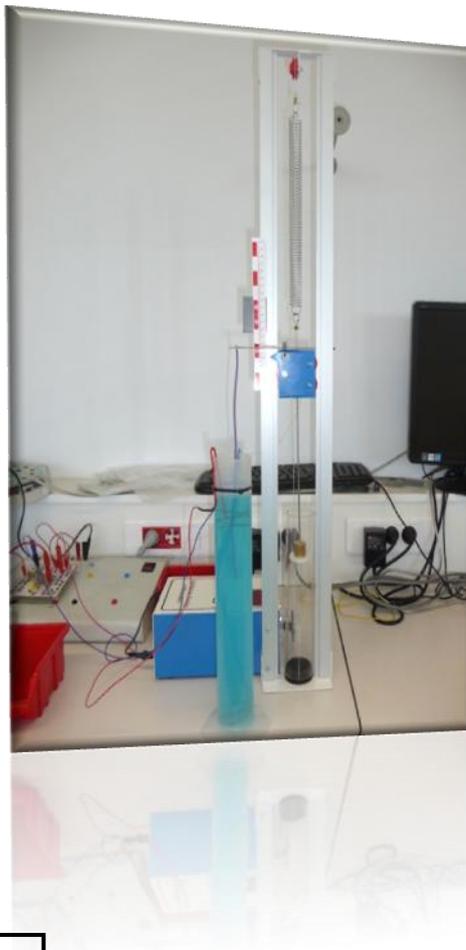


## Étude d'un oscillateur (cas du système masse-ressort)



Enseignant responsable du projet : Bernard GLEYSE

**Etudiants :**

Justine FEREY

Monya HAROUEL

Karine LESAGE

Aurore THIOT

Qian ZHANG

Yidong ZHAO

*Cette page est laissée intentionnellement vierge.*

Date de remise du rapport : 18/06/11

Référence du projet : STPI/P6-3/2011 – 7

Intitulé du projet : *Étude d'un oscillateur (cas du système masse-ressort)*

Type de projet : *Modélisation et expérimentation*

Objectifs du projet

*Etudier dans un premier temps par l'expérience le mouvement d'un solide masse-ressort, calculer les solutions théoriques, modéliser les solutions à l'aide de méthodes numériques et comparer les résultats.*

## TABLE DES MATIERES

1. Introduction .....	6
2. Méthodologie / Organisation du travail.....	7
3. Travail réalisé et résultats .....	9
3.1 Travail théorique.....	9
3.1.1 Système soumis uniquement aux forces de gravité.....	9
3.1.2 Système soumis aux forces de frottements fluides.....	10
3.1.3 Système entretenu soumis aux forces de frottements fluides.....	11
3.2 Travail pratique.....	11
3.3 Méthode numérique .....	14
3.3.1 Méthode d'Euler.....	14
3.3.2 Méthode RK4 .....	18
3.4 Partie simulation .....	20
4. Conclusions et perspectives .....	25
5. Bibliographie.....	26
6. Annexes (non obligatoire – exemples ci-dessous) .....	27
6.1 Documentation technique .....	27
6.2 Listings des programmes réalisés .....	27

## NOTATIONS, ACRONYMES

RK4 : Runge Kutta 4.

## 1. Introduction

Dans le cadre de notre formation d'ingénieur à l'INSA de Rouen, nous sommes régulièrement amenés à réaliser des projets en groupe, dont ce projet de P6-3. Ce dernier s'inscrit dans l'optique de l'INSA de former des ingénieurs capables de travailler en groupe, et ce, sur un large éventail de domaines.

C'est pourquoi, durant treize séances, nous nous sommes concentrés sur l'étude d'un système masse-ressort. L'étude de ce dernier est un projet relativement complet, qui rassemble à la fois de la programmation, de la physique, de l'analyse numérique mais aussi des mathématiques. Pour pouvoir étudier ce système, nous nous sommes aidés du TP que nous avons fait au semestre 3, intitulé « résonance mécanique ». L'objectif de notre projet n'était pas de refaire simplement le TP ou bien d'étudier d'autres phénomènes physiques mais plutôt de réaliser une modélisation mathématique, permettant de résoudre les équations différentielles d'ordre 2 avec Maple et la programmation en C++.

Lors du projet, nos séances se sont déroulées dans la salle informatique à raison d'1h30 par semaine. Lors des cinq premières séances, nous avons pris connaissance du sujet. Nous avons travaillé tous ensemble sur la partie théorique : la méthode d'Euler et de Runge Kutta 4, qui sont des méthodes d'analyse numérique d'approximation de solutions d'équations différentielles. Puis à partir de la sixième séance, nous nous sommes intéressés à la programmation (C++) et à la modélisation. Puis dans les dernières séances, nous avons réalisé le TP afin d'obtenir tous les mêmes valeurs.

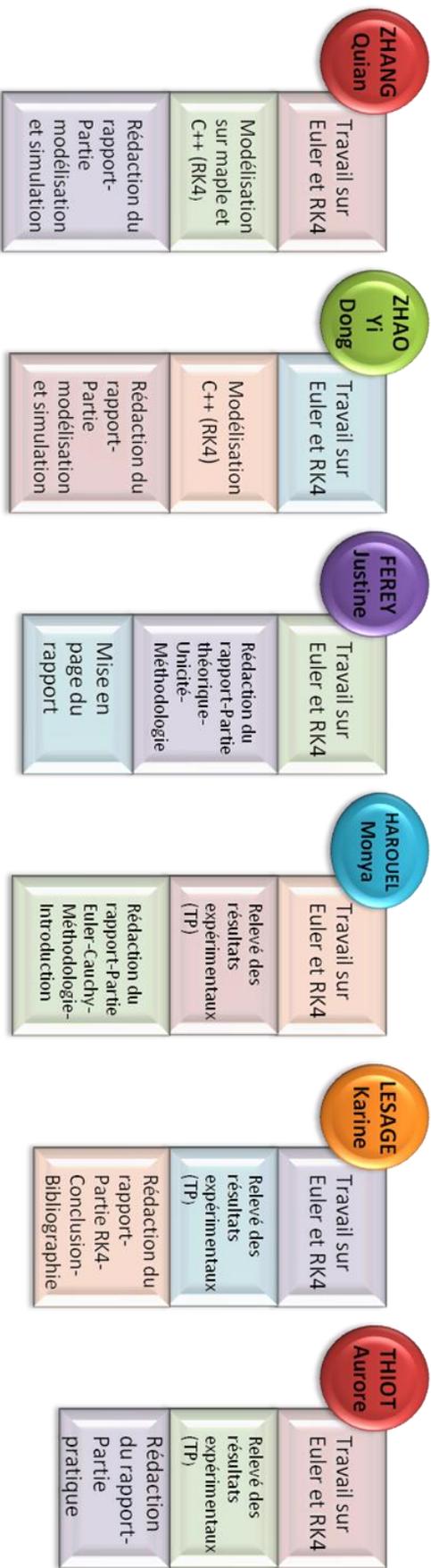
Le fait que l'on soit dans différentes thématiques a enrichi notre projet, nous étions complémentaires en termes de connaissance et de savoir-faire.

Nous allons dans un premier temps, à travers ce dossier, présenter les méthodes et l'organisation du groupe dans le suivi du travail. Ensuite, nous passerons à l'étude théorique de ce système ainsi qu'à la partie pratique, en présentant le travail expérimental qui a été réalisé. Par la suite, nous nous attacherons à l'aspect mathématique et aux différentes méthodes d'analyse numérique établies afin d'obtenir une solution approchée de nos équations différentielles, qui nous permettra de les modéliser numériquement. Enfin, nous ferons une comparaison entre les résultats de nos expériences, nos résultats théoriques et ceux obtenus numériquement.

## 2. Méthodologie / Organisation du travail

Ce sujet nous a été attribué après différents sondages faits auprès des élèves en fonction des goûts et des horaires possibles. Au cours de ce semestre, nous avons appris à travailler ensemble pour cibler les différentes affinités de chacun. Nous nous sommes aperçus que la plupart d'entre nous n'était pas orientée vers la programmation ce qui aurait pu nous poser quelques problèmes.

Après quelques interrogations concernant notre projet, l'enseignant nous a expliqué la méthodologie et le déroulement à suivre pour avancer dans ce projet. Au début, nous avons travaillé tous ensemble sur les mêmes tâches ce qui nous ralentissait. Nous faisons de petites réunions de mise au point à la fin de chaque séance pour se rendre compte du travail effectué et du travail à faire. Après quelques séances, nous avons commencé à travailler sur des tâches différentes. Nous nous sommes répartis le travail selon l'organigramme suivant :



## Répartition des tâches

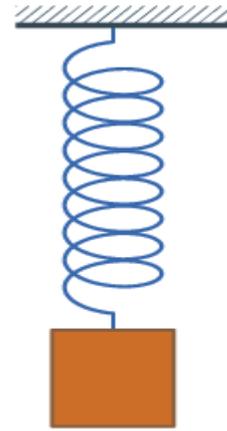
### 3. Travail réalisé et résultats

Un système masse ressort est un système mécanique à un degré de liberté. Il y a donc une seule grandeur qui varie dans ce système, ici c'est la position selon l'axe Oz. Il est constitué par une masse accrochée à un ressort. Son mouvement est dû à trois forces :

- une force de rappel
- une force d'amortissement  $F_a$
- une force extérieure  $F_e$

#### 3.1 Travail théorique

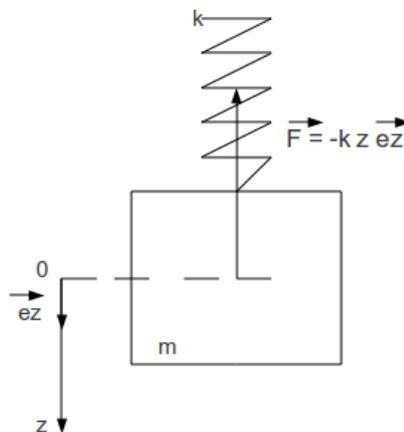
Nous allons établir dans cette partie les différentes équations différentielles de la position de ce système associées aux différents systèmes et allons résoudre ces équations. Commençons par un système soumis uniquement aux forces de gravité.



