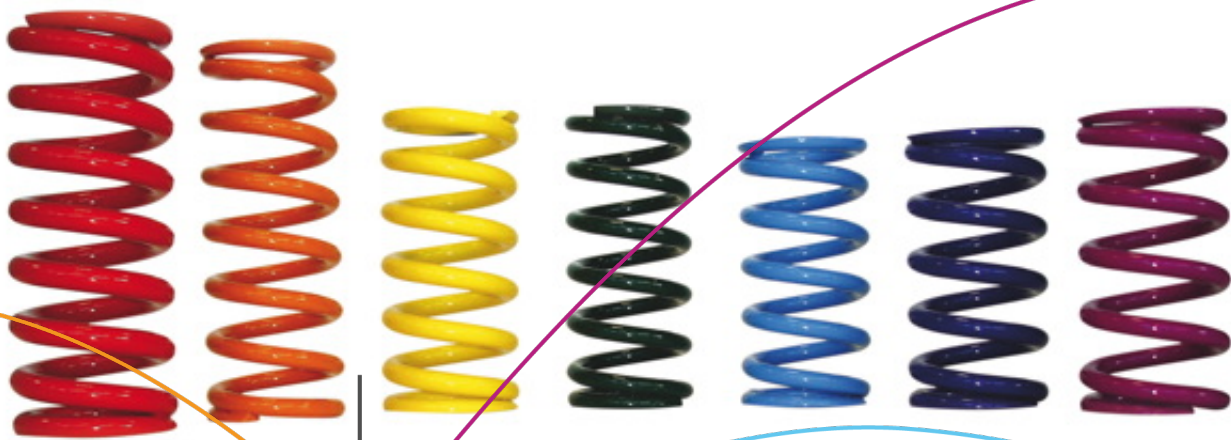


## Oscillateur Masses Ressorts



Enseignant responsable :  
Bernard Gleyse

Étudiants :  
BIZZOTTO Stéphane  
RIDEL Morgan

TERRIEN Loan  
BENKIRANE Amine

**Date de remise du rapport :** 12/06/2016

**Référence du projet :** STPI<sup>1</sup>/P6/2015 – 2016

**Intitulé du projet :** Projet n°13 : Oscillateur masses-ressorts

**Type de projet :** *modélisation mathématique*

**Objectifs du projet :** *L'objectif du projet est de modéliser le mouvement des masses d'une oscillateur harmonique avec une à deux masses mobiles en deux dimensions. On cherche donc à obtenir les équations de mouvement des masses mobiles d'un système donné puis de modéliser leur mouvement numériquement.*

**Remerciements :** *Nos remerciements s'adressent à M. Gleyse pour son encadrement durant toute la durée du projet.*

---

1. INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE ROUEN  
DÉPARTEMENT SCIENCES ET TECHNIQUES POUR L'INGÉNIEUR  
685 AVENUE DE L'UNIVERSITÉ BP 08- 76801 SAINT-ETIENNE-DU-ROUVRAY  
TÉL : +33 2 32 95 66 21 - FAX : +33 2 32 95 66 31

# Table des matières

<b>Introduction</b>	<b>4</b>
<b>1 Méthodologie, organisation du travail</b>	<b>5</b>
<b>2 Système à deux masses et trois ressorts</b>	<b>6</b>
2.1 Mise en équation du système . . . . .	6
2.2 Résolution : . . . . .	7
<b>3 Système à une masse et 2 ressorts en bidimensionnel :</b>	<b>11</b>
3.1 Résolution dans $\mathbb{R}^2$ : . . . . .	11
3.2 Résolution complexe : ( Identification de $\mathbb{R}^2$ par plan complexe ) $\mathbb{R}^2 \rightarrow \mathbb{C}$ : . . . . .	12
3.3 Représentation graphique . . . . .	12
<b>4 Système à 3 masses et 3 ressorts en Triangle :</b>	<b>14</b>
<b>5 Système à une masse et quatre ressorts</b>	<b>20</b>
<b>6 Simulation</b>	<b>25</b>
6.1 Présentation du programme . . . . .	25
6.2 Explication de l'algorithme de Runge-Kutta 4 . . . . .	26
<b>7 Conclusion</b>	<b>28</b>
<b>8 Bibliographie</b>	<b>29</b>
<b>9 Annexe</b>	<b>30</b>
9.1 Programme Pascal pour une masse . . . . .	30

# Introduction

Ce projet a été réalisé dans le cadre de l'EC P6, qui consiste à travailler en groupe sur un sujet parmi tous ceux proposés. Notre groupe est tombé sur l'oscillateur masses-ressorts. Contrairement à l'étude de l'année dernière, notre étude devait prendre en compte un mouvement en deux dimensions. Nous avons pu travailler sur des systèmes similaires dans le cadre de la mécanique du point étudiée en première année mais ce projet a approfondi le sujet et nous a demandé de faire des recherches et des calculs pendant ces derniers mois. Ces calculs nous amèneront à des équations de mouvement que l'on utilisera pour modéliser graphiquement le mouvement des masses selon les raideurs de ressorts données et les conditions initiales.

Le travail est donc réparti en deux parties majeures. L'une consiste en la détermination des équations à l'aide des outils mathématiques que nous avons assimilés depuis l'année dernière. L'autre consiste en une modélisation s'appuyant sur le travail précédent, à l'aide d'un programme codé en Pascal utilisant la méthode de Runge-Kutta.

Tous les ans, ce projet reprends les idées du travail de l'année précédentes, en approfondissant celui-ci. Notre changement représente le fait d'ajouter une autre dimension dans le mouvement, permettant d'analyser le mouvement et de voir clairement les résultats des modélisations sous forme de graphiques.

Les parties de résolution et de simulation ont été séparées entre les quatre membres de notre groupe et nous ont permis de travailler simultanément et en complément les uns des autres.

# Partie 1

## Méthodologie, organisation du travail

Afin de pouvoir accomplir le travail des deux différentes parties efficacement, on nous a demandé de travailler en petits groupes sur chaque partie. De plus il a fallu choisir un correspondant permettant la communication entre le professeur et le groupe. Ce rôle a été rempli par Morgan. Le travail a été réparti en binôme de cette façon :

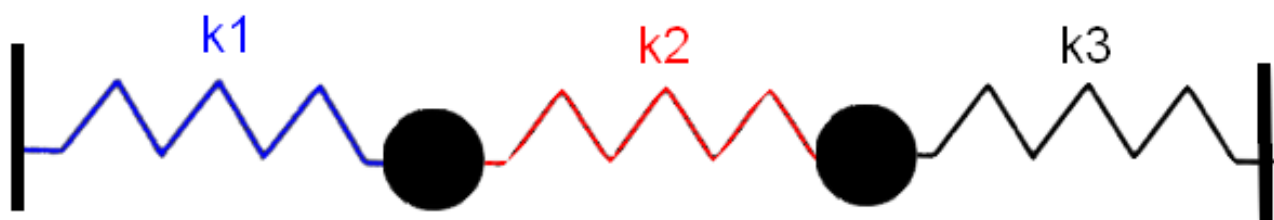
- BIZZOTTO Stéphane et BENKIRANE Amine ont travaillé sur la résolution mathématique des système mécaniques par le biais de l'analyse différentielle.
- TERRIEN Loan et RIDEL Morgan ont réalisé la simulation numérique, utilisant un programme Pascal qui calcule et affiche la trajectoires des masses dans les systèmes.

Notre organisation durant les heures de travail en classe était plutôt simple, pendant que le groupe gérant la partie mathématique travaillait sur les systèmes d'équations différentielles, l'autre groupe s'attelait à la réalisation du programme et à travailler sur la méthode de Runge-Kutta. Une fois le projet un peu avancé, nos deux binômes ont pu compléter leur travail. En effet, à partir d'un certain stade, nous avons pu commencer à utiliser le programme Pascal d'un binôme afin de visualiser les trajectoires des masses à partir des equations trouvées par l'autre binôme. Ainsi nous avons un vrai aperçu visuel de notre travail et nous avons pu également vérifier la cohérence des résultats.

En dehors des heures dédiées en projets, un certain travail personnel nous a permis d'apporter de nouvelles idées et d'avancer dans nos raisonnements. Pour cela, nous avons pris contact par le biais d'Internet, pour pouvoir échanger des informations entre les différentes séances. Quant à la rédaction du rapport, chaque partie de celui-ci a été rédigé par le binôme ayant travaillé sur la partie concerné afin d'avoir une rédaction précise.

## Partie 2

### Systeme à deux masses et trois ressorts



#### 2.1 Mise en équation du système

Soit  $k_{14}$ ,  $k_{12}$  et  $k_{23}$  les constantes de raideurs des ressorts.

