Analyse Numérique
Equations Différentielles Ordinaires
Mouvement keplérien et double pendule

Axel Bruyere, Dorian Fabregue
4ETI-IMI

28 mars 2021

Objectifs – • Compréhension des EDO d'ordre 1 (Problème de Cauchy)
• Implémentation des méthodes d'Euler et de Runge-Kutta (ordre 2 et 4)
• Application de ces méthodes à l'étude du mouvement képlerien et du double pendule
• Compréhension des défauts de ces méthodes pour chaque étude
• Implémentation de nouveaux algorithmes afin d'améliorer les résultats de l'analyse numérique

1 Contexte

Les équations différentielles ordinaires (EDO) interviennent dans de nombreux domaines de la modélisation mathématique. Nous allons nous intéresser ici en particulier à deux modèles mécaniques particuliers, à savoir :

- Etude du mouvement képlerien (trajectoire d'un satellite orbitant autour d'un astre)
- Etude du mouvement d'un double pendule non-amorti

Ces cas-là sont malheureusement trop complexes pour être résolus par des méthodes analytiques car ces situations sont modélisées par des systèmes d'équations différentielles non linéaires et ne présentent pas de solutions exactes.

Pour résoudre ces systèmes d'équations, nous allons mettre en place différents modèles de résolution, à savoir la méthode d'Euler et la méthode de Runge-Kutta (ordres 2 et 4). Nous analyserons les caractéristiques de chacun de ces modèles et mettrons en place, si possible, un nouveau modèle de résolution plus performant.

Table des matières

2 Etude du mouvement képlerien

2.1 Introduction

En astronomie, plus précisément en mécanique céleste, le mouvement képlérien correspond à une description du mouvement d'un astre par rapport à un autre respectant les trois lois de Kepler (1571-1630). La première de ces trois lois, la "loi des orbites", énonce que les objets célestes gravitant autour d'une étoile décrivent des trjactoires elliptiques. C'est ce type de mouvement que nous allons essayer de modéliser dans cette partie.

Dans notre cas, nous supposons l'orbite elliptique avec une forte excentricité. 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 ce cas, le mouvement de la planète $P = (x(t), y(t))$ est modélisé par le système d'équations différentielles suivant :

$$\begin{cases} x''(t) + \mathscr{G}M\dfrac{x(t)}{[x(t)^2 + y(t)^2]^{3/2}} = 0 \\ y''(t) + \mathscr{G}M\dfrac{y(t)}{[y(t)^2 + y(t)^2]^{3/2}} = 0 \end{cases} \tag{1}$$

où $\mathscr{G}$. est la constante de gravitation universelle, elle apparait dans la troisième loi de Kepler :

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

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.

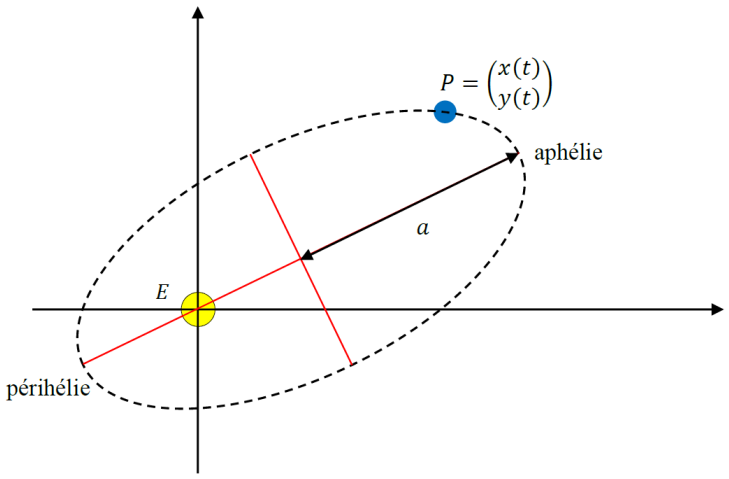

FIGURE 1 – Orbite elliptique (avec forte excentricité) d'une planète P autour d'une étoile E

Afin de résoudre ce système d'équations différentielles, nous allons mettre en place un programme utlisant différentes méthodes d'analyse numériques et essayer d'obtenir un résultat satisfaisant, à savoir une ellipse constante comme sur la figure ci-dessus.