##### 3.1.1 Système soumis uniquement aux forces de gravité

Supposons une masse  $m$  suspendue à un ressort de constante de raideur  $k$ . Cette masse est soumise à la gravité. Nous noterons  $g$  la valeur de l'accélération de la gravité. La masse  $m$  est soumise à deux forces :

- la force  $P$  (poids de l'objet) générée par la gravité, qui s'exerce toujours dans le même sens (vers le bas), et l'amplitude est égale au produit de la masse  $m$  par l'accélération de la pesanteur  $g$  tel que  $P=mg$ .
- La force de rappel  $F$  du ressort ; dont la valeur dépend de la différence entre l'allongement du ressort  $L$  et sa longueur  $L_0$ . L'orientation de cette force change selon que le ressort est comprimé ou dilaté et dépend de  $k$ , la constante de raideur du ressort.



Appliquons maintenant le principe fondamental de la dynamique. Nous obtenons alors :

$$\sum \text{Forces extérieures} = m\vec{a}$$

Avec  $m$  la masse de l'objet. Nous obtenons :

$$P+F=ma \implies P+F=m.\ddot{y}$$

Les relations donnant les valeurs des deux forces sont :

- Le poids :  $P= mg$
- la force du ressort :  $F = -kz$

En reportant ces équations dans l'expression du principe fondamental de la dynamique, nous obtenons :

$$mz'' = mg - kz \text{ avec } z=(z-z_0)$$

$$\Leftrightarrow mz'' = mg - k(z-z_0)$$

Equation différentielle de la position d'un système masse ressort soumis uniquement aux forces de gravité :

$$mz''+kz = mg + kz_0$$

### 3.1.2 *Système soumis aux forces de frottements fluides*

On suppose maintenant que la masse est soumise à une force de frottement  $F_1$  de type fluide. Cette force agit toujours dans le sens contraire au déplacement de l'objet. De plus, elle est proportionnelle à la vitesse de l'objet. Cette force est définie par la formule suivante :

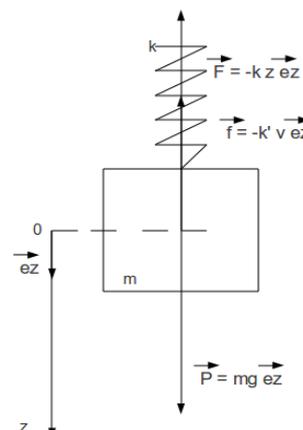
$$F_1 = -\alpha v \vec{e}_z$$

Nous allons maintenant tenter d'établir l'équation différentielle soumise à ce type de système. Ajoutons alors cette force aux autres forces exercées dans l'équation du mouvement. Nous obtenons :

$$T+P+F_1 = ma$$

$$mg - k(z-z_0) - \alpha v = mz''$$

$$mg - k(z- z_0) - \alpha z' = mz''$$



Equation différentielle de la position d'un système masse ressort soumis aux forces de gravité et aux forces de frottements fluide :

$$z'' + \frac{\alpha}{m} z' + \frac{k}{m} z = g + \frac{kz_0}{m}$$

### 3.1.3 Système entretenu soumis aux forces de frottements fluides

Prenons maintenant le cas de mouvement de frottement fluide entretenu avec une force sinusoïdale de la forme  $\cos(\omega t)$ .

Nous obtenons alors l'équation différentielle suivante :

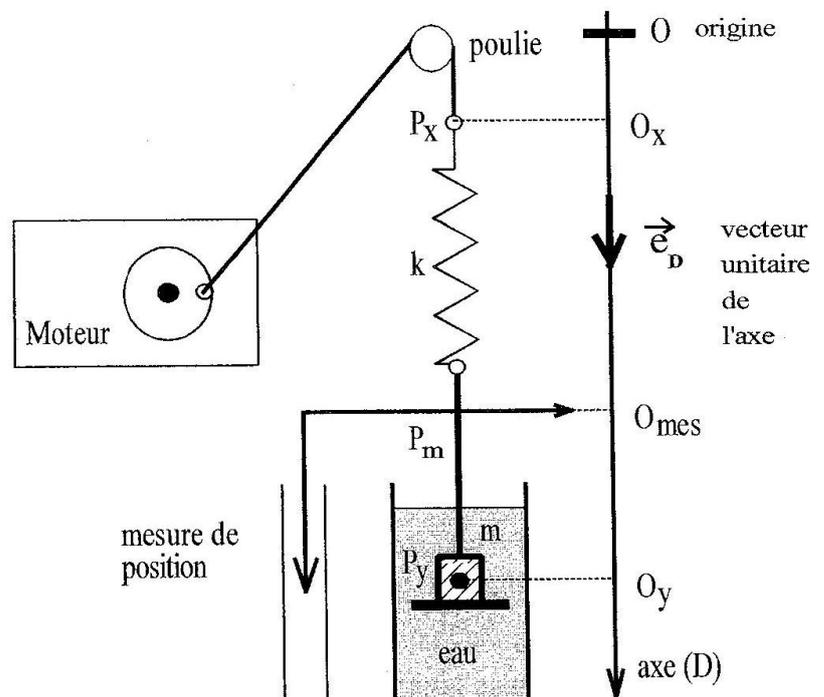
$$z'' + \frac{\alpha}{m} z' + \frac{k}{m} z = g + \frac{kz_0}{m} + \cos(\omega t)$$

## 3.2 Travail pratique

Pour comparer la pratique et la modélisation, nous avons repris les résultats du TP « résonance mécanique » effectué lors de l'U.V. P6-2 au semestre dernier. En effet ce TP nous offre les courbes expérimentales ainsi que les paramètres des équations du mouvement.

Il faut noter que nous avons voulu refaire le TP pour obtenir de meilleures courbes mais le changement de dispositif et les problèmes techniques que nous avons rencontrés nous ont freinés dans notre démarche.

### Schéma du montage



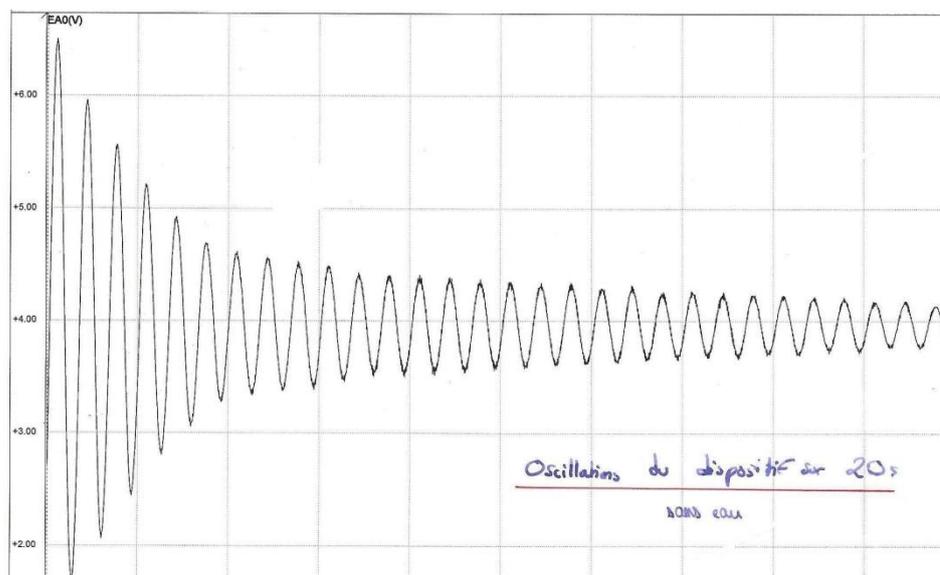
Avec  $P_x$ , le point d'ancrage de l'excitateur,  $P_y$  le centre d'inertie de la masse mobile et  $P_m$  le point de mesure du mouvement.

La mesure de position s'effectue grâce à une éprouvette contenant une solution de sulfate de cuivre à 20g/L avec des électrodes de cuivres. Une électrolyse de type à anode soluble se produit, ainsi il n'y a pas de création d'une force électromotrice. Du coup, l'ensemble se comporte comme une résistance. Ainsi la tension entre l'électrode soluble et la masse donne une indication linéaire sur la position sur l'axe vertical.

