

# Analyse Numérique - TP1 EDO

Adrien Hernandez

Emma Ortiz

Février 2022

## 1 Introduction

Dans cet exercice, on étudie le mouvement d'un pendule simple amorti, problème de physique classique étudié en classes préparatoires. Pour cela, on considère une bille de masse  $m$  suspendue à un fil de longueur  $l$  que l'on éloigne de sa position d'équilibre d'un angle  $\theta_0$  et que l'on lance avec une vitesse angulaire initiale  $\theta'_0$ . La bille est soumise à trois forces (son poids  $P$ , la tension du fil  $T$ , et des forces de frottements de l'air  $f$ ) et après l'application du Principe Fondamental de la Dynamique, on obtient l'équation suivante :

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

Dans le cas où les frottements de l'air ne sont pas négligeables, la fonction  $\theta(t)$  obéit à une équation différentielle non-linéaire d'ordre 2 qui n'admet pas de solution analytique. Le but de cet exercice est donc de résoudre cette équation numériquement par le biais de 3 méthodes vues en cours : la méthode d'Euler explicite et les méthodes de Runge-Kutta d'ordre 2 et 4. De plus, on trace sur une même figure, la trajectoire de la bille dans l'espace réel et dans l'espace des phases et également l'énergie de cette bille au cours du temps pour ensuite commenter ces courbes et en interpréter les résultats.

## 2 Méthode

Pour résoudre l'équation (1), il faut se ramener à un système de deux équations différentielles d'ordre 1 de la façon suivante :

$$\begin{aligned}\theta'(t) &= z(t) \\ z(t) &= -\gamma\theta'(t) - \omega^2 \sin(\theta(t)) = -\gamma z(t) - \omega^2 \sin(\theta(t))\end{aligned}$$

On pose alors les fonctions  $f$  et  $g$  telles que :

$$\begin{aligned}f(t, \theta, z) &= z \\ g(t, \theta, z) &= -\gamma z - \omega^2 \sin(\theta)\end{aligned}$$

Grâce à ces fonctions, on arrive à une solution numérique avec les trois méthodes différentes. Cependant, certaines sont plus précises car l'itération de la suite est différente selon la méthode utilisée.

### 2.1 Méthode d'Euler en dimension 2

On considère la suite suivante :

$$\begin{aligned}\theta_{n+1} &= \theta_n + hf(t_n, \theta_n, z_n) \\ z_{n+1} &= z_n + hg(t_n, \theta_n, z_n)\end{aligned}$$

Cette suite est codée de la manière suivante sous Matlab et porte le nom de *Euler\_2D* :

```

1 function [x,y,t]=Euler_2D(x0,y0,tmin,tmax,pas,F,G)
2     t = tmin:pas:tmax;
3
4     x = zeros(1,length(t));
5     y = zeros(1,length(t));
6     x(1) = x0;
7     y(1) = y0;
8
9     for k = 1:length(t)-1
10         x(k+1) = x(k) + pas*F(t(k),x(k),y(k));
11         y(k+1) = y(k) + pas*G(t(k),x(k),y(k));
12     end
13
14 end

```

## 2.2 Méthode Runge-Kutta d'ordre 2 en dimension 2

On considère cette fois la suite suivante :

$$\theta_{n+1} = \theta_n + h[(1 - \beta)k_{1x} + \beta k_{2x}]$$

$$z_{n+1} = z_n + h[(1 - \beta)k_{1y} + \beta k_{2y}]$$

Avec :

$$k_{1x} = f(t_n, \theta_n, z_n)$$

$$k_{1y} = g(t_n, \theta_n, z_n)$$

$$k_{2x} = f\left(t_n + \frac{h}{2\beta}, \theta_n + \frac{h}{2\beta}k_{1x}, z_n + \frac{h}{2\beta}k_{1y}\right)$$

$$k_{2y} = g\left(t_n + \frac{h}{2\beta}, \theta_n + \frac{h}{2\beta}k_{1x}, z_n + \frac{h}{2\beta}k_{1y}\right)$$

Cette suite est codée de la manière suivante sous Matlab et porte le nom de *RK2\_2D* :

```

1 function [x,y,t]=RK2_2D(x0,y0,tmin,tmax,pas,beta,F,G)
2     t = tmin:pas:tmax;
3     x = zeros(1,length(t));
4     y = zeros(1,length(t));
5     x(1) = x0;
6     y(1) = y0;
7
8     for k = 1:length(t)-1
9         k1x = F(t(k), x(k), y(k));
10        k1y = G(t(k), x(k), y(k));
11        k2x = F(t(k)+pas/(2*beta), x(k)+k1x*pas/(2*beta), y(k)+k1y*pas/(2*beta));
12        k2y = G(t(k)+pas/(2*beta), x(k)+k1x*pas/(2*beta), y(k)+k1y*pas/(2*beta));
13
14        x(k+1) = x(k)+pas*((1-beta)*k1x+beta*k2x);
15        y(k+1) = y(k)+pas*((1-beta)*k1y+beta*k2y);
16
17    end
18 end

