Modelisation des systèmes périodiques

Cavazzoni Christophe

2024-2025 - Institut Champollion

Table des matières

Chapitre 1

Introduction

1.0.1 Definitions

Dans ce rapport, nous allons nous pencher sur l'étude et la modélisation de **systèmes dynamiques** particuliers appelés **systèmes périodiques**. Tout d'abord nous définissons les concepts en jeu.

- On appelle **système dynamique** un ensemble d'éléments qui intéragissent entre eux et dont l'évolution dans le temps est décrite par une loi, qui pourra être soit physique, soit chimique, etc... Nous nous intéresseront au cas particulier des systèmes périodiques qui sont des sytèmes dynamiques régis par les lois physiques élémentaires, et dont les interactions varient **continument**, ie ce sont des **systèmes dynamiques à temps continu.**
- On appelle **systèmes périodiques** un système dynamique tel qu'il évolue de part et d'autres d'un point d'équilibre donné.

1.0.2 Cas étudiés

Nous étudiront principalement 2 cas de sytèmes périodiques :

- Le cas du **système masse-ressort**.
- Le cas du **pendule simple non-linéaire**.

L'objectif étant d'identifier les paramêtres du système, modéliser leur mouvement via des équations différentielles, et étudier ces équations et les propriétés de leurs solutions (portrait de phase, points d'équilibre, solutions périodiques et si le temps le permet, stabilité.).

1.0.3 Techniques de modélisation utilisées

Les systèmes que l'on veut modéliser sont des systèmes basées sur les lois de la physique, on utilisera donc principalement les concepts suivants pour modéliser les phénomènes :

- Le **principe fondamental de la dynamique** qui nous permettra d'établir que $a = \frac{1}{m} \sum \vec{F}$.
- On devra donc souvent effectuer le **bilan des forces** qui s'appliquent à notre objet.
- Certains objets on des propriétés particulières qui demanderont d'autres concepts physiques (ressorts notamment).

Ces techniques nous permettront alors de nous ramener à l'étude d'une équation différentielle de la forme :

$$y''(t) = F(t, y(t), y'(t))$$

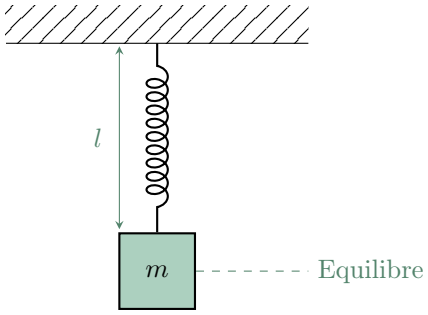
Et on pourra alors utiliser l'arsenal mathématique à notre disposition pour l'étudier.

Chapitre 2

Systême masse-ressort

2.0.1 Modélisation

On considère un objet de masse m relié à un mur par un ressort de longueur l, tous deux des réels positifs. On considère aussi que la masse est à l'équilibre, voici un schéma qui illustre la situation :



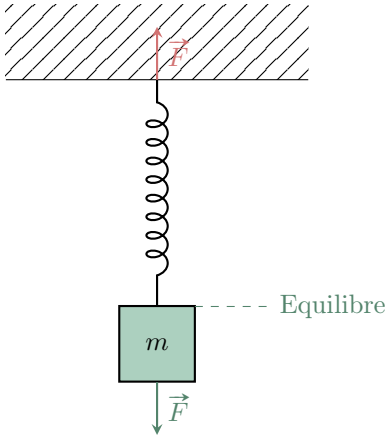
Le système qu'on cherche à modéliser est le mouvement de la masse si on déplace la masse en dehors du point d'équilibre. On peut tout d'abord identifier les **paramétres du modèle** :

- La masse de l'objet.
- La longeur du ressort.
- La force de gravité.

