^{[n+1]}$), puis on calcule la valeur correspondante de $\boldsymbol{f}$ (soit $\overline{\boldsymbol{f}}^{[n+1]}$) et enfin, on subsitue cette valeur dans un schéma implicite (on obtient alors une valeur *corrigée*).

pour n variant de 0 à $N-1$ faire
Calculer une valeur approchée $\overline{\boldsymbol{y}}^{[n+1]}$ par un schéma explicite ;
Evaluer $\overline{\boldsymbol{f}}^{[n+1]} = \boldsymbol{f}(t^{n+1}, \overline{\boldsymbol{y}}^{[n+1]})$;
$\boldsymbol{y}^{[n+1]}$) à l'aide d'un schéma implicite en remplaçant l'inconnue par $\overline{\boldsymbol{y}}^{[n+1]}$);
finpour

1.3 Travail à effectuer

Le but est de représenter graphiquement les erreurs données par plusieurs schémas et de retouver numériquement leur ordre. Pour celà il faudra pouvoir connaitre explicitement la solution du problème de Cauchy étudié. Voir l'annexe 3.1 pour plusieurs exemples de problèmes de Cauchy avec solutions.

Q. 1 *1. Ecrire les cinq fonctions Matlab suivantes correspondant à la résolution d'un problème de Cauchy :*

- *[t,Y]=redEUP(f,a,b,yo,N) : schéma d'Euler progressif.*
- *[t,Y]=redTGA(f,a,b,yo,N) : schéma de la tangente améliorée.*
- *[t,Y]=redRK4(f,a,b,yo,N) : schéma de Runge et Kutta d'ordre 4.*
- *[t,Y]=redAB3(f,a,b,yo,N) : schéma d'Adams-Bashforth d'ordre 3.*
- *[t,Y]=redPC4(f,a,b,yo,N) : schéma de type prédiction-correction utilisant les schémas d'Adams-Bashforth et d"Adams-Moulton d'ordre 4.*

Ici les paramètres f, a, b, yo correspondent respectivement aux $\boldsymbol{f}, t^0, t^0 + T, \boldsymbol{y}^{[0]}$ *du problème de Cauchy (1-2). Enfin, Y est le tableau contenant les* $\boldsymbol{y}^{[n]}$, $n \in \{0, \cdots, N\}$ *et t est le tableau contenant les N+1 nombres* t^n, $n \in \{0, \cdots, N\}$

2. *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.

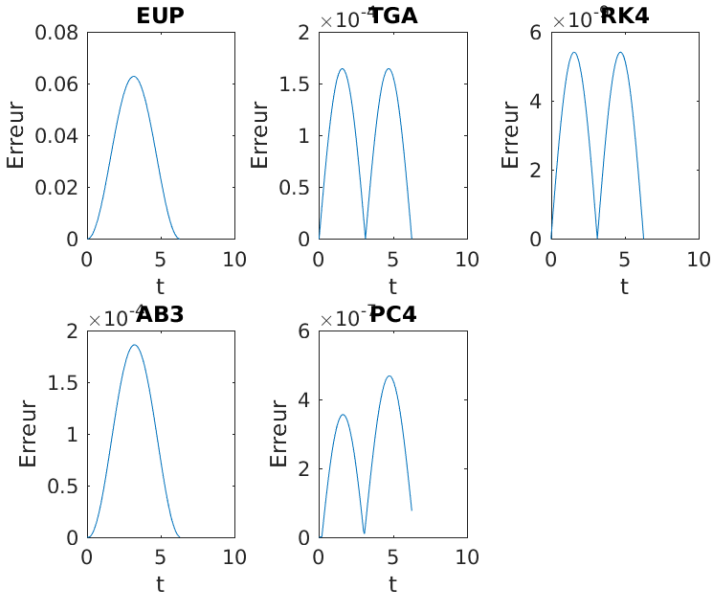


FIGURE 1 – Valeurs absolues des erreurs des 5 schémas

3. *Ecrire le programme principal (fichier* `ordre.m`*) permettant de calculer numériquement l'ordre des 5 schémas et de le représenter.*
Voir la figure 2 pour un exemple de tracé.

