Energétique I | **Méthodes Numériques II**
Sup'Galilée | **Année 2013-2014**

TRAVAUX PRATIQUES - E.D.O.

Durée : 8h00
8h30-12h30 et 13h30-17h30

Travail individuel et personnel

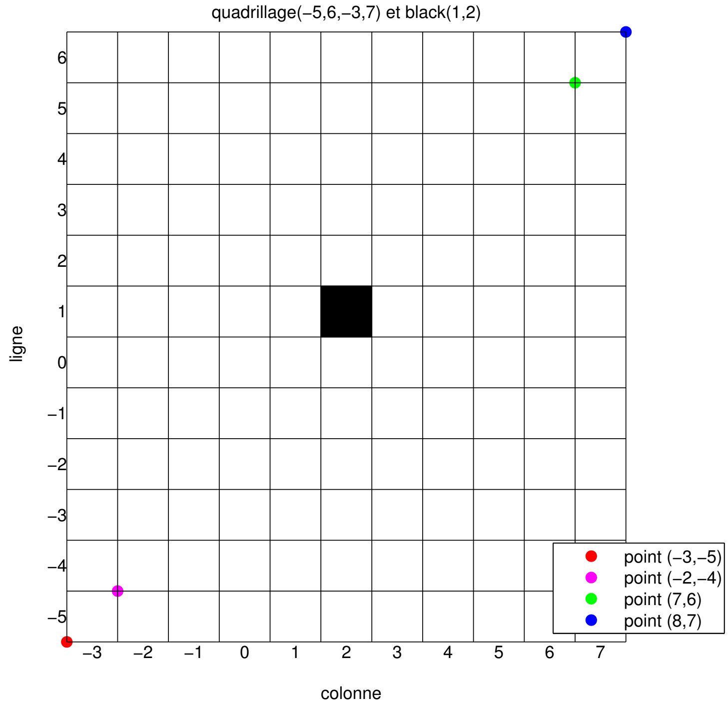
Table des matières

Listings

1 Tests Algorithmique et Matlab

On dispose, entre autres, de la fonction `black` et du programme `Quadrillagefigure` fournis. Dans le programme `Quadrillagefigure` l'appel à la fonction `Quadrillage` a été mis en commentaire.

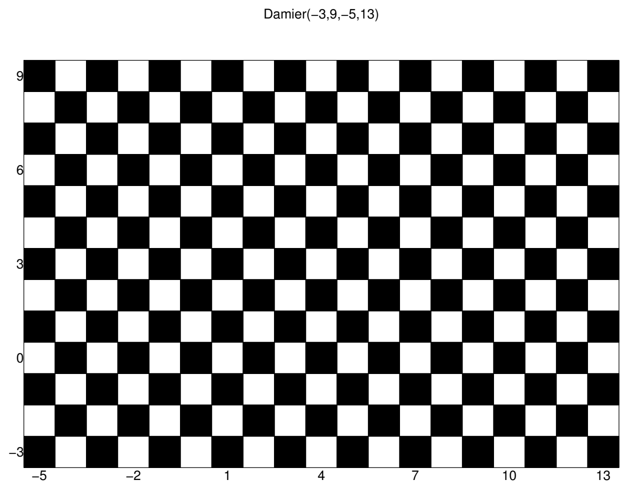
Q. 1 *Ecrire la fonction Matlab* `Quadrillage(imin,imax,jmin,jmax)` *permettant de générer un quadrillage pour les lignes* `imin` *à* `imax` *et les colonnes* `jmin` *à* `jmax`*. Voici un exemple avec la commande* `Quadrillage(-5,6,-3,7)` *représentant uniquement les traits noirs sur la figure :*



On peut tester cette fonction avec le programme `Quadrillagefigure` *fourni pour obtenir la figure précédante.*■

On dispose de la fonction `black(i,j)` qui dessine un pavé noir en ligne i et colonne j d'un quadrillage. Voici le résultat de la commande `black(1,2)` sur le quadrillage précédant.

