Equations Différentielles Ordinaires (EDO)

TP Matlab

Exercice 1

On considère le problème de Cauchy suivant :

$$(P) \quad \begin{cases} (1+t^2)y' - 2ty = 3t - 1, & t \in [0,2] \\ y(0) = 1 \end{cases}$$

1) Montrer que la résolution de (P) par la méthode d'Euler <u>implicite</u> revient à calculer les termes de la suite (y_n) définie par :

$$\begin{cases} y_{n+1} = \dfrac{(1+t_{n+1}^2)y_n + h(3t_{n+1} - 1)}{1 + t_{n+1}^2 - 2ht_{n+1}} \\ y_0 = 1 \end{cases}$$

où h est le pas de calcul et $t_{n+1} = t_n + h$.

2) Compléter le programme Matlab ci-dessous afin que celui-ci permette de résoudre le problème de Cauchy (P) à l'aide de :

- la méthode d'Euler explicite
- la méthode d'Euler implicite
- la méthode de Runge-Kutta d'ordre 2 (on prendra $\beta = 1/2$)
- la méthode de Runge-Kutta d'ordre 4

```
clear variables;
close all;

% fonction f définie par y'=f(t,y)
f=@(t,y)(_____________________________________________);

% initialisation
tmin=_______;
tmax=_______;
h=0.01;                    % pas de calcul
t=tmin:h:tmax;
y=zeros(4,length(t));      % 4 lignes : 1 ligne par méthode
y(:,1)=_____;
beta=0.5;                  % paramètre de la méthode de Runge-Kutta d'ordre 2

for k=1:__________________

    % Euler explicite
    ____________________________________________________________________________

    % Euler implicite
    ____________________________________________________________________________

    % Runge-Kutta d'ordre 2 (3 lignes de code)
    ____________________________________________________________________________

   % Runge-Kutta d'ordre 4 (5 lignes de code)
    ____________________________________________________________________________
end

% affichage
figure(1);hold on;
plot(t,y(1,:),'c');
plot(t,y(2,:),'m');
plot(t,y(3,:),'r');
plot(t,y(4,:),'b');
lg=legend('Euler explicite','Euler implicite','RK2','RK4');
```

Exercice 2

On s'intéresse à la résolution du problème de Cauchy suivant :

$$\begin{cases} y'(t) + (4t^3 + 5)y(t) = t^3 e^{-5t}, & t \in [0;1] \\ y(0) = 1 \end{cases} \quad (1)$$

La solution exacte de ce problème est :

$$y(t) = \frac{1}{4}\left(e^{t^4} + 3\right)e^{-t(t^3+5)}$$

1) Résoudre ce problème à l'aide de la méthode d'Euler et de la méthode de Runge-Kutta d'ordre 2. On suivra les consignes suivantes :
   - Récupérer le fichier `TP_EDO_Ex2.m` sur Teams
   - Compléter la ligne 5 contenant la fonction f définie par $y'(t) = f(t,y)$
   - Compléter la ligne 6 contenant la fonction définie par la solution exacte de l'équation
   - Coder les algorithmes de la méthode d'Euler et de la méthode de Runge-Kutta d'ordre 2 ; pour cela on créera deux nouvelles fonctions dont les prototypes sont les suivants :

     ```
     [yEuler,t]=fct_Euler(y0,tmin,tmax,h,f)
     [yRK,t]=fct_RK2(y0,tmin,tmax,h,beta,f)
     ```

     où les paramètres d'entrée sont :
     - `y0` : condition initiale
     - `tmin, tmax` : valeurs minimale et maximale de t
     - `h` : pas de calcul
     - `f` : fonction définie par $y'(t) = f(t,y)$
     - `beta` : paramètre spécifique à la méthode de Runge-Kutta d'ordre 2 (cf. cours)

     et les parmètres de sortie sont :
     - `yEuler` / `yRK2` : vecteur contenant la solution approchée
     - `t` : vecteur contenant les valeurs de t entre `tmin` et `tmax` avec un pas de `h`

     Ces fonctions seront codées dans 2 nouveaux fichiers Matlab, `fct_Euler.m` et `fct_RK2.m`, que l'on placera dans le même répertoire que TP_EDO_Ex2.m
   - Afficher dans une même fenêtre mais sur 2 graphiques différents (voir Fig. 1 ci-après) les résultats suivants :
     - 1er graphique
       - La solution exacte
       - Les solutions approchées par la méthode d'Euler pour des pas de 0.1 et 0.05
       - La solution approchée par la méthode de Runge-Kutta d'ordre 2 pour un pas de 0.05
     - 2nd graphique
       - les trois courbes d'erreur associées aux trois solutions approchées calculées précédemment
   - Commenter les résultats