2.2 Méthode d'Euler

2.2.1 Problème de Cauchy

Peu importe l'ordre de notre système d'EDO, il peut être ramener à un système d'EDO d'ordre 1. De façon générale, les EDO admettent une infinité de solutions. Ce n'est que lorsqu'on impose une condition initiale $z(t_0) = z_0$ (comme c'est le cas ici avec $(x(t_0), y(t_0)) = (0.5, 0)$) que l'on détermine l'une des solutions de l'EDO. On considèrera alors la solution du problème appelé « problème de Cauchy » qui consiste à déterminer la fonction $z = (x, y) : I \subset \mathbb{R}^2 \to \mathbb{R}^2$ telle que :

$$\begin{cases} z'(t) = u(t, z(t)), \quad \forall t \in I \\ z(t_0) = z_0 \end{cases} \tag{3}$$

Afin de vérifier l'unicité de la solution, on admettra que $u(t, z)$ vérifie deux conditions :
- elle est continue par rapport à la variable t et continue par rapport à la variable $z = (x, y)$
- elle est k-lipschitzienne par rapport à la variable $z = (x, y)$

Les 3 méthodes analytiques que nous allons implémenter pour l'étude du mouvement keplérien seront : la méthode d'Euler, la méthode de Runge-Kutta 4 et la méthode d'Euler-Richardson. Celle-ci devront être adaptées à notre système d'EDO d'orde 2 à deux dimensions.

2.2.2 Méthode d'Euler explicite

On calcule la solution approchée (x_{n+1}, y_{n+1}) au point t_{n+1} à partir de la solution approchée y_n au point t_n. La discrétisation du problème de Cauchy à deux dimensions s'écrit :

$$\begin{cases} f(t_n, x(t_n), y(t_n)) = \dfrac{x(t_{n+1}) - x(t_n)}{t_n + 1 - t_n} (\approx x'(t_n)) \\ g(t_n, x(t_n), y(t_n)) = \dfrac{y(t_{n+1}) - y(t_n)}{t_n + 1 - t_n} (\approx y'(t_n)) \end{cases} \tag{4}$$

ou encore, en approchant $(x(t_n), y(t_n))$ par (x_n, y_n), et en introduisant les notation $f_n = f(t_n, x(t_n), y(t_n))$ et $g_n = g(t_n, x(t_n), y(t_n))$, nous obtenons le schéma explicite de la méthode d'Euler :

$$\begin{cases} x_{n+1} = x_n + h f_n, \quad n = 0, 1, ...N_h - 1 \\ y_{n+1} = y_n + h g_n \end{cases} \tag{5}$$

avec h le résultat de la discrétisation de l'intervalle $I = [a; b]$ divisé en N_h intervalles d'amplitude h suffisemment faible (ici $h = 0.01ua$) par rapport au mouvement étudié.

Sachant que nous avons ici une EDO du second ordre, on introduit de nouvelles variables $(v_x(t_n), v_y(t_n))$ telles que $\forall n \in [0, 1, ..., N_h - 1]$:

$$\begin{cases} v_x(t_n) = x'(t_n) \\ v_y(t_n) = y'(t_n) \\ v_x(0) = x'(0) = 0 \\ v_y(0) = y'(0) = 11.5 \end{cases} \tag{6}$$

Cela nous permet, en résolvant deux EDO d'ordre 2 à chaque itération, d'écrire le programme suivant :

```
function [x,y,t]=fct_Euler_Kepler(x0,y0,vx0,vy0,tmin,tmax,h,f,g)

%%Initialisation
t = tmin:h:tmax;
x = zeros(1,length(t)); x(1) = x0;
y = zeros(1,length(t)); y(1) = y0;
vx = zeros(1,length(t)); vx(1) = vx0;
vy = zeros(1,length(t)); vy(1) = vy0;

%%Calcul des valeurs n+1
for k = 1:length(t)-1
    x(k+1) = x(k) + h*vx(k);
    vx(k+1) = vx(k) + h*f(t(k),x(k),y(k));
    y(k+1) = y(k) + h*vy(k);
    vy(k+1) = vy(k) + h*g(t(k),x(k),y(k));
end
```

2.2.3 Méthode de Runge-Kutta 4 (en 2D)

Les méthodes de RK sont des méthodes numériques à un pas comme la méthode d'Euler dont elles sont une généralisation en considérant cette fois plusieurs évaluations des fonctions (f, g) par intervalle $[t_n; t_{n+1}]$. Nous avons pour RK4 les équations suivantes :

$$\begin{cases} x_{n+1} = x_n + \dfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad n = 0, 1, ...N_h - 1 \\ k_1 = f(t_n, x_n) \\ k_2 = f(t_n + \dfrac{h}{2}, x_n + \dfrac{h}{2}k_1) \\ k_3 = f(t_n + \dfrac{h}{2}, x_n + \dfrac{h}{2}k_2) \\ k_4 == f(t_n, x_n + hk_3) \end{cases} \tag{7}$$