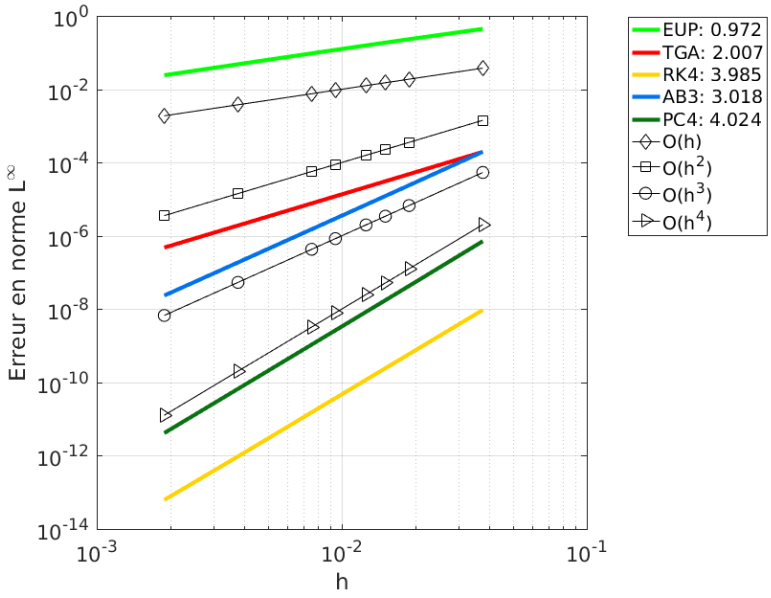


FIGURE 2 – Ordre des 5 schémas

2 Le système solaire

Une archive compressée au format **zip**

`www.math.univ-paris13.fr/~cuvelier/docs/Enseignements/macs2/TPs-EDP/18-19/TP3_CodesFournis_SystemeSolaire.zip`

ou au format **tar.gz**

`www.math.univ-paris13.fr/~cuvelier/docs/Enseignements/macs2/TPs-EDP/18-19/TP3_CodesFournis_SystemeSolaire.tar.gz`

est disponible en ligne. Il faut télécharger l'archive et la décompresser dans un répertoire.

2.1 Position du problème et équations différentielles

Le but de cette partie est de prédire la position de planètes dans le système solaire à partir de positions et de vitesses initiales des planètes. Les unités choisies sont : les masses sont relatives au soleil, les distances sont en unités astronomiques (1 U.A. = 149 597 870 km) et le temps en jour terrestre.

En astrophysique, la masse solaire est l'unité de masse conventionnellement utilisée pour exprimer les masses des corps célestes massifs et des structures formées d'étoiles (amas, superamas, galaxies, etc.). Son symbole et sa valeur sont :

$$M_\odot = 1,9891 \times 10^{30} \text{ kg}.$$

La masse solaire peut aussi être déterminé par l'Unité astronomique (U.A.), l'année et le Constante gravitationnelle (G) :

$$M_\odot = \frac{4\pi^2 \times (1\text{UA})^3}{G \times (1\text{ans})^2}.$$

La constante gravitationnelle est alors

$$G = \frac{4\pi^2}{(365,25636567)^2} \approx 2.959130713485796 \times 10^{-4},$$

une année sidérale correspondant à $365,25636567$ jours.

On note $\boldsymbol{q}_i(t)$ la position du ième corps au temps $t.$, $\boldsymbol{v}_i(t)$ sa vitesse au temps t et m_i sa masse. Les données initiales pour le soleil (corps $i = 1$) sont $\boldsymbol{q}_1 = (0,0,0)^t$ et $\boldsymbol{v}_1 = (0,0,0)^t$. Toutes les autres données sont fournies dans [1] page 14, regroupées dans le fichier `SysSolDataHairer.m` et correspondent aux positions de différentes planètes le 5 septembre 1994 à 0h00m00s. D'autres données, récupérées sur JPL Solar System Dynamics sont disponibles dans le fichier `SysSolData.m` et correspondent aux positions de différentes planètes le 13 mai 2011 à 0h00m00s.