Aussi, on peut remarque que si quand la masse est à l'équilibre et si on note l_0 la longueur du ressort à vide, alors la force de gravité est proportionelle à l'élongation du ressort, ie on a :

$$mg = k(l - l_0)$$

On appelle alors k la **constante de raideur** du ressort. Notons y la fonction qui donne la position au temps t du centre de la masse. Pour modéliser le mouvement de la masse si on la sort de la position d'équilibre, on doit modéliser les forces en action. D'aprés la 3em loi de Newton, on a la situation suivante :



En effet grâce à la relation de proportionnalité ci-dessus, la gravité n'a pas d'impact sur le mouvement du ressort, elle ne fait que déplacer le point d'équilibre vers le bas. En outre, la force qu'exercerce le ressort dans la direction de l'équilibre est donc linéairement proportionelle à la distance avec l'équilibre, et donc d'aprés le **principe fondamental de la dynamique**, ie on a :

$$\vec{a}(t) = \frac{1}{m}\sum \vec{F}(t)$$

Et donc on obtient en faisant le bilan des forces, dont il ne resulte qu'une force proportionnelle à l'élongation :

$$\boxed{y''(t) = -\frac{k}{m}y(t)}$$

Cette équation modélise donc bien le mouvement de la masse sur le ressort.

2.0.2 Résolution

L'équation précédente et facilement résoluble, en effet c'est une equation linéaire et on peut remarquer facilement que les fonctions solutions sont de la forme :

$$y(t) = c_1\cos(\omega t) + c_2\sin(\omega t)\ ;\ \omega = \sqrt{\frac{k}{m}}$$

Où c_1, c_2 sont les constantes d'intégration à déterminer, on a :

— On a directement $y(0) = c_1$ donc c_1 est la position initale de la masse.

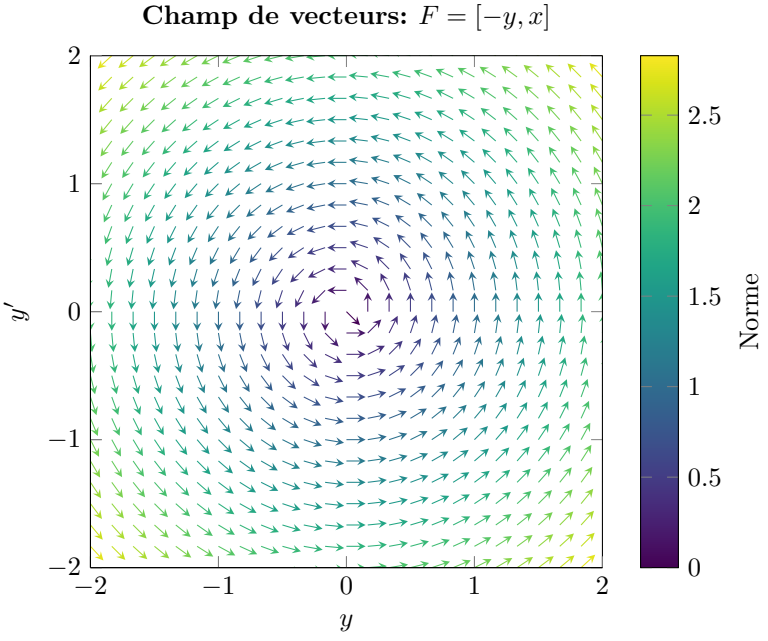
— On a en dérivant que $y'(0) = \omega c_2$, donc $c_2 = \frac{v_0}{\omega}$, c'est une constante proportionelle à la vitesse initiale.

Finalement on a la solution donnée par :