Q. 2 *Ecrire la fonction* `Damier(imin,imax,jmin,jmax)` *permettant d'obtenir un damier, sur le quadrillage associé (commande* `Quadrillage(imin,imax,jmin,jmax)`*), sachant que le carré en bas à gauche est noir. Voici un exemple d'utilisation :*



■

Q. 3 *Ecrire la fonction* `mosaiqueB(n)` *permettant de créer la mosaïque représentée en Figure 3 sur le quadrillage* `Quadrillage(-n,n,-n,n)`

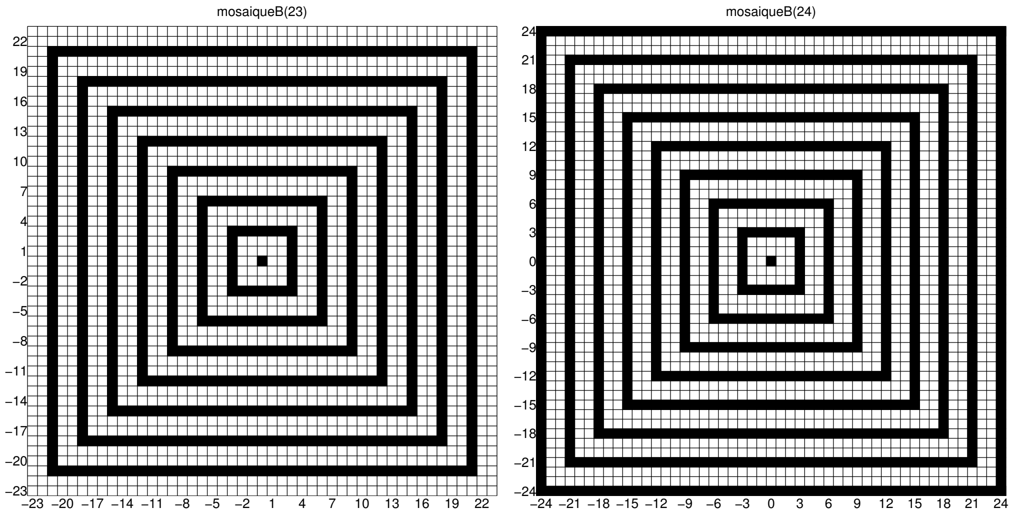


Figure 1 – Résultat des commandes `mosaiqueB(23)` et `mosaiqueB(24)`

Q. 4 *Ecrire la fonction* `mosaiqueC(n)` *permettant de créer la mosaïque représentée en Figure 4 sur le quadrillage* `Quadrillage(-n,n,-n,n)`