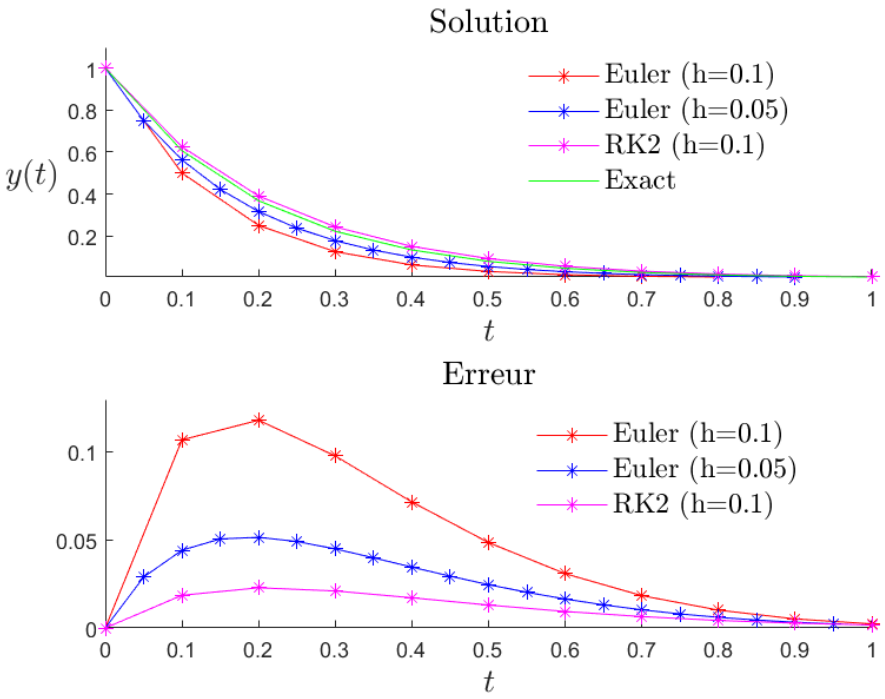

Figure 1 : Solutions du problème de Cauchy (1) par la méthode d'Euler et la méthode de Runge-Kutta d'ordre 2.

2) L'objectif de cette question est d'étudier et de comparer les convergences des méthodes d'Euler et de Runge-Kutta d'ordre 2. Pour cela, calculer la solution approchée du problème pour des valeurs de pas h_i allant de 0.02 à 0.1 avec un pas de 0.01 (9 valeurs de pas différentes). Pour chaque pas h_i, calculer l'erreur maximale entre la solution et la solution approchée, on la notera $maxerr(h_i)$, puis afficher les points de coordonnées $(h_i, maxerr(h_i))$ dans une nouvelle fenêtre (voir Fig. 2 ci-après). Commenter le résultat obtenu.

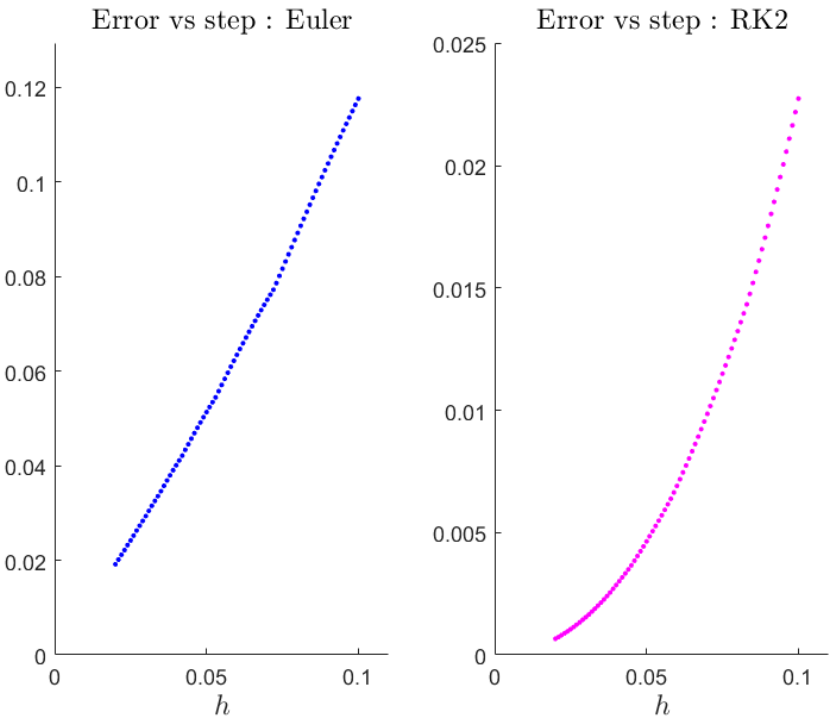

Figure 2 : Erreur (en fonction du pas) des méthodes d'Euler et Runge-Kutta d'ordre 2 pour la résolution du problème de Cauchy (1).

Exercice 3

On considère le problème de Cauchy suivant :

$$\begin{cases} y^{(3)} + \dfrac{1}{y} = te^{-t^2}, & t \in \mathbb{R} \\ y(0) = 1 \\ y'(0) = 0 \\ y''(0) = 1 \end{cases}$$

Expliquez comment vous feriez pour résoudre ce problème de Cauchy par la méthode d'Euler explicite puis écrire un programme Matlab permettant de trouver la solution et de l'afficher.