De manière classique, un problème à N corps se modélise par

$$\frac{d^2\boldsymbol{q}_i}{dt^2}(t) = -G \sum_{j=1,\ j\neq i}^{N} m_j \frac{\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)}{\|\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)\|^3},\ \forall i \in [\![1, N]\!].$$

et l'énergie du système ,correspondant à la somme des énergies cinétiques et des énergies potentielles gravitationnelle est donnée par

$$\mathcal{E}(t) = \frac{1}{2}\sum_{i=1}^{N} m_i \|\boldsymbol{v}_i(t)\|^2 - G\sum_{i=1}^{N}\sum_{j=1}^{i-1} \frac{m_i m_j}{\|\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)\|}.$$

2.2 Résolution numérique classique

Q. 2 (Sur feuille) *1. Ecrire de manière détaillée le problème de Cauchy associé à un problème à N corps sachant que la fonction vectorielle inconnues $\boldsymbol{y}$ est définie par*

$$\boldsymbol{y}(t) = \begin{pmatrix} \boldsymbol{q}_1(t) \\ \vdots \\ \boldsymbol{q}_N(t) \\ \hline \boldsymbol{q}'_1(t) \\ \vdots \\ \boldsymbol{q}'_N(t) \end{pmatrix} \in \mathbb{R}^{6N}$$

2. Quelles sont les données du problème de Cauchy obtenu ? (avec leur type détaillé : entier, réel, complexe, vecteur, matrice, fonction, ...)

3. Quelles sont les inconnues du problème de Cauchy obtenu ? (avec leur type détaillé : entier, réel, complexe, vecteur, matrice, fonction, ...) ▪

2.3 Système à 3 corps : Soleil, Saturne et Jupiter

Pour récupérer les données initiales de ce système, on peut utiliser les commandes Matlab :

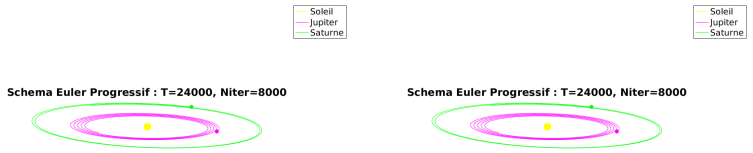
```
SysSol3=SysSolData();
SysSol3.planetes=SysSol3.planetes(1:3);
```

Les deux fonctions fournies plotSysSol et PlotPlanets (necessite PlotPlanet) permettent de représenter, après calculs, respectivement l'orbite des planetes de la structure SysSol (voir fonctions SysSolData ou SysSolDataHairer) et les planètes à un instant donné. Leur syntaxe est la suivante plotSysSol(T,Y,SysSol) et PlotPlanets(SysSol,q) avec

- (T,Y) tableaux retounés lors de la résolution du problème de Cauchy associé,
- SysSol structure contenant divers renseignements sur le système solaire,
- q positions des planetes.

Q. 3 (Matlab) *1. Ecrire la fonction Matlab fSysSol3 (fichier `fSysSol3.m`) correspondant à la fonction $\boldsymbol{f}$ du problème de Cauchy associé au problème à 3 corps (Soleil, Saturne et Jupiter). On pourra utiliser une variable globale pour les paramètres physiques (voir fichiers `SysSolDataHairer.m` ou `SysSolData.m`).*

2. Ecrire le programme prgSysSol3 (fichier `prgSysSol3.m`) qui résoud le problème de Cauchy par resEUP et resRK4 puis représente, pour chaque méthode, les orbites des 3 planètes ainsi que leurs positions à l'instant final avec comme données $T = 24000$ et $N = 8000$ (voir figure 3) On représentera aussi l'énergie du système pour chaque méthode.



■

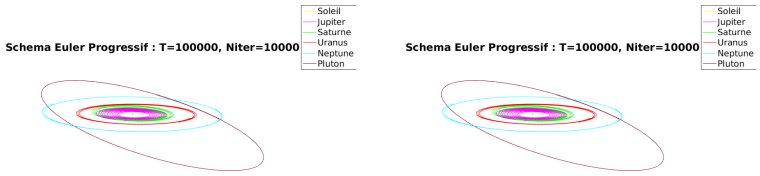
FIGURE 3 – Orbites Soleil-Saturne-Jupiter

2.4 Système à 6 corps

On étudie ici le système composé des 6 planetes Soleil, Saturne, Jupiter, Uranus, Neptune et Pluton.

