Institutional Repository - Research Portal
Dépôt Institutionnel - Portail de la Recherche
researchportal.unamur.be
University of Namur

THESIS / THÈSE

MASTER EN SCIENCES MATHÉMATIQUES À FINALITÉ DIDACTIQUE

Identification d'équations aux dérivées partielles

PIETQUIN, Julien

Award date:
2022

Awarding institution:
Universite de Namur

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 02. May. 2026

Université de Namur

Faculté des sciences

Mémoire
Identification d'équations aux dérivées partielles

Mémoire présenté pour l'obtention

du grade académique de master en "Sciences mathématiques"

Julien PIETQUIN

Juin 2022

Remerciements

Ce travail conclut mes cinq années d'études en Sciences mathématiques à l'Université de Namur. Ce fut un parcours riche en apprentissages et surtout en rencontres. Je tiens à remercier tous les étudiants que j'ai croisés sur mon chemin et avec qui nous avons beaucoup échangé.

Je remercie tous mes professeurs, assistants, tous les intervenants de l'Université de Namur qui m'ont accompagné et tout particulièrement *Monsieur le Professeur Alexandre Mauroy* qui a cru en mes capacités, m'a soutenu et encadré dans la réalisation de ce travail.

Résumé

Ce travail a pour objectif d'améliorer la méthode d'identification des équations aux dérivées partielles de fonctions de base connues à l'aide de l'opérateur de Koopman. En effet, cette méthode nécessite le choix d'une fonction de poids. Nous commençons par comparer cette méthode à d'autres méthodes existantes et clarifier la notion d'opérateur de Koopman dans le cadre des équations aux dérivées partielles ainsi que des méthodes d'approximations de cet opérateur et du générateur de Lie en dimension finie. Constatant qu'un choix automatique de fonction de poids est nécessaire pour le problème d'identification, des méthodes d'optimisation sont envisagées et comparées avec des tentatives d'améliorations en terme de variabilité et de rapidité de calculs. Ces analyses sont réalisées dans le cas des données non bruitées ainsi que dans le cas des données bruitées.

Mots clés

identification, équation aux dérivées partielles, système, trajectoire, régression, moindres carrés, Ridge, LASSO, Gurevich, OFR, différences finies, erreur de troncature, erreur d'arrondis, flot, Koopman, Lie, semi-groupe, non-linéarité, dérivée de Gâteaux, fonctionnelle, fonctionnelles de base, fonction de poids, pseudo-inverse de Moore-Penrose, EDMD, validation croisée, optimisation, Nelder-Mead, gradient stochastique, interpolation de Lagrange, condition d'arrêt.

Abstract

The objective of this work is to improve the method of identification of partial differential equations with known basis functions using the Koopman operator. Indeed, this method requires the choice of a weight function. We start by comparing this method to other existing methods and clarifying the notion of Koopman operator in the context of partial differential equations as well as the methods of approximation of this operator and the Lie generator in finite dimension. Noting that an automatic choice of weight function is necessary for the identification problem, optimization methods are considered and compared with attempts at improvements in terms of variability and speed of calculation. These analyses are carried out in the case of noiseless data as well as in the case of noisy data.

Key words

identification, partial differential equation, system, trajectory, regression, least squares, Ridge, LASSO, Gurevich, OFR, finite differences, truncation error, round-off error, flow, Koopman, Lie, semigroup, nonlinearity, Gâteaux derivative, functional, basis functionals, weighting function, Moore-Penrose pseudoinverse, EDMD, cross-validation, optimization, Nelder-Mead, stochastic gradient, Lagrange interpolation, break condition.

Table des matières

Introduction

L'emploi à un système dynamique du semi-groupe de Koopman défini depuis le début des années trente [9] (dont le caractère fortement continu dépend de l'espace considéré) prend de l'essor depuis moins de vingt ans [13]. En effet, cet opérateur de Koopman, également nommé opérateur de composition, permet de linéariser le système dynamique sous-jacent en passant de l'espace initial d'états à un espace fonctionnel de dimension infinie. Il est donc très intéressant à appliquer à des équations différentielles non nécessairement linéaires à identifier en fonction de trajectoires connues. Ce dernier procédé permet notamment de comprendre un système dynamique et de prédire ses trajectoires. L'application de l'opérateur de Koopman à des équations différentielles ordinaires a récemment été employée pour identifier ces dernières [12]. Ce procédé a également été employé dans le cas des équations aux dérivées partielles [11] afin d'élaborer une méthode d'identification et ce présent mémoire vise à améliorer cette dernière, notamment en choisissant judicieusement la fonction de poids faisant partie des paramètres de la méthode.

Comme énoncé précédemment, le problème consiste à identifier des équations aux dérivées partielles non nécessairement linéaires sur base d'un ensemble de trajectoires du système dynamique sous-jacent. Tout au long du mémoire, nous allons considérer que l'équation à identifier est telle que pour chaque trajectoire associée, sa dérivée temporelle est une combinaison linéaire de fonctions de bases supposées connues qui dépendent de cette trajectoire ainsi que de ses dérivées spatiales. Il ne reste alors plus qu'à identifier les coefficients de la combinaison linéaire en question. Des méthodes d'identification d'équations aux dérivées partielles existent déjà mais nécessitent des approximations des dérivées partielles et ne sont pas robustes au bruit des données.

L'objectif de ce mémoire consiste alors à s'appuyer sur la méthode d'identification de l'article [11] afin de l'améliorer. En effet, cette méthode dépend d'une fonction de poids à choisir judicieusement. Il s'agira en particulier de comparer des méthodes de choix de fonctions de poids en tenant compte de la variabilité des données, de sélectionner la méthode la plus adéquate, d'améliorer les performances et de tester sa robustesse.