### Avec masse et frottements de l'air

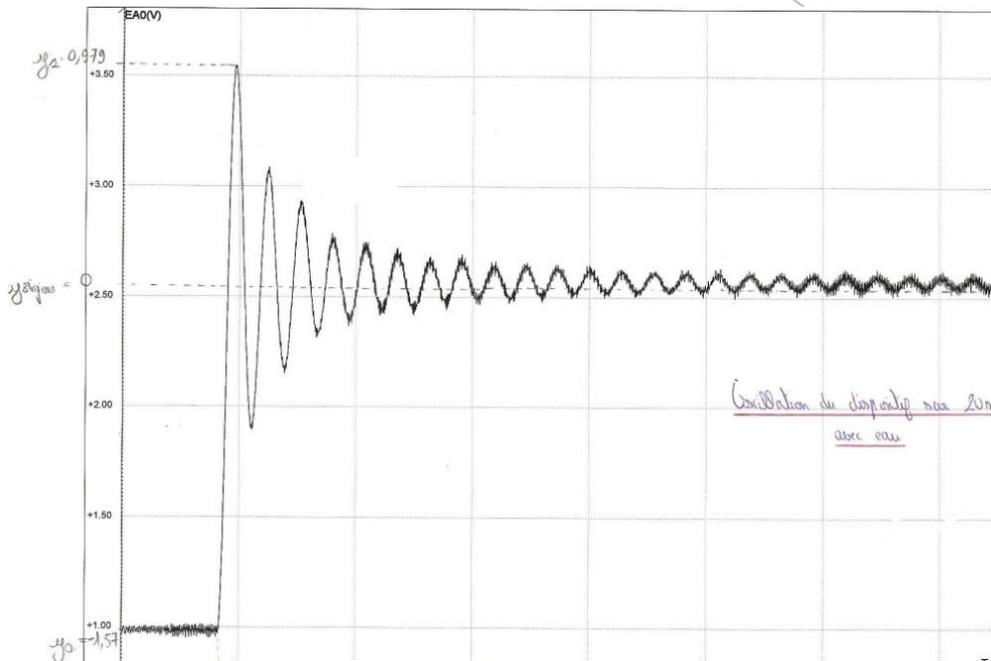
Pour commencer, nous avons pesé la masse qui était de 0,165 Kg. Puis nous avons déduit  $k$ , la constante de raideur du ressort grâce à une surcharge en mesurant l'allongement du ressort. Avec la formule  $k = \frac{\Delta P}{\Delta l}$  où  $P$  est le poids (mg en N) et  $l$  la longueur du ressort (en m), on obtient  $k = 24,9$  N/m.

On enregistre les oscillations sans eau en configurant Synchronie (2 000 points avec 20s d'enregistrement). On obtient un régime amorti :



### En rajoutant des frottements dû au fluide (eau)

Maintenant on rajoute une force de frottement due au fluide ici de l'eau. On enregistre aussi les oscillations libres sous synchronie.



Cette fois ci on observe une décroissance des oscillations beaucoup plus rapide. La force de frottement de l'eau amortit bien le système jusqu'à l'obtention d'un régime permanent.

Le mouvement  $y(t)$  est donné par 
$$y(t) = \frac{y_0}{\cos(\Phi)} e^{-\alpha \omega_0 t} \cos\left(\omega_0 \sqrt{1 - \alpha^2} t + \Phi\right)$$

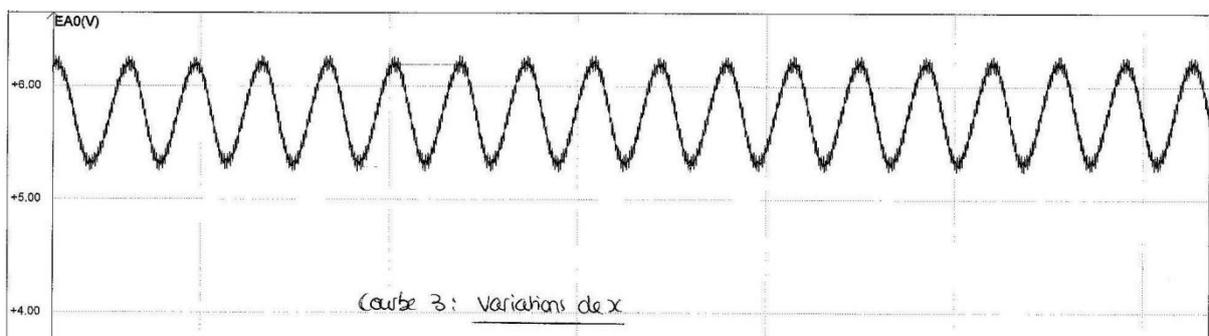
On relève les différences de potentiel  $y_0 = -1,57$  V et  $y_1 = 0,979$  V.

Enfin on calcule alpha à partir de sa formule 
$$\alpha = \frac{\ln\left(-\frac{y_1}{y_0}\right)^2}{\pi^2 + \ln\left(-\frac{y_1}{y_0}\right)}$$
 et on obtient le

coefficient d'amortissement  $\alpha = 0,15$ .

### En ajoutant un moteur

Le moteur entretient le mouvement. On obtient donc un régime périodique.



Et  $\Phi = 0$  (le décalage à  $t=0s$ ).

On obtient donc  $x(t) = 0.91 \sin(98.72 t)$

### 3.3 Méthode numérique

#### 3.3.1 Méthode d'Euler

##### Présentation de la méthode d'Euler :

La méthode d'Euler est une procédure numérique pour résoudre par approximation des équations différentielles du premier ordre avec une condition initiale. C'est la plus simple des méthodes de résolution numérique des équations différentielles.

La méthode d'Euler permet d'obtenir une valeur approchée d'une valeur d'une fonction en un point lorsque la fonction elle-même n'est pas connue explicitement, mais en connaissant sa valeur en un autre point et sa dérivée.

Elle permet alors également la construction d'une représentation graphique approchée de la fonction étudiée.

##### Soit l'équation différentielle d'ordre 2 suivante :

$$(1) \quad aZ'' + bZ' + cZ = f(t) \quad \text{sur l'intervalle } [t_0, T].$$

Il est nécessaire d'avoir deux conditions initiales puisque c'est une équation différentielle d'ordre 2.

$$\begin{cases} Z(t_0) = Z_0 & \text{(position)} \\ Z'(t_0) = Z'_0 & \text{(vitesse)} \end{cases}$$

Nous devons nous ramener à une équation d'ordre 1 pour pouvoir utiliser la méthode d'Euler ou de Runge Kutta 4.

##### Passage du scalaire en vectoriel :

Nous posons donc :

$$\begin{aligned} y' &= F(t, y) & y'(t) &\in \mathbb{R}^2 \\ y(t_0) &= y_0 & y(t) &\in \mathbb{R}^2 \end{aligned}$$

En effet, nous avons :

$$y : [t_0, t] \mapsto \mathbb{R}^2 \\ t \mapsto \mathbb{R}^2$$

$$y \begin{vmatrix} Z \\ Z' \end{vmatrix} \quad \text{et} \quad y' \begin{vmatrix} Z' \\ Z'' \end{vmatrix}$$

On a donc  $F(t, Z, Z')$ .

Posons  $Z' = u$ .

Nous obtenons :

$$y \begin{vmatrix} Z \\ u \end{vmatrix} \quad \text{et} \quad y' \begin{vmatrix} u \\ u' \end{vmatrix} \quad \text{et} \quad F(t, Z, u).$$

L'équation différentielle devient :

$$(2) au' + bu + cZ = f(t).$$

$$y' = \begin{pmatrix} u \\ u' \end{pmatrix} = F(t, Z, u) = \begin{pmatrix} f_1(t, z, u) \\ f_2(t, z, u) \end{pmatrix}$$

On a donc :  $u' = f_2(t, z, u)$ .

Si  $a \neq 0$ .

Dans (2)  $u' = 1/a (f - cZ - bu) = f_2(t, z, u)$

Et on a :  $u = f_1(t, z, u)$ .

Euler :

Nous devons donc résoudre l'équation différentielle du premier ordre :

$$(2) au' + bu + cZ = f(t).$$

$$\begin{cases} y' = F(t, y) \\ y(t_0) = y_0 \end{cases} \quad \left\{ \begin{array}{ll} y(t_0) = z(t_0) = z_0 & \text{(position)} \\ z'(t_0) = u(t_0) = z'_0 = u_0 & \text{(vitesse)} \end{array} \right.$$

Si on applique la méthode d'Euler, on trouve alors :

$$\begin{cases} y_{i+1} = y_i + hF(t_i, y_i) \\ Y_0 = y(t_0) \end{cases} \quad \text{avec } h=1/N, h \text{ étant le pas.}$$

Or, on est en vectoriel, on obtient donc :

$$\begin{cases} Z_{i+1} = z_i + \frac{hf_1(t_i, z_i, u_i)}{u_i} \\ U_{i+1} = u_i + hf_2(t_i, z_i, u_i) = 1/a (f(t_i) - cz_i - bu_i) \end{cases}$$

Rappel sur la méthode :

Soit une équation différentielle de premier ordre :  $\frac{dy}{dx} = f(x, y)$ .

A partir de la connaissance de la valeur  $y = y_0$  pour une valeur de  $x = x_0$ , on peut calculer la

valeur de  $\frac{dy}{dx}$  en ce point soit  $\left(\frac{dy}{dx}\right)_0$ . La valeur estimée de  $y$  pour  $x = x_0 + dx$  sera prise égale

$$\text{à } y_0 + dy = y_0 + \left(\frac{dy}{dx}\right)_0 dx.$$

C'est une méthode itérative. La valeur  $y_{i+1}$  est déterminée en ajoutant  $\Delta y_i$  à la valeur  $y_i$ .

$$y_{i+1} = y_i + \Delta y_i = y_i + \Delta x \times f(x_i, y_i).$$

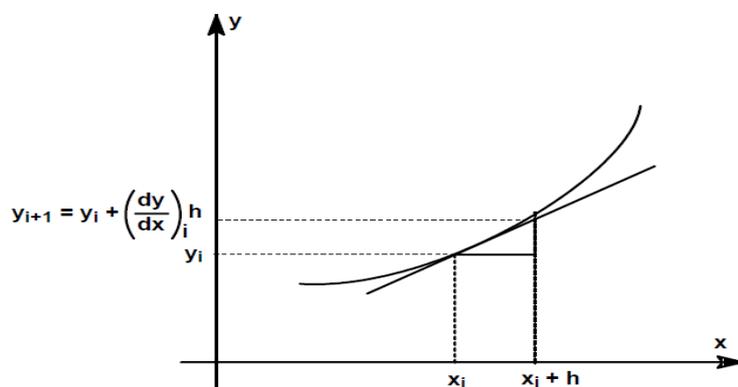
Si on note  $h$  le pas de discrétisation en  $x$ , la méthode d'Euler définit deux suites :

-une première qui définit les valeurs de  $x$  : terme initial :  $x_0$

$$\text{terme de récurrence : } x_{i+1} = x_i + h$$

-une deuxième qui permet d'évaluer les valeurs de  $y$  : terme initial  $y_0$

$$\text{terme de récurrence : } y_{i+1} = y_i + hf(x_i, y_i)$$



Les valeurs estimées seront d'autant plus proches des valeurs exactes que le pas  $h$  est petit.