Exercice 4

On propose d'étudier le système d'équations différentielles de Lokta-Volterra suivant, encore appelé « modèle proies-prédateurs » (cf. cours) :

$$\begin{cases} x'(t) = x(t)\big(\alpha - \beta y(t)\big) \\ y'(t) = -y(t)(\gamma - \delta x(t)) \end{cases} \quad (2)$$

avec $\alpha = 1,\ \beta = 0{,}5,\ \gamma = 2$ et $\delta = 1$.

Indications :

- Récupérer le fichier TP_EDO_Ex4.m sur Teams
- Coder l'algorithme de la méthode d'Euler dans la fonction `fct_Euler_2D` dont le prototype est le suivant :

```
function [x,y,t]=Euler_2D(x0,y0,tmin,tmax,pas,F,G)
```

- Résoudre les équations de Lokta-Volterra sur l'intervalle $[0; 15]$ avec les conditions initiales $x(0) = 2$ et $y(0) = 0{,}5$ et pour un pas temporel de 0,01
- Afficher dans 2 figures différentes :
  - les courbes des populations des proies et des prédateurs en fonction du temps (voir Fig. 3 ci-après)
  - les trajectoires proies-prédateurs en superposition du champ de vecteurs défini par la fonction $(x, y) \mapsto \big(x(\alpha - \beta y), -y(\gamma - \delta y)\big)$ (voir Fig. 4 ci-après)
- Commenter les résultats
- Coder l'algorithme de la méthode de Runge Kutta dans la fonction `fct_RK2_2D` dont le prototype est le suivant :

```
function [x,y,t]=RK2_2D(x0,y0,tmin,tmax,pas,beta,F,G)
```

- Résoudre à nouveau les équations avec la méthode de Runge-Kutta (même intervalle et même pas temporels) pour les conditions initiales $x(0) = 2$ et $y(0) = 0{,}5$, puis $x(0) = 3$ et $y(0) = 2$
- Afficher les trajectoires obtenues dans une 3ème figure (voir Fig. 5 ci-après)
- Commenter les résultats

Pour quelles valeurs de (x_0, y_0), en fonction de $\alpha, \beta, \gamma, \delta$, la population de proies et de prédateurs reste constante au cours du temps ?

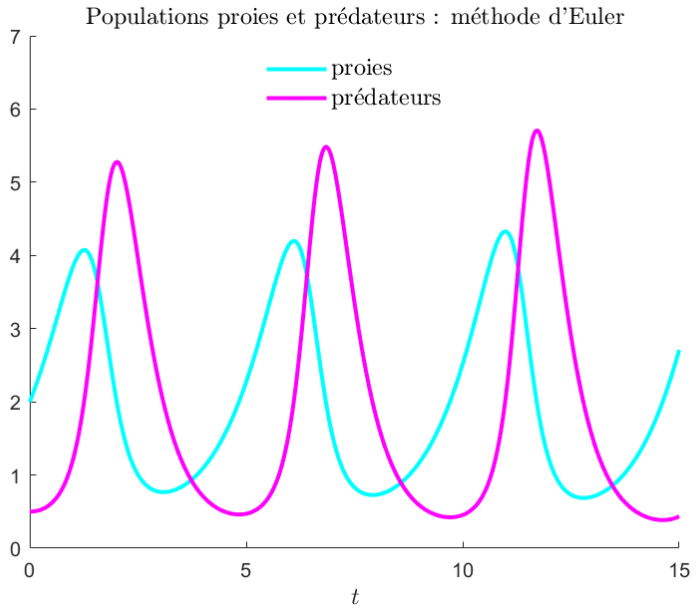

Figure 3 : Evolutions en fonction du temps des populations des proies et des prédateurs modélisées par le système d'équations différentielles (2) résolu de façon approchée à l'aide de la méthode d'Euler.