Afin de mener à bien ces objectifs, un premier chapitre sera consacré à la description des méthodes d'identification d'équations aux dérivées partielles ainsi que de leurs inconvénients en commençant au préalable par fixer le cadre du problème évoqué précédemment. Ensuite, nous allons commencer la description de ma contribution personnelle s'appuyant sur la méthode de l'article [11] à l'aide d'un second chapitre se portant sur les premiers résultats de la méthode où nous allons notamment calculer l'estimation moyenne des coefficients de la

combinaison linéaire de l'équation aux dérivées partielles grâce à plusieurs fonctions de poids. Constatant qu'il est nécessaire d'opter pour une méthode automatique de choix de fonctions de poids, un prochain chapitre sera consacré à une recherche de ces fonctions de poids dans un espace fonctionnel par des problèmes d'optimisation en tenant compte de la variabilité des données. Les problèmes d'optimisation se basent en particulier sur la minimisation de l'écart des coefficients à identifier et sur la minimisation de l'écart entre les trajectoires estimées sur base des coefficients estimés et les véritables trajectoires. Au terme de ce chapitre, une méthode d'optimisation sera choisie en fonction des résultats obtenus et nous constaterons que cette méthode choisie améliore bel et bien les performances de l'identification de manière significative. Ensuite, le chapitre suivant consistera à tenter d'améliorer les performances de la méthode choisie en estimant rapidement le générateur du système dynamique via une interpolation de Lagrange. Enfin, nous conclurons le travail en comparant les résultats et conclusions des chapitres précédents avec les objectifs fixés dans cette introduction.

Chapitre 1

Méthodes numériques d'identification d'équations aux dérivées partielles

L'objectif de ce chapitre consiste à décrire les grandes méthodes numériques de résolution du problème d'identification décrit dans l'introduction de ce travail. Pour ce faire, nous commencerons par une section présentant plus précisément le problème. Ensuite, nous allons décrire l'état de l'art, c'est-à-dire la description des méthodes déjà existantes afin de les comparer avec la méthode décrite dans l'article [11] se basant sur l'opérateur de Koopman linéarisant le système étudié. La section suivante sera dédiée à la description et à l'illustration des inconvénients qu'engendre l'approximation des dérivées à l'aide des différences finies. Ensuite, nous développerons la notion d'opérateurs de Koopman d'un système dynamique en rappelant les concepts importants dans le cas d'un système décrit par une équation différentielle ordinaire et en les comparant avec un système décrit par une équation aux dérivées partielles. Enfin, une dernière section décrira la méthode de l'article [11] qui limite l'approximation de dérivées partielles en utilisant cet opérateur de Koopman.

1.1 Présentation du problème

Tout au long du mémoire, nous travaillerons avec des systèmes de la forme

$$\frac{\partial u}{\partial t} = \sum_{i=1}^{n} c_i W_i(u) \tag{1.1}$$

où les opérateurs W_i sont supposés connus, non nécessairement linéaires et faisant intervenir la variable u ainsi que les dérivées spatiales et où les coefficients c_i sont à identifier sur base des trajectoires sous-jacentes. Le fait de supposer les opérateurs W_i connus est une hypothèse tout à fait raisonnable dans les applications. En effet, nous savons que l'équation de la chaleur en une dimension s'écrit sous la forme

$$\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial x^2}$$

où la variable spatiale x appartient à l'intervalle compact $[0; L]$ et où la variable temporelle t est positive. Clairement, si nous disposons de trajectoires régies par cette équation avec des conditions initiales et des conditions de bords données, alors le système sera identifié uniquement en identifiant le coefficient c^2.

Nous allons également supposer que l'application W_1 est l'opérateur identité. De nouveau, cette hypothèse supplémentaire est très peu restrictive. En effet, dans ce cas, si le système réel s'écrit sous la forme

$$\frac{\partial u}{\partial t} = \sum_{i=2}^{n} c_i W_i(u),$$

c'est-à-dire si le vrai modèle ne fait pas intervenir l'application identité W_1, alors nous nous attendons à ce que l'identification numérique des scalaires c_i estime le coefficient c_1 comme étant un scalaire très proche de zéro. Si le vrai modèle ne fait pas intervenir l'opérateur W_1, alors le coefficient c_1 sera ignoré après identification numérique.

1.2 Classes de méthodes existantes

Décrivons à présent les méthodes identifiant le système (1.1) existant avant l'apparition de la méthode de l'article [11] utilisant l'opérateur de Koopman en les répartissant selon les grandes classes de méthodes : des méthodes se basant sur la régression linéaire sur base d'un estimateur de Ridge [15] et d'un estimateur LASSO [10] généralisant l'estimateur des moindres carrés, une méthode décrite dans l'article [7] qui utilise une notion similaire à la fonction de poids de la méthode de Koopman [11] et enfin un algorithme de régression orthogonale en avant (Orthogonal Forward Regression - OFR) similaire à la méthode présentée dans l'article [6]. Toutes ces méthodes supposent comme connues les mêmes données construites sur base d'un ensemble de trajectoires u_k pour k allant de 1 à N_u s'exprimant sous la forme

$$U := \begin{pmatrix} u_1(x_1; t_0) \\ u_1(x_1; t_1) \\ \vdots \\ u_1(x_1; t_{N_t}) \\ u_1(x_2; t_0) \\ \vdots \\ u_1(x_{N_x}; t_{N_t}) \\ u_2(x_1; t_0) \\ \vdots \\ u_{N_u}(x_{N_x}; t_{N_t}) \end{pmatrix}$$

où l'ensemble discrétisé de temps $\{t_0; t_1; \dots t_{N_t}\}$ ainsi que l'ensemble discrétisé d'états $\{x_1; x_2; \dots x_{N_x}\}$ sont déterminés à l'avance. Ces données sont le vecteur Y étant un vecteur colonne représentant l'approximation de $\frac{\partial u}{\partial t}$ via une différence finie et une matrice X appartenant à l'ensemble des matrices réelles $\mathbb{R}^{(N_u N_x (N_t+1)) \times n}$ où les composantes à la i-ème ligne et à la j-ème colonne de la matrice X sont définies par $W_j(U_i)$.