**Avantage et inconvénient de la méthode d'Euler :**

Avantage	Inconvénient
Cette méthode est plus simple que la méthode RK4 (RK4 à l'ordre 1).	Cette méthode tend moins vite et la précision est moins bonne.

Les méthodes de Runge-Kutta utilisent le même principe que celle d'Euler. Elles apportent cependant une meilleure précision.

Théorème de Cauchy-Lipschitz :

Le théorème de Cauchy-Lipschitz assure l'existence *locale* et l'unicité de la solution d'une équation différentielle.

Enoncé :

Soit une fonction de deux variables :

$$f : U \times I \rightarrow \mathbb{R} \quad \text{où } U \subset \mathbb{R} \quad \text{et } I \subset \mathbb{R}$$

$$(x, t) \rightarrow f(x, t)$$

Considérons l'équation différentielle du premier ordre :

$$(1) \quad \frac{dx(t)}{dt} = f(x(t), t) \quad \text{avec comme condition initiale : } x(t_0) = x_0 \text{ où } t_0 \in I \text{ et } x_0 \in U.$$

Si  $f$  est continue et  $k$ -Lipschitzienne en  $x$ , c'est-à-dire,  $f$  vérifie la *condition de Lipschitz* :

$$\exists k > 0, \forall t \in I, \forall (x, y) \in U^2, |f(x, t) - f(y, t)| \leq k|x - y|$$

alors il existe une unique solution  $x(t)$  de l'équation différentielle définie pour tout  $t \in J, J \subset I$ .  $I$  étant un intervalle centré sur  $t_0$ , vérifiant la condition initiale donnée.

Nous cherchons ici à démontrer l'unicité de l'équation différentielle du TP.

L'équation différentielle est de la forme :  $y''(t) + 2\alpha\omega_0 y'(t) + \omega_0^2 y(t) = \omega_0^2 x(t)$

Après homogénéisation de l'équation, nous avons :  $\frac{1}{\omega_0^2} y''(t) + 2\alpha\frac{1}{\omega_0} y'(t) + y(t) = x(t)$

Or  $\alpha$  est le coefficient d'amortissement et  $\omega_0$  la fréquence.

Avant de chercher une solution de cette équation, il faut démontrer l'existence et l'unicité de cette solution afin de nous assurer que la solution trouvée sera unique.

Étudions le théorème suivant :

**Théorème :** Si  $F$  est continue et lipschitzienne sur  $I^* \mathbb{R}^2$  alors le problème de Cauchy a une solution unique  $y$  sur  $I$  telle que  $y(t_0) = y_0$

La condition de Lipschitz est suffisante mais non nécessaire pour démontrer l'existence et l'unicité de la solution.

Comme vu précédemment,

$$F(t, z, u) = \begin{bmatrix} f_1(t, z, u) \\ f_2(t, z, u) \end{bmatrix} = \begin{bmatrix} u(t, z, u) \\ \omega_0^2(x(t) - z - 2\alpha\frac{1}{\omega_0} u) \end{bmatrix}$$

Afin de démontrer l'unicité de cette fonction, il faut démontrer que les fonctions  $f_1$  et  $f_2$  sont des fonctions lipschitziennes sur  $I^* \mathbb{R}^2$ . Or, si  $F$  est différentiable et si  $J_f(t, y)$  matrice carrée

d'ordre 2, appelée jacobienne, de coefficients  $\frac{\partial f_i}{\partial y_j}(t, y)$  est bornée  $\|J_f\| \leq M_1$  sur  $I^* \mathbb{R}^2$  alors  $f$  est lipschitzienne.

$$\text{Dans notre cas, nous avons } J_f = \begin{bmatrix} \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial z} & \frac{\partial f_2}{\partial u} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial z} & \frac{\partial u}{\partial u} \\ \frac{\partial(\omega_0^2(x(t) - z - 2\alpha\frac{1}{\omega_0} u))}{\partial z} & \frac{\partial(\omega_0^2(x(t) - z - 2\alpha\frac{1}{\omega_0} u))}{\partial u} \end{bmatrix}$$

$$\text{Nous avons donc : } J_f = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\alpha\omega_0 \end{bmatrix}$$

Calculons maintenant la norme 1 de cette matrice afin de démontrer que cette norme est majorée. Nous avons :  $\|J_f\| = \max \sum |j_{ij}| = \max \{ 1, -\omega_0^2 - 2\alpha\omega_0 \} = 1$ .  
Nous pouvons donc dire que la solution est unique.

### Généralisation au rang n :

Dans le cas général, une équation différentielle d'ordre n est du type :

$$(2) \quad \frac{d^n x(t)}{dt^n} = g \left( t, x(t), \frac{dx}{dt}(t), \dots, \frac{d^{n-1}x}{dt^{n-1}}(t) \right)$$

$$y(t) = \left( x(t), \frac{dx}{dt}(t), \dots, \frac{d^{n-1}x}{dt^{n-1}}(t) \right)$$

Cette équation peut se lire comme une équation de variable :

L'équation (2) devient  $\frac{dy}{dt}(t) = f(t, y(t))$

$$f(t, y(t)) = \left( \frac{dx}{dt}(t), \frac{d^2x}{dt^2}(t), \dots, \frac{d^{n-1}x}{dt^{n-1}}(t), g \left( t, x(t), \frac{dx}{dt}, \dots, \frac{d^{n-1}x}{dt^{n-1}}(t) \right) \right)$$

### 3.3.2 Méthode RK4

#### Méthode de résolution Runge-Kutta 4 (RK4)

La méthode Runge-Kutta est une généralisation de la méthode d'Euler-Cauchy.

Elle permet de résoudre les équations différentielles de la forme :  $y'(t)=f(t,y(t))$  avec t compris dans l'intervalle  $[t_0, T]$  et  $y(t_0)=y_0$ . Cet intervalle nous permet de définir le pas h de l'itération, avec la relation :  $h = (T-t_0)/n$

La méthode RK4 utilise plusieurs points intermédiaires pour calculer la valeur de  $y_{i+1}$  à partir de la valeur de  $y_i$ .

Le cas classique est donné par l'équation :  $y_{i+1} = y_i + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)$

Avec :

$$k_1 = hf(t_i, y_i)$$

$$k_2 = hf \left( t_i + \frac{1}{2} h, y_i + \frac{1}{2} h k_1 \right)$$

$$k_3 = hf \left( t_i + \frac{1}{2} h, y_i + \frac{1}{2} h k_2 \right)$$

$$k_4 = hf(t_i + h, y_i + h k_3)$$

Cette méthode se base sur l'approximation des  $y_{i+1}$  par les valeurs actuelles des  $y_i$  et du produit de la taille du pas  $h$  par la pente estimée. La pente est obtenue par moyenne pondérée des pentes  $k_1, k_2, k_3$  et  $k_4$ .

$$pente = \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

On donne un poids plus important aux pentes prises au milieu de l'intervalle.

Comme son nom l'indique, la méthode RK4 est une méthode de résolution d'ordre 4, ce qui veut dire qu'à chaque étape une erreur de l'ordre de  $h^5$  est commise et que l'erreur totale accumulée sur toute la résolution est de l'ordre de  $h^4$ .

Ces formules sont aussi valables pour des fonctions à valeurs vectorielles.

Pour passer du cas scalaire au cas vectoriel on décompose la fonction  $y$  en deux fonctions  $z$  et  $u$ .

On pose :

On a alors :

$$\begin{array}{l} y_i = z_i \\ u_i \end{array} \quad \begin{array}{l} z_{i+1} = z_i + hf(t_i, z_i, u_i) \\ u_{i+1} = u_i + hf(t_i, z_i, u_i) \end{array}$$

On veut désormais une équation en deux dimensions, modélisée avec une matrice comportant  $i$  lignes et  $j$  colonnes.