```

## 2.3 Méthode de Runge-Kutta d'ordre 4 en dimension 2

Enfin, on considère cette dernière suite :

$$\theta_{n+1} = \theta_n + \frac{h}{6}(k_{1x} + 2k_{2x} + 2k_{3x} + k_{4x})$$

$$z_{n+1} = z_n + \frac{h}{6}(k_{1y} + 2k_{2y} + 2k_{3y} + k_{4y})$$

Avec :

$$\begin{aligned}
 k_{1x} &= f(t_n, \theta_n, z_n) \\
 k_{1y} &= g(t_n, \theta_n, z_n) \\
 k_{2x} &= f\left(t_n + \frac{h}{2}, \theta_n + \frac{h}{2}k_{1x}, z_n + \frac{h}{2}k_{1y}\right) \\
 k_{2y} &= g\left(t_n + \frac{h}{2}, \theta_n + \frac{h}{2}k_{1x}, z_n + \frac{h}{2}k_{1y}\right) \\
 k_{3x} &= f\left(t_n + \frac{h}{2}, \theta_n + \frac{h}{2}k_{2x}, z_n + \frac{h}{2}k_{2y}\right) \\
 k_{3y} &= g\left(t_n + \frac{h}{2}, \theta_n + \frac{h}{2}k_{2x}, z_n + \frac{h}{2}k_{2y}\right) \\
 k_{4x} &= f(t_n + h, \theta_n + hk_{3x}, z_n + hk_{3y}) \\
 k_{4y} &= g(t_n + h, \theta_n + hk_{3x}, z_n + hk_{3y})
 \end{aligned}$$

Cette suite est codée de la manière suivante sous Matlab et porte le nom de *RK4\_2D* :

```

1 function [x,y,t]=RK4_2D(x0,y0,tmin,tmax,pas,F,G)
2     t = tmin:pas:tmax;
3
4     x = zeros(1,length(t));
5     y = zeros(1,length(t));
6     x(1) = x0;
7     y(1) = y0;
8
9     for k = 1:length(t)-1
10         k1x = F(t(k), x(k), y(k));
11         k1y = G(t(k), x(k), y(k));
12
13         k2x = F(t(k)+pas/2, x(k)+k1x*pas/2, y(k)+k1y*pas/2);
14         k2y = G(t(k)+pas/2, x(k)+k1x*pas/2, y(k)+k1y*pas/2);
15
16         k3x = F(t(k)+pas/2, x(k)+k2x*pas/2, y(k)+k2y*pas/2);
17         k3y = G(t(k)+pas/2, x(k)+k2x*pas/2, y(k)+k2y*pas/2);
18
19         k4x = F(t(k)+pas, x(k)+k3x*pas, y(k)+k3y*pas);
20         k4y = G(t(k)+pas, x(k)+k3x*pas, y(k)+k3y*pas);
21
22         x(k+1) = x(k) + (pas/6)*(k1x+2*k2x+2*k3x+k4x);
23         y(k+1) = y(k) + (pas/6)*(k1y+2*k2y+2*k3y+k4y);
24     end
25
26 end

```

C'est avec l'aide de ces formules que l'on peut calculer  $(t, z)$  de trois façons différentes en appelant les différentes fonctions. A partir de là, on peut visualiser le mouvement du pendule dans le réel et dans le domaine des phases ainsi que l'évolution de l'énergie de la bille.

Code Matlab pour calculer les différentes énergies :

```

1 %energies
2 E_c = 0.5*m*(l)^2*(z).^2;
3 E_p = m*gr*l*(1-cos(theta));
4 E_tot = E_c + E_p;