Q. 4 (Matlab) *1. Ecrire la fonction Matlab fSysSol6 (fichier `fSysSol6.m`) correspondant à la fonction $\boldsymbol{f}$ du problème de Cauchy associé au problème à 6 corps (Soleil, Saturne, Jupiter, Uranus, Neptune et Pluton). On pourra utiliser une variable globale pour les paramètres physiques (voir fichiers `SysSolDataHairer.m` ou `SysSolData.m`).*

2. Ecrire le programme prgSysSol6 (fichier `prgSysSol6.m`) qui résoud le problème de Cauchy par resEUP et resRK4 puis représente, pour chaque méthode, les orbites des 6 planètes ainsi que leurs positions à l'instant final avec comme données $T = 100000$ et $N = 10000$ (voir figure 4) On représentera aussi l'energie du système pour chaque méthode.



■

FIGURE 4 – Orbites Soleil, Saturne, Jupiter, Uranus, Neptune et Pluton

2.5 Système à 7 corps

On étudie ici le système composé des 7 planetes Soleil, Saturne, Jupiter, Uranus, Neptune, Pluton et Terre.

Q. 5 (Matlab) *1. Ecrire la fonction Matlab fSysSol7 (fichier `fSysSol7.m`) correspondant à la fonction $\boldsymbol{f}$ du problème de Cauchy associé au problème à 7 corps (Soleil, Saturne, Jupiter, Uranus, Neptune, Pluton et Terre). On pourra utiliser une variable globale pour les paramètres physiques (voir fichiers `SysSolDataHairer.m` ou `SysSolData.m`).*

2. Ecrire le programme prgSysSol7 (fichier `prgSysSol7.m`) qui résoud le problème de Cauchy par resEUP et resRK4 puis représente, pour chaque méthode, les orbites des 7 planètes ainsi que leurs positions à l'instant final avec comme données $T = 100000$ et $N = 10000$. On représentera aussi l'energie du système pour chaque méthode. ▪

2.6 Résolution numérique par intégrateur symplectique

Les méthodes usuelles de résolution d'un problème à N corps peuvent poser des problèmes lors de la résolution en temps long : Celles-ci ne respectent pas forcément la physique du problème considéré : il n'y a pas conservation de l'énergie. On propose ici de regarder les méthodes numériques de résolution d'E.D.O. par intégrateur symplectique qui, *par construction*, conserve l'énergie et permettent de résoudre sur des temps beaucoup plus long et avec un pas de temps plus grand (pour plus de détails voir [1] et [2]). Voici en résumé, le mode opératoire.

Pour un problème à N corps, on pose $\boldsymbol{p}_i(t) = m_i \boldsymbol{v}_i(t) = m_i \frac{d\boldsymbol{q}_i}{dt}(t), \forall i \in [\![1, N]\!]$, et

$$\boldsymbol{q}(t) = \begin{pmatrix} \boldsymbol{q}_1(t) \\ \vdots \\ \boldsymbol{q}_N(t) \end{pmatrix} \text{ et } \boldsymbol{p}(t) = \begin{pmatrix} \boldsymbol{p}_1(t) \\ \vdots \\ \boldsymbol{p}_N(t) \end{pmatrix}$$

On note $\mathcal{H}$ l'hamiltonien du système, correspondant à la somme des energies cinétiques et des energies potentielles gravitationnelle, défini par

$$\mathcal{H}(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{2} \sum_{i=1}^{N} \frac{1}{m_i} \|\boldsymbol{p}_i\|^2 - G \sum_{i=1}^{N} \sum_{j=1}^{i-1} \frac{m_i m_j}{\|\boldsymbol{q}_i - \boldsymbol{q}_j\|}.$$

et vérifiant

$$\left\{ \begin{array}{rcl} \frac{d\boldsymbol{p}}{dt}(t) & = & -\mathcal{H}_q(\boldsymbol{p}(t), \boldsymbol{q}(t)) \\ \frac{d\boldsymbol{q}}{dt}(t) & = & \mathcal{H}_p(\boldsymbol{p}(t), \boldsymbol{q}(t)) \end{array} \right.$$

avec

$$\mathcal{H}_q(\boldsymbol{p}, \boldsymbol{q}) = G \begin{pmatrix} h_1(\boldsymbol{q}) \\ \vdots \\ h_N(\boldsymbol{q}) \end{pmatrix}, \; h_i(\boldsymbol{q}) = m_i \sum_{j=1,\; j \neq 1}^{N} m_j \frac{\boldsymbol{q}_i - \boldsymbol{q}_j}{\|\boldsymbol{q}_i - \boldsymbol{q}_j\|^3}$$

et

$$\mathcal{H}_p(\boldsymbol{p}, \boldsymbol{q}) = \begin{pmatrix} \frac{\boldsymbol{p}_1}{m_1} \\ \vdots \\ \frac{\boldsymbol{p}_N}{m_N} \end{pmatrix}$$