On applique le principe fondamental de la dynamique (PFD) au système :

$$m_1 \cdot \ddot{z}_1 = k_{14}(z_4 - z_1) + k_{12}(z_2 - z_1)$$

$$m_2 \cdot \ddot{z}_2 = k_{12}(z_1 - z_2) + k_{23}(z_3 - z_2)$$

Soit :

On a :

$$m_1 \cdot \ddot{z}_1 = k_{14}(z_4 - z_1) + k_{12}(z_2 - z_1)$$

$$m_2 \cdot \ddot{z}_2 = k_{12}(z_1 - z_2) + k_{23}(z_3 - z_2)$$

On sait que,

$$z_j = \begin{pmatrix} x_j \\ y_j \end{pmatrix}$$

On pose le changement de variable suivant :

$$w_j = x_j + iy_j$$

On réécrit sous forme matricielle :

$$\begin{pmatrix} m_1.\ddot{w}_1 \\ m_2.\ddot{w}_2 \end{pmatrix} = \begin{pmatrix} -k_{14} - k_{12} & k_{12} \\ k_{12} & -k_{12} - k_{23} \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + \begin{pmatrix} k_{14}.w_4 \\ k_{23}.w_3 \end{pmatrix} \quad (2.1)$$

On pose  $m_1 = m_2$  et  $k_{12} = k_{23} = k_{14}$

On obtient donc :

$$m. \begin{pmatrix} \ddot{w}_1 \\ \ddot{w}_2 \end{pmatrix} = -k. \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + k. \begin{pmatrix} w_4 \\ w_3 \end{pmatrix} \quad (2.2)$$

On pose :

$$K = k. \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \quad (2.3)$$

Donc

$$m. \begin{pmatrix} \ddot{w}_1 \\ \ddot{w}_2 \end{pmatrix} = -K. \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + k. \begin{pmatrix} w_4 \\ w_3 \end{pmatrix} \quad (2.4)$$

Soit  $w$  tel que  $p = w - weq$ ;

On cherche un  $Weq$  pour effectuer une translation tel que :  $W = Weq$

On a donc :

$$m. \begin{pmatrix} \ddot{w}_1 \\ \ddot{w}_2 \end{pmatrix} = -K. \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \quad (2.5)$$

## 2.2 Résolution :

Valeurs propres :

$$\det(M - \lambda Id) = -\frac{k}{m} \begin{vmatrix} 2 - \lambda & -1 \\ -1 & 2 - \lambda \end{vmatrix} = -\frac{k}{m} [(2 - \lambda)^2 - 1] \quad (2.6)$$

Pour  $-\frac{k}{m} [(2 - \lambda)^2 - 1] = 0$ , on trouve  $\lambda_1 = 3$  ou  $\lambda_2 = 1$ .

Matrice de passage :

Pour  $\lambda_1 = 3$  :

$$\begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3x \\ 3y \end{pmatrix} \quad (2.7)$$

On peut donc écrire le système d'équation :

$$\begin{cases} 2x - y = 3x \\ -x + 2y = 3y \end{cases} \iff \begin{cases} -y = x \\ -x = y \end{cases}$$

On obtient un vecteur propre :  $u_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

De la même manière, pour  $\lambda = 1$  on trouve un vecteur propre  $u_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

On peut dire que la matrice de passage P est égale à :  $P = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$

et que la matrice diagonale D est égale à :  $D = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}$

Matrice M' :

On sait que  $M' = P^{-1}MP$

et on a  $P^{-1} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$

On trouve :

$$M' = \frac{-k}{m} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \quad (2.8)$$

On reprend le système précédent en remplaçant M par M' :

$$\begin{pmatrix} \ddot{u}_1(t) \\ \ddot{u}_2(t) \end{pmatrix} = M' \cdot \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} \quad (2.9)$$

$$\iff \begin{pmatrix} \ddot{u}_1(t) \\ \ddot{u}_2(t) \end{pmatrix} = \frac{-k}{m} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} \quad (2.10)$$

Posons  $C = \frac{k}{m}$

On obtient le système d'équations différentielles de second ordre avec second membre :

$$\begin{cases} \ddot{u}_1(t) = -3Cu_1(t) \\ \ddot{u}_2(t) = -Cu_2(t) \end{cases}$$



Pour  $\ddot{u}_1(t) = -3Cu_1(t)$ , on obtient  $u_1(t) = \lambda \cos(\sqrt{3C}t) + \mu \sin(-\sqrt{3C}t)$

Pour  $\ddot{u}_2(t) = -Cu_2(t)$ , on obtient  $u_2(t) = \lambda' \cos(\sqrt{C}t) + \mu' \sin(-\sqrt{C}t)$

On utilise maintenant la matrice de passage pour repasser dans la base initiale :

$$Pu = \underline{w}$$

avec :

$$u = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} \lambda \cos(\sqrt{3C}t) + \mu \sin(-\sqrt{3C}t) \\ \lambda' \cos(\sqrt{C}t) + \mu' \sin(-\sqrt{C}t) \end{pmatrix} \quad (2.11)$$

$$\underline{w} = \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} \quad (2.12)$$

$$P = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \quad (2.13)$$

On obtient donc :

$$\underline{w} = \begin{pmatrix} \lambda \cos(\sqrt{3C}t) + \mu \sin(-\sqrt{3C}t) + \lambda' \cos(\sqrt{C}t) + \mu' \sin(-\sqrt{C}t) \\ -\lambda \cos(\sqrt{3C}t) - \mu \sin(-\sqrt{3C}t) + \lambda' \cos(\sqrt{C}t) + \mu' \sin(-\sqrt{C}t) \end{pmatrix} \quad (2.14)$$

Pour trouver les constantes, on utilise les conditions initiales. On sait que à  $t = 0$  :

$$\underline{w}_1(0) = x_1(0) + iy_1(0) , \quad \dot{\underline{w}}_1(0) = \dot{x}_1(0) + i\dot{y}_1(0)$$

$$\underline{w}_2(0) = x_2(0) + iy_2(0) , \quad \dot{\underline{w}}_2(0) = \dot{x}_2(0) + i\dot{y}_2(0)$$

.

On dérive  $\underline{w}$  :

$$\dot{\underline{w}} = \begin{pmatrix} -\lambda\sqrt{3C} \sin(\sqrt{3C}t) - \mu\sqrt{3C} \cos(-\sqrt{3C}t) - \lambda'\sqrt{C} \sin(\sqrt{C}t) + \mu'\sqrt{C} \cos(-\sqrt{C}t) \\ \lambda\sqrt{3C} \sin(\sqrt{3C}t) + \mu\sqrt{3C} \cos(-\sqrt{3C}t) - \lambda'\sqrt{C} \sin(\sqrt{C}t) - \mu'\sqrt{C} \cos(-\sqrt{C}t) \end{pmatrix} \quad (2.15)$$

Donc :

$$\begin{cases} \underline{w}_1(0) = x_1(0) + iy_1(0) = \lambda + \lambda' \\ \dot{\underline{w}}_1(0) = \dot{x}_1(0) + i\dot{y}_1(0) = -\mu\sqrt{3C} - \mu'\sqrt{C} \end{cases}$$

$$\begin{cases} \underline{w}_2(0) = x_2(0) + iy_2(0) = -\lambda + \lambda' \\ \dot{\underline{w}}_2(0) = \dot{x}_2(0) + i\dot{y}_2(0) = \mu\sqrt{3C} - \mu'\sqrt{C} \end{cases}$$

$$\begin{cases} \underline{w}_1(0) = x_1(0) + iy_1(0) = \lambda + \lambda' = A \\ \dot{\underline{w}}_1(0) = \dot{x}_1(0) + i\dot{y}_1(0) = -\mu\sqrt{3C} - \mu'\sqrt{C} = B \\ \underline{w}_2(0) = x_2(0) + iy_2(0) = -\lambda + \lambda' = A' \\ \dot{\underline{w}}_2(0) = \dot{x}_2(0) + i\dot{y}_2(0) = \mu\sqrt{3C} - \mu'\sqrt{C} = B' \end{cases}$$