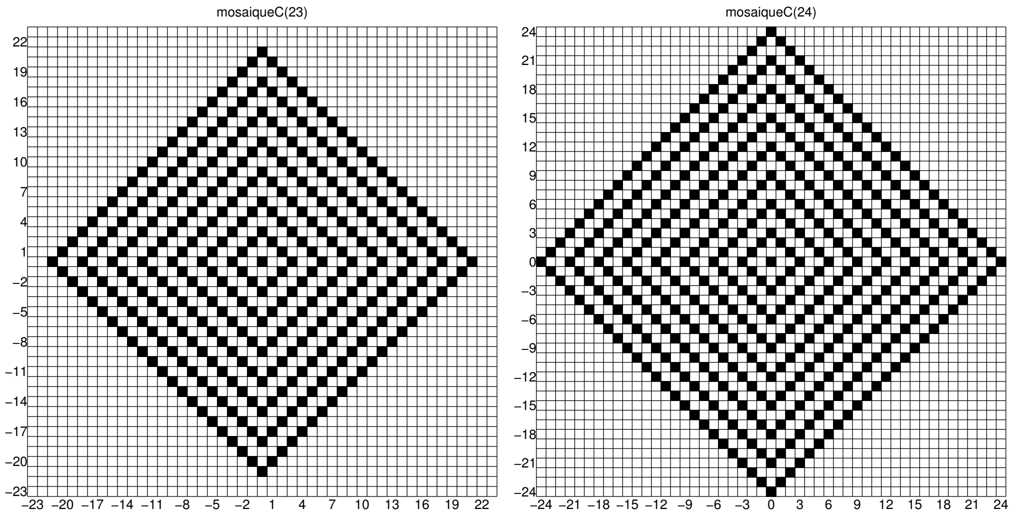


Figure 2 – Résultat des commandes `mosaiqueC(23)` et `mosaiqueC(24)`

A faire avant 10h30

- ◇ Créer une archive compressée (format `zip` ou `tar.gz`) nommée `<NOM>-TP1-Q1a4` contenant les fichiers `Quadrillage.m`, `black.m`, `Damier.m`, `mosaiqueB.m`, `mosaiqueC.m` et tout autre fichier permettant l'exécution de `Damier.m`, `mosaiqueB.m` et `mosaiqueC.m`. Ici `<NOM>` correspond évidemment à votre nom.
- ◇ Envoyer un mail à votre enseignant de TP ayant pour **sujet** `"<NOM> TP1 Q1a4"` et en fichier joint l'archive compressée créée précédement.

2 Résolution numérique d'équations différentielles ordinaires

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

2.1 Rappels de schémas numériques usuels

Définition 2.1 (problème de Cauchy) *Soit $\boldsymbol{f}$ l'application continue définie par*

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

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

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

continue et dérivable, telle que

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

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

■

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

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

Ce schéma est d'ordre 1.

2.1.2 Schéma de la tangente améliorée

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

Ce schéma est d'ordre 2.

2.1.3 Schéma de Runge-Kutta d'ordre 4

$$\begin{array}{lcl} \boldsymbol{k}_1^{[n]} & = & \boldsymbol{f}(t^n, \boldsymbol{y}^{[n]}) \\ \boldsymbol{k}_2^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{k}_1^{[n]}) \\ \boldsymbol{k}_3^{[n]} & = & \boldsymbol{f}(t^n + \frac{h}{2}, \boldsymbol{y}^{[n]} + \frac{h}{2}\boldsymbol{k}_2^{[n]}) \\ \boldsymbol{k}_4^{[n]} & = & \boldsymbol{f}(t^n + h, \boldsymbol{y}^{[n]} + h\boldsymbol{k}_3^{[n]}) \\ \hline \boldsymbol{y}^{[n+1]} & = & \boldsymbol{y}^{[n]} + \frac{h}{6}(\boldsymbol{k}_1^{[n]} + 2\boldsymbol{k}_2^{[n]} + 2\boldsymbol{k}_3^{[n]} + \boldsymbol{k}_4^{[n]}). \end{array} \tag{2.5}$$

2.1.4 Méthodes d'Adams-Bashforth

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

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{2}\left(3\boldsymbol{f}^{[n]} - \boldsymbol{f}^{[n-1]}\right). \tag{2.6}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{12}\left(23\boldsymbol{f}^{[n]} - 16\boldsymbol{f}^{[n-1]} + 5\boldsymbol{f}^{[n-2]}\right). \tag{2.7}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{24}\left(55\boldsymbol{f}^{[n]} - 59\boldsymbol{f}^{[n-1]} + 37\boldsymbol{f}^{[n-2]} - 9\boldsymbol{f}^{[n-3]}\right). \tag{2.8}$$