Le schéma d'**Euler Symplectique** proposé dans [1] et [2] est donné par

$$\left\{ \begin{array}{rcl} \boldsymbol{p}^{[n+1]} & = & \boldsymbol{p}^{[n]} - h\mathcal{H}_q(\boldsymbol{p}^{[n]}, \boldsymbol{q}^{[n]}) \\ \boldsymbol{q}^{[n+1]} & = & \boldsymbol{q}^{[n]} + h\mathcal{H}_p(\boldsymbol{p}^{[n+1]}, \boldsymbol{q}^{[n+1]}) \end{array} \right.$$

Il est à noter que, pour notre problème, ce schéma est explicite.

Le schéma de **Störmer-Verlet Symplectique** proposé dans [1] et [2] est donné par

$$\left\{ \begin{array}{rcl} \boldsymbol{p}^{[n+1/2]} & = & \boldsymbol{p}^{[n]} - \frac{h}{2}\mathcal{H}_q(\boldsymbol{p}^{[n+1/2]}, \boldsymbol{q}^{[n]}) \\ \boldsymbol{q}^{[n+1]} & = & \boldsymbol{q}^{[n]} + \frac{h}{2}\left(\mathcal{H}_p(\boldsymbol{p}^{[n+1/2]}, \boldsymbol{q}^{[n]}) + \mathcal{H}_p(\boldsymbol{p}^{[n+1/2]}, \boldsymbol{q}^{[n+1]})\right) \\ \boldsymbol{p}^{[n+1]} & = & \boldsymbol{p}^{[n+1/2]} - \frac{h}{2}\mathcal{H}_q(\boldsymbol{p}^{[n+1/2]}, \boldsymbol{q}^{[n+1]}) \end{array} \right.$$

Il est à noter que, pour notre problème, ce schéma est explicite.

Q. 6 (Matlab) *1. Ecrire les fonctions Matlab HSysSol, HpSysSol et HqSysSol correspondant associé au problème à 6 corps (Soleil, Saturne, Jupiter, Uranus, Neptune et Pluton), la structure SysSol étant déclarée en variable globale.*

*2. Ecrire les fonctions Matlab SymplecticEuler et SymplecticStormerVerlet permettant de résoudre ce problème respectivement par le schéma d'**Euler Symplectique** et par schéma de **Störmer-Verlet Symplectique**. L'appel de ses fonctions se fera par*

```
[T,P,Q]=SymplecticEuler(@Hp,@Hq,Pini,Qini,Tini,Tfin,Niter);
[T,P,Q]=SymplecticStormerVerlet(@Hp,@Hq,Pini,Qini,Tini,Tfin,Niter);
```

la structure SysSol étant déclarée en variable globale. Les variables correspondent à :

- *Tini : temps initial (en jours)*
- *Tfinal : temps final (en jours)*
- *Qini : correspond à $\boldsymbol{q}$ au temps initial*
- *Pini : correspond à $\boldsymbol{p}$ au temps initial*
- *Niter : nombre de pas de temps pour le schéma*
- *T : discretisation de l'intervalle de temps,*
- *P : tableau contenant l'approximation de $\boldsymbol{p}$ à tous les temps de discrétisation.*
- *Q : tableau contenant l'approximation de $\boldsymbol{q}$ à tous les temps de discrétisation.*

3. Ecrire le programme prgSysSolSymp permettant de comparer les schémas d'Euler progressif, de Runge-Kutta 4, d'Euler Symplectique et de Störmer-Verlet Symplectique appliqué au problème à 6 corps. On représentera aussi l'énergie du système pour chaque méthode. ▪

3 Annexes

3.1 Quelques E.D.O. du premier ordre

3.1.1 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 $y(t) = \sin(t) + \alpha$.

3.1.2 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 $y(t) = -\cos(t) + 1 + \beta$.

3.1.3 Exemple 3

L'E.D.O.

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

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

Références

[1] C. Lubich E. Hairer and G. Wanner. *Geometrical Numerical Integration*. Springer, 2006.

[2] S. Trilles J. Anton and P. Mercier. Study of integration schemes suited for the long-term extrapolation.