Pour cela on définit :

$$Y_i = F(t_i, z_i, u_i)$$

avec :

$$F(t_i, z_i, u_i) = \begin{pmatrix} f_1(t_i, z_i, u_i) \\ f_2(t_i, z_i, u_i) \end{pmatrix}$$

Par exemple pour  $k_1$  si l'on suit cette méthode on pose :  $K_1 = hF(t_i + h, y_i + K_3)$

$$K_1 = \begin{bmatrix} k_{11} \\ k_{12} \end{bmatrix}$$

On a alors :

Et on obtient :

$$\begin{array}{l} k_{11} = hf_1(t_i, z_i, u_i) \\ k_{12} = hf_2(t_i, z_i, u_i) \end{array}$$

En généralisant aux  $k^{ij}$  avec  $j=1,2$  :

$$k_{1,j} = hf_j(t_i, z_i, u_i)$$

On peut ensuite procéder de la même manière pour  $k^2$ ,  $k^3$  et  $k^4$ .  
On obtient alors les équations suivantes en vectoriel : (avec  $j=1,2$ )

$$k_{2j} = hf_j \left( t_i + \frac{1}{2} h, z_i + \frac{1}{2} k_{11}, u_i + \frac{1}{2} k_{12} \right)$$

$$k_{3j} = hf_j \left( t_i + \frac{1}{2} h, z_i + \frac{1}{2} k_{21}, u_i + \frac{1}{2} k_{22} \right)$$

$$k_{4j} = hf_j \left( t_i + h, z_i + k_{31}, u_i + k_{32} \right)$$

On a également :

$$y_{i+1} \quad z_{i+1} = z_i + \frac{1}{6} (k_{11} + 2k_{21} + 2k_{31} + k_{41})$$

$$u_{i+1} = u_i + \frac{1}{6} (k_{12} + 2k_{22} + 2k_{32} + k_{42})$$

### 3.4 Partie simulation

Equation différentielle homogène du second ordre du système masse-ressort sans frottements et sans moteur (expression expérimentale) :

$$y(t) = y_0 \cos(\omega_0 t)$$

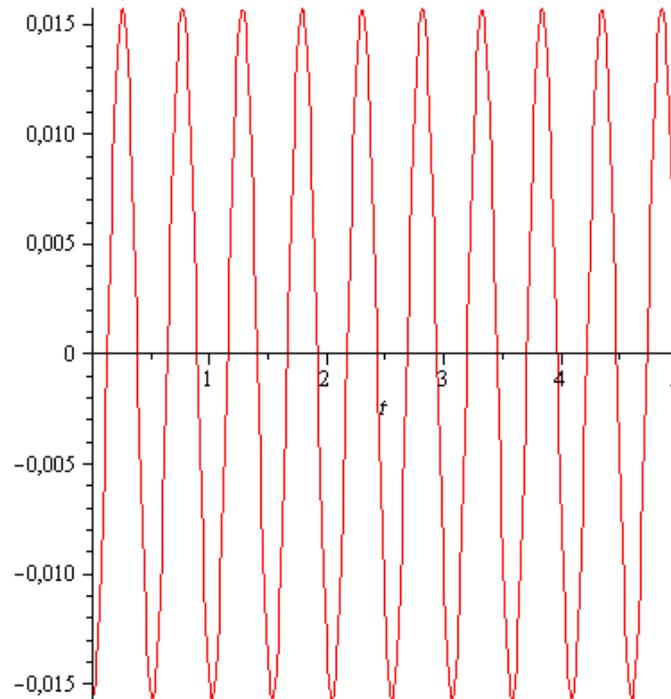
Equation différentielle homogène du second ordre du système masse-ressort sans frottements et sans moteur (expression théorique) :

$$m := \frac{33}{200} \quad k := 24.9 \quad s0 := y(t) = -\frac{157}{10000} \cos\left(\frac{2}{11} \sqrt{4565} t\right)$$

$$\omega_0 := 12.28450613 \quad T := 0.1628067078 \pi$$

Après modélisation nous obtenons : (le programme Maple complet est joint en annexe)

**Régime libre**



Equation différentielle homogène du second ordre du système Masse-Ressort avec frottement.  
Régime pseudo-périodique.

$$m := \frac{33}{200}$$

$$k := 24.9$$

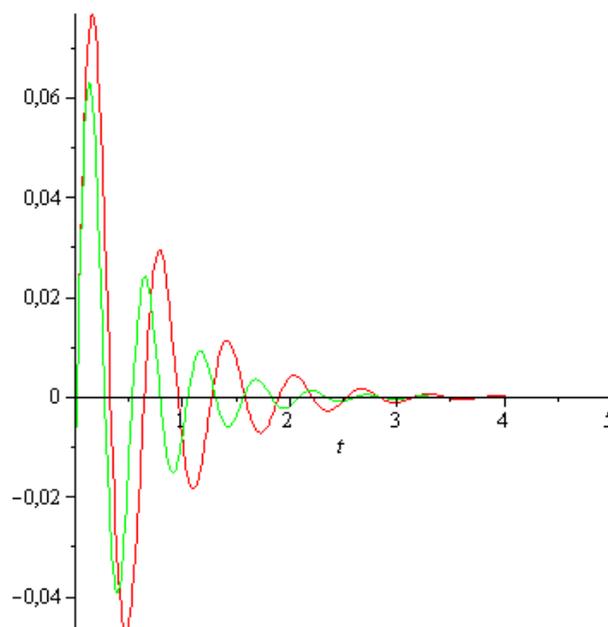
$$\alpha := 0.15$$

$$\omega_0 := 10.13$$

$$h := 0.6080830536$$

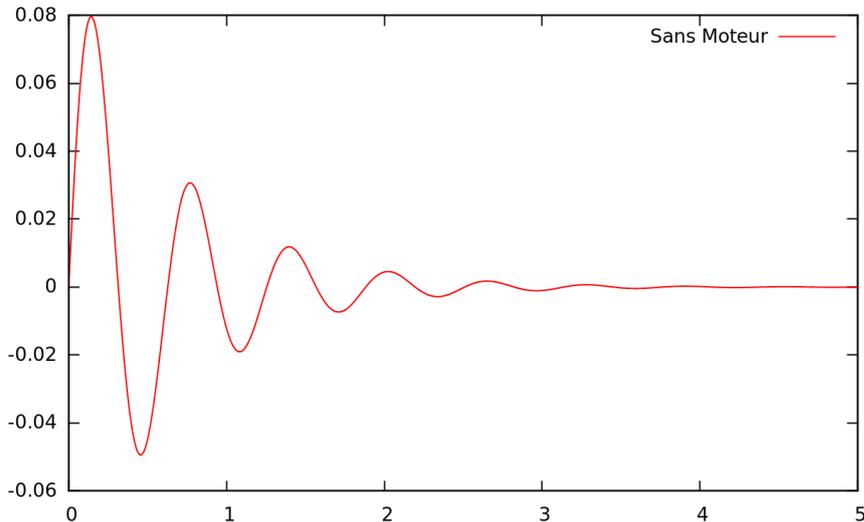
$$\Delta_2 := -16.06423500$$

- La courbe théorique et la courbe expérimentale avec Maple :



La courbe rouge correspond à la courbe expérimentale, et la courbe verte correspond à la courbe théorique. Nous pouvons donc remarquer que les deux courbes ont la même allure, elles sont juste un peu décalées.

- La courbe expérimentale avec C++ :



Nous remarquons que le système masse-ressort est amorti. En effet il n'y a pas de moteur qui permettrait d'entretenir les oscillations. C'est un régime pseudo-périodique.

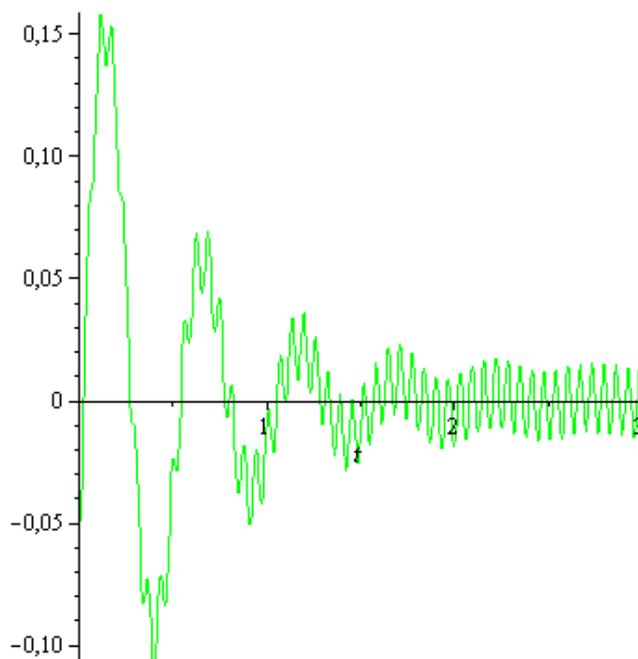
Equation différentielle non homogène du second ordre du système Masse-Ressort avec frottements et avec le moteur

- La courbe théorique avec Maple :

$$m := \frac{33}{200}$$

$$k := 24.9$$

$$h := 0.6080830536$$

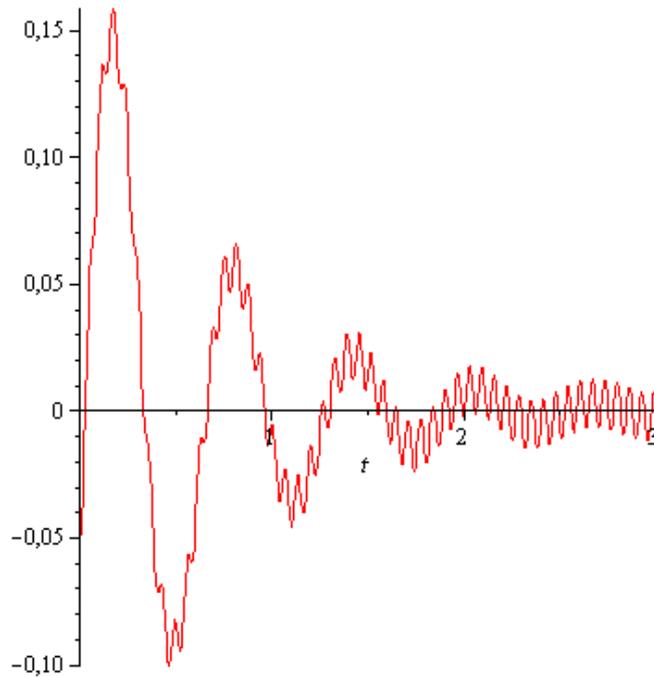


- La courbe expérimentale avec Maple :