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

2.1.5 Méthodes d'Adams-Moulton

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{2}\left(\boldsymbol{f}^{[n+1]} + \boldsymbol{f}^{[n]}\right). \tag{2.9}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{12}\left(5\boldsymbol{f}^{[n+1]} + 8\boldsymbol{f}^{[n]} - \boldsymbol{f}^{[n-1]}\right). \tag{2.10}$$

$$\boldsymbol{y}^{[n+1]} = \boldsymbol{y}^{[n]} + \frac{h}{24}\left(9\boldsymbol{f}^{[n+1]} + 19\boldsymbol{f}^{[n]} - 5\boldsymbol{f}^{[n-1]} + \boldsymbol{f}^{[n-2]}\right). \tag{2.11}$$

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

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 : on fournit explicitement une valeur approchée de la solution au $n^{ième}$ pas (soit $\bar{\boldsymbol{y}}^{[n+1]}$), puis on calcule la valeur correspondante de $\boldsymbol{f}$ (soit $\bar{\boldsymbol{f}}^{[n+1]}$) 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 ȳ[n+1] par un schéma explicite ;
Evaluer f̄[n+1] = f(t^{n+1}, ȳ[n+1]);
y[n+1]) à l'aide d'un schéma implicite en remplaçant l'inconnue par ȳ[n+1]);
finpour
```

2.3 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 4.2 pour plusieurs exemples de problèmes de Cauchy avec solutions.

Q. 5 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 (fichier* `redEUP.m`*).*
- `[t,Y]=redTGA(f,a,b,yo,N)`: *schéma de la tangente améliorée (fichier* `redTGA.m`*).*
- `[t,Y]=redRK4(f,a,b,yo,N)`: *schéma de Runge et Kutta d'ordre 4 (fichier* `redRK4.m`*).*
- `[t,Y]=redAB4(f,a,b,yo,N)`: *schéma d'Adams-Bashforth d'ordre 4 (fichier* `redAB4.m`*).*