1.2.1 Estimateur de Ridge et estimateur LASSO

L'article [15] propose une estimation du vecteur $c = \begin{pmatrix} c_1 & \dots & c_n \end{pmatrix}^\top$ en minimisant

$$\|Y - Xc\|_2 + \lambda \|c\|_2^2 \tag{1.2}$$

où la valeur du paramètre positif λ est fixé à l'avance par l'utilisateur. Intuitivement, le vecteur de coefficients c représente un compromis entre deux exigences. La première concerne l'ajustement de l'expression Xc au vecteur Y décrit par le premier terme représentant le problème des moindres carrés classiques tandis que la seconde pénalise système (1.1) admettant beaucoup d'opérateurs W_i. Cette pénalisation est représentée par le second terme nommé *terme de pénalisation*. L'article [15] va même plus loin dans la pénalisation de la complexification du système à identifier. En effet, les coefficients c_i de valeur absolue inférieure à une certaine tolérance fixée par l'utilisateur sont considérés comme étant nuls afin que le système (1.1) soit identifié avec une complexité affaiblie, c'est-à-dire avec moins de termes. Les coefficients restants c_i sont mis à jour à nouveau avec la minimisation de (1.1) avec les colonnes de la matrice X associées aux opérateurs W_j dont les coefficients c_i associés ne sont pas annulés. Ces itérations seront répétées un nombre de fois fixé à l'avance par l'utilisateur. L'article [10] propose une idée similaire en remplaçant (1.2) par

$$\|Y - Xc\|_2 + \lambda \|c\|_1 . \tag{1.3}$$

Remarquons que ces deux estimateurs généralisent l'estimateur des moindres carrés en posant le paramètre λ comme étant égal à zéro.

1.2.2 Méthode de Gurevich

Afin d'estimer le vecteur $c = \begin{pmatrix} c_1 & \dots & c_n \end{pmatrix}^\top$, la première étape de la méthode de l'article [7] consiste en la construction d'une certaine matrice $Q \in \mathbb{R}^{(N_u p)\times(n+1)}$. Ses éléments de la première colonne à la $(k-1)\,p + i$-ème ligne sont définis par

$$\int_{x_1}^{x_{N_x}} \int_{t_0}^{t_{N_t}} -\frac{\partial u_k}{\partial t}(x;t)\, w_i(x;t)\, \mathrm{d}t\mathrm{d}x$$

où les p fonctions w_i sont fixées à l'avance par l'utilisateur tandis que les autres éléments de cette matrice Q sont définis à la $(k-1)\,p + i$-ème ligne et à la j-ème colonne par

$$\int_{x_1}^{x_{N_x}} \int_{t_0}^{t_{N_t}} W_{j-1}(u_k)(x;t)\, w_i(x;t)\, \mathrm{d}t\mathrm{d}x.$$

La deuxième étape de l'algorithme est de réaliser la *décomposition en valeurs singulières* de la matrice construite à l'étape précédente. Ainsi,

$$Q = USV^\top.$$

Par conséquent, sous réserve que l'élément $V(1, n+1)$ soit *non nul*, l'article [7] estime le vecteur c comme étant

$$-\frac{V(2:(n+1), n+1)}{V(1,(n+1))}.$$

Remarque 1. *Dans cette méthode, la dérivée temporelle des trajectoires u_k n'intervient que dans la première colonne de la matrice Q. Certes, l'approximation de ces dérivées peut être évitée à l'aide de l'intégration par partie. En revanche, en plus des fonctions de poids w_i, l'utilisateur devra également fournir ses dérivées temporelles.*

L'article [7] propose également l'annulation successive de coefficients c_i dans le cas où $N_u p \geq n+1$. Pour ce faire, supposons que k de ces coefficients sont déjà annulés. Alors nous désigneront $U_{(k)}S_{(k)}V_{(k)}^\top$ comme étant la décomposition en valeurs singulières de la matrice Q définie dans cette sous-section en ne considérant que les opérateurs W_j associées aux coefficients du vecteur c qui ne font pas partie des k coefficients déjà annulés. Supposons que l'élément de la dernière colonne de $V_{(k)}$ ayant la plus petite valeur en module se situe sur la $\ell_{(k)}$-ème ligne. Par conséquent, la méthode de l'article [7] annule le coefficient $c_{\ell_{(k)}}$ si

$$\Sigma_{(k+1)}((n+1):(n+1)) \leq \gamma \Sigma_{(k)}((n+1):(n+1))$$

où γ est un paramètre strictement positif fixé à l'avance par l'utilisateur et termine l'algorithme dans le cas contraire sous réserve que l'élément $V_{(k)}(1,(n+1))$ soit non nul en estimant les coefficients restants c_i sous forme d'un vecteur comme étant

$$-\frac{V_{(k)}(2:(n+1),(n+1))}{V_{(k)}(1,(n+1))}.$$

1.2.3 Méthode OFR

Enfin, l'article [6] propose une régression linéaire à l'aide de l'estimateur des moindres carrés en ajoutant successivement les régresseurs en partant d'un modèle sans régresseurs. Le vecteur x de coefficients des régresseurs sélectionnés se déterminera à l'aide de la résolution du système linéaire

$$Ax = b$$

où la matrice A sera une matrice triangulaire supérieure unité.

Supposons que $v-1$ régresseurs soient déjà sélectionnés dont le k-ème régresseur sélectionné est le vecteur colonne $X(:, \ell_k)$, que les $v-1$ premières colonnes de la matrice A soient déjà initialisées ainsi que les $v-1$ premières composantes du vecteur b et les vecteurs $w_1^0, \ldots, w_{v-1}^0$.

La première étape de la v-ème itération consiste à définir la matrice $w_{(v)}$ de taille $(N_u N_x (N_t+1)) \times n$ dont la j-ème colonne est définie par

$$X(:,j) - \sum_{k=1}^{v-1} \frac{{w_k^0}^\top Y}{{w_k^0}^\top w_k^0} w_k^0.$$

Remarquons que pour la première itération, la somme sur k est toujours par convention nulle, ce qui signifie en particulier qu'à ce stade, aucun vecteur w_k^0 n'est nécessaire.