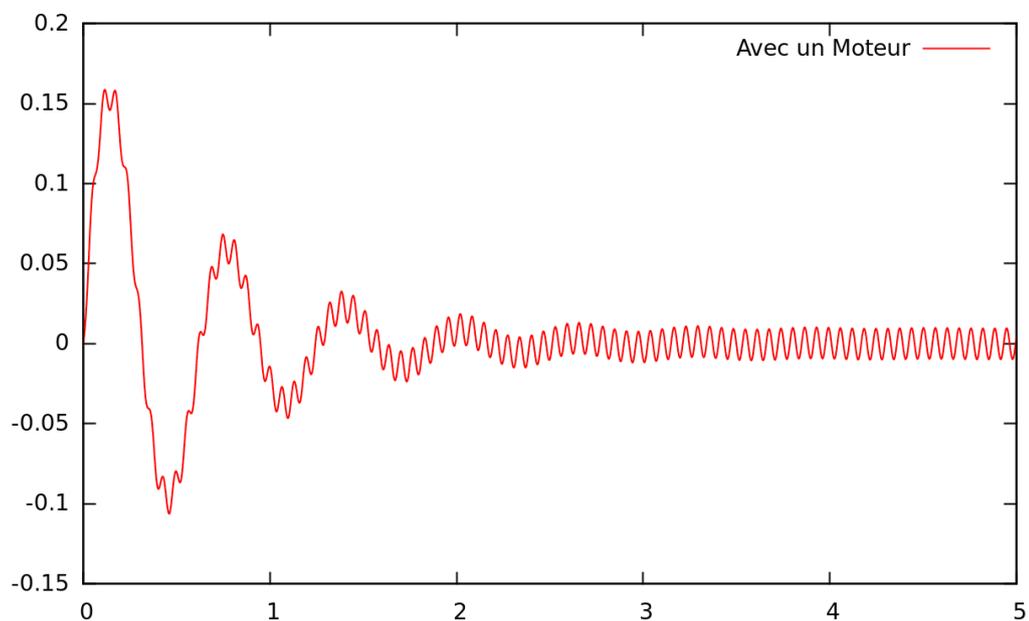
$$\alpha := 0.15$$

$$\omega_0 := 10.13$$

$$x := t \rightarrow 0.91 \sin(98.27 t)$$

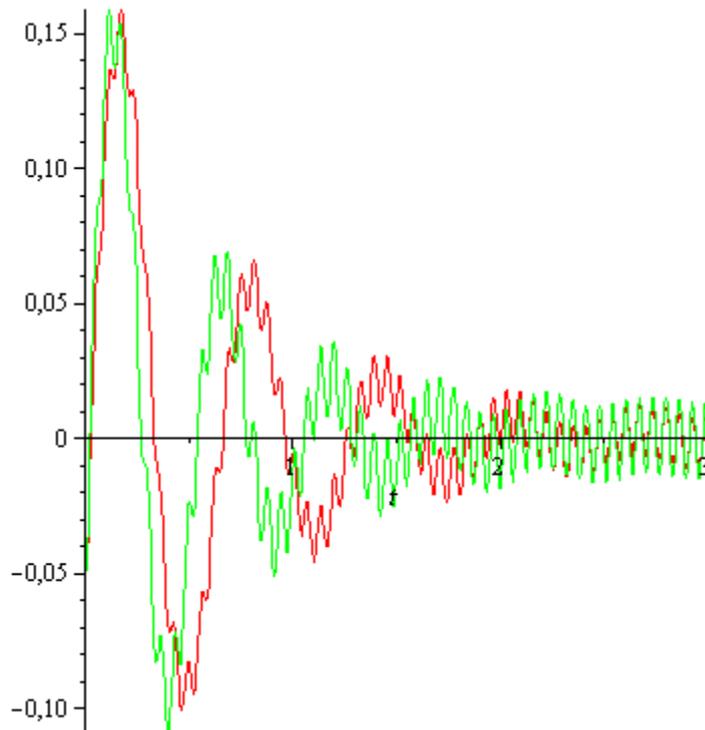


- La courbe expérimentale avec le programme en C++ :



La modélisation montre la présence d'un régime transitoire avant d'atteindre le régime périodique. Ce régime est dû au lâcher de la masse, avant la stabilisation des oscillations par le moteur.

- La courbe théorique et la courbe expérimentale avec Maple :



La courbe rouge correspond à la courbe expérimentale, et la courbe verte correspond à la courbe théorique. Nous pouvons donc remarquer que les deux courbes ont la même allure, elles sont juste un peu décalées.

Pendant notre TP, nous avons obtenu des valeurs expérimentales ( $\alpha$ ,  $\omega_0$ ..).

Grâce au logiciel Maple, nous avons pu modéliser les courbes théoriques et les courbes que nous avons obtenues dans le TP. Puis, nous avons pu les comparer.

Avec le logiciel Maple, nous avons pu calculer directement les solutions exactes des équations différentielles et nous avons modélisé les deux courbes.

Avec le langage C, nous avons modélisé la courbe expérimentale avec la méthode d'approximation de RK4.

Nous pouvons en conclure que nous avons réussi à modéliser les solutions à l'aide de Maple et du langage C.

Pour modéliser les courbes théoriques nous avons utilisé l'équation :

$$z''(t) + \frac{h}{m} z'(t) + \frac{k}{m} z(t) = \frac{k}{m} x(t)$$

Nous avons obtenu  $\omega_0$  théorique par identification avec l'équation suivante :

$$z''(t) + 2\alpha\omega_0 z'(t) + \omega_0^2 z(t) = \omega_0^2 x(t)$$

Nous avons alors la relation :

$$\omega_{0_{th}} = \sqrt{\frac{k}{m}}$$

Grâce au du poids de la masse fournie lors du TP et à la mesure de l'allongement  $k$ , nous avons pu obtenir  $\omega_{0_{th}}$ .

#### 4. Conclusions et perspectives

Tout d'abord, ce projet traitant du système masse-ressort nous a quelque peu déstabilisés. En effet, nous avons dès le début commencé à résoudre l'équation différentielle du système alors que nous pensions étudier le système à partir de la réalisation d'un TP. Nous avons cependant été informés que nous devions reprendre le TP réalisé au troisième semestre. Au fil des séances ce projet nous a alors davantage paru comme un projet mathématique que comme un projet physique.

Ensuite nous avons eu au départ quelques difficultés à nous partager le travail. Étant un groupe de six et ne nous connaissant pas, nous avons beaucoup travaillé sur les mêmes notions au cours des premières semaines. Ce n'est que lorsque nous avons commencé à travailler davantage ensemble que nous avons réussi à mieux nous partager le travail.

Comme nous vous l'avons précisé, ce projet étant très mathématique nous avons dû revoir des notions des semestres précédents, mais nous avons également dû beaucoup nous informer sur les notions que nous ne connaissions pas. Pour la programmation en C++, nous avons aussi revu les notions propres à ce langage, et nous avons recherché les fonctions nécessaires pour la modélisation, que nous n'avions jamais utilisées. Néanmoins nous tenons à préciser que tout ce processus d'apprentissage s'est déroulé dans une bonne ambiance, rendant ainsi notre travail le plus agréable possible.

Au final nous avons acquis de nouvelles connaissances grâce à ce projet, mais nous regrettons son caractère un peu trop abstrait. Si ce projet est maintenu les années suivantes, nous pensons qu'il serait plus intéressant d'aller au-delà du TP du troisième semestre afin de pouvoir observer et analyser d'autres phénomènes physiques.

## 5. Bibliographie

### **Méthode de résolution : Euler-Cauchy**

<http://web2.uqat.ca/lerene/Webcours/gen-0135/manuel/m12-0135.pdf>

<http://rfv.insa-lyon.fr/~jolion/ANUM/node34.html>

<http://math.cmaisonneuve.qc.ca/plantagne/Maple/Euler-Cauchy/Euler-Cauchy3.html>

<http://www.ac-nancy-metz.fr/enseign/physique/divers/MethodNum/euler/Euler.htm>

<http://www.ac-nancy-metz.fr/enseign/physique/divers/MethodNum/Schw-doc/EULER03.pdf>

[http://fr.wikipedia.org/wiki/%C3%89quation\\_diff%C3%A9rentielle\\_d'Euler](http://fr.wikipedia.org/wiki/%C3%89quation_diff%C3%A9rentielle_d'Euler)

<http://www.sosmath.com/diffeq/second/euler/euler.html>

### **Méthode de résolution : Runge-Kutta**

[http://www.ac-nancy-metz.fr/enseign/physique/divers/MethodNum/Schw-doc/RK4\\_02.pdf](http://www.ac-nancy-metz.fr/enseign/physique/divers/MethodNum/Schw-doc/RK4_02.pdf)

[http://www.ac-nancy-metz.fr/enseign/physique/divers/MethodNum/Schw-doc/RK2\\_02.pdf](http://www.ac-nancy-metz.fr/enseign/physique/divers/MethodNum/Schw-doc/RK2_02.pdf)

<http://rfv.insa-lyon.fr/~jolion/ANUM/node36.html>

### **Développement de Taylor**

[http://www.mgi.polymtl.ca/JeanGuerin/Mth2210A/ch1-4\\_developpement\\_taylor.pdf](http://www.mgi.polymtl.ca/JeanGuerin/Mth2210A/ch1-4_developpement_taylor.pdf)

### **Méthode de résolution générale des équations différentielles du 2<sup>nd</sup> degré**

<http://www.iut-bethune.univ-artois.fr/sokol/cours/equadif2/equadif2.html>

[http://fr.wikipedia.org/wiki/%C3%89quation\\_diff%C3%A9rentielle](http://fr.wikipedia.org/wiki/%C3%89quation_diff%C3%A9rentielle)