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad n = 0, 1, ...N_h - 1 \\ k_1 = f(t_n, y_n) \\ k_2 = f(t_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}k_1) \\ k_3 = f(t_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}k_2) \\ k_4 == f(t_n, y_n + hk_3) \end{cases} \tag{8}$$

Sachant que nous avons ici une EDO du second ordre, comme pour la méthode d'Euler, on introduit de nouvelles variables $(v_x(t_n), v_y(t_n))$.Cela nous permet, en résolvant deux EDO d'ordre 2 à chaque itération, d'écrire le programme suivant :

```
function [x,y,t]=fct_RK4_2D_Kepler(x0,y0,vx0,vy0,tmin,tmax,h,f,g)

%%Initialisation
t = tmin:h:tmax;
x = zeros(1,length(t)); x(1) = x0;
y = zeros(1,length(t)); y(1) = y0;
vx = zeros(1,length(t)); vx(1) = vx0;
vy = zeros(1,length(t)); vy(1) = vy0;
```

```
%%Algo keplerien
for k = 1:length(t)-1
    %%k1
    kf1 = f(t(k),x(k),y(k));
    kg1 = g(t(k),x(k),y(k));
    %%k2
    kf2 = f(t(k) + h/2, x(k) + (h/2)*kf1, y(k) + h/2*kg1);
    kg2 = g(t(k) + h/2, x(k) + (h/2)*kf1, y(k) + h/2*kg1);
    %%k3
    kf3 = f(t(k) + h/2, x(k) + (h/2)*kf2, y(k) + h/2*kg2);
    kg3 = g(t(k) + h/2, x(k) + (h/2)*kf2, y(k) + h/2*kg2);
    %%k4
    kf4 = f(t(k) + h, x(k) + h*kf3, y(k) + h*kg3);
    kg4 = g(t(k) + h, x(k) + h*kf3, y(k) + h*kg3);
    %%Calcul des nouvelles valeures n+1
    x(k+1) = x(k) + h*vx(k);
    vx(k+1) = vx(k) + (h/6) * (kf1 + 2*kf2 + 2*kf3 + kf4);
    y(k+1) = y(k) + h*vy(k);
    vy(k+1) = vy(k) + (h/6) * (kg1 + 2*kg2 + 2*kg3 + kg4);

end
```

L'intérêt principal de cette méthode est d'avoir un pas plus flexible que le pas constant de la méthode d'Euler, on se rendra vite compte que ça ne sera pas suffisant pour obtenir une ellipse non-divergente.

2.2.4 Méthode d'Euler-Richardson

Nous prenons ici comme base la méthode d'Euler vue précédemment. Nous y introduisons cependant un pas temporel h adaptatif afin de garantir la qualité de l'analyse numérique. Résumons le principe de l'algorithme (pour plus de détails, consulter *femto-physique.com*) :
Soit la méthode d'Euler avec un pas $h = t_{n+1} - t_n$, envisageons maintenant un évolution en deux étapes de pas $h/2$, elles nous permettent d'obtenir le schéma numérique suivant :

$$\begin{cases} k_n = f'(t_n, Y_n) \\ k'_n = f(t_n + h/2) \\ Y_0 = Y(0) \\ Y_{n+1} = Y_n + hk'_n \end{cases} \tag{9}$$

et d'avoir accès à l'erreur $\varepsilon = \dfrac{h}{2}|k'_n - k_n|$, qui nous servira d'indicateur pour savoir si l'on peut augmenter le pas ou s'il faut le diminuer. Il suffit alors de fixer un seuil de précision ε_{seuil} selon lequel nous augmenterons ou diminuerons la valeur du pas h.