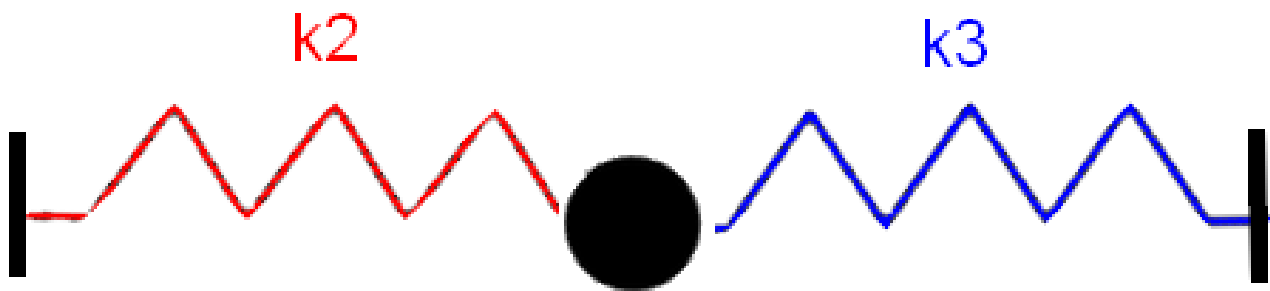
$$\begin{cases} \lambda = \frac{A-A'}{2} \\ \lambda' = \frac{A+A'}{2} \\ \mu = \frac{-B'-B}{2\sqrt{3C}} \\ \mu' = \frac{B'-B}{2\sqrt{C}} \end{cases}$$

Remplaçons dans l'équation de  $\underline{W}$  :

$$\underline{W} = \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} = \begin{pmatrix} \frac{A-A'}{2} \cos(\sqrt{3C}t) + \frac{-B'-B}{2\sqrt{3C}} \sin(\sqrt{3C}t) + \frac{A+A'}{2} \cos(\sqrt{C}t) + \frac{B'-B}{2\sqrt{C}} \sin(\sqrt{C}t) \\ -\frac{A-A'}{2} \cos(\sqrt{3C}t) - \frac{-B'-B}{2\sqrt{3C}} \sin(\sqrt{3C}t) + \frac{A+A'}{2} \cos(\sqrt{C}t) + \frac{B'-B}{2\sqrt{C}} \sin(\sqrt{C}t) \end{pmatrix} \quad (2.16)$$

## Partie 3

Systeme à une masse et 2 ressorts en bidimensionnel :



### 3.1 Résolution dans $\mathbb{R}^2$ :

Selon le Principe fondamental de la dynamique, on a :

$$m_1 \ddot{z}_1 = k_3(z_1 - z_2) + k_2(z_1 - z_2)$$

On prend  $k_2 = k_3$  ;

$$m_1 \ddot{z}_1 = -2.k.z_1 + k(z_3 + z_2)$$

Avec :

$$(z_3 + z_2) = 0$$

Donc :

$$m_1 \ddot{z}_1 = -2.k.z_1$$

$$(E) : \theta^2 + \frac{2k}{m} = 0$$

Donc :

$$w = \pm \sqrt{\frac{2k}{m}}$$

La solution est donc :

$$z_1(t) = C_1 \cos(wt) + C_2 \sin(wt); \quad C_1, C_2 \in \mathbb{R}$$

Conditions initiales :

$$z_1(0) = \begin{pmatrix} x_1(0) \\ y_2(0) \end{pmatrix} \quad ; \quad \dot{z}_1(0) = \begin{pmatrix} \dot{x}_1(0) \\ \dot{y}_1(0) \end{pmatrix} \quad (3.1)$$

D'où :

$$\boxed{z_1(t) = \cos(wt) \begin{pmatrix} x_1(0) \\ y_2(0) \end{pmatrix} + \frac{\sin(wt)}{w} \begin{pmatrix} \dot{x}_1(0) \\ \dot{y}_1(0) \end{pmatrix}} \quad (3.2)$$

La trajectoire est donc une Ellipse.

### 3.2 Résolution complexe : ( Identification de $\mathbb{R}^2$ par plan complexe ) $\mathbb{R}^2 \rightarrow \mathbb{C}$ :

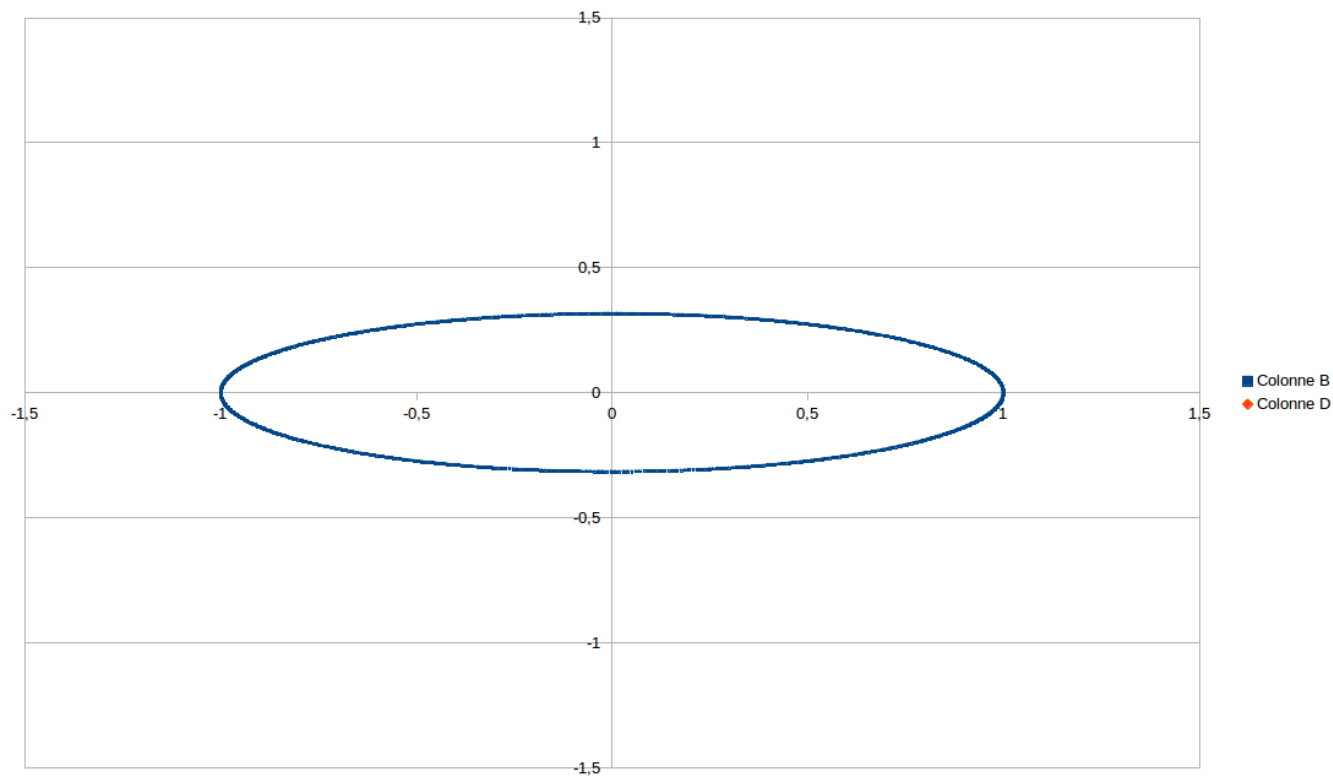
Soit :  $w_1 = x_1 + iy_1$

$$(E) : \ddot{w}_1 + \frac{2k}{m}w_1 = 0$$

D'où :

$$\boxed{w_1(t) = C_1 \cos(wt) + C_2 \sin(wt); \quad C_1, C_2 \in \mathbb{C}}$$

### 3.3 Représentation graphique



Avec  $k_2 = k_3$  et une position initiale tel que :

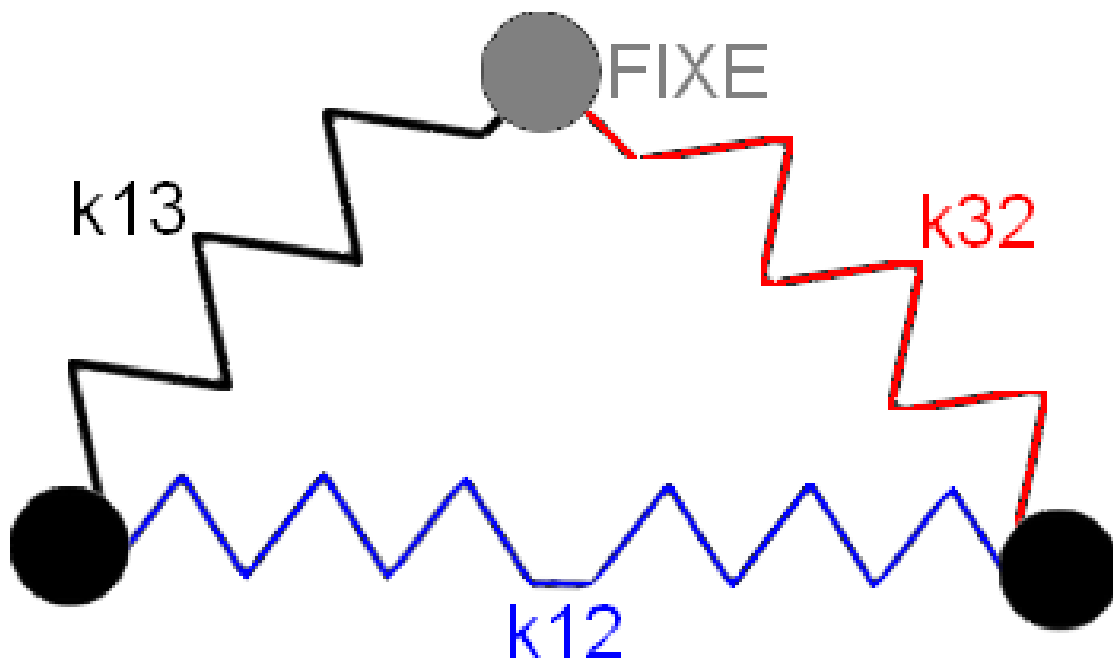
$$x = 1$$

$$y = 0$$

en donnant une légère vitesse initiale verticale à la masse on voit que la trajectoire est une ellipse et que cette trajectoire est périodique.