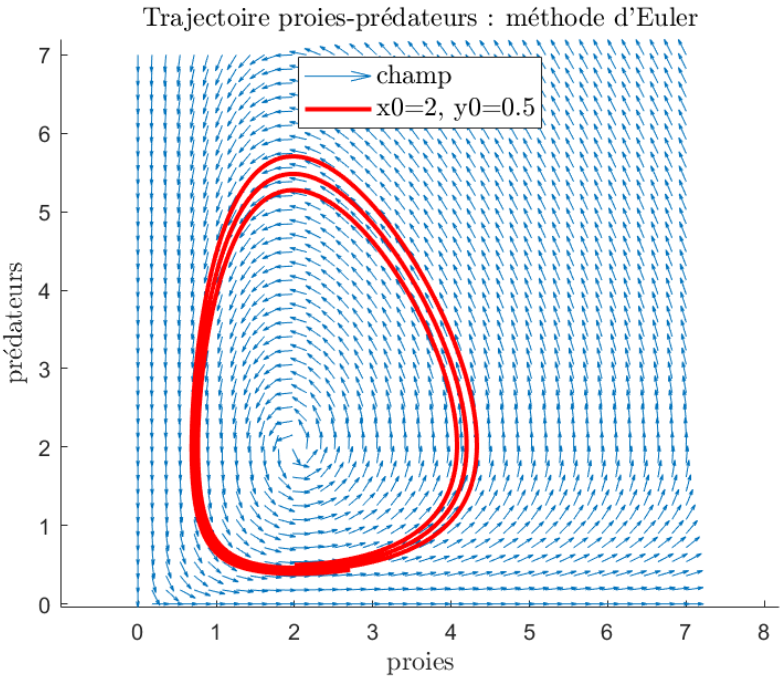

Figure 4 : Trajectoire du couple proie-prédateur dans le champ de vecteurs. Les calculs sont réalisés à l'aide de la méthode d'Euler.

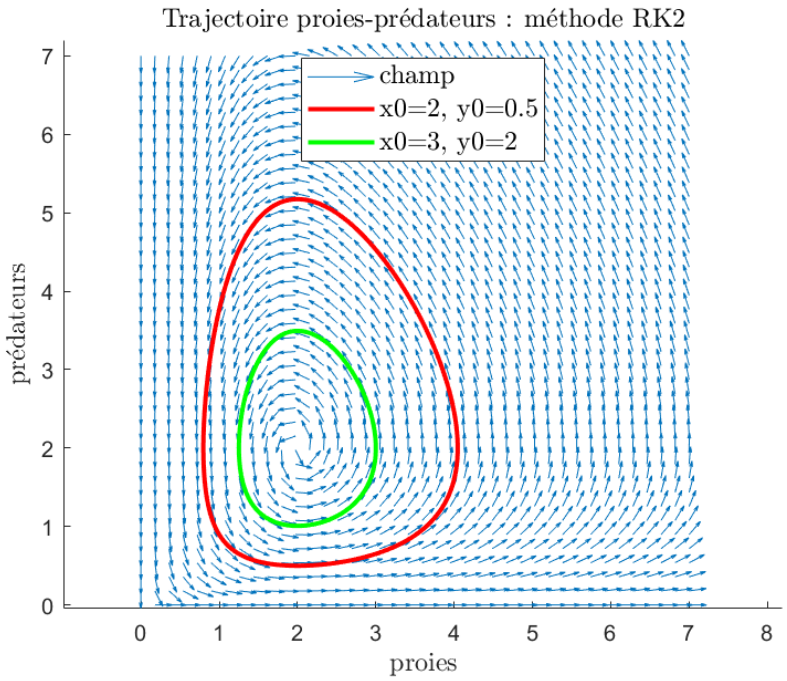

Figure 5 : Deux trajectoires du couple proie-prédateur dans le champ de vecteurs pour des conditions initiales différentes. Les calculs sont réalisés à l'aide de la méthode de Runge-Kutta d'ordre 2.

Exercice 5 : Pendule simple amorti

On considère un pendule simple composé d'une bille de masse m suspendue à un fil de longueur l (et de masse négligeable) que l'on éloigne de sa position d'équilibre d'un angle θ_0 et que l'on lance avec une vitesse angulaire initiale θ_0'.

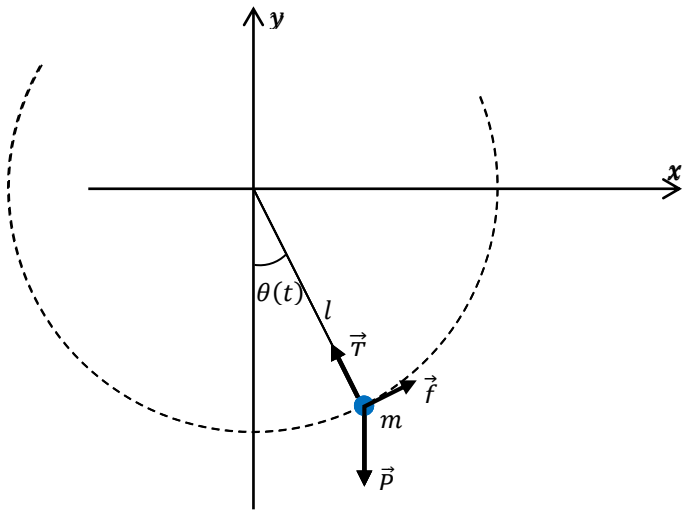

Figure 6 : Pendule simple amorti

La bille est soumise à trois forces : son poids ($\vec{P}$), la tension du fils ($\vec{T}$) et des forces de frottement de l'air ($\vec{f}$). On s'intéresse à l'évolution de l'angle θ en fonction du temps. On montre que la fonction $\theta(t)$ obéit à l'équation différentielle suivante :

$$\theta''(t) + \gamma\theta'(t) + \omega^2 \sin\theta(t) = 0 \qquad (1)$$

où

- γ est un coefficient tenant compte de la résistance de l'air (en toute rigueur $\gamma = \frac{6\pi r\eta}{m}$ où r est le rayon de la bille et $\eta = 1{,}8.10^{-5}\ kg.m^{-1}.s^{-1}$ est la viscosité de l'air à 20° C)
- $\omega = \sqrt{l/g}$ est la fréquence propre d'oscillation du pendule, c'est-à-dire la fréquence d'oscillation lorsque $\gamma = 0$ (g est l'accélération de la pesanteur)