L'algorithme d'Euler-Richardson se déroule de la manière suivante :
1 - Initialisation du pas h, de la durée T et du seuil de précision ε_{seuil},
2 - Initialisation des conditions initiales : $t = 0$ et $y = (0)$
3 - Tant que $t \leq T$, faire :

a. Calcul de $k = f(t, Y)$
b. Calcul de $k' = f(t + h/2, Y + kh/2)$
c. Calcul de $\varepsilon = \frac{h}{2}|k - k'|$ et $\alpha = \frac{\varepsilon}{\varepsilon_{seuil}}$
d. Si $\alpha > 1$, faire $h = 0,9h/\sqrt{\alpha}$
d. Si $\alpha < 1$, faire $h = 0,9h/\sqrt{\alpha}; Y = Y + k'h; t = t + h;$

Dans le cas de notre étude de mouvement keplérien, nous avons :

2.3 Résultats

2.3.1 Euler

Nous appelons dans notre *main* (cf. annexe) la fonction *fct_ Euler_ Kepler.m* avec les données de l'énoncé et obtenons la figure suivante :

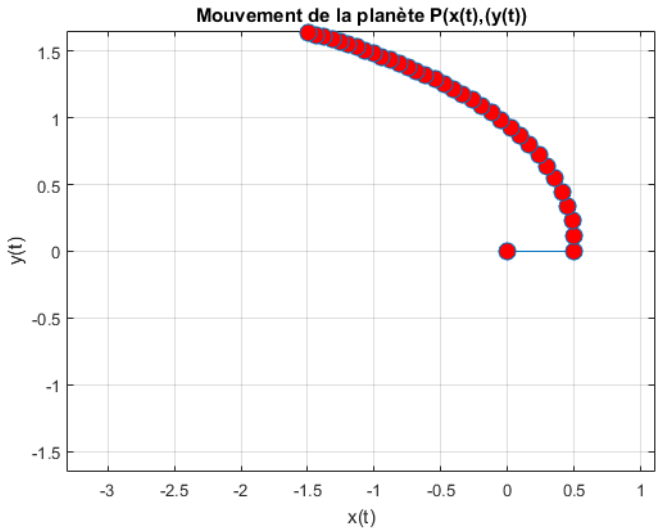

FIGURE 2 – Mouvement de l'astre P avec la méthode d'Euler ($h = 0.01$; durée=$2ua$)

On observe que l'astre dévit de son orbite rapidement et ne le rejoindra plus jamais. On pourrait penser corriger ça en réduisant la vitesse initiale $y'(0)$, en se disant qu'il est normal qu'un astre soit ejecté de son orbite s'il a trop de vitesse. On tente le coup avec $y'(0) = 3$ soit une vitesse 4 fois plus faible et obtenons la figure 3 :

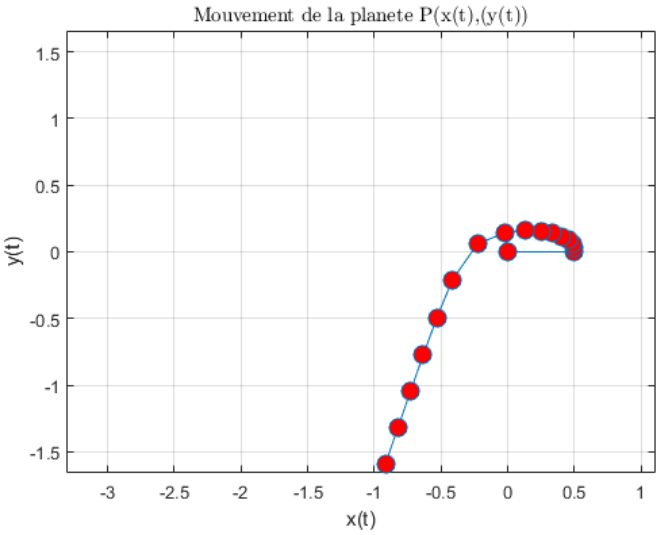

FIGURE 3 – Mouvement de l'astre P avec la méthode d'Euler ($y'(0) = 3$) et ($h = 0.01$; durée=$2ua$)

On voit que cela ne fonctionne pas peu importe la vitesse initialle choisie, on en déduit que ce n'est pas un problème de données erronées. On pourrait modifier la position initiale (sans succès évidemment), mais si l'on commence à trop changer les données de l'enoncé, nous perdons de vue l'objectif de départ.