- `[t,Y]=redPC3(f,a,b,yo,N)`: *schéma de type prédiction-correction utilisant les schémas d'Adams-Bashforth et d"Adams-Moulton d'ordre 3 (fichier* `redPC3.m`*).*

Ici les paramètres `f,a,b,yo` *correspondent respectivement aux* $\boldsymbol{f}, t^0,\ t^0+T,\ \boldsymbol{y}^{[0]}$ *du problème de Cauchy (2.1-2.2). Enfin,* `Y` *est le tableau contenant les* $\boldsymbol{y}^{[n]},\ n \in \{0, \cdots, N\}$ *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 3 pour un exemple de tracé.
3. *Ecrire le programme principal (fichier* `ordre.m`*) permettant de calculer numériquement l'ordre des 5 schémas et de le représenter.*
Voir la figure 4 pour un exemple de tracé.

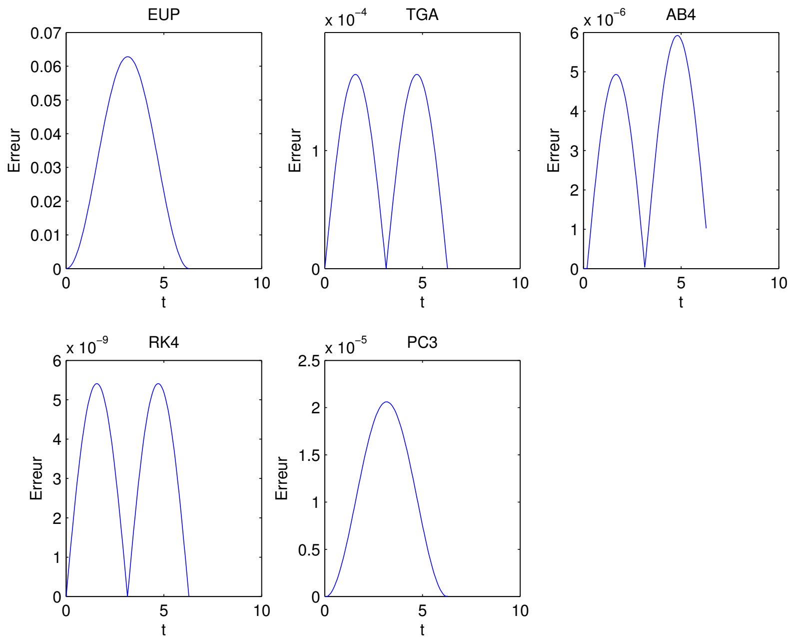


FIGURE 3 – Valeurs absolues des erreurs des 5 schémas

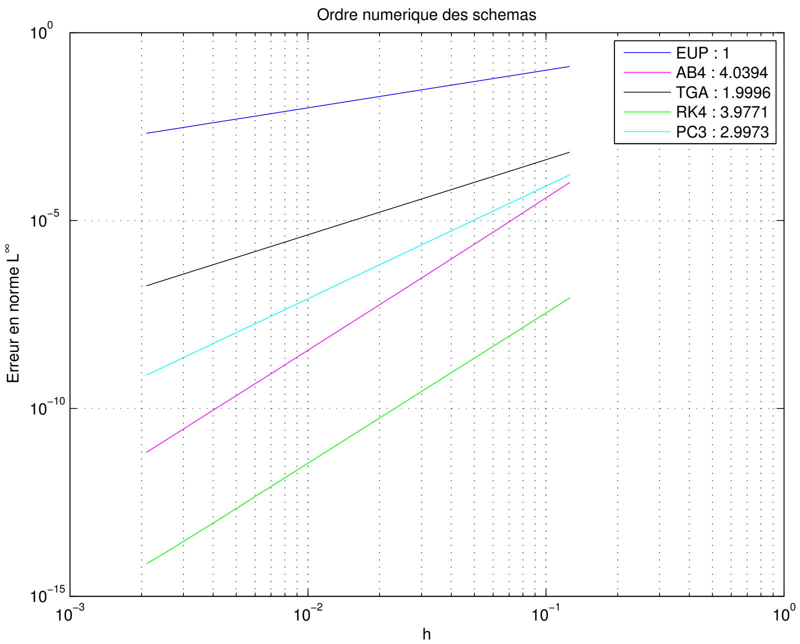


FIGURE 4 – Ordres des 5 schémas

A faire avant 12h30

- ◇ Créer une archive compressée nommée `<NOM>-TP1-Q5` contenant les fichiers `redEUP.m`, `redTGA.m`, `redAB4.m`, `redRK4.m`, `redPC3.m`, `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 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.

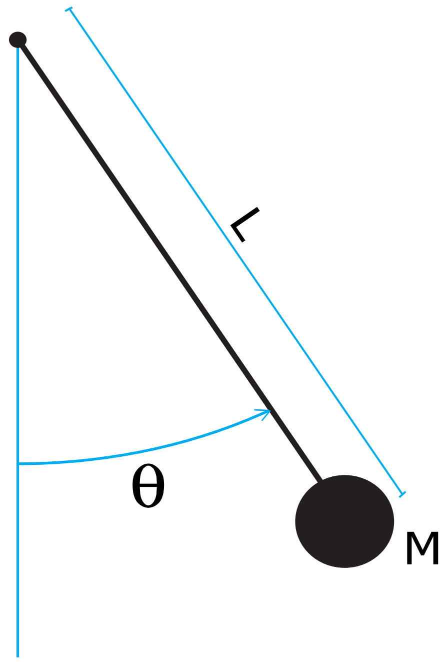


Figure 5 – Pendule pesant

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 (Sur feuille) *1. Ecrire de manière détaillée le problème de Cauchy associé à (3.1-3.2).*

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, ...)*
4. *Expliquer, en détaillant, comment résoudre numériquement (3.1-3.2) par la méthode de Runge-Kutta d'ordre 4.* ∎

Q. 7 (Matlab) *1. 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.)*