L'énergie totale d'un tel système est une fonction du temps, elle est la somme de l'énergie cinétique et de l'énergie potentielle :

$$E(t) = E_c(t) + E_p(t)$$

avec

$$E_c(t) = \frac{1}{2}ml^2\theta'(t)^2 \quad : \text{énergie cinétique}$$

et

$$E_p(t) = mgl(1 - \cos\theta(t)) \quad : \text{énergie potentielle}$$

Lorsque les frottements sont négligeables le pendule oscille indéfiniment à la fréquence angulaire ω sans aucun amortissement. Dans ce cas l'énergie totale est constante au cours du temps car les pertes d'énergie potentielle compensent les gains d'énergie cinétique et réciproquement.

L'équation différentielle (1) est une équation non linéaire d'ordre 2 qui n'admet pas de solution analytique. Nous proposons ici de résoudre cette équation à l'aide de 3 méthodes numériques : la méthode d'Euler explicite et les méthodes de Runge-Kutta d'ordre 2 et 4.

Pour cela, on propose de compléter le programme Matlab ci-dessous de façon à ce qu'il permette de :

- visualiser la trajectoire de la bille
  - dans l'espace réel (voir Fig. 7a)
  - dans l'espace des phases, c'est-à-dire un plan d'abscisse θ et d'ordonnées θ' (voir Fig. 7b)
- visualiser l'évolution de l'énergie cinétique, de l'énergie potentielle et de l'énergie totale en fonction du temps (voir Fig. 7c)

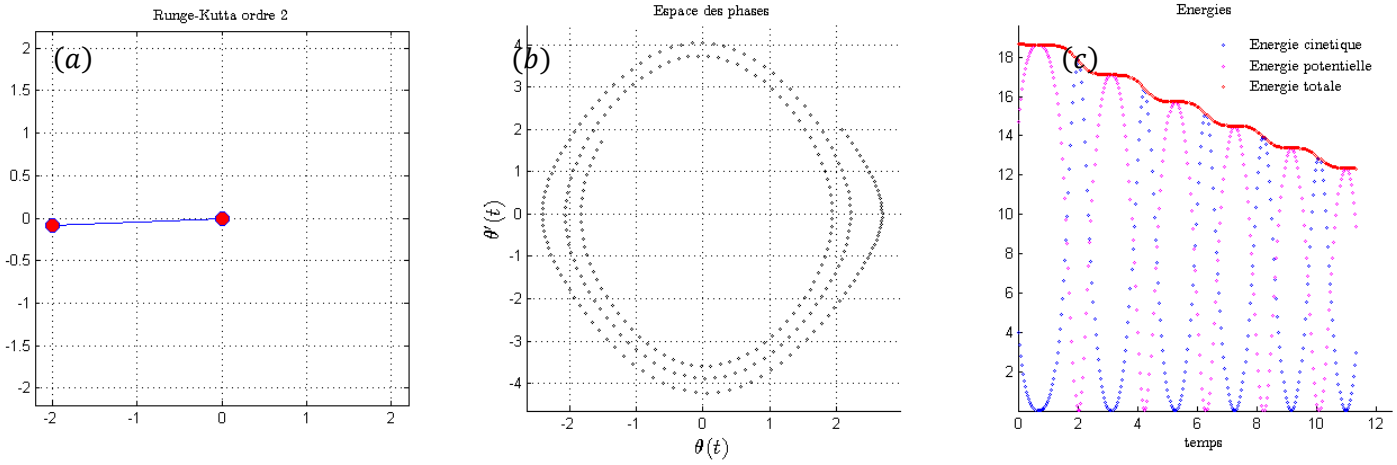

Figure 7 : Trajectoires dans l'espace réel (a), dans l'espace des phases (b) et énergies (c) d'un pendule simple de masse par la méthode de Runge-Kutta d'ordre 2.

Indication : utiliser les fonctions `fct_Euler_2D`, `fct_RK2_2D` développées pour le modèle proies-prédateurs, et coder l'algorithme de la méthode de Runge-Kutta d'ordre 4 dans une fonction qu'on appellera `fct_RK4_2D`.