La deuxième étape de cette itération consiste à sélectionner le régresseur $X(:,\ell_v)$ où l'indice ℓ_v minimise la quantité

$$\frac{\left(w_{(v)}(:,\ell_v)^\top Y\right)^3}{Y^\top Y\left(w_{(v)}(:,\ell_v)^\top w_{(v)}(:,\ell_v)\right)^2}$$

parmi les régresseurs qui ne sont pas encore sélectionnés. Cette quantité ainsi minimisée sera notée E_v.

La dernière étape de cette itération met à jour les éléments suivants pour tout naturel k non nul et strictement inférieur à v

$$\begin{cases} w_v^0 &= w_{(v)}(:,\ell_v) \\ A(k,v) &= \frac{{w_k^0}^\top X(:,\ell_v)}{{w_k^0}^\top w_k^0} \\ b(v) &= \frac{{w_k^0}^\top Y}{{w_k^0}^\top w_k^0} \end{cases}.$$

Dans le cas où certains régresseurs ne sont pas encore sélectionnés, une itération supplémentaire sera réalisée lorsque

$$1-\sum_{k=1}^{v} E_k \geq \rho$$

où le paramètre ρ est fixé à l'avance par l'utilisateur.

1.3 Inconvénients des différences finies

Cette section consiste à rappeler la méthode principale pour approximer une dérivée temporelle pour donner le vecteur Y défini dans la section précédente et des inconvénients que cela engendre. L'approximation de cette dérivée temporelle se base sur la définition de la dérivée partielle $\frac{\partial u}{\partial t}$ qui est

$$\lim_{\delta\to 0}\frac{u(x,t+\delta)-u(x,t)}{\delta}.$$

Par conséquent, la dérivée partielle sera approximée en choisissant un δ assez proche de zéro. Le choix de ce réel sera déterminant dans la qualité de son approximation. En arithmétique idéale, plus le réel δ est proche de zéro, meilleure est la qualité de l'approximation de la dérivée partielle. Cependant, en arithmétique en virgule flottante, lorsque le réel δ est trop proche de zéro, les erreurs d'arrondis vont dominer les erreurs de troncature sur δ à cause de la division par δ et la division par un nombre trop proche de zéro est source de très importantes instabilités. La figure 1.1, réalisée à l'aide du langage de programmation `Julia`,

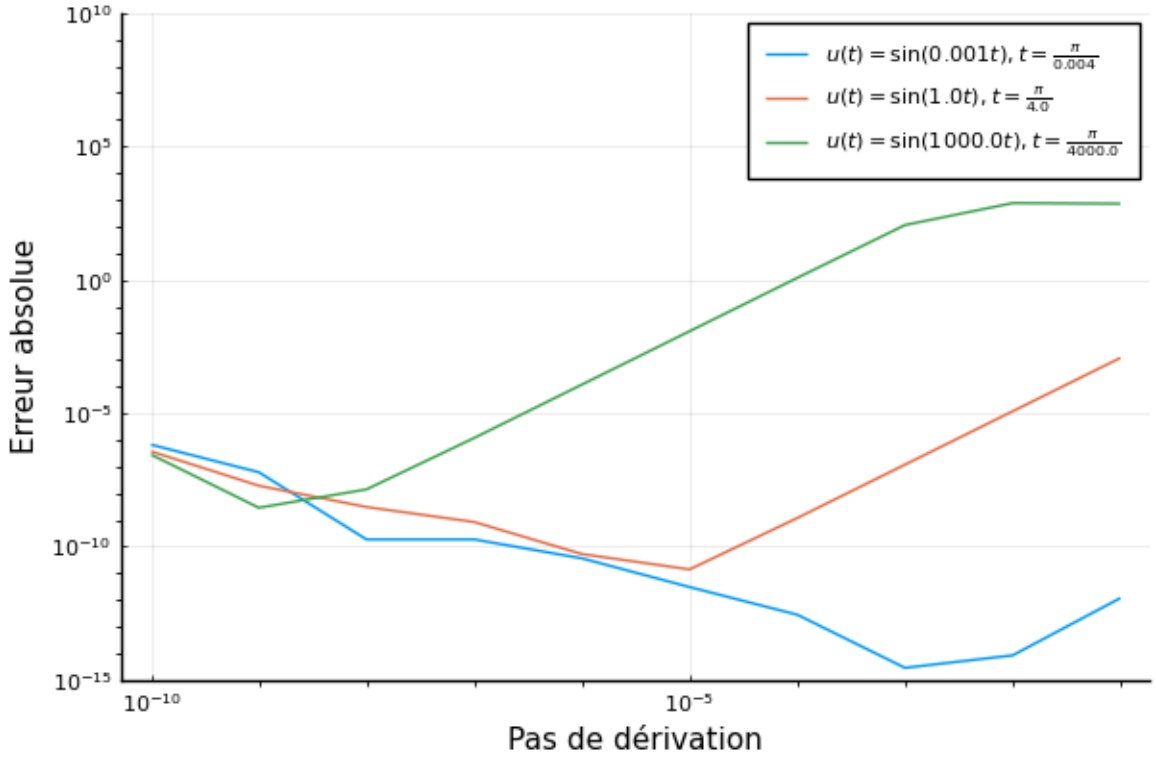


FIGURE 1.1 – Représentation de l'erreur absolue de l'approximation de la dérivée par différences finies centrées d'ordre deux pour différentes fonctions, réalisée à l'aide du langage de programmation `Julia`.

confirme ce constat avec une décroissance pour une très petite valeur du pas de dérivation δ suivie d'une croissance pour des pas de dérivations supérieurs. Elle illustre également une problématique majeure lors de l'approximation des dérivées par différences finies : le pas de dérivation δ à choisir dépend de la fonction u à dériver selon l'impact des erreurs de troncature de la méthode des différences finies et des erreurs d'arrondis dues à l'arithmétique en virgule flottante. De plus, si les données sont bruitées, il est clair que l'approximation des dérivées partielles seront encore plus médiocres, surtout si ce bruitage rend la continuité (et donc la différentiabilité) très peu vraisemblable. Par conséquent, il est tout à fait souhaitable d'envisager des méthodes qui ne demandent pas le calcul explicite des dérivées. C'est pourquoi la méthode de l'article [11] faisant l'objet de la section suivante a été conçue, bien qu'elle nécessite encore le calcul des dérivées temporelles.