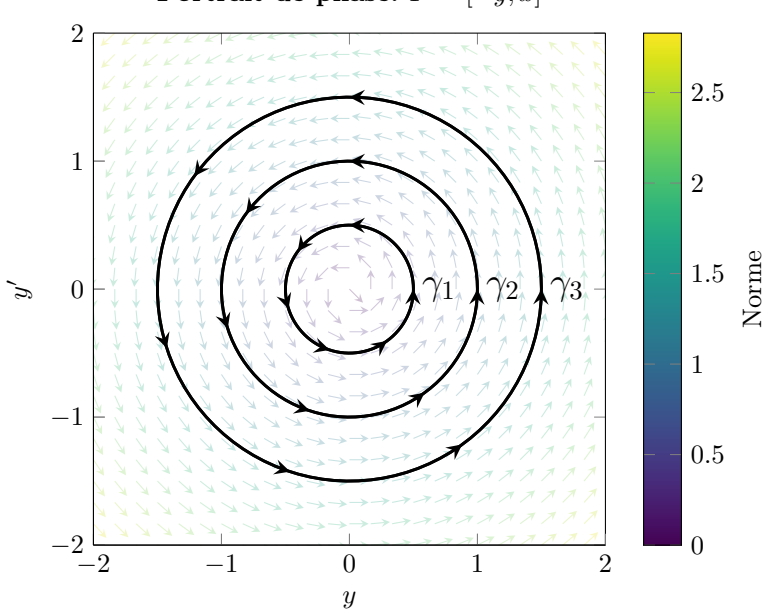
$$\boxed{y(t) = y_0\cos(\omega t) + \frac{v_0}{\omega}\sin(\omega t)\ ;\ \omega = \sqrt{\frac{k}{m}}}$$

2.0.3 Champ de vecteur associé et portrait de phase

Si on ramène l'équation différentielle d'ordre 2 à une équation différentielle vectorielle d'ordre 1, on obtient que la position de la masse est exactement caractérisée par sa position et sa vitesse, ie l'espace des phases est le plan repéré par (y, y'), et la dynamique est caractérisée par le champ de vecteur associé trouvé ci-dessus. Par exemple pour $k = 1$, on a le champ de vecteur suivant :



Traçons quelques trajectoires possibles du système, pour les conditions initiales $(y_0, v_0) = (\frac{i}{2}, 0)$ pour $i \in \{1, 2, 3\}$, on trouve les trajectoires suivantes :



On remarque donc la propriété suivante :

Toutes les trajectoires sont périodiques et le seul point d'équilibre correspond à la masse au repos.

2.0.4 Ajout d'un terme d'amortissement

On essaie maintenant de rendre le modèle plus réaliste en ajoutant la contribution des frottements de l'air. Alors, en première approximation, on peut imaginer rajouter un terme proportionnel à la vitesse qui ralentit la masse, ie on considère la nouvelle équation suivant, pour $\lambda > 0$ un coefficient de frottements :

$$y''(t) = -\frac{k}{m}y(t) - \lambda y'(t)$$

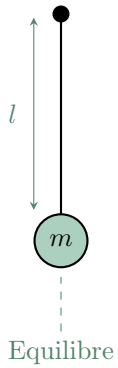
Et la c'est la merde, 3 cas, régime apériodique, critique, et pseudo-périodique ... dépendant du Δ de l'équation caractéristique de l'équation (C LE POLYNOME CARACTERISTIQUE DE LA MATRICE), le dernier ok, les autres sont relou + technique de resolution d'eqdiff relou aussi ...

Chapitre 3

Pendule simple

3.0.1 Modélisation

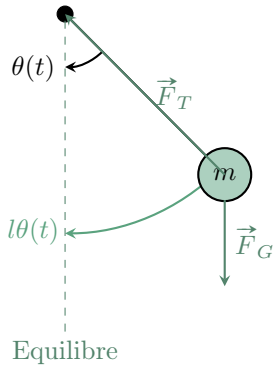
On considère un objet de masse m relié à un point par une corde rigide de longueur l, tous deux des réels positifs. On considère aussi que la masse est à l'équilibre, voici un schéma qui illustre la situation :



Le système qu'on cherche à modéliser est le mouvement de la masse si on déplace la masse en dehors du point d'équilibre. On peut tout d'abord identifier les **paramétres du modèle** :

- La masse de l'objet.
- La longeur du ressort.
- La constante de gravité.