Programme à compléter :

```
clear variables;
close all;

% paramètres physiques
m=0.7;                % masse de la bille (kg)
r=0.035;              % rayon de la bille (m)
eta=0.000018;         % coeff. de viscosité de l'air à 20°C (kg.m^-1.s^-1)
gamma=6*pi*r*eta/m;   % frottements (s^-1)
gr=9.8;               % accéleration de la pesanteur (m.s^-2)
l=2;                  % longueur du fil (m)
omega=sqrt(gr/l);     % fréquence propre (rad.s^-1)
T0=2*pi/omega;        % (pseudo-)période du pendule (s)

% autres paramètres
tmin=0;               % instant initial
tmax=4*T0;            % instant final
pas=0.001;            % pas de calcul
fprintf('Durée de l''expérience physique : %1.2f\n',tmax-tmin);

% fonctions Y'=F(Y) avec ici Y=(theta,z) et F(Y)=(f,g)
f=@(___,_____,__)(_____________________________);
g=@(___,_____,__)(_____________________________);

% conditions initiales
theta0=2*pi/3;    % angle initial (rad)
thetap0=0;        % vitesse angulaire initiale (rad/s)

% choix de la méthode
method=1;    % 1 : Euler
             % 2 : Runge-Kutta ordre 2
             % 3 : Runge-Kutta ordre 4

% calculs numériques
switch method
    case 1
        [theta,z,t]=fct_Euler_2D(____,____,____,____,____,____,____);
        strTitle='Euler';
    case 2
        beta=1;
        [theta,z,t]=fct_RK2_2D(____,____,____,____,____,____,____);
        strTitle='Runge-Kutta ordre 2';
    case 3
        [theta,z,t]=fct_RK4_2D(____,____,____,____,____,____,____);
        strTitle='Runge-Kutta ordre 4';
end

% énergies du pendule
Ec=_____________________________;      % énergie cinétique
Ep=_____________________________;      % énergie potentielle
E=_____________________________;       % énergie totale

% affichage des résultats
figure(1);
xmin=-1;xmax=1;
ymin=-1;ymax=1;

tic;
```

```
for k=1:65:length(theta)  % régler le pas de sorte à obtenir un mouvement réaliste

    % espace réel
    subplot(131);
    x=_____________;
    y=_____________;
    plot([0,x],[0,y],'Marker','o','MarkerFacecolor','r','MarkerSize',10);
    axis('equal');
    axis(1.1*[xmin,xmax,ymin,ymax]);
    grid 'on';
    t1=title(strTitle);
    set(t1,'interpreter','latex');


    % espace des phases
    subplot(132);hold on;
    t3=title('Espace des phases');
    set(t3,'interpreter','latex');
    h=plot(_____________,_____________,'ok');
    set(h,'MarkerSize',2);
    axis(1.1*[min(theta),max(theta),min(z),max(z)]);
    grid on;

    % énergies
    subplot(133);hold on;
    t2=title('Energies');
    set(t2,'interpreter','latex');
    h=plot(t(k),________,'ob',t(k),________,'om',t(k),_______,'or');
    set(h,'MarkerSize',2);
    axis(1.1*[tmin,tmax,min(Ec),max(Ec)]);

    pause(0.0001);
end

cputime=toc;
fprintf('Durée de la simulation numérique : %1.2f\n',cputime);

% labels des axes de la figure du portrait de phase
subplot(132);
l1=xlabel('$\theta(t)$','interpreter','latex');
set(l1,'FontSize',14);
l2=ylabel('$\theta''(t)$','interpreter','latex');
set(l2,'FontSize',14);

% légende de la figure des énergies
subplot(133);
g1=legend('Energie cinetique','Energie potentielle','Energie totale');
legend('boxoff');
set(g1,'interpreter','latex');
xlabel('temps','interpreter','latex');
```

Exercice 6 : Etude du mouvement képlerien

On considère la trajectoire d'une planète orbitant autour d'une étoile. On suppose que l'orbite est elliptique avec une forte excentricité (voir figure 8). Cela signifie que la distance entre la planète et l'étoile varie fortement entre l'aphélie (point le plus éloigné du l'étoile) et le périhélie (point le plus proche). Dans l'hypothèse où la masse de la planète (notée m) est très inférieure à celle de l'étoile (notée M), le centre de gravité du système {étoile-planète} se situe à proximité du centre de l'étoile. Dans ce cas, le mouvement de la planète $P = \big(x(t), y(t)\big)$ est modélisé par le système d'équations différentielles suivant :

$$\begin{cases} x''(t) + \mathcal{G}M \dfrac{x(t)}{[x(t)^2 + y(t)^2]^{3/2}} = 0 \\ y''(t) + \mathcal{G}M \dfrac{y(t)}{[x(t)^2 + y(t)^2]^{3/2}} = 0 \end{cases}$$

où $\mathcal{G}$ est la constante de gravitation universelle, elle apparait dans la troisième loi de Képler :

$$\frac{a^3}{T^2} = \frac{\mathcal{G}M}{4\pi^2}$$

où a est le demi-grand axe de l'orbite elliptique et T est la période de révolution de la planète autour de l'étoile.