1.4 Opérateur de Koopman

Dans cette section, nous définissons un concept fondamental à la compréhension de la méthode de l'article [11] : l'opérateur de Koopman. La première sous-section rappelle cette

notion dans le cas d'un système décrit par une équation différentielle ordinaire et la seconde sous-section compare ce cas avec le cas qui nous intéresse dans le cadre du mémoire, à savoir celui d'un système décrit par une équation aux dérivées partielles. Dans chacune de ces sous-sections, nous allons commencer au préalable par définir le flot du système considéré qui sera incontournable lors de la définition de l'opérateur de Koopman. Ensuite, nous allons énoncer la définition et les propriétés importantes de l'opérateur de Koopman qui motiveront son emploi lors de l'identification de systèmes. Enfin, dans l'objectif d'affiner les interprétations de l'algorithme de l'article [11], nous allons développer la notion de générateur de Lie.

1.4.1 Système décrit par une équation différentielle ordinaire

Comme énoncé au début de la section, commençons par rappeler la définition du flot dans le cas d'un système décrit par une équation différentielle ordinaire.

Définition 1. *Soit une application F qui est* continument différentiable *définie sur l'ensemble des n-uplets réels noté $\mathbb{R}^n$. Elle définit alors pour toute condition initiale u_0 de l'ensemble $\mathbb{R}^n$ un système dont la trajectoire u vérifie pour tout temps positif t le système*

$$\begin{cases} \dot{u}(t) &= F(u(t)) \\ u(0) &= u_0 \end{cases}.$$

Alors, le flot de ce système noté φ est défini comme étant une application *représentant en tout temps t positif et pour toute condition initiale u_0 dans l'ensemble $\mathbb{R}^n$, la solution u du système qui sera dès lors notée $\varphi^t(u_0)$.*

Remarquons que tout système dont le problème de Cauchy est donné par le système

$$\begin{cases} \dot{v}(t) &= G(v(t), t) \\ v(0) &= v_0 \end{cases}$$

peut se réexprimer par un système de la Définition 1 avec la variable

$$u(t) = \begin{pmatrix} v(t) & t \end{pmatrix}^\top.$$

En effet, l'équation différentielle ordinaire devient

$$\dot{u}(t) = \begin{pmatrix} \dot{v}(t) & \dot{t} \end{pmatrix}^\top = \begin{pmatrix} G(v(t), t) & 1 \end{pmatrix}^\top =: F(u(t))$$

et la condition initiale s'exprime comme étant

$$u(0) = \begin{pmatrix} v(0) & 0 \end{pmatrix}^\top = \begin{pmatrix} v_0 & 0 \end{pmatrix}^\top =: u_0.$$

Voici à présent une propriété très intéressante concernant les flots qui va avoir une conséquence directe sur la famille d'opérateurs de Koopman que nous allons définir plus tard dans cette sous-section. Étant donné que cette propriété reste valable dans le cas d'un système

décrit par une équation aux dérivées partielles comme nous le verrons plus tard, cette propriété sera démontrée dans ce dernier cas car la démonstration de cette propriété dans le cas de cette sous-section admet exactement la même structure.

Propriété 1. *Soit le flot φ décrit par la Définition 1. Alors il représente un* semi-groupe, *c'est-à-dire que la fonction φ^0 est la fonction identité et que tous les scalaires positifs t et s vérifient la relation*

$$\varphi^t \circ \varphi^s = \varphi^{t+s}.$$

Maintenant que nous avons développé la notion de flot d'un système décrit par une équation différentielle ordinaire, nous sommes maintenant capables de définir son opérateur de Koopman.

Définition 2. *Soient un réel t positif et le flot φ d'un système décrit par une* équation différentielle ordinaire. *Alors l'opérateur de Koopman U^t appliqué à une* fonction f *appelée* observable *est défini pour toute condition initiale u_0 par l'équation*

$$\left(U^t f\right)(u_0) := f\left(\varphi^t(u_0)\right).$$

Afin de rendre ces définitions plus concrètes, illustrons-les à l'aide d'exemples.

Exemple 1. *Commençons par un système simple décrit par une équation différentielle ordinaire linéaire et une condition initiale de la forme*

$$\begin{cases} \dot{x}(t) &= x(t) + y(t) \\ \dot{y}(t) &= x(t) - y(t) \\ x(0) &= x_0 \in \mathbb{C} \\ y(0) &= y_0 \in \mathbb{C} \end{cases}.$$

Ce système peut se réécrire sous la forme

$$\begin{cases} \dot{v}(t) &= Av(t) \\ v(0) &= v_0 \in \mathbb{C} \end{cases}.$$

où la variable v représente le couple $\begin{pmatrix} x & y \end{pmatrix}^\top$, où la condition initiale v_0 est égale au couple $\begin{pmatrix} x_0 & y_0 \end{pmatrix}^\top$ et où la matrice A du système linéaire est égale à la matrice

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Pour résoudre ce système, commençons par calculer les valeurs propres λ de la matrice A. Par définition, ces valeurs propres doivent vérifier la relation

$$0 = \det(\lambda I - A) = \det\begin{pmatrix} \lambda - 1 & -1 \\ -1 & \lambda + 1 \end{pmatrix} = (\lambda - 1)(\lambda + 1) - 1 = \lambda^2 - 2.$$