## Partie 4

Systeme à 3 masses et 3 ressorts en Triangle :



On fait le bilan des forces sur  $m_1$  et  $m_2$ .

$$\begin{cases} m_1 \ddot{z}_1 = k_{13}(z_3 - z_1) + k_{12}(z_2 - z_1) \\ m_2 \ddot{z}_2 = k_{32}(z_3 - z_2) + k_{12}(z_1 - z_2) \end{cases}$$

$$\begin{cases} m_1 \ddot{z}_1 = -(k_{13} + k_{12})z_1 + k_{12}z_2 + k_{13}z_3 \\ m_2 \ddot{z}_2 = k_{12}z_1 - (k_{32} + k_{12})z_2 + k_{32}z_3 \end{cases}$$

On sait que,

$$z_j = \begin{pmatrix} x_j \\ y_j \end{pmatrix}$$

On pose le changement de variable suivant :

$$w_j = x_j + iy_j$$

On obtient alors de système suivant dans  $\mathbb{C}^2$  (avant dans  $\mathbb{R}^4$ ) :

$$\begin{cases} m_1 \ddot{w}_1 = -(k_{13} + k_{12})w_1 + k_{12}w_2 + k_{13}w_3 \\ m_2 \ddot{w}_2 = k_{12}w_2 - (k_{32} + k_{12})w_2 + k_{32}w_3 \end{cases}$$

$$\begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix} \begin{pmatrix} \ddot{w}_1 \\ \ddot{w}_2 \end{pmatrix} = \begin{pmatrix} -(k_{13} + k_{12}) & k_{12} \\ k_{12} & -(k_{23} + k_{12}) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + \begin{pmatrix} k_{13}w_3 \\ k_{32}w_3 \end{pmatrix} \quad (4.1)$$

On choisit que  $k_{12} = k_{23} = k_{13} = k$  et aussi,  $m_1 = m_2 = m$ ,

$$\begin{pmatrix} \ddot{w}_1 \\ \ddot{w}_2 \end{pmatrix} = \frac{-k}{m} \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + k \begin{pmatrix} w_3 \\ w_3 \end{pmatrix} \quad (4.2)$$

On pose un autre changement de variable tel que notre système soit à l'équilibre :  $\underline{w} = w - w_{eq}$ , avec  $\underline{w} \in \mathbb{C}^2$

A l'équilibre  $\ddot{\underline{w}} = 0$ , donc,

$$0 = \frac{-k}{m} \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} w_{eq} + k \begin{pmatrix} w_3 \\ w_3 \end{pmatrix} \quad (4.3)$$

$$m \begin{pmatrix} w_3 \\ w_3 \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} w_{eq} \quad (4.4)$$

On pose  $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$  ce qui nous donne le système suivant,

$$A^{-1}m \begin{pmatrix} w_3 \\ w_3 \end{pmatrix} = A^{-1}Aw_{eq} \quad (4.5)$$

Or  $A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$  et où  $w_3 = (i; i)$ , on a alors,

$$w_{eq} = \frac{m}{3} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} i \\ i \end{pmatrix} \quad (4.6)$$

De plus, pour la simulation, nous prendrons  $m = 1$ , ce qui nous donne,

$$w_{eq} = \begin{pmatrix} i \\ i \end{pmatrix} \quad (4.7)$$

Après ces transformations, on obtient un système centré sur l'origine, tel que,

$$\begin{pmatrix} \ddot{w}_1 \\ \ddot{w}_2 \end{pmatrix} = \frac{-k}{m} \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \quad (4.8)$$

avec :

$$M = \frac{-k}{m} \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \quad (4.9)$$

On cherche maintenant à changer de base pour pouvoir simplifier les calculs. On commence par chercher les valeurs propres de la matrice M.

$$\det(M - \lambda Id) = -\frac{k}{m} \begin{vmatrix} 2 - \lambda & -1 \\ -1 & 2 - \lambda \end{vmatrix} = -\frac{k}{m} [(2 - \lambda)^2 - 1] \quad (4.10)$$

Pour  $-\frac{k}{m} [(2 - \lambda)^2 - 1] = 0$ , on trouve  $\lambda_1 = 3$  ou  $\lambda_2 = 1$ .

Puis on cherche la matrice de passage qui nous permettra de changer de base.

Pour  $\lambda_1 = 3$  :

$$\begin{pmatrix} 2 & -1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3x \\ 3y \end{pmatrix} \quad (4.11)$$

On peut donc écrire le système d'équation :

$$\begin{cases} 2x - y = 3x \\ -x + 2y = 3y \end{cases} \iff \begin{cases} -y = x \\ -x = y \end{cases}$$

On obtient un vecteur propre :  $u_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

De la même manière, pour  $\lambda = 1$  on trouve un vecteur propre  $u_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

On peut dire que la matrice de passage P est égale à :  $P = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$  et que la matrice dia-

gonale D est égale à :  $D = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}$

On cherche maintenant à exprimer notre matrice M dans la nouvelle base. Nous nommerons cette matrice M'.



On sait que  $M' = P^{-1}MP$  et on a  $P^{-1} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$

$$M' = \frac{-k}{m} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \quad (4.12)$$

On reprend le système précédent en remplaçant  $M$  par  $M'$  :

$$\begin{pmatrix} \ddot{y}_1(t) \\ \ddot{y}_2(t) \end{pmatrix} = M' \cdot \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \quad (4.13)$$

$$\iff \begin{pmatrix} \ddot{y}_1(t) \\ \ddot{y}_2(t) \end{pmatrix} = \frac{-k}{m} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} \quad (4.14)$$

En posant  $C = \frac{k}{m}$ , on obtient le système d'équations différentielles de second ordre avec second membre :

$$\begin{cases} \ddot{y}_1(t) = -3Cy_1(t) \\ \ddot{y}_2(t) = -Cy_2(t) \end{cases}$$

On obtient,

$$\underline{Y} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \begin{pmatrix} \lambda \cos(\sqrt{3C}t) + \mu \sin(\sqrt{3C}t) \\ \lambda' \cos(\sqrt{C}t) + \mu' \sin(\sqrt{C}t) \end{pmatrix} \quad (4.15)$$

On utilise maintenant la matrice de passage pour repasser dans la base initiale :

$$P\underline{Y} = \underline{W}$$

avec :

$$\underline{W} = \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} \quad (4.16)$$

$$P = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \quad (4.17)$$

On obtient donc :

$$\underline{W} = \begin{pmatrix} \lambda \cos(\sqrt{3C}t) + \mu \sin(\sqrt{3C}t) + \lambda' \cos(\sqrt{C}t) + \mu' \sin(\sqrt{C}t) \\ -\lambda \cos(\sqrt{3C}t) - \mu \sin(\sqrt{3C}t) + \lambda' \cos(\sqrt{C}t) + \mu' \sin(\sqrt{C}t) \end{pmatrix} \quad (4.18)$$

Pour trouver les constantes, on utilise les conditions initiales d'où :