```

### 3 Description et interprétation des résultats

#### 3.1 Méthode d'Euler

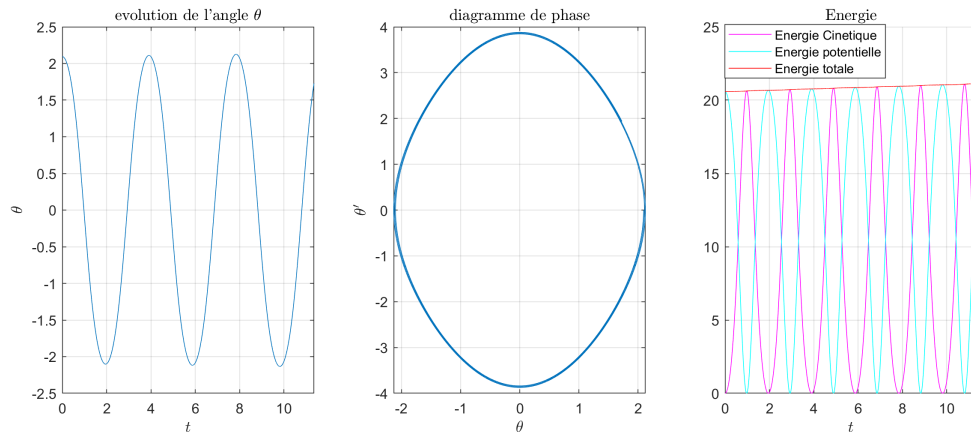


FIGURE 1 – Résultat de la simulation avec la méthode de Runge-Kutta

On peut voir qu'avec la méthode d'Euler, la simulation donne un résultat qui n'est pas physique. En effet, on peut constater que l'énergie totale du système augmente au cours du temps. Cette erreur est liée à l'utilisation d'une méthode numérique. Une résolution numérique par définition fait des approximations. C'est le cumul de celles-ci qui fait que notre modélisation renvoi une solution divergente. C'est pour cela que nous sommes contraint de trouver une méthode numérique avec une erreur plus faible. Nous nous tournons donc vers la méthode de Runge-Kutta.

#### 3.2 Méthode de Runge-Kutta

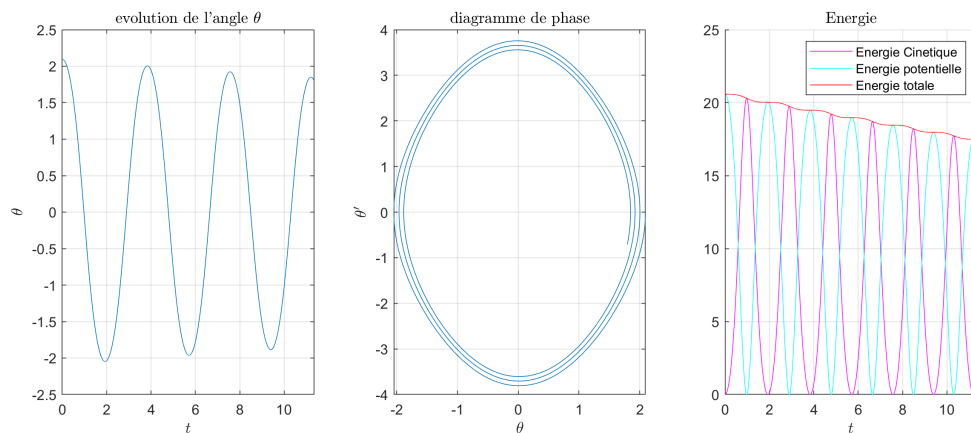


FIGURE 2 – Résultat de la simulation avec la méthode de Runge-Kutta

Avec la méthode de Runge-Kutta, on peut voir que le résultat obtenu semble coïncider avec la physique du problème. En effet on peut voir que l'énergie totale du pendule diminue ce qui est lié aux frottements avec l'air. Cette méthode de résolution permet également d'atténuer les erreurs numériques de la méthode d'Euler. En effet l'erreur de la méthode d'Euler est de  $\varepsilon_n \leq Kh$  alors que la méthode de Runge-Kutta comporte une erreur de  $\varepsilon_n \leq Kh^2$ . L'ordre supérieur de la méthode de Runge-Kutta permet donc d'avoir une résolution numérique plus adéquate pour le problème du pendule simple.

## 4 Conclusion

Tout au long de ce TP, nous avons travaillé sur le problème du pendule simple. Nous avons pour problème l'inexistence d'une solution analytique pour ce problème. C'est pour cela que nous avons mis en oeuvre des méthodes de résolution numérique. Ces méthodes sont la méthode d'Euler explicite et les méthodes de Runge-Kutta d'ordre 2 et 4.

Nous avons pu mettre en évidence les limites des méthodes numérique. En effet ces méthodes nécessitent des approximations et donc cela entraîne des erreurs. Nous avons pu mettre en évidence également que la méthode de Runge-Kutta possède une erreur plus faible que la méthode d'Euler.