On note alors θ la fonction qui donne la position angulaire en radians au temps t du centre de la masse. Pour modéliser le mouvement de la masse si on la sort de la position d'équilibre, on doit modéliser les forces en action. D'aprés la 3em loi de Newton, on a la situation suivante :



En utilisant les faits géométriques que je n'ai pas encore écrit et le **principe fondamental de la dynamique**, on a :

$$\vec{a}(t) = \frac{1}{t} \sum \vec{F}(t)$$

Et donc finalement la composante d'acceleration dans le sens de la course du pendule est donnée par :

$$\boxed{\theta''(t) = -\frac{g}{l} \sin(\theta)}$$

3.0.2 Résolution

Malheureusement cette équation différentielle est **trés difficile** à résoudre analytiquement, le seul cas où la résolution est possible est pour θ_0 petit, alors on a :

$$\sin(\theta) \sim \theta$$

Et donc on se ramène au cas simple d'oscillateur harmonique :

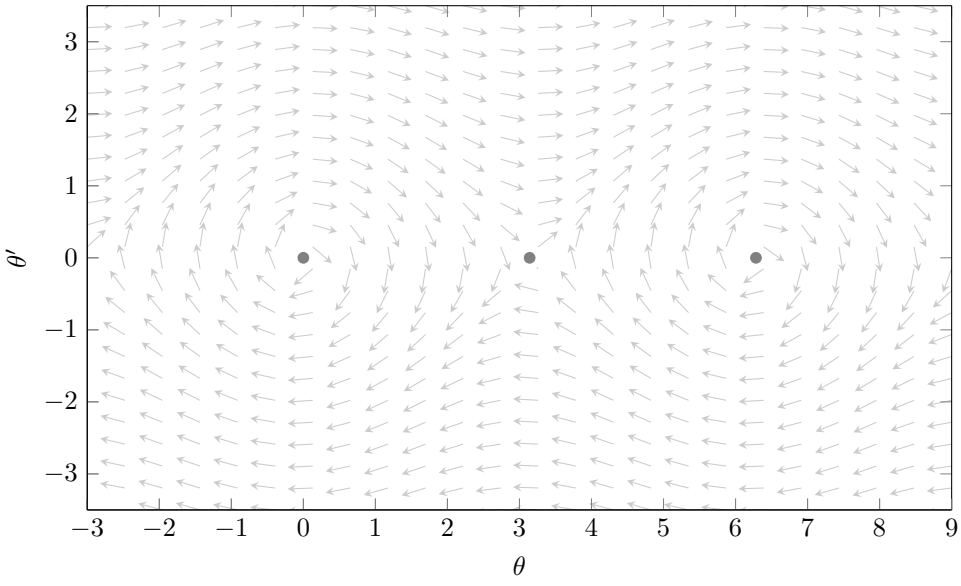
$$\theta''(t) = -\frac{g}{l}\theta$$

Néanmoins, on peut explorer le problème non-linéaire via les méthodes numériques, ce qu'on verra dans la section suivante.

3.0.3 Champ de vecteur associé et portrait de phase

Si on ramène l'équation différentielle d'ordre 2 à une équation différentielle vectorielle d'ordre 1, on obtient que la position de la masse est exactement caractérisée par sa position angulaire et sa vitesse angulaire, ie l'espace des phases est le plan repéré par (θ, θ'), et la dynamique est caractérisée par le champ de vecteur associé trouvé ci-dessus. Par exemple [1] pour $g = 1$ et $l = 1$, on a le champ de vecteur suivant :

Champ de vecteurs: $F = [y, -\sin(x)]$



1. Ici on utilise $g = 1$ pour faciliter la lisibilité du champ de vecteur.

3.0.4 Points remarquables

On remarque graphiquement des types de points d'équilibres :

— Les points où l'angle et la vitesse sont nuls, c'est point sont évidemment stables. Ils correspondent à la situation où le pendule est à l'arrêt.

— Les points "exotiques" où l'angle est exactement égal à π et où la vitesse est nulle. Ils correspondent à la situation où le pendule en en équilibre à l'envers. Ces points sont instables.

Chapitre 4

Implémentations & traitement des données

Dans cette partie, je partage les différent moyens informatiques utilisés pour réaliser ce projet.