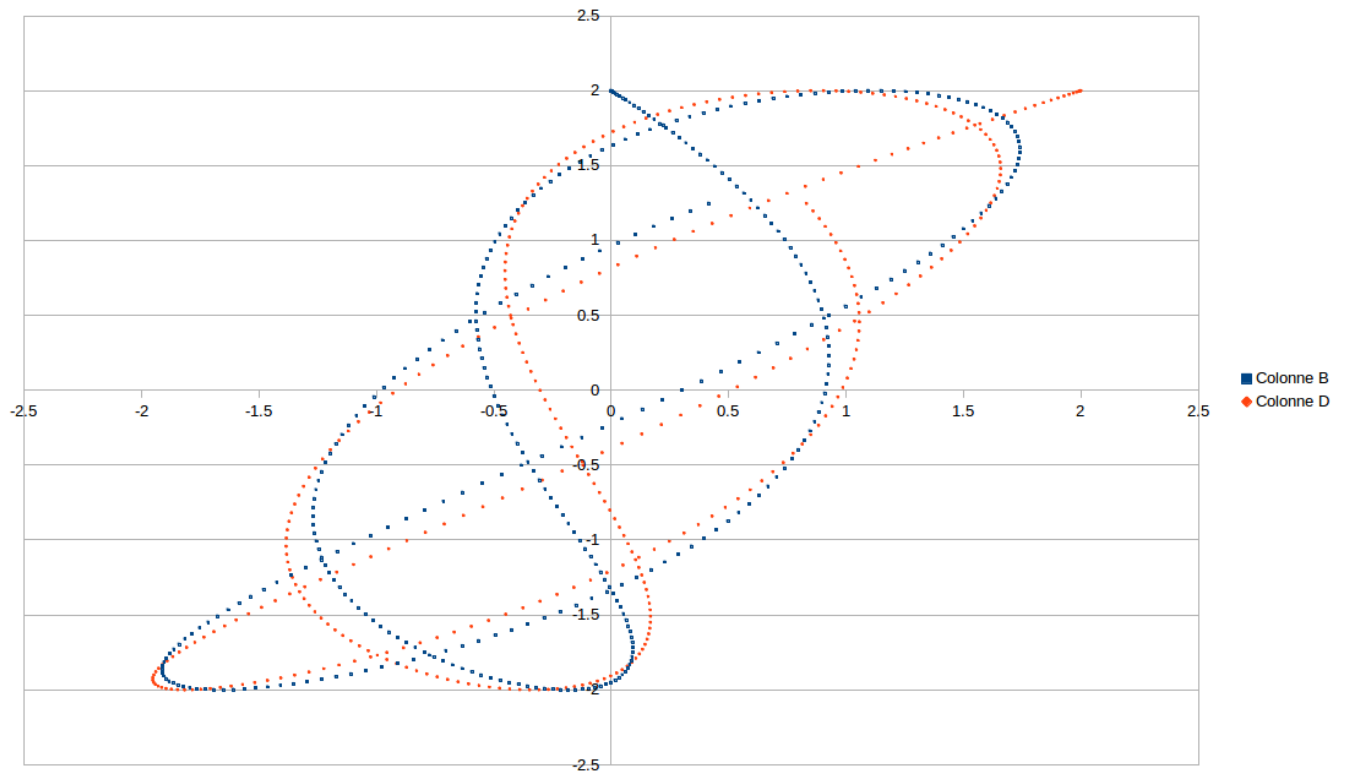
$$\begin{cases} \underline{w}_1(0) = x_1(0) + iy_1(0) = \lambda + \lambda' = A \\ \dot{\underline{w}}_1(0) = \dot{x}_1(0) + i\dot{y}_1(0) = -\mu\sqrt{3C} - \mu'\sqrt{C} = B \\ \underline{w}_2(0) = x_2(0) + iy_2(0) = -\lambda + \lambda' = A' \\ \dot{\underline{w}}_2(0) = \dot{x}_2(0) + i\dot{y}_2(0) = \mu\sqrt{3C} - \mu'\sqrt{C} = B' \end{cases}$$

$$\begin{cases} \lambda = \frac{A-A'}{2} \\ \lambda' = \frac{A+A'}{2} \\ \mu = \frac{-B'-B}{2\sqrt{3C}} \\ \mu' = \frac{B'-B}{2\sqrt{C}} \end{cases}$$

Remplaçons dans l'équation de  $\underline{W}$  :

$$\underline{W} = \begin{pmatrix} w_1(t) \\ w_2(t) \end{pmatrix} = \begin{pmatrix} \frac{A-A'}{2} \cos(\sqrt{3C}t) + \frac{-B'-B}{2\sqrt{3C}} \sin(\sqrt{3C}t) + \frac{A+A'}{2} \cos(\sqrt{C}t) + \frac{B'-B}{2\sqrt{C}} \sin(\sqrt{C}t) \\ -\frac{A-A'}{2} \cos(\sqrt{3C}t) - \frac{-B'-B}{2\sqrt{3C}} \sin(\sqrt{3C}t) + \frac{A+A'}{2} \cos(\sqrt{C}t) + \frac{B'-B}{2\sqrt{C}} \sin(\sqrt{C}t) \end{pmatrix} \quad (4.19)$$

## Représentation graphique



La constante de raideur est la même pour les 3 ressorts. On a pris en positions initiales :

$$x_1 = 0$$

$$y_1 = 2$$

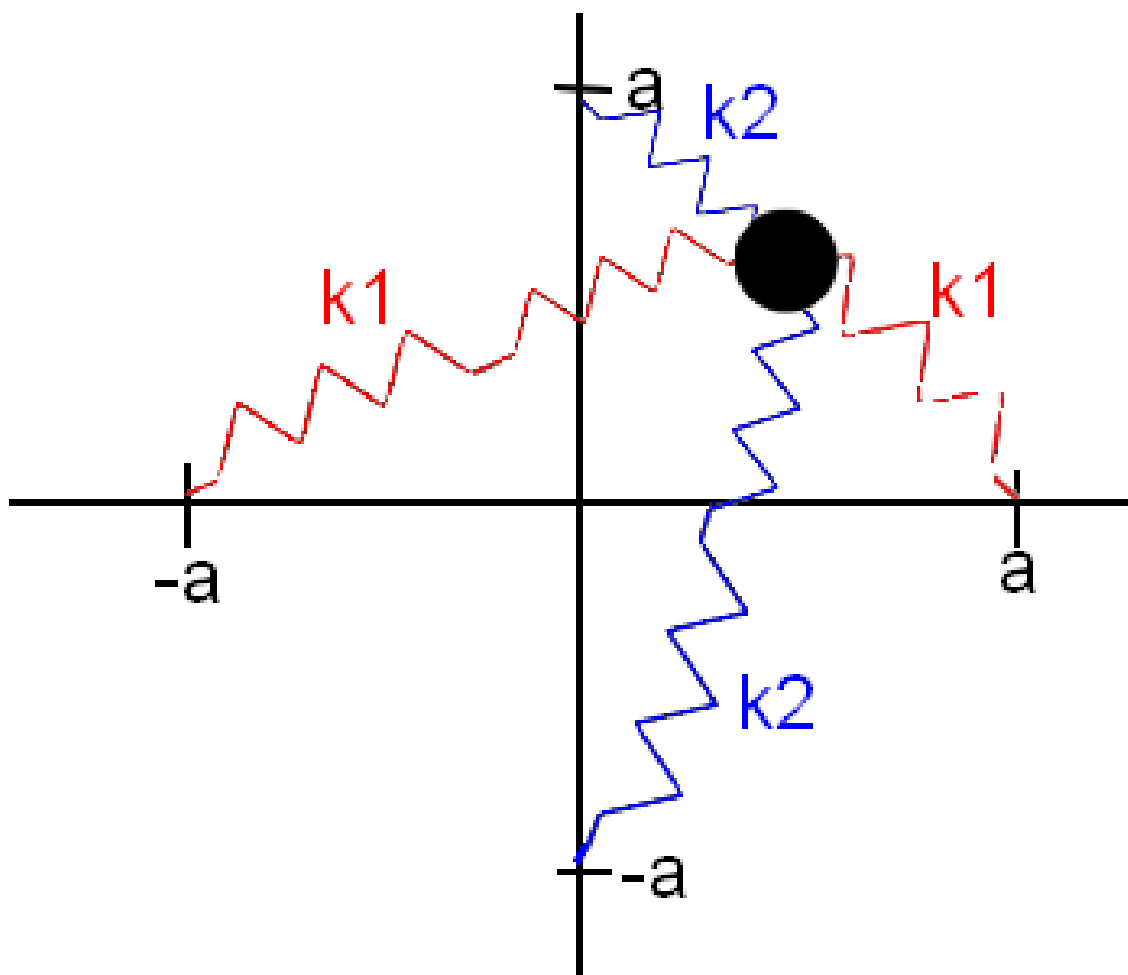
$$x_2 = 2$$

$$y_2 = 2$$

On voit que la trajectoires des masses n'est pas périodique et qu'elle dépend de la position de départ de l'autre masse. C'est cohérent avec la forme en triangle qui fait que les deux masses sont liées entre elle en plus de l'être à un point fixe. La trajectoire des masses ne semble pas être périodique et les masses ne repasseront semblablement pas par leur positions initiales.

## Partie 5

### Systeme à une masse et quatre ressorts



Soit  $k_1$  et  $k_2$  le constante de raideurs des ressorts.