Les valeurs propres de la matrice A étant les valeurs propres distinctes $-\sqrt{2}$ et $\sqrt{2}$, nous déduisons que la solution s'écrit sous la forme

$$\left\{\begin{array}{rcl} x\left(t\right) & = & \alpha \mathrm{e}^{\sqrt{2}t} + \beta \mathrm{e}^{-\sqrt{2}t} \\ y\left(t\right) & = & \gamma \mathrm{e}^{\sqrt{2}t} + \delta \mathrm{e}^{-\sqrt{2}t} \end{array}\right.$$

où les scalaires α, β, γ et δ sont à déterminer. Pour ce faire, nous savons que, d'une part, les conditions initiales imposent le système de relations

$$\left\{\begin{array}{rcccccc} x_0 & = & x\left(0\right) & = & \alpha \mathrm{e}^{\sqrt{2}\cdot 0} + \beta \mathrm{e}^{-\sqrt{2}\cdot 0} & = & \alpha + \gamma \\ y_0 & = & y\left(0\right) & = & \gamma \mathrm{e}^{\sqrt{2}\cdot 0} + \delta \mathrm{e}^{-\sqrt{2}\cdot 0} & = & \beta + \delta \end{array}\right.$$

ou, autrement dit, le système de relations

$$\left\{\begin{array}{rcl} \beta & = & x_0 - \alpha \\ \gamma & = & y_0 - \delta \end{array}\right.$$

et que, d'autre part, l'équation différentielle à résoudre impose notamment la relation

$$\sqrt{2}\alpha \mathrm{e}^{\sqrt{2}t} - \sqrt{2}\left(x_0 - \alpha\right)\mathrm{e}^{-\sqrt{2}t} = x'\left(t\right) = x\left(t\right) + y\left(t\right) = \left(\alpha + y_0 - \delta\right)\mathrm{e}^{\sqrt{2}t} + \left(x_0 - \alpha + \delta\right)\mathrm{e}^{-\sqrt{2}t}$$

ou, autrement dit, le système de relations

$$\left\{\begin{array}{rcl} \sqrt{2}\alpha & = & \alpha + y_0 - \delta \\ -\sqrt{2}\left(x_0 - \alpha\right) & = & x_0 - \alpha + \delta \end{array}\right. .$$

En remplaçant la dernière équation par la somme des équations du système multipliée par $\sqrt{2}$, nous obtenons

$$\left\{\begin{array}{rcl} \delta & = & \left(1 - \sqrt{2}\right)\alpha + y_0 \\ 4\alpha - 2x_0 & = & \sqrt{2}x_0 + \sqrt{2}y_0 \end{array}\right. .$$

Par conséquent, la dernière équation permet de déduire

$$\alpha = \frac{2 + \sqrt{2}}{4}x_0 + \frac{\sqrt{2}}{4}y_0.$$

Ainsi, l'expression de δ en fonction de α permet d'exprimer le scalaire γ initialement exprimée en fonction de δ à l'aide de la relation

$$\gamma = \left(\sqrt{2} - 1\right)\alpha.$$

Nous pouvons donc nous appuyer sur l'expression de α et des scalaires β, γ et δ exprimés en fonction du scalaire α pour réexprimer la solution de l'équation différentielle comme étant

$$\boxed{\left\{\begin{array}{rcl} x\left(t\right) & = & \left(\frac{2+\sqrt{2}}{4}x_0 + \frac{\sqrt{2}}{4}y_0\right)\mathrm{e}^{\sqrt{2}t} + \left(\frac{2-\sqrt{2}}{4}x_0 - \frac{\sqrt{2}}{4}y_0\right)\mathrm{e}^{-\sqrt{2}t} \\ y\left(t\right) & = & \left(\frac{\sqrt{2}}{4}x_0 + \frac{2-\sqrt{2}}{4}y_0\right)\mathrm{e}^{\sqrt{2}t} + \left(\frac{2+\sqrt{2}}{4}y_0 - \frac{\sqrt{2}}{4}x_0\right)\mathrm{e}^{-\sqrt{2}t} \end{array}\right.}.$$

Maintenant que nous avons trouvé la solution analytique $\begin{pmatrix} x(t) & y(t) \end{pmatrix}^\top$, *la Définition 1 implique que le flot de ce système linéaire est défini comme étant*

$$\varphi^t(v_0) = \begin{pmatrix} \frac{2+\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \\ \\ \frac{\sqrt{2}}{4} & \frac{2-\sqrt{2}}{4} \end{pmatrix} v_0 \mathrm{e}^{\sqrt{2}t} + \begin{pmatrix} \frac{2-\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ \\ -\frac{\sqrt{2}}{4} & \frac{2+\sqrt{2}}{4} \end{pmatrix} v_0 \mathrm{e}^{-\sqrt{2}t}.$$

Par conséquent, la Définition 2 fournit

$$\left(U^t f\right)(v_0) = f\left(\varphi^t(v_0)\right) = f\left(\begin{pmatrix} \frac{2+\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \\ \\ \frac{\sqrt{2}}{4} & \frac{2-\sqrt{2}}{4} \end{pmatrix} v_0 \mathrm{e}^{\sqrt{2}t} + \begin{pmatrix} \frac{2-\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ \\ -\frac{\sqrt{2}}{4} & \frac{2+\sqrt{2}}{4} \end{pmatrix} v_0 \mathrm{e}^{-\sqrt{2}t}\right)$$

pour toute fonction observable f. *En particulier, si nous définissons une fonction de projection*

$$\begin{array}{cccc} \pi_1 & \mathbb{R}^2 & \to & \mathbb{R} \\ & (r,s) & \rightsquigarrow & r \end{array},$$

alors, nous obtenons

$$\left(U^t \pi_1\right)(v_0) = \begin{pmatrix} \frac{2+\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \end{pmatrix} v_0 \mathrm{e}^{\sqrt{2}t} + \begin{pmatrix} \frac{2-\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \end{pmatrix} v_0 \mathrm{e}^{-\sqrt{2}t}.$$

Ce dernier cas particulier correspond exactement à la réponse ρ *du système commandé linéaire temps-invariant décrit par le système*

$$\begin{cases} \dot{x} &= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \end{pmatrix} u \\ \\ \rho &= \begin{pmatrix} 1 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \end{pmatrix} u \end{cases}.$$

Cet exemple illustre donc bien l'interprétation principale très intéressante de l'opérateur de Koopman : si un système est décrit par un flot φ^t en tout temps positif t et si l'application f est une fonction d'observation de la trajectoire sur base d'une condition initiale u_0, alors $\left(U^t f\right)(u_0)$ représente l'observation de l'état de ce système de la même condition initiale mais après un temps t.