Une donnée sur laquelle on pourrait influer serait celle du pas temporel h, en se disant que notre analyse est incorrecte car trop peu précise. Nous prenons donc $h = 0.001ua$ (soit une valeur 10 fois plus petite) et obtenons la figure 4 :

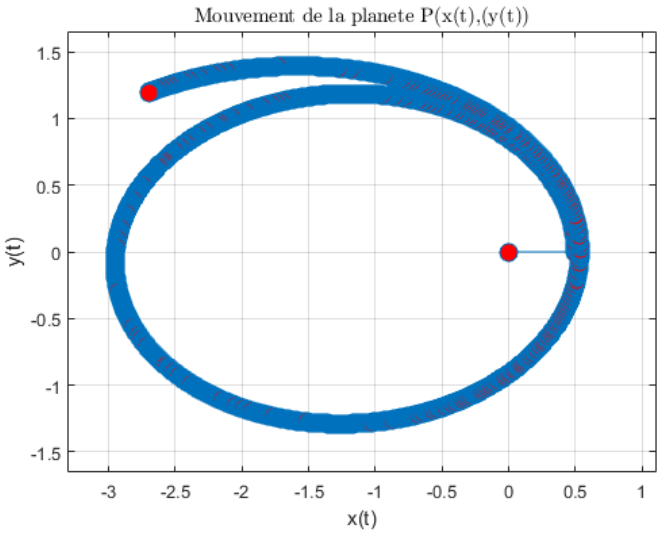

FIGURE 4 – Mouvement de l'astre P avec la méthode d'Euler ($h = 0.001ua$ et durée=$3ua$)

En affichant le mouvement de la planète P sur une durée de $3ua$, nous parvenons à obtenir un tracé plus elliptique mais qui finit par diverger au bout d'un moment.
Après toutes les modifications que nous avons effectuées, nous ne sommes pas parvenus à obtenir le tracé elliptique souhaité. Nous en déduisons que la méthode d'Euler, telle quelle, est trop imprécise.

2.3.2 Runge-Kutta 4

Nous appelons dans notre *main* (cf. annexe) la fonction *fct_ RK4_ 2D_ kepler.m* avec les données de l'énoncé et obtenons la figure 5 suivante :

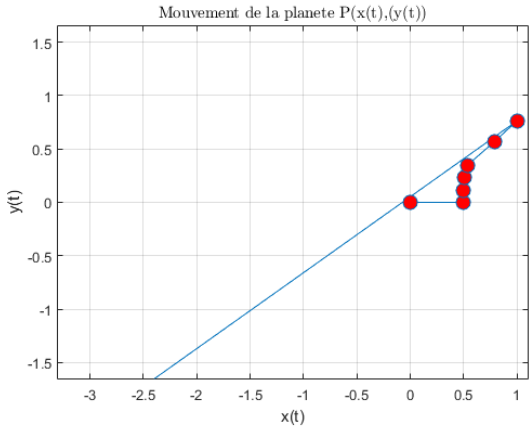

FIGURE 5 – Mouvement de l'astre P avec la méthode de Runge-Kutta 4 ($h = 0.01$; durée=$2ua$)

On remarque de suite l'instabilité de l'analyse proposée par cette méthode. On suppose que cette instabilité est causée par des variations trop grandes de valeurs, et proposons donc de prendre un pas $h = 0.001$ pour tenter de contourner ce problème. Nous obtenons la figure 6 suivante :

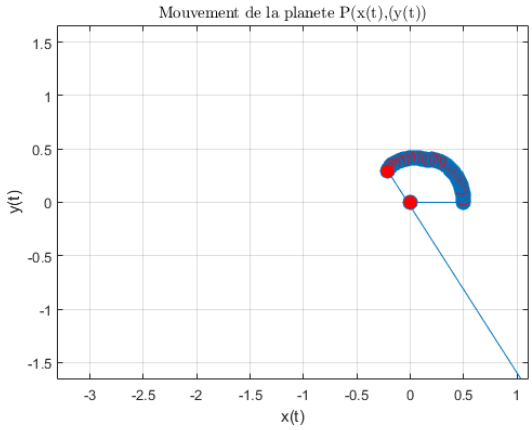

FIGURE 6 – Mouvement de l'astre P avec la méthode de Runge-Kutta 4 ($h = 0.001$; durée=$2ua$)

On remarque que l'analyse reste stable plus longtemps, cependant elle diverge au bout d'un certain temps. Nous en déduisons qu'il ne s'agit pas d'une bonne méthode d'analyse pour ce problème. Cependant on remarque, comme avec la méthode d'Euler, qu'influer sur le pas temporel h permet d'améliorer la qualité de l'analyse. C'est pour cela que nous allons analyser ce problème avec la méthode d'Euler-Richardson dans la prochaine partie.