On applique le principe fondamental de la dynamique (PFD) au système :

$$\begin{aligned} m.\ddot{z} &= k_1(z_1 - z) + k_1(z'_1 - z) + k_2(z_2 - z) + k_2(z'_2 - z) \\ \Leftrightarrow m.\ddot{z} &= -2(k_1 + k_2)z + k_1 \underbrace{(z_1 + z'_1)}_a + k_1 \underbrace{(z_2 + z'_2)}_{-a} \\ \Leftrightarrow m.\ddot{z} &= -2(k_1 + k_2)z \end{aligned}$$

On obtient :

$$m \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = -2(k_1 + k_2) \begin{pmatrix} x \\ y \end{pmatrix}$$

Si on prend  $K = k_1 + k_2$ , on a :

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \frac{-2K}{m} \begin{pmatrix} x \\ y \end{pmatrix}$$

Si on considère chaque équation de façon indépendante, on peut écrire le système suivant

$$\begin{cases} \ddot{x} + \left(\frac{2K}{m}\right)x = 0 \\ \ddot{y} + \left(\frac{2K}{m}\right)y = 0 \end{cases}$$

Pour résoudre ces équations, on trouve l'équation caractéristique qui est la même pour les deux.

$$\begin{aligned} r^2 + \frac{2K}{m} &= 0 \\ r &= \pm i \sqrt{\frac{2K}{m}} \end{aligned}$$

On obtient de système suivant,

$$\begin{cases} x(t) = \alpha \cos \frac{2K}{m}t + \beta \sin \frac{2K}{m}t \\ y(t) = \alpha' \cos \frac{2K}{m}t + \beta' \sin \frac{2K}{m}t \end{cases}$$

Si on considère maintenant les conditions initiales suivantes :

$$(z_0) = \begin{pmatrix} x_0 \\ 0 \end{pmatrix}$$

$$(\dot{z}_0) = \begin{pmatrix} 0 \\ v_0 \end{pmatrix}$$

On considère aussi que

$$\omega = \sqrt{\frac{2K}{m}}$$

on a alors,

$$\begin{cases} \alpha = x_0 \\ \alpha' = 0 \\ \beta = 0 \\ \beta' = \frac{v_0}{\omega} \end{cases}$$

donc, on a les solutions suivantes,

$$\begin{cases} x(t) = x_0 \cos \omega t \\ y(t) = \frac{v_0}{\omega} \sin \omega t \end{cases}$$

Si on prends  $x \ll a$  et  $y \ll a$ , par approximation, on obtient,

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \frac{1}{m} \begin{pmatrix} -2k_1 & 0 \\ 0 & -2k_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{5.1}$$

On obtient les solutions suivantes,

$$\begin{cases} x(t) = x_0 \cos \omega_1 t \\ y(t) = \frac{v_0}{\omega_2} \sin \omega_2 t \end{cases}$$

avec  $\omega_1^2 = \frac{2k_1}{m}$  et  $\omega_2^2 = \frac{2k_2}{m}$

Une autre solution pour résoudre aurait été de rester avec

$$z(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

On aurait eu le système d'équation suivant,

$$\begin{cases} z(t) = \alpha \cos \omega t + \beta \sin \omega t \\ \dot{z}(t) = -\alpha' \omega \sin \omega t + \beta' \omega \cos \omega t \end{cases}$$

Or à  $t=0$ , on a :

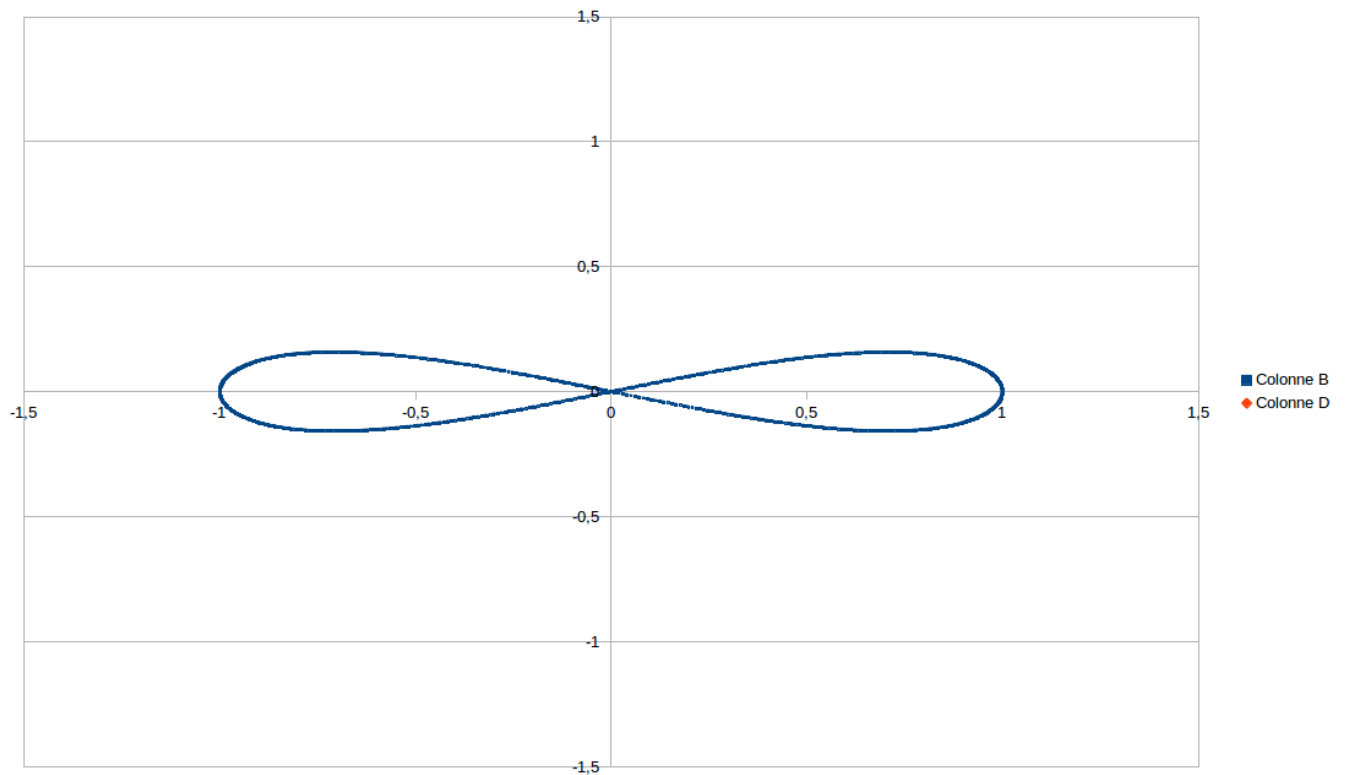
$$z_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \alpha$$

$$\dot{z}_0 = \begin{pmatrix} \dot{x}_0 \\ \dot{y}_0 \end{pmatrix} = \omega \beta'$$

On obtient donc au final,

$$z(t) = \begin{cases} x(t) = x_0 \cos \omega t + \frac{\dot{x}_0}{\omega} \sin \omega t \\ y(t) = y_0 \cos \omega t + \frac{\dot{y}_0}{\omega} \sin \omega t \end{cases}$$

## Représentation graphique



Avec une position initiale :