Le prochain exemple montre que nous pouvons très bien appliquer l'opérateur de Koopman sur un système qui soit non linéaire.

Exemple 2. *Soit le système décrit par l'équation différentielle ordinaire qui s'exprime comme étant*

$$\begin{cases} \dot{x}(t) &= x(t)\left(x(t)+1\right) \\ x(0) &= x_0 \in \mathbb{C} \end{cases}.$$

Alors, pour déterminer le flot de ce système, nous devons d'abord résoudre le problème de Cauchy. Pour ce faire, la séparation des variables sera utilisée. Tout d'abord, nous observons que l'équation différentielle ordinaire peut se réexprimer comme étant

$$\frac{\dot{x}(t)}{x(t)(x(t)+1)} = 1.$$

Ensuite, l'intégration par rapport au temps nous fournit la relation

$$\int_0^t \frac{\dot{x}(\tau)}{x(t)(x(\tau)+1)} \mathrm{d}\tau = \int_0^t \mathrm{d}\tau = t.$$

Or, nous pouvons effectuer sur le premier membre de cette équation un changement de variable de la forme $u = x(\tau)$. Par conséquent, cette équation devient

$$\int_{x_0}^{x(t)} \frac{\mathrm{d}u}{u(u+1)} = t.$$

L'étape suivante consiste à remarquer que nous pouvons décomposer la fonction en u à intégrer en fractions simples. Ce qui donne la relation

$$\int_{x_0}^{x(t)} \left(\frac{1}{u} - \frac{1}{u+1}\right) \mathrm{d}u = t.$$

Cette décomposition nous permet de trouver une primitive de la fonction à intégrer pour ainsi fournir

$$\int_{x_0}^{x(t)} \frac{\mathrm{d}}{\mathrm{d}u} \left(\ln|u| - \ln|u+1|\right) \mathrm{d}u = t.$$

Par propriétés sur les fonctions logarithmiques, cette équation se réduit à

$$\int_{x_0}^{x(t)} \frac{\mathrm{d}}{\mathrm{d}u} \left(-\ln\left|1+u^{-1}\right|\right) \mathrm{d}u = t.$$

L'application du Théorème Fondamental du calcul intégral nous donne

$$\ln\left|1+x_0^{-1}\right| - \ln\left|1+x(t)^{-1}\right| = t.$$

Cette équation nous fournit finalement comme solution

$$\boxed{x(t) = \frac{1}{\mathrm{e}^{-t}\left(1+x_0^{-1}\right)-1}}.$$

De nouveau, maintenant que nous avons trouvé la solution $x(t)$ du système considéré, la Définition 1 indique que le flot s'exprime comme étant

$$\varphi^t(x_0) = \frac{1}{\mathrm{e}^{-t}\left(1+x_0^{-1}\right)-1}$$

tandis que la définition 2 définit l'opérateur de Koopman à l'aide de la relation

$$\left(U^t f\right)(x_0) = f\left(\varphi^t(x_0)\right) = f\left(\frac{1}{\mathrm{e}^{-t}\left(1+x_0^{-1}\right)-1}\right)$$

pour toute fonction observable f. En particulier, si nous définissons une fonction d'observabilité sur un intervalle compact

$$\begin{array}{rccl} \pi_{[-2;0]} & \mathbb{R} & \to & [-2;0] \\ & u & \rightsquigarrow & u\chi_{[-2;0]}(u) - 2\chi_{]-\infty;-2[}(u) \end{array},$$

alors la Définition 2 fournit à nouveau

$$\left(U^t \pi_{[-2;0]}\right)(x_0) = \frac{\chi_{[-2;0]}\left(\frac{1}{\mathrm{e}^{-t}\left(1+x_0^{-1}\right)-1}\right)}{\mathrm{e}^{-t}\left(1+x_0^{-1}\right)-1} - 2\chi_{]-\infty;-2[}\left(\frac{1}{\mathrm{e}^{-t}\left(1+x_0^{-1}\right)-1}\right). \tag{1.4}$$

Remarquons toutefois que nous pouvons simplifier grandement la relation (1.4) et en particulier l'expression des fonctions caractéristiques. En effet, d'une part, pour que le flot soit négatif pour un temps fixé t, il est nécessaire et suffisant de vérifier

$$\mathrm{e}^{-t}\left(1+x_0^{-1}\right)-1 < 0$$

ou autrement dit, par positivité stricte de l'exponentielle

$$x_0^{-1} < \mathrm{e}^t - 1. \tag{1.5}$$

D'autre part, pour que le flot soit strictement inférieur à -2, il suffit de résoudre en la variable x_0^{-1} l'inéquation

$$\frac{1}{\mathrm{e}^{-t}\left(1+x_0^{-1}\right)-1} < -2$$

qui s'exprime de manière équivalente par (1.5)

$$1 < -2\left(\mathrm{e}^{-t}\left(1+x_0^{-1}\right)-1\right).$$

Nous obtenons donc immédiatement toujours grâce à la positivité de l'exponentielle

$$x_0^{-1} > \frac{\mathrm{e}^t}{2} - 1. \tag{1.6}$$

Par conséquent, les inéquations (1.5) et (1.6) permettent d'exprimer l'équation (1.4) comme étant

$$\left(U^t \pi_{[-2;0]}\right)(x_0) = \frac{\chi_{\left]-\infty;\frac{\mathrm{e}^t}{2}-1\right]}\left(x_0^{-1}\right)}{\mathrm{e}^{-t}\left(1+x_0^{-1}\right)-1} - 2\chi_{\left]\frac{\mathrm{e}^t}{2}-1;\mathrm{e}^t-1\right[}\left(x_0^{-1}\right).$$

Notons au passage que l'expression des fonctions caractéristiques en fonction des variables x_0 ne simplifie pas davantage l'expression de cette dernière équation étant donné que la fonction inverse n'est pas une fonction monotone.