2. *Reprendre le fonction* `redRK4.m` *(déjà écrite dans le cas 1d) et la modifier si besoin pour l'adapter à la résolution de problèmes de Cauchy correspondant à des systèmes de d E.D.O. du premier ordre, $d > 1$.*
3. *Ecrire le programme* `prg4` *(fichier* `prg4.m`*) permettant de représenter la position et la vitesse du pendule au cours du temps.*
4. *En s'aidant du programme* `PenduleMovie.m` *et de la fonction* `PlotPendule.m` *(voir annexe 4.1), écrire le programme* `PenduleVideo.m` *permettant de réaliser une vidéo représentant le pendule en mouvement au cours du temps.* ∎

A faire avant 15h30

- ◇ Rendre la réponse à la question 6. N'oubliez pas d'écrire votre nom sur les documents rendus!
- ◇ Créer une archive compressée nommée `<NOM>-TP1-Q7` contenant les fichiers `fCauchy.m`, `prg4.m`, `redrk4.m` et `PenduleMovie.m`, ainsi que tout autre code nécessaire à l'exécution de `prg4` et `PenduleMovie`. Ici `<NOM>` correspond évidemment à votre nom.
- ◇ Envoyer un mail à `cuvelier@math.univ-paris13.fr` ayant pour **sujet** "`<NOM> TP1 Q7`" et en fichier joint l'archive compressée créée précédement.

3.3 Portrait de phase du pendule

Le portait de phase du pendule (M, k et L fixé) est la représentation des courbes paramétrées $(x(t), y(t)) = (\theta(t), \theta'(t))$ calculée avec un grand nombre de données initiales (voir les figures 6 et 7)

Q. 8 (Sur feuille) *Soit le portrait de phase d'un pendule sans viscosité représenté en figure 6. A partir de celui-ci,*

1. *décrire le mouvement du pendule lorsque $(\theta(0); \theta'(0)) = A = (5; 0)$,*