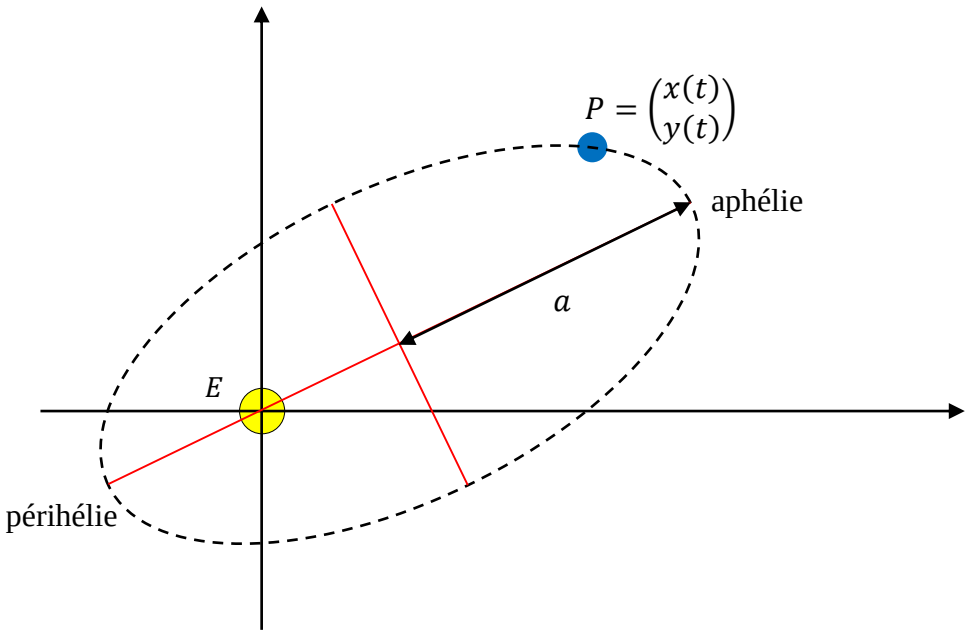

Figure 8 : Orbite elliptique (avec forte excentricité)
d'une planète P autour d'une étoile E

1) Développer un programme permettant de calculer et d'afficher (sous la forme d'une animation) la trajectoire de la planète autour de l'étoile en utilisant :
   a. la méthode d'Euler
   b. la méthode de Runge-Kutta d'ordre 4
   On prendra les valeurs numériques suivantes :
   $$a = 1\ ua, \qquad T = 1\ \text{an}, \qquad M = 1\ \text{masse solaire}$$
   Ainsi, dans ce système d'unité, on a :
   $$\mathcal{G} = 4\pi^2$$

On se placera dans les conditions initiales suivantes :

$$\begin{cases} x(0) = 0{,}5 \\ y(0) = 0 \\ x'(0) = 0 \\ y'(0) = 11{,}5 \end{cases}$$

On calculera la trajectoire sur l'intervalle de temps égale à 2 périodes.

Pour information :

- ua signifie « unité astronomique », $1\ ua =$ distance Terre-Soleil
- dans le système international (SI) on a :

  $1\ ua \simeq 1{,}5.10^{11}\ m, \qquad M_{\text{Soleil}} \simeq 2.10^{30}\ kg, \qquad m_{\text{Terre}} \simeq 6.10^{24}\ kg, \qquad \mathcal{G} \simeq 6{,}67.10^{-11}\ m^3 kg^{-1} s^{-2}$

  [ces valeurs sont données à titre d'information, elle ne sont pas directement utiles pour résoudre le problème, elle ne doivent donc pas figurer dans votre code]

2) Les méthodes d'Euler et de Runge-Kutta n'étant pas satisfaisantes, résoudre le problème par la méthode d'Euler-Richardson (cf. https://femto-physique.fr/omp/euler-richardson.php).

Exercice 7 : pendule double non-amorti

On considère le double pendule constitué d'une bille de masse m_1 suspendue à un fil de longueur l_1 (et de masse négligeable) et d'une seconde bille de masse m_2 suspendue à un fil accrochée à la première masse, ce fil est de longueur l_2 (et de masse négligeable). On ne tient pas compte de l'amortissement des mouvements des billes en supposant que les forces de frottement de l'air sont négligeables. Le système peut être représenté par la figure suivante :