$$x = 1$$

$$y = 0$$

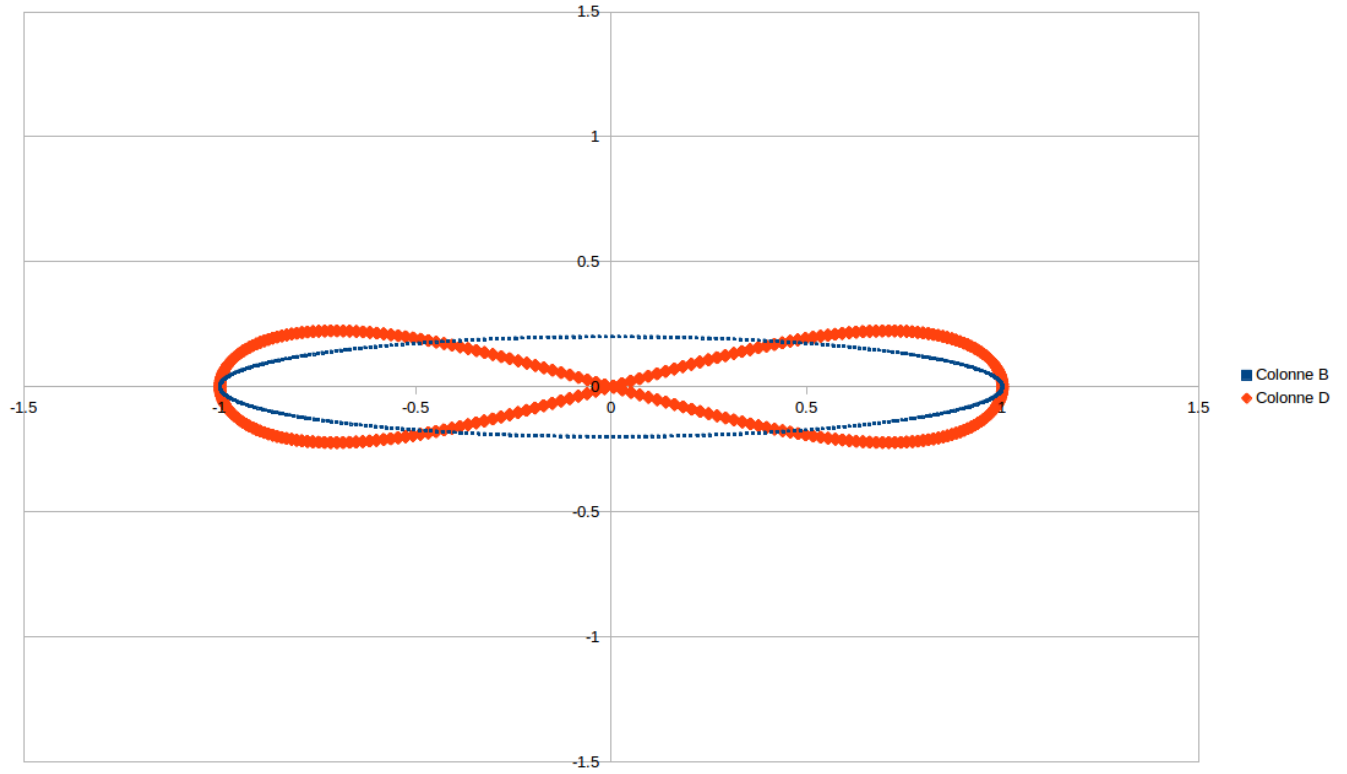
Ici  $k_1$  et  $k_2$  sont différent avec :

$$k_2 = 4 * k_1$$

On obtient une trajectoire périodique intéressante. Nous avons voulu tenter d'obtenir une trajectoire différente et avons recalculé la trajectoire avec cette fois avec un unique  $k$  tel que :

$$k = k_1 + k_2$$

En superposant les 2 courbes on obtient :



On retrouve une ellipse similaire au cas avec 2 ressorts plus haut. On voit donc que le changement des constantes de raideur peut influencer la forme de la trajectoire d'une masse dans ce cas mais que la trajectoire semble rester périodique.



# Partie 6

## Simulation

### 6.1 Présentation du programme

Afin d'avoir une vision concrète du mouvements des masses pour chaque système, nous avons du réaliser une simulation numérique nous permettant d'obtenir des graphiques représentant ces mouvements en deux dimensions. C'est avec cette simulation que nous avons produits les graphiques présent dans la partie d'analyse mathématique. Dans cette partie nous détaillerons comment est réalisé le programme Pascal et la façon d'obtenir les graphiques.

Nous avons choisi de réaliser ce programme en Pascal car nous avons déjà une connaissance de la syntaxe, pratiquant ce langage de programmation depuis 2 ans. De plus, nous avons choisi de réaliser deux programmes différents. Un qui prend en compte une masse, et l'autre deux. Le principe du programme est d'utiliser la méthode de Runge-Kutta afin d'obtenir les positions des masses dans le temps à partir des équations différentielles des masses (deux par masses donc une pour chaque dimensions) et de conditions initiales prédéfinies dans le programme mais modifiable (vitesses et positions initiales en  $x$  et  $y$ ).

En réalisant ce programme, un des problème est l'approximation que peut faire l'ordinateur en calculant les valeurs. C'est afin de réduire au maximum ces erreurs que nous utilisons la méthode de Runge-Kutta qui calcule la vitesse et la position à l'aide de 4 valeurs intermédiaires.

Par rapport à l'affichage graphique de ces trajectoires, le programme écrit les positions successives des masses dans un fichier texte. Nous avons fait un programme qui utilise la bibliothèque graphique SDL qui affichait donc directement un graphique à l'exécution du programme. Malheureusement, les axes n'étaient pas gradués et nous avons donc du trouver une alternative. Nous utilisons une feuille de tableur à laquelle on donne les valeurs stockées dans le fichier texte, le tableur dessinera donc le graphique sans aucun problème. En configurant la figure pour que les axes soient orthonormés, il est possible d'avoir un aperçu clair de la trajectoire des masses.

## 6.2 Explication de l’algorithme de Runge-Kutta 4

Des équations de mouvement seules ne permettent pas de visualiser la réalité des faits. Si l’on possède les équations des positions des masses en fonction du temps (en d’autres termes, si les équations différentielles de mouvement sont solvables), il est très facile de mettre en place un programme traçant à chaque milliseconde par exemple la position d’une masse, nous permettant de visualiser sa trajectoire. En revanche, si les équations différentielles de mouvement ne peuvent être résolues, il nous faut alors utiliser une méthode numérique qui permet une approximation la plus précise possible de la trajectoire d’une masse. Nous avons donc utilisé l’algorithme dit de ‘Runge-Kutta 4’ (RK4). Celui-ci est une méthode très adaptée à la résolution de notre problème.

La méthode RK4 consiste en l’approximation de la solution d’une équation de type  $y'(t) = F(y(t), t)$  avec une condition initiale connue  $y(t_0) = y_0$ . A chaque instant  $t_n$ , l’algorithme RK4 permet d’approximer la solution à  $t_{n+h}$ , en approximant les pentes de la fonction  $y(t)$  à quatre reprises :

*entre  $t_n$  et  $t_{n+0.25h}$ ,  $t_{n+0.25h}$  et  $t_{n+0.5h}$ ,  $t_{n+0.5h}$  et  $t_{n+0.75h}$ , et enfin  $t_{n+0.75h}$  et  $t_{n+h}$*

(h étant le pas choisi entre chaque point).

Si l’on nomme  $k_1$ ,  $k_2$ ,  $k_3$  et  $k_4$  les quatre pentes citées précédemment, la méthode RK4 indique que :

$$\begin{aligned} k_1 &= F(y(t), t) \\ k_2 &= F(y(t) + k_1 * 0.5h, t + 0.5h) \\ k_3 &= F(y(t) + k_2 * 0.5h, t + 0.5h) \\ k_4 &= F(y(t) + k_3 * h, t + h) \end{aligned}$$

Et que :

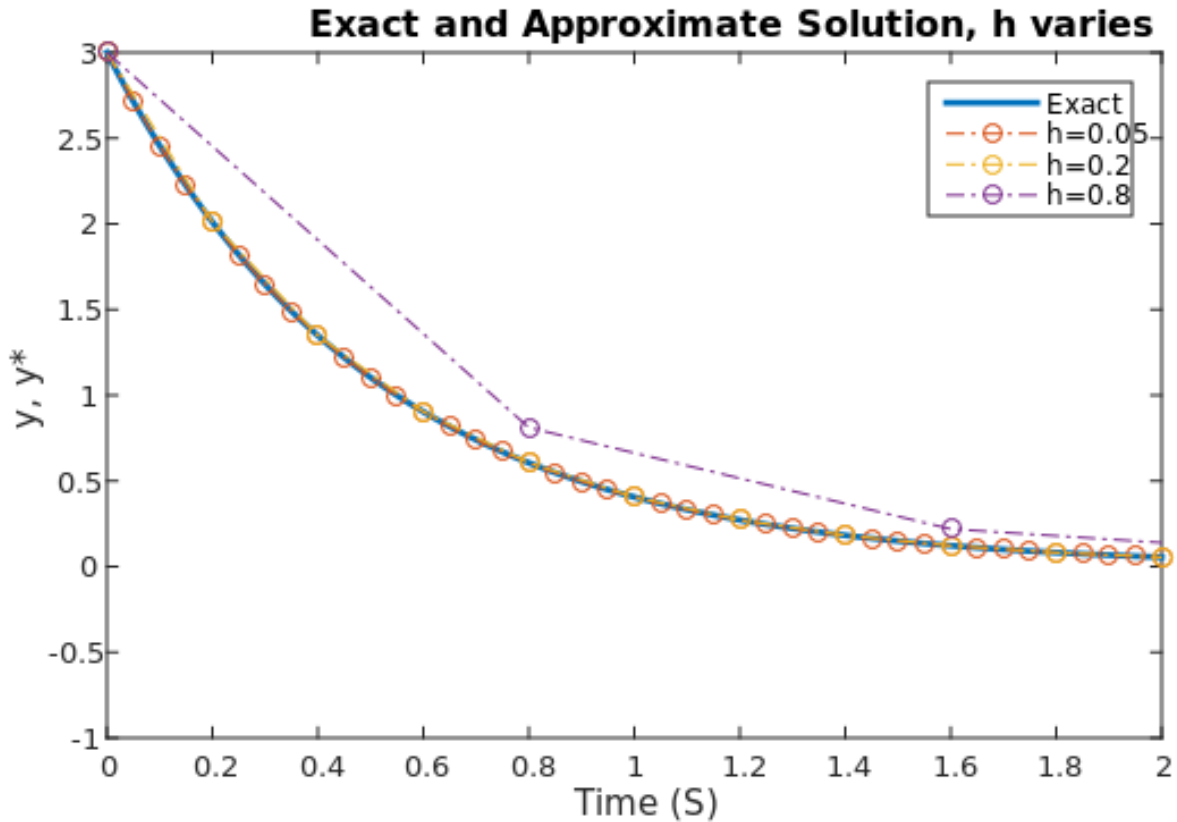
$$y(t + h) = y(t) + \frac{(k_1 + 2k_2 + 2k_3 + k_4)}{6} * h$$