2. *décrire le mouvement du pendule lorsque $(\theta(0); \theta'(0)) = B = (\pi; 1)$,*
3. *décrire le mouvement du pendule lorsque $(\theta(0); \theta'(0)) = C = (\pi; -1)$.* ∎

Q. 9 (Sur feuille) *Soit le portrait de phase d'un pendule avec viscosité représenté en figure 7. A partir de celui-ci,*

1. *décrire le mouvement du pendule lorsque $(\theta(0); \theta'(0)) = E = (0; 11)$,*
2. *décrire le mouvement du pendule lorsque $(\theta(0); \theta'(0)) = F = (6\pi; -10)$.* ∎

Q. 10 (Matlab) *1. Ecrire le programme* `PPSansViscosite.m` *permettant de représenter le portrait de phase du pendule de paramètres $M = L = 1$ et $k = 0$ (figure 6).*

2. *Ecrire le programme* `PPAvecViscosite.m` *permettant de représenter le portrait de phase du pendule de paramètres $M = L = 1$ et $k = 0.5$ (figure 7).* ∎

A faire avant 17h30

- ◇ Rendre les réponses aux questions 8 et 9. N'oubliez pas d'écrire votre nom sur les documents rendus!
- ◇ Créer une archive compressée nommée `<NOM>-TP1-Q10` contenant les fichiers `PPSansViscosite.m` et `PPSansViscosite.m`, ainsi que tout autre code nécessaire à leur exécution. Ici `<NOM>` correspond évidemment à votre nom.
- ◇ Envoyer un mail à `cuvelier@math.univ-paris13.fr` ayant pour **sujet** "`<NOM> TP1 Q10`" et en fichier joint l'archive compressée créée précédement.

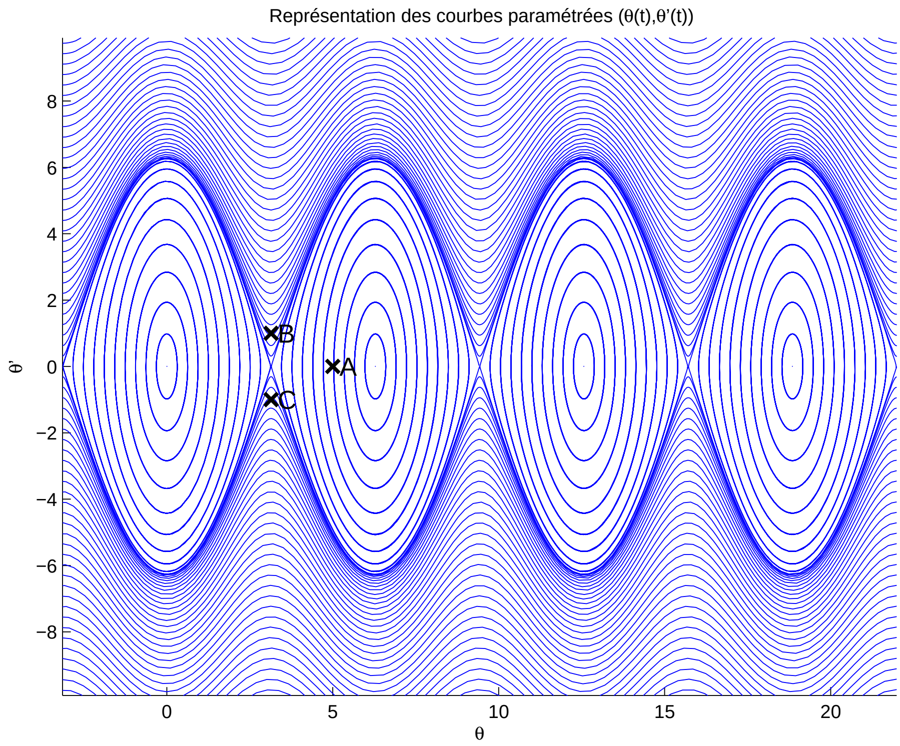


FIGURE 6 – Pendule pesant sans viscosité - Portrait de phase

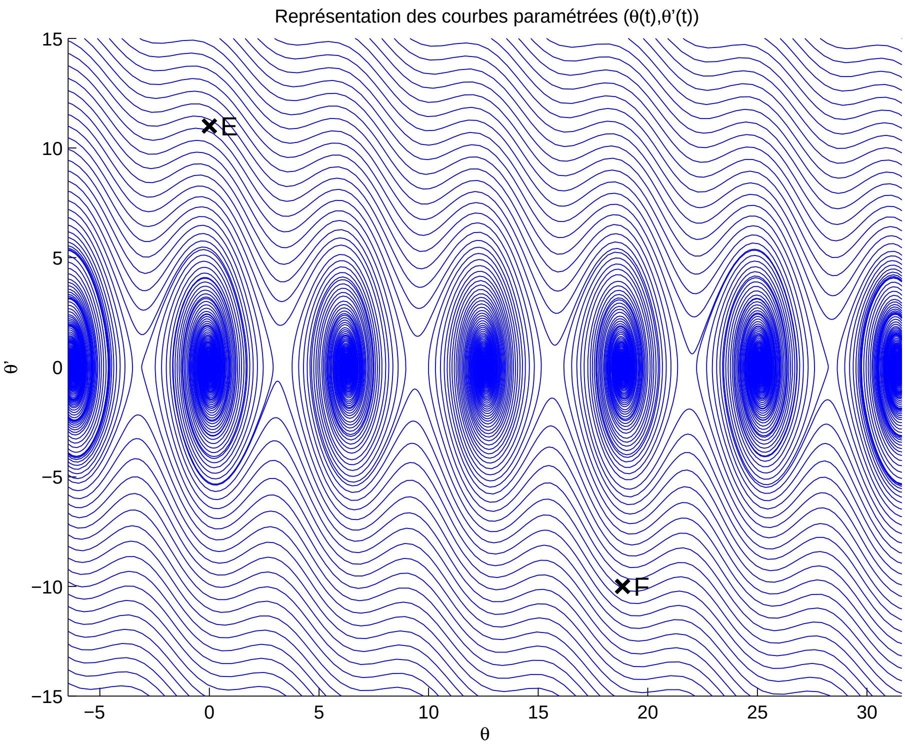


FIGURE 7 – Pendule pesant avec viscosité - Portrait de phase

4 Annexes

4.1 listings

Listing 1 – edo1d.m