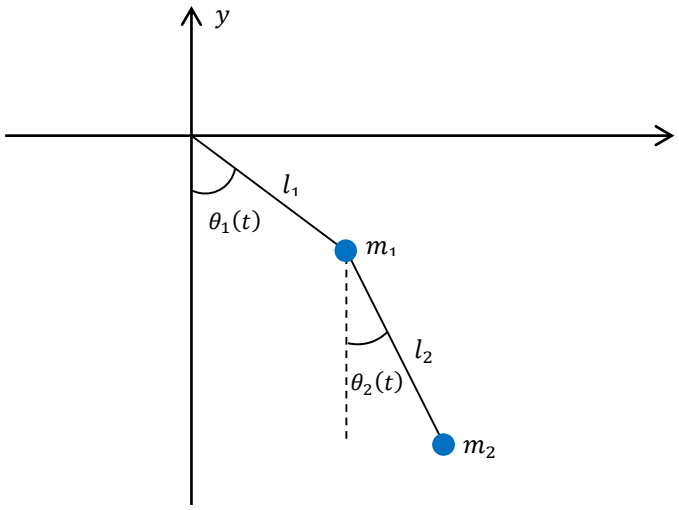

Figure 3 : Pendule double non-amorti

On montre alors que les angles $\theta_1(t)$ et $\theta_2(t)$ obéissent au système d'équations différentielles suivant :

$$\begin{cases} (m_1 + m_2) l_1 \theta_1''(t) + m_2 l_2 \theta_2''(t) \cos\big(\theta_1(t) - \theta_2(t)\big) + m_2 l_2 \theta_2'(t)^2 \sin\big(\theta_1(t) - \theta_2(t)\big) + (m_1 + m_2) g \sin\big(\theta_1(t) - \theta_2(t)\big) = 0 \\ l_1 \theta_1''(t) \cos\big(\theta_1(t) - \theta_2(t)\big) + l_2 \theta_2''(t) - l_1 \theta_1'(t)^2 \sin\big(\theta_1(t) - \theta_2(t)\big) + g \sin\big(\theta_2(t)\big) = 0 \end{cases}$$

Comme pour le pendule simple, ce sont des équations différentielles non linéaires d'ordre 2 ; ce système n'admet pas de solution exacte. On peut toutefois le résoudre de façon approchée à l'aide d'une méthode numérique comme la méthode de Runge-Kutta par exemple. Pour cela on transforme le système précédent en un système de 4 équations différentielles (non linéaires) d'ordre 1 que l'on peut écrire sous le forme suivante :

$$\begin{cases} \theta_1' = f_1(t, \theta_1, \theta_2, z_1, z_2) \\ \theta_2' = f_2(t, \theta_1, \theta_2, z_1, z_2) \\ z_1' = g_1(t, \theta_1, \theta_2, z_1, z_2) \\ z_2' = g_2(t, \theta_1, \theta_2, z_1, z_2) \end{cases}$$

avec

$$f_1(t, \theta_1, \theta_2, z_1, z_2) = z_1$$

$$f_2(t, \theta_1, \theta_2, z_1, z_2) = z_2$$

$$g_1(t, \theta_1, \theta_2, z_1, z_2) = -\frac{g(2m_1 + m_2)\sin\theta_1 + m_2[g\sin(\theta_1 - 2\theta_2) + 2(l_2 z_2^2 + l_1 z_1^2 \cos(\theta_1 - \theta_2))\sin(\theta_1 - \theta_2)]}{2l_1[m_1 + m_2\sin^2(\theta_1 - \theta_2)]}$$

$$g_2(t, \theta_1, \theta_2, z_1, z_2) = \frac{(m_1 + m_2)(l_1 z_1^2 + g\cos\theta_1) + l_2 m_2 z_2^2 \cos(\theta_1 - \theta_2)}{l_2[m_1 + m_2\sin^2(\theta_1 - \theta_2)]}\sin(\theta_1 - \theta_2)$$

1) Coder l'algorithme de la méthode de Runge-Kutta d'ordre 4 en dimension 4 (s'inspirer de la méthode de Runge-Kutta d'ordre 4 en dimension 2) ; pour cela on créera un nouveau fichier Matlab, `fct_RK4_4D.m` que l'on placera dans le même répertoire que le programme principal.
2) Coder le programme principal ; ce dernier doit fonctionner selon deux modes :
   - mode 1 : visualisation de l'évolution d'un double pendule (cf. vidéo `DouplePendule_mode1.avi` sur Teams)
   - mode 2 : visualisation de l'évolution de deux doubles pendules partant avec des conditions initiales très proches (cf. vidéo `DouplePendule_mode2.avi` sur Teams)

Ce sujet est relativement libre et ouvert ; ne pas hésiter à proposer des idées d'évolution du programme principal qui ne sont pas explicitement demandées dans l'énoncé. Vous pouvez éventuellement joindre à votre rapport des séquences vidéo (vos noms doivent alors figurer dans les noms des fichiers vidéo), surtout si elles apportent des informations supplémentaires et/ou des comportements non observés dans les 2 vidéos déjà proposées sur Teams. Voici les quelques lignes de code permettant de générer une séquence vidéo de l'exécution d'un programme Matlab :

```
clear variables;
close all;

v=VideoWriter('seq_video.avi');
open(v);

t=-5:0.01:5;
for a=5:-0.02:0.01
    plot(t,exp(-(t.^2)/a),'linewidth',2);
    drawnow;
    thisframe=getframe(gcf);
    writeVideo(v,thisframe);
end

close(v);
```