On peut remarquer que le poids des pentes du milieu ( $k_2$  et  $k_3$ ) est plus élevé que celui des pentes extrêmes ( $k_1$  et  $k_4$ ). Cela est dû au fait qu’il est considéré qu’elles sont de meilleures approximations pour la pente finale de  $y(t)$  à  $y(t+h)$ . L’entière fonctionnalité de l’algorithme RK4 repose sur la connaissance de la condition initiale  $y(t_0) = y_0$ . Prenons un exemple simple pour s’en rendre bien compte. Si l’on a l’équation différentielle  $y'(t) + 2y(t) = 0$  avec  $y(t_0) = 3$ . L’équation différentielle s’écrit également :

$$y'(t) = -2y(t) = F(y(t), t).$$

Ainsi on remarque que  $k_1 = -2y(t)$  soit pour la première approximation  $-2y(t_0) = -2 * 3 = -6$ . Ensuite, l’algorithme permet de calculer  $k_2$  puisque  $k_1$  est maintenant connu, et ainsi de suite jusqu’à l’approximation de  $y(t_0 + h)$ . Il suffit ensuite de recommencer en prenant  $y(t) = y(t_0 + h)$ .

On pourrait penser qu’en utilisant une méthode itérative comme RK4, c’est à dire calculer une première estimation de la solution pour calculer une seconde estimation, et ainsi de suite, perdrait en précision à chaque étape. Cependant, l’erreur commise à chaque étape est de l’ordre de  $O(h^5)$  et l’erreur globale de l’ordre de  $O(h^4)$ . Ainsi, l’algorithme RK4 prend tout son sens avec une valeur de pas la plus faible possible, comme le prouve ce graphique :



Notre programme Pascal reprend donc l’algorithme RK4 en permettant de modifier les conditions initiales, et va ainsi calculer les approximations des solutions des équations différentielles de mouvement de chaque masse d’un système (RK4 devant être appliqué autant de fois que le nombre de points voulus, et les équations différentielles étant propres à chaque masse et chaque système).

Ainsi, le programme Pascal nous a permis de modéliser efficacement la trajectoire des masses. Cette vision obtenue est faillible à cause de l’imprécision de l’ordinateur mais les erreurs sont contrôlées et réduites grâce à la méthode de Runge-Kutta. Nous obtenons ainsi une modélisation proche de la réalité physique.

# Partie 7

## Conclusion

Ce projet comme tous les autres projets de ce semestre nous a appris à travailler en groupe. C'est d'ailleurs le projet que nous avons fait cette année avec le plus grand nombre de membre. Contrairement aux autres projets où nous travaillons souvent tous sur la même chose, ici bien que le sujet soit le même, nous avons dû nous organiser binôme travaillant sur des aspect différents de la problématique. Ce qui à donné nos deux parties : analyse mathématique et simulation.

De plus, nous avons pu utiliser les connaissances que nous avons assimilées pendant ces deux premières années à l'INSA. Il est intéressant de voir que ce que nous avons appris dans la théorie est essentiel pour travailler sur cette problématique qui est encore d'actualité aujourd'hui.

Ainsi, ce projet fut un excellent moyen de combiner application des connaissances, recherches et organisation de groupe, toutes ces compétences qui nous seront utiles à la fois dans la suite de nos études et dans notre future carrière.

# Partie 8

## Bibliographie

### Documents internet

#### Documentation mathématique et physique

- Explication de la méthode de Runge-Kutta <http://www.cinam.univ-mrs.fr/klein/teach/mip/numeriq/node46.html>
- Étude de système masse-ressort [http://www.ruzzene.gatech.edu/CISM\\_Course/Course\\_Slides\\_files/Ruzzene02.pdf](http://www.ruzzene.gatech.edu/CISM_Course/Course_Slides_files/Ruzzene02.pdf)
- Étude d'un oscillateur harmonique 2D <http://farside.ph.utexas.edu/teaching/336k/Newtonhtml/node28.html>
- Simulation d'une masse attachée à un ressort et resolution de système <http://www.myphysicslab.com/spring2d.html>
- Etude d'un mouvement en 2 dimensions <http://farside.ph.utexas.edu/teaching/336k/Newtonhtml/node28.html>

### Logiciels

- Compilateur Pascal **Freepascal** <http://www.freepascal.org/>

# Partie 9

## Annexe

### 9.1 Programme Pascal pour une masse

```
program runge1masse;  
uses sysutils;  
  
const  
  CONSOLE = 0; //si 0 sortie copiable pour tableur  
  
  KRAID1 = 1;  
  KRAID2 = 1;  
  MASS = 0.2;  
  NB_POINTS = 1000;  
  
var n : Integer;  
  x,y,v,w,h,vn1,wn1, yn1, xn1 : real;  
  t : real;  
  fichier : Text;  
  //format: TFloatFormat;  
  
function f(A,B : real) : real;  
  //var r : real;  
  begin  
    //r := sqrt(A*A+B*B);  
    f := (-2*(KRAID1)*A)/MASS;  
  end;  
  
function g(A,B : real) : real;  
  //var r : real;  
  begin  
    //r := sqrt(A*A+B*B);  
    g := (-2*(KRAID2)*B)/MASS  
  end;  
  
procedure RK4D(var xn,yn,vn,wn : real);  
  var k11,k12,k13,k14,k21,k22,k23,k24,  
      k31,k32,k33,k34,k41,k42,k43,k44 : real;
```

```

37 begin
38
39     k11 := h*vn;
40     k12 := h*f(xn,yn);
41     k13 := h*wn;
42     k14 := h*g(xn,yn);
43
44     k21 := h*(vn+(k12/2));
45     k22 := h*f(xn+(k11/2),yn+(k13/2));
46     k23 := h*(wn+(k14/2));
47     k24 := h*g(xn+(k11/2),yn+(k13/2));
48
49     k31 := h*(vn+(k22/2));
50     k32 := h*f(xn+(k21/2),yn+(k23/2));
51     k33 := h*(wn+(k24/2));
52     k34 := h*g(xn+(k21/2),yn+(k23/2));
53
54     k41 := h*(vn+k32);
55     k42 := h*f(xn+k31,yn+k33);
56     k43 := h*(wn+k34);
57     k44 := h*g(xn+k31,yn+k33);
58
59     xn1 := xn + (k11 + (2*k21) + (2*k31) + k41)/6;
60     vn1 := vn + (k12 + (2*k22) + (2*k32) + k42)/6;
61     yn1 := yn + (k13 + (2*k23) + (2*k33) + k43)/6;
62     wn1 := wn + (k14 + (2*k24) + (2*k34) + k44)/6;
63
64 end;
65
66 begin
67     x := 1; y := 0; v := 0; w := 1;
68     h := 0.01;
69     t := 0;
70     //format := ffGeneral;
71     assign(fichier, 'position.txt');
72     rewrite(fichier);

```

```

73 for n:=1 to NB_POINTS do
74     begin
75         RK4D(x,y,v,w);
76
77         if CONSOLE = 1 then
78             begin
79                 writeln('t : ',t:0:3);
80                 writeln('x : ', x:0:5);
81                 writeln('y : ', y:0:5);
82                 writeln('v : ', v:0:5);
83                 writeln('w : ', w:0:5);
84                 writeln();
85             end
86         else
87             begin
88                 //Sortie excel
89                 writeln(x:0:5,' ', y:0:5);
90                 writeln(fichier, x:0:5,' ', y:0:5)
91
92                 //writeln(fichier, stringreplace(FloatToStrF(x,format,5,5),'.','',[rfReplaceAll]),' ', stringreplace(FloatToStrF(y,f
93
94             end;
95
96             x := xn1;
97             y := yn1;
98             v := vn1;
99             w := wn1;
100
101             t := t + h
102
103
104         end;
105     close(fichier);
106 end.
107
108

```