```
% Resolution du probleme de Cauchy (scalaire, m=1)
%   Trouver y solution de
%       y'(t)=f(t,y(t)), pour t dans [a,b]
%       y(a)=ya un reel donne.
% Utilisation du schema d'Euler pour la resolution numerique et
% representation graphique de la solution exacte et de la solution
% approchee.

clear all
close all

f=@(t,y) -y*sin(t);
yex=@(t) exp(cos(t));

a=0;b=6*pi;
[t,y]=redEUL1d(f,a,b,yex(a),2000);
yEx=yex(t);

figure(1)
plot(t,y,'b',t,yEx,'r')
legend('Euler','Exacte')
xlabel('t')
print -depsc2 edo1d.eps
```

Listing 2 – redEUL1d.m

```
function [t,Y]=redEUL1d(f,a,b,Y0,n)
% Resolution du probleme de Cauchy (scalaire, m=1)
%   |  Trouver la fonction Y : [a,b] -> R telle que
%   |      Y'(t)=f(t,Y(t)), pour t dans [a,b]
%   |  avec Y(a)=Y0 un reel donne.
% par le schema d'Euler explicite.
% f : fonction du probleme de Cauchy
% n : nombre d'intervalles dans la discretisation de [a,b]
h=(b-a)/n;
t=a:h:b;
Y=zeros(1,n+1);
Y(1)=Y0;
for i=1:n
  Y(i+1)=Y(i)+h*f(t(i),Y(i));
end
```

Listing 3 – PenduleMovie.m

```
% Creation d'un fichier avi
%   Matlab version >= 2010b ?
%   Teste avec version 2012b
clear all
close all

FrameRate=30;Step=10;
T=5;        % temps final
N=T*FrameRate*Step;
L=3;R=0.2; % Parametres graphiques du pendule

% pour simuler la rotation : physiquement faux (sauf sans viscosite et en apenseteur!)
theta=@(t) pi*t;
t=0:T/N:T;

n=length(t);
figure(1);

%
writerObj = VideoWriter('VideoTest.avi','Motion JPEG AVI');
set(writerObj,'FrameRate',FrameRate);
open(writerObj);
```

```
for j=1:Step:n
  cla
  PlotPendule(L,R,theta(t(j)));
  title(sprintf('\\theta(%f)=%f',t(j),theta(t(j)) ))

  axis on
  drawnow
  writeVideo(writerObj,getframe(gcf));
end

close(writerObj);
```

Listing 4 – `PlotPendule.m`

```
function PlotPendule(L,R,theta)
% Representation d'un pendule de parametre L (longueur de la tige)
% et R (rayon du disque) en position angulaire theta
hold on
% axe de rotation
plot(0,0,'ro')
y=[-L:1/100:0];

% Coordonnee de la tige
X=-y*sin(theta);
Y=y*cos(theta);
plot(X,Y,'r')
C=[L*sin(theta);-L*cos(theta)]; % Centre du disque
t=0:pi/100:2*pi;
% Coordonnee du cercle
XC=C(1)+R*cos(t);
YC=C(2)+R*sin(t);
fill(XC,YC,'r');
LL=L+2*R;
axis equal
axis([-LL LL -LL LL])
axis off
hold off
```

4.2 Quelques E.D.O. du premier ordre

4.2.1 Exemple 1

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

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

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

4.2.2 Exemple 2

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

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

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

4.2.3 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 $y(t) = \exp(\cos(t))$.