### **Documentation sur différentes fonctions en C**

<http://www.gnuplot.info/>

### **Documentation sur les matrices**

<http://dms.umontreal.ca/~math1600/6Supplement/Normematricielle-1/Normematricielle-1.html>

## 6. Annexes (non obligatoire – exemples ci-dessous)

### 6.1 Documentation technique

### 6.2 Listings des programmes réalisés

#### Programme Maple

#### Equation différentielle homogène du second ordre du système masse-ressort sans frottements et sans moteur (expression expérimentale)

> restart

> dsolve({diff(y(t), t, t) +  $\omega_0^2$ ·y(t) = 0, D(y)(0) = 0, y(0) =  $y_0$ }, y(t));

$$y(t) = y_0 \cos(\omega_0 t) \quad (1)$$

#### Equation différentielle homogène du second ordre du système masse-ressort sans frottements et sans moteur(expression théorique)

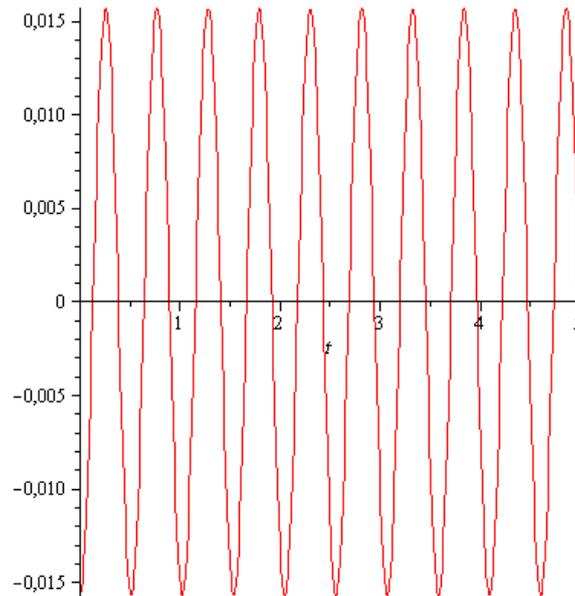
> m :=  $\frac{165}{1000}$ ; k := 24.9; s0 := dsolve({m·diff(y(t), t, t) + k·y(t) = 0, D(y)(0) = 0, y(0) =  $-\frac{1.57}{100}$ }, y(t));

$$\begin{aligned} m &:= \frac{33}{200} \\ k &:= 24.9 \\ s0 := y(t) &= -\frac{157}{10000} \cos\left(\frac{2}{11} \sqrt{4565} t\right) \end{aligned} \quad (2)$$

>  $\omega_0 := \text{sqrt}\left(\frac{k}{m}\right)$ ; evalf(%); T :=  $2 \cdot \frac{\pi}{\omega_0}$ ; evalf(%);

$$\begin{aligned} \omega_0 &:= 12.28450613 \\ &12.28450613 \\ T &:= 0.1628067078 \pi \\ &0.5114723572 \end{aligned} \quad (3)$$

> plot(rhs(s0), t = 0 .. 5);



### Régime libre

Equation différentielle homogène du second ordre du système Masse-Ressort avec frottement . Régime pseudo-périodique.

$$m := \frac{165}{1000}; k := 24.9; \alpha := 0.15; \omega_0 := 10.13; h := 2 \cdot \alpha \cdot \sqrt{\frac{k}{m}} \cdot m; \Delta 2 := h^2 - 4 \cdot k \cdot m;$$

$$s2a := dsolve\left(\left\{m \cdot \frac{d^2}{dt^2}(y(t), t, t) + h \cdot \frac{d}{dt}(y(t), t) + k \cdot y(t) = 0, D(y)(0) = 1, y(0) = \frac{-1.57}{100}\right\}, y(t)\right);$$

$$s2b := dsolve\left(\left\{\frac{d^2}{dt^2}(y(t), t, t) + 2 \cdot \alpha \cdot \omega_0 \cdot \frac{d}{dt}(y(t), t) + \omega_0^2 \cdot y(t) = 0, D(y)(0) = 1, y(0) = \frac{-1.57}{100}\right\}, y(t)\right)$$

$$m := \frac{33}{200}$$

$$k := 24.9$$

$$\alpha := 0.15$$

$$\omega_0 := 10.13$$

$$h := 0.6080830536$$

$$\Delta 2 := -16.06423500$$

$$s2a := y(t)$$

$$= \frac{121383748507}{2535390624988083890000} \sqrt{2788929687486892279} e^{-\frac{23033449}{12500000} t} \sin\left(\frac{1}{137500000}\right)$$

$$\sqrt{2788929687486892279} t) - \frac{157}{10000} e^{-\frac{23033449}{12500000} t} \cos\left(\frac{1}{137500000} \sqrt{2788929687486892279} t\right)$$

$$s2b := y(t) = \frac{19522877}{3960830000} \sqrt{391} e^{-\frac{3039}{2000} t} \sin\left(\frac{1013}{2000} \sqrt{391} t\right) - \frac{157}{10000} e^{-\frac{3039}{2000} t} \cos\left(\frac{1013}{2000} \sqrt{391} t\right)$$

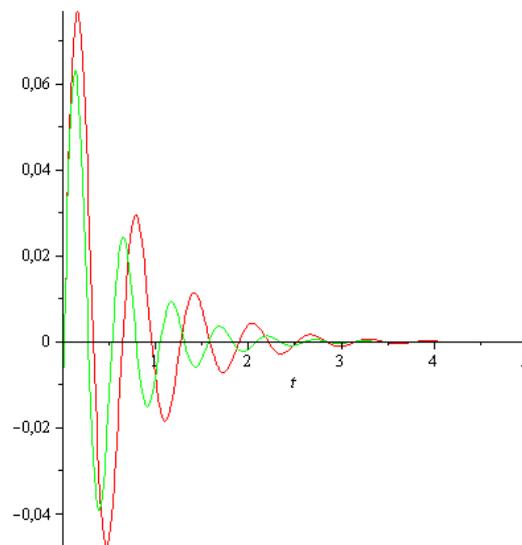
(4)

> with (plots)

[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions, setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot]

(5)

> display ( (plot(rhs(s2a), t=0..5, color = green), plot(rhs(s2b), t=0..5, color = red) ) )



La courbe rouge correspond à la courbe expérimentale, et la courbe verte correspond à la courbe théorique

. On peut donc remarquer que les deux courbes ont la même allure; elles sont juste un peu décalées.

## Equation différentielle non homogène du second ordre du système Masse-Ressort avec frottements et avec le moteur

### la courbe théorique

>  $m := \frac{165}{1000}; k := 24.9; h := 2 \cdot \alpha \cdot \text{sqrt}\left(\frac{k}{m}\right) \cdot m; x := t \rightarrow 0.91 \cdot \sin(98.27 \cdot t);$   
 $s5 := \text{dsolve}\left(\left\{\frac{d^2}{dt^2}(y(t), t, t) + \frac{h}{m} \cdot \frac{d}{dt}(y(t), t) + \frac{k}{m} \cdot y(t) = \frac{k}{m} \cdot x(t), D(y)(0) = 1, y(0) = \frac{-5.3}{100}\right\}, y(t)\right);$

$$m := \frac{33}{200}$$

$$k := 24.9$$

$$h := 0.6080830536$$

$$x := t \rightarrow 0.91 \sin(98.27 t)$$

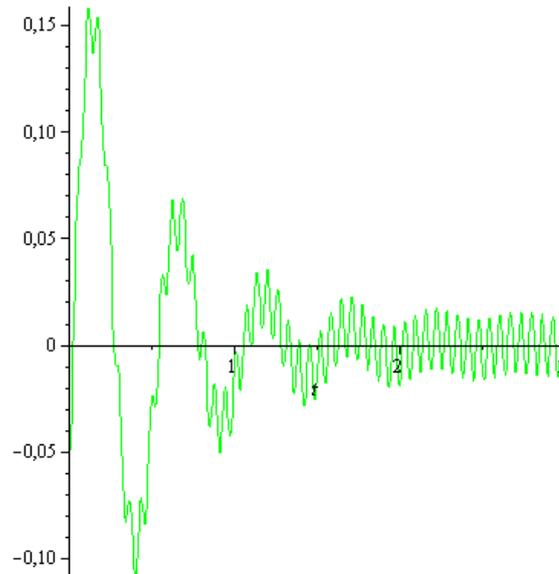
$s5 := y(t)$

$$= \frac{4102273223900161604930514137432425023}{3259157814960724178900837352023956586065459000} e^{-\frac{23033449}{12500000} t} \sin\left(\frac{1}{12500000} \sqrt{23049005680289399} t\right) - \frac{7416555362810457148570326873}{141401232667829461160859941000} e^{-\frac{23033449}{12500000} t} \cos\left(\frac{1}{12500000} \sqrt{23049005680289399} t\right) - \frac{77709968584504292955250}{141401232667829461160859941} \cos\left(\frac{9827}{100} t\right) - \frac{2039757130564578814953125}{141401232667829461160859941} \sin\left(\frac{9827}{100} t\right) \quad (6)$$

>  $\text{evalf}(\%);$

$$y(t) = 0.1910932249 e^{-1.842675920 t} \sin(12.14551919 t) - 0.05245042934 e^{-1.842675920 t} \cos(12.14551919 t) - 0.0005495706587 \cos(98.27000000 t) - 0.01442531364 \sin(98.27000000 t) \quad (7)$$