2.3.3 Euler-Richardson

Nous appelons dans notre *main* (cf. annexe) la fonction *fct_ Euler_ Richardson.m* avec les données de l'énoncé et obtenons la figure 7 suivante :

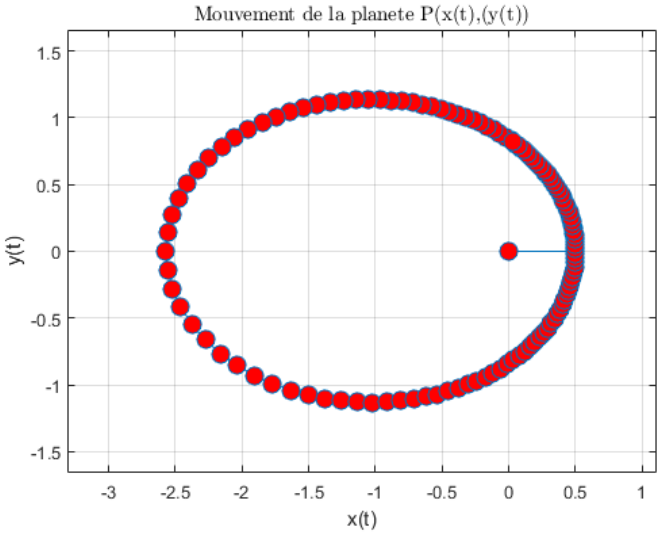

FIGURE 7 – Mouvement de l'astre P avec Euler-Richardson (durée=$2ua$; $\varepsilon_{seuil} = 0.01$)

On remarque de suite la précision de l'analyse et obtenons bien une ellipse comme attendu. On peut par ailleurs approximer la valeur de la période du mouvement elliptique $T_{priode} \approx 2ua$. Par précaution, nous vérifions ce résultat pour une durée=$20ua$ et obtenons la figure 8 suivante :

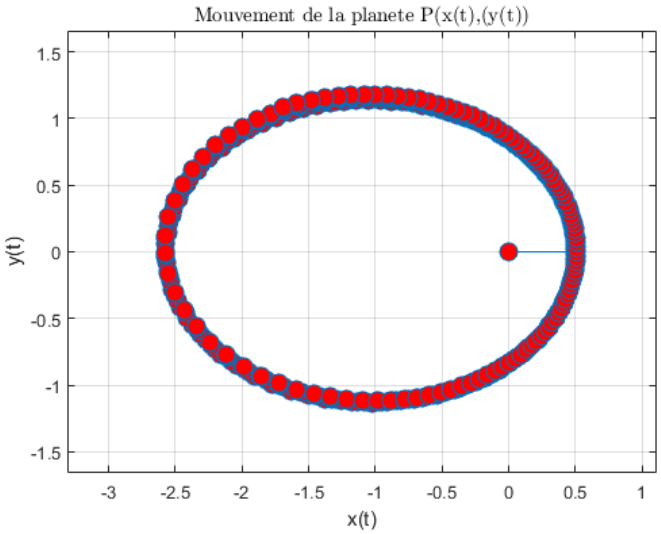

FIGURE 8 – Mouvement de l'astre P avec Euler-Richardson (durée=$20ua$; $\varepsilon_{seuil} = 0.01$)

Même après plus de 10 périodes T_{priode}, nous avons une analyse stable avec une valeur de seuil $\varepsilon_{seuil} = 0.01$.

Un autre point à noter est le nombre de points nécessaires au tracé de l'analyse. Avec la méthode d'Euler, pour avoir un tracé un semblant correct sur $2ua$ nous avions besoin de plus de 10000

points, alors qu'avec cette méthode de pas adaptif (et avec $\varepsilon_{seuil} = 0.01$), nous n'avons besion que de 140 points. On réduit considérablement le temps de d'exécution du programme.

Nous allons maintenant essayer d'afficher le résultat de la figure 7 ($h = 0.01$, durée=$2ua$) avec des seuils d'erreur ε_{seuil} différentes :

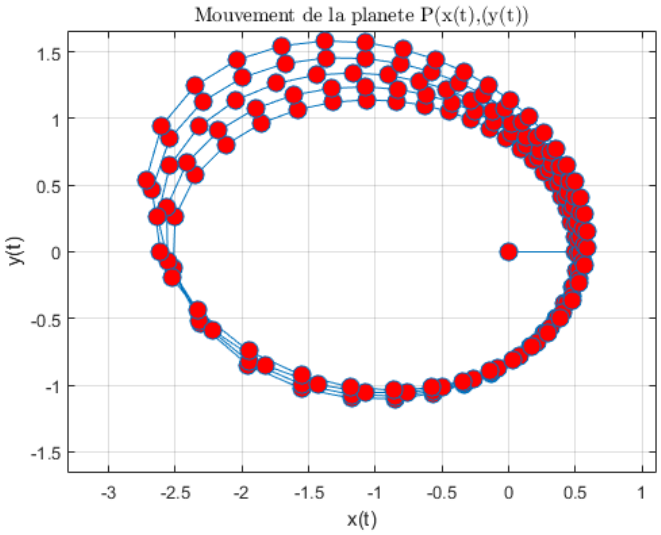

FIGURE 9 – Mouvement de l'astre P avec Euler-Richardson (durée=$20ua$; $\varepsilon_{seuil} = 0.1$)