4.0.1 Méthode d'Euler

Pour calculer les différents portraits de phase, il a fallu être capable de générer les points des courbes solutions d'équations, différentielles, pour cela, on utilise la méthode bien connue d'Euler, qu'on généralise en une fonction qui à tout **champ de vecteur** et **conditions initiales** retourne une courbe intégrale sur un intervalle de temps donné, voici le code en question implémenté en Sagemaths :

Listing 4.1 – Méthode d'Euler générale

```
def eulerMethod(vField, time, stepSize, *initialConditions):
# Prends un champ de vecteur et approxime une ligne de flot pour des
# conditions initiales donnees et une duree t

    n = floor(time / stepSize)
    # Nombre d'iterations necessaire pour completer l'intervalle de temps

    L = []
    P0 = vector(initialConditions)
    for i in range(n):
        L.append(P0)
        P0 = vector(tuple(approx(P0[i]) for i in range(len(initialConditions))))
        P0 = P0 + stepSize*vField(*P0)
        # gamma(t0 + h) ~ gamma(t0) + hgamma'(t0) = gamma(t0) + hF(gamma(t0))

    return L # Retourne une liste de points
```

4.0.2 Traitement et affichage des données

Par la suite, on va utiliser cet algorithme pour générer les 8 courbes intégrales du pendule (seul exemple non trivial), les formater et les envoyer sous forme de fichier data dans le dossier LaTeX qui s'occupera alors de l'affichage :

Listing 4.2 – Méthode d'Euler générale

```
# 2D Pendulum
def computePendulumData():
    # Precision presque ideale: 0.0025
    acc = 0.0025
    # Trajectoires aperiodiques
    aperiodicFlowPoints1 = eulerMethod(H, 12.5, acc, -3, 2) # Longueur d'arc 12.5
    aperiodicFlowPoints2 = eulerMethod(H, 13, acc, -3, 1) # Longueur d'arc 13
    aperiodicFlowPoints3 = eulerMethod(H, 13, acc, 9, -1) # Longueur d'arc 13
    aperiodicFlowPoints4 = eulerMethod(H, 12.5, acc, 9, -2) # Longueur d'arc 12.5

    # Trajectoires periodiques
    periodicFlowPoints1 = eulerMethod(H, 12, acc, 2.14, 0) # Longueur d'arc 12
    periodicFlowPoints2 = eulerMethod(H, 7, acc, 1.14, 0) # Longueur d'arc 7
    periodicFlowPoints3 = eulerMethod(H, 12, acc, 4.14, 0) # Longueur d'arc 12
    periodicFlowPoints4 = eulerMethod(H, 7, acc, 5.14, 0) # Longueur d'arc 7
    return [
        aperiodicFlowPoints1, aperiodicFlowPoints2,
        aperiodicFlowPoints3, aperiodicFlowPoints4,
        periodicFlowPoints1, periodicFlowPoints2,
        periodicFlowPoints3, periodicFlowPoints4
    ]

def sendFormattedPendulumData(L):
    pointListToPgfPlot(L[0], 'C:\\.....\\data\\aperiodic1.dat')
    pointListToPgfPlot(L[1], 'C:\\.....\\data\\aperiodic2.dat')
    pointListToPgfPlot(L[2], 'C:\\.....\\data\\aperiodic3.dat')
    pointListToPgfPlot(L[3], 'C:\\.....\\data\\aperiodic4.dat')

    pointListToPgfPlot(L[4], 'C:\\.....\\data\\periodic1.dat')
    pointListToPgfPlot(L[5], 'C:\\.....\\data\\periodic2.dat')
    pointListToPgfPlot(L[6], 'C:\\.....\\data\\periodic3.dat')
    pointListToPgfPlot(L[7], 'C:\\.....\\data\\periodic4.dat')

sendFormattedPendulumData(computePendulumData()) # Envoie les donnees
```

Finalement, une fois ceci fait, la package TikZ récupère directement les données dans le dossier "data" et trace la courbe sur le graphique du champ de vecteur. Cette démarche de fournir à LaTeX des données précalculées à l'avance était nécessaire car LaTeX étant un langage de typographie, il ne sait pas calculer (ou alors que des calculs trés simple) et il ne sait bien sûr pas résoudre ou tracer les solutions des équations différentielles considérées.