$\text{plot}(\text{rhs}(s5), t = 0..3, \text{color} = \text{green});$



> la courbe expérimentale

>  $\alpha := 0.15; \omega_0 := 10.13; x := t \rightarrow 0.91 \cdot \sin(98.27 \cdot t);$

$s6 := dsolve\left(\left\{\frac{d^2y}{dt^2}(y(t), t, t) + 2 \cdot \alpha \cdot \omega_0 \cdot \frac{dy}{dt}(y(t), t) + \omega_0^2 \cdot y(t) = \omega_0^2 \cdot x(t), D(y)(0) = 1, y(0) = \frac{-5.3}{100}\right\}, y(t)\right);$

$\alpha := 0.15$

$\omega_0 := 10.13$

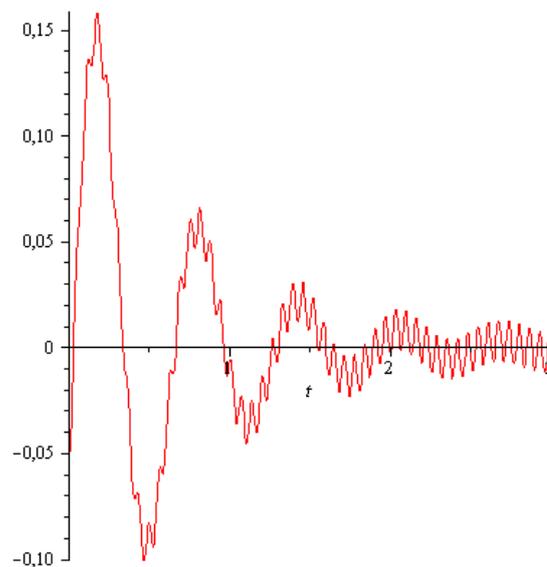
$x := t \rightarrow 0.91 \sin(98.27 t)$

$$s6 := y(t) = \frac{1144903099150966191257899}{120640660801710185076249000} \sqrt{391} e^{-\frac{3039}{2000} t} \sin\left(\frac{1013}{2000} \sqrt{391} t\right) - \frac{16050008728519645259}{304584293700336003000} e^{-\frac{3039}{2000} t} \cos\left(\frac{1013}{2000} \sqrt{391} t\right) - \frac{929588375981629}{3045842937003360030} \cos\left(\frac{9827}{100} t\right) - \frac{2974002687881680}{304584293700336003} \sin\left(\frac{9827}{100} t\right) \quad (8)$$

>  $evalf(\%);$

$$y(t) = 0.1876564094 e^{-1.519500000 t} \sin(10.01538914 t) - 0.05269480095 e^{-1.519500000 t} \cos(10.01538914 t) - 0.0003051990517 \cos(98.27000000 t) - 0.009764136725 \sin(98.27000000 t) \quad (9)$$

>  $plot(rhs(s6), t = 0..3, color = red);$



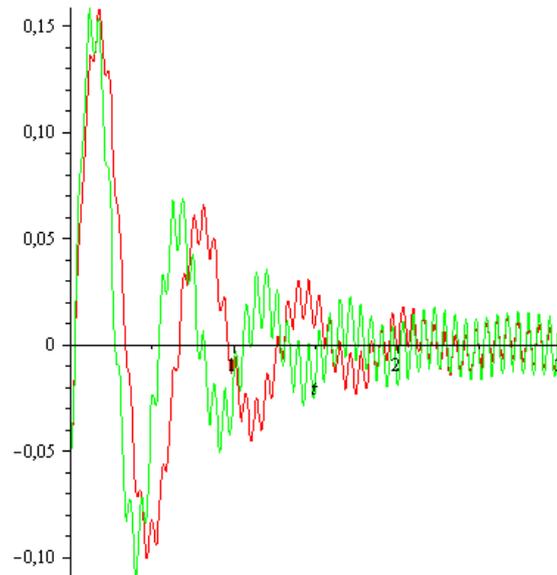
>  $with(plots)$

[*animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicatplot, implicatplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra supported, polyhedraplot,*

(10)

`rootlocus, semilogplot, setcolors, setoptions, setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot]`

`> display( (plot(rhs(s5), t=0..3, color = green), plot(rhs(s6), t=0..3, color = red)) )`



*> La courbe rouge correspond à la courbe expérimentale, et la courbe verte correspond à la courbe théorique . On peut donc remarquer que les deux courbes ont la même allure, elles sont juste un peu décalées.*

## Programmation de Runge Kutta 4 en C++

Nous avons adapté le programme en C++ que le professeur nous a donné aux valeurs expérimentales, ça veut dire que  $\alpha = 0,15$ ,  $\omega_0 = 10,13$ ,  $x(t) = 0,91 \sin(98,72 t)$ , la position initiale=0m, la vitesse initiale=1m/s.

L'équation différentielle originale  $y'' + 2\alpha\omega_0 y' + (\omega_0)^2 y = (\omega_0)^2 x(t)$  devient  $y'' + 3,039y' + 102,6169y = 93,3814 * \sin(98,72 t)$ .

Dans le cas où il y a un moteur, l'équation est bien

$$y'' + 3,039y' + 102,6169y = 93,3814 * \sin(98,72 t) ;$$

Dans l'autre cas où il n'y en a pas, l'équation est  $y'' + 3,039y' + 102,6169y = 0$ .

Programme :

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
```

```
double rK4(double x, double yPos, double yVis, double pas, int k);
double renvoiVitesse(double x, double yPos, double yVis);
double renvoiAcceleration(double x, double yPos, double yVis);
```

```
int main ( int argc, char** argv )
```

```
{
    int i, j;

    double PAS = 0.0005; /*Permet de changer le pas*/

    double yPosition = 0.; /*Conditions initiales*/
    double yVitesse = 1.;

    double yPositionSave=0.;

    double x = 0.;
```

```
double valeurTracer [100000][2]; /*Taleau contenant les valeurs de la solutions de
l'équation differentielle.*/
```

```
char* nomdefichier[6];
nomdefichier[0] = "RK4v1Application";
/*nomdefichier[1] = &quot;RK4v2&quot;;*/
```

```
valeurTracer[0][0] = yPosition;
valeurTracer[0][1] = x;
```

```
for(j=0; j<1; j++) /*Cette boucle sert si on souhaite résoudre plusieurs équations
différentielles -- Ici nous avons qu'un application physique donc j= 1*/
```

```
{
    // on intialise le fichier qui prend les valeurs
```

```
FILE* fichier = NULL;
fichier = fopen(nomdefichier[j], "w+"); // suppression du contenu au préalable ! //
fprintf(fichier, "%f\t%f \n", x, yPosition);
```

```
if(fichier!=NULL)
```

```
{
    for(i=1; i<=10000; i++) /*Nombre de points ? calculer de la solution de
l'équation différentielle par l'ordinateur*/
```

```
{
    x = x + PAS;
    yPositionSave = yPosition;
    yPosition = rK4(x, yPosition, yVitesse, PAS, 0);
    yVitesse = rK4(x, yPositionSave, yVitesse, PAS, 1);
```

```
valeurTracer[i][0] = yPosition;
valeurTracer[i][1] = x;
```

```
fprintf(fichier, "%f\t%f \n", x, yPosition);
printf ("x = %f\t|tf(x) = %f", x , yPosition);
```

```
//Permet d'écrire les valeurs de la solution sur la console //
```

```
}
```

```
    fclose(fichier);
        yPosition = 0.;
        x = 0.;
        printf("Ecriture dans %s termine.", nomdefichier[j]);
    }
else
    printf("Echec de l'ouverture du fichier.");
}
return 0;
}
```

```
double rK4(double x, double yPos, double yVis, double pas, int k)
{
    double renvoiVitesse(double x, double yPos, double yVis);
    double renvoiAcceleration(double x, double yPos, double yVis);

    double k1x, k1y, k2x, k2y, k3x, k3y, k4x, k4y;

    double x2, yp2, yv2;
    double x3, yp3, yv3;
    double x4, yp4, yv4;

    //Calcul de K1
    k1x = pas * renvoiVitesse(x, yPos, yVis);
    k1y = pas * renvoiAcceleration(x, yPos, yVis);

    //Calcul de K2
    x2=x + 0.5*pas;
    yp2=yPos + 0.5*k1x;
    yv2=yVis + 0.5*k1y;

    k2x = pas * renvoiVitesse(x2, yp2, yv2);
    k2y = pas * renvoiAcceleration(x2, yp2, yv2);
```

```

//Calcul de K3
x3=x + 0.5*pas;
yp3=yPos + 0.5*k2x;
yv3=yVis + 0.5*k2y;
k3x = pas * renvoiVitesse(x3, yp3, yv3);
k3y = pas * renvoiAcceleration(x3, yp3, yv3);

//Calcul de K4
x4=x + pas;
yp4=yPos + k3x;
yv4=yVis + k3y;
k4x = pas * renvoiVitesse(x4, yp4, yv4);
k4y = pas * renvoiAcceleration(x4, yp4, yv4);

//On calcule yi+1
if(k==0)
{
    return yPos + ((k1x + 2*k2x + 2*k3x + k4x)/6); //Renvoie la position
}
else
{
    return yVis + ((k1y + 2*k2y + 2*k3y + k4y)/6); //Renvoie la vitesse
}
}
double renvoiVitesse(double x, double yPos, double yVis)
{
    return yVis;
}
double renvoiAcceleration(double x, double yPos, double yVis)
{
    return 93.3814*sin(98.72*x)- 3.039*yVis - 102.6169*yPos;
} //équation différentielle

```

Pour les deux cas différents, on change l'équation différentielle à la fin du programme.