On remarque de suite la divergence qui se crée si on prend $\varepsilon_{seuil} = 0.1$. Ce qui est un résultat attendu.

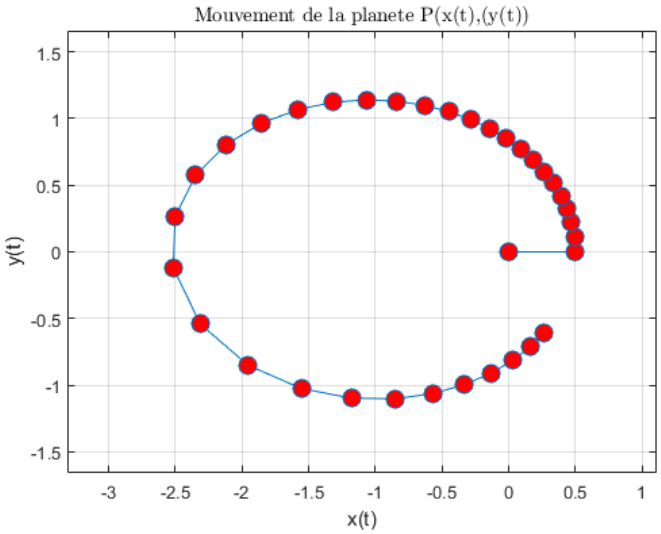

FIGURE 10 – Mouvement de l'astre P avec Euler-Richardson (durée=$2ua$; $\varepsilon_{seuil} = 0.1$)

On remarque aussi qu'avec un seuil d'erreur plus important $\varepsilon_{seuil} = 0.1$, l'astre P semble parcourir moins de distance (même pas une ellipse complète) qu'avec $\varepsilon_{seuil} = 0.01$ (cf. figure 7). Il faut garder en tête que dans cette analyse, le temps est discrétisé et est incrémenté à chaque boucle $t = [t, t(i) + h]$; avec un pas adaptatif $h = 0.9h \times \sqrt{\varepsilon_{seuil}/\varepsilon_2}$ qui augmente plus ε_{seuil} est grand.

Essayons maintenant avec $\varepsilon_{seuil} = 0.001$ sur une durée de $2ua$, nous avons la figure 11 suivante :

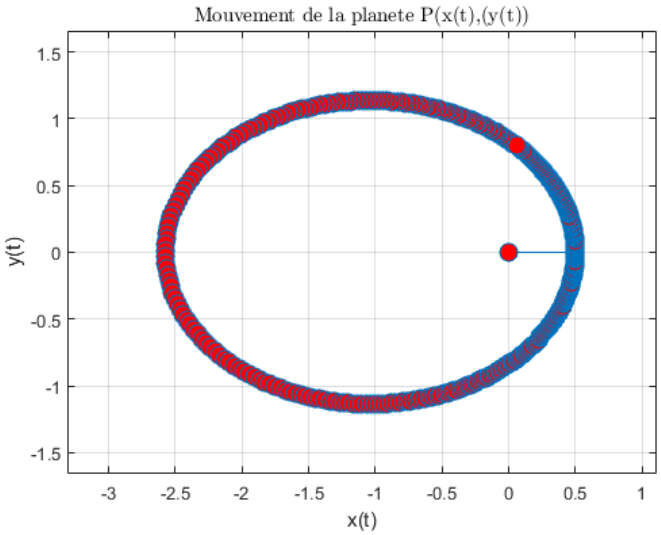

FIGURE 11 – Mouvement de l'astre P avec Euler-Richardson (durée=$2ua$; $\varepsilon_{seuil} = 0.001$)

On retrouve un résultat semblable à la figure 7 ($\varepsilon_{seuil} = 0.01$) avec un nombre de points $n = 438$. On aurait pu s'attendre à un nombre de points plus important si l'on pensait qu'avoir une précision ε_{seuil} 10 fois plus fine impliquerait la création de 10 fois plus de points. On va donc visualiser l'évolution du nombre n de points par rapport à la valeur de ε_{seuil} sur une durée de $2ua$:

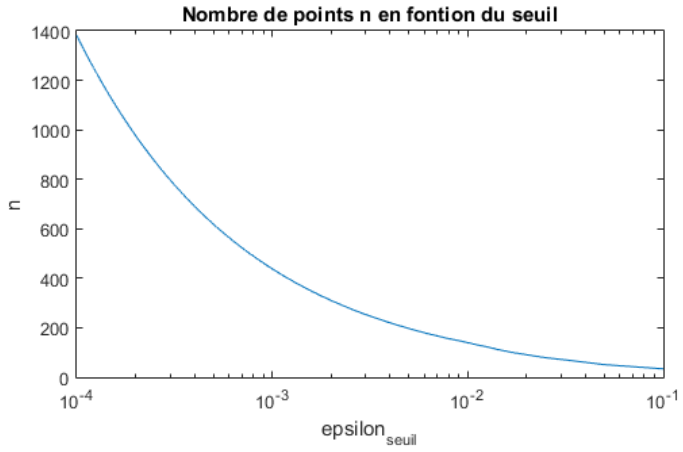

FIGURE 12 – Evolution du nombre de points n en fonction de ε_{seuil} (durée=$2ua$)

On remarque que la méthode d'Euler-Richardson nous permet d'accéder à des précisions très élevées ($\varepsilon_{seuil} = 10^{-4}$) sans pour autant augmenter drastiquement le nombre de points n et donc de calculs à effectuer. On notera cependant que la courbe semble suivre une loi qui ressemble à une loi inverse, et que si l'on souhaite des précisions encore plus fines (10^{-8} par exemple) on aurait alors un nombre de points n qui augmenterait drastiquement et on perdrait l'intérêt de cette méthode.

3 Double pendule non-amorti

3.1 Introduction

More text.

3.2 Méthode

More text.

3.3 Résultats

More text.

4 Annexe

4.1 Mouvement Keplérien - fonction *main*

```
clear variables;
close all;

% Initialisation de l'intervalle temporel (en ua)
tmin=0;tmax=2;

% parametres du modele keplerien
a=1;
T=1;
GM = 4* pi^2 *T^2 /a^3;

% conditions initiales
x0=0.5*a;
y0=0;
vx0=0;
vy0=11.5;

% second membre de l'equ. diff. (x'',y'')=(vx',vy')=(f(t,x,y),g(t,x,y))
f = @(t,x,y)(-GM * x ./ ((x.^2 + y.^2).^1.5));
g = @(t,x,y)(-GM * y ./ ((x.^2 + y.^2).^1.5));

% pas temporel
h=0.01;

%% 1. methode d'Euler
% [X,Y,t] = fct_Euler_Kepler(x0,y0,vx0,vy0,tmin,tmax,h,f,g);

%% 2. methode RK4
% [X,Y,t] = fct_RK4_2D_Kepler(x0,y0,vx0,vy0,tmin,tmax,h,f,g);

% 3. methode Euler-Richardson
N = zeros(1,1000);
nx = zeros(1,1000);
for j=10:10:10000
[X,Y,t]= fct_Euler_Richardson(x0,y0,vx0,vy0,tmin,tmax,h,f,g,1/j);
T_total=t(end);
N(j/10)=length(t);
nx(j/10)= 1/j;
end
semilogx(nx,N);
title('Nombre de points n en fontion du seuil')
xlabel('epsilon_{seuil}')
ylabel('n')
xlim([0.0001 0.1]);
% affichage des resultats
% figure(1);
```

```
47 % xmin=-3*a; xmax=a;
48 % ymin=-1.5*a; ymax=1.5*a;
49 % tic;
50 % for k=1:65:length(t)
51 % x= X * a;
52 % y= Y * a;
53 % plot([0,x],[0,y],'Marker','o','MarkerFacecolor','r','MarkerSize',10);
54 % axis('equal');
55 % axis(1.1*[xmin,xmax,ymin,ymax]);
56 % grid 'on';
57 % t1=title('Mouvement de la planete P(x(t),(y(t))');
58 % xlabel('x(t)')
59 % ylabel('y(t)')
60 % set(t1,'interpreter','latex');
61 % end
62 % cputime=toc;
63 % fprintf('Duree de la simulation numerique : %1.2f\n',cputime);
```