Avant de passer à la notion de générateur de Lie, regardons trois propriétés importantes de cet opérateur. De nouveau, ces propriétés restant tout à fait valables dans le cas d'un système décrit par une équation aux dérivées partielles, elles seront démontrées dans ce dernier cas pour les mêmes raisons. La première consiste à donner des informations sur sa structure.

Propriété 2. *La famille d'opérateurs de Koopman pour un système de flot φ décrit par la Définition 1 est un* semi-groupe*, c'est-à-dire que l'opérateur U^0 est l'opérateur identité et que tous les scalaires t et s vérifient*

$$U^t U^s = U^{t+s}.$$

La seconde propriété fournit une description encore plus fine de la famille d'opérateur de Koopman dans un cas particulier qui reste très fréquent dans les applications : le caractère de continuité ponctuelle du semi-groupe.

Propriété 3. *Supposons que chaque opérateur de Koopman pour un système de flot φ décrit par la Définition 1 admet un domaine commun qui est l'ensemble des fonctions continues à valeur complexe de norme supremum finie et que les fonctions φ^s et $t \rightsquigarrow \varphi^t(u)$ soient continues pour tout temps réel positif s et pour toute condition initiale u. Alors, le semi-groupe d'opérateur de Koopman est ponctuellement continu, c'est-à-dire que toutes les fonctions f du domaine et toutes les conditions initiales u_0 vérifient la relation*

$$\lim_{t \downarrow 0^+} \left\| \left(U^t f\right)(u_0) - f(u_0) \right\| = 0.$$

La dernière propriété quant à elle, rend l'utilisation de l'opérateur de Koopman très intéressante comme nous le verrons dans la prochaine sous-section.

Propriété 4. *Soient un réel t positif et le flot φ d'un système décrit par une équation différentielle ordinaire. Alors l'opérateur de Koopman U^t est linéaire.*

Maintenant que nous avons développé la notion d'opérateur de Koopman d'un système dynamique décrit par une équation différentielle ordinaire, nous allons à présent introduire la notion de générateur de Lie. Cette notion dans le cas d'un système décrit par une équation aux dérivées partielles va permettre d'affiner les interprétations de l'algorithme de l'article [11] qui sera développé dans la section suivante. Commençons tout d'abord par rappeler la définition du générateur de Lie d'un système dynamique décrit par une équation différentielle ordinaire.

Définition 3. *Soit un système dynamique dont l'opérateur de Koopman est décrit par la Définition 2. Alors le générateur de Lie L est défini pour toute condition initiale u_0 par l'égalité*

$$(Lf)(u_0) = \lim_{t \downarrow 0^+} \frac{(U^t f)(u_0) - f(u_0)}{t}$$

pour toute fonction f du domaine de définition de l'opérateur de Koopman telle que la limite précédente soit bien définie.

Notons que l'expression du générateur de Lie dans le cas d'un système décrit par une équation différentielle ordinaire de la Définition 1 où la fonction F est continue et avec une fonction observable f qui soit continument différentiable se simplifie grandement. En effet, la définition de l'opérateur de Koopman implique que

$$(Lf)(u_0) = \lim_{t \downarrow 0^+} \frac{f(\varphi^t(u_0)) - f(u_0)}{t}.$$

Ensuite, la différentiabilité de la fonction observable f fournit la relation

$$\lim_{t \downarrow 0^+} \frac{f(\varphi^t(u_0)) - f(u_0)}{t} = \lim_{t \downarrow 0^+} \frac{\mathrm{d}}{\mathrm{d}t} f\left(\varphi^t(u_0)\right).$$

La prochaine étape consiste à utiliser la dérivation en chaîne pour obtenir l'égalité

$$\lim_{t \downarrow 0^+} \frac{\mathrm{d}}{\mathrm{d}t} f\left(\varphi^t(u_0)\right) = \lim_{t \downarrow 0^+} \left(\nabla f\left(\varphi^t(u_0)\right)^\top \frac{\mathrm{d}}{\mathrm{d}t} \varphi^t(u_0)\right).$$

Ce dernier membre a du sens. En effet, la Définition 1 indique que

$$\lim_{t \downarrow 0^+} \left(\nabla f\left(\varphi^t(u_0)\right)^\top \frac{\mathrm{d}}{\mathrm{d}t} \varphi^t(u_0)\right) = \lim_{t \downarrow 0^+} \left(\nabla f\left(\varphi^t(u_0)\right)^\top F\left(\varphi^t(u_0)\right)\right).$$

Ensuite, la continuité du flot φ (grâce à sa dérivabilité), de la fonction F ainsi que du gradient de la fonction observable f fournissent l'équation

$$\lim_{t \downarrow 0^+} \left(\nabla f\left(\varphi^t(u_0)\right)^\top F\left(\varphi^t(u_0)\right)\right) = \nabla f\left(\varphi^0(u_0)\right)^\top F\left(\varphi^0(u_0)\right).$$

Enfin, la Propriété 1 implique que

$$\nabla f\left(\varphi^0(u_0)\right)^\top F\left(\varphi^0(u_0)\right) = \nabla f(u_0)^\top F(u_0).$$

Ces six dernières équations indiquent que le générateur de Lie appliqué à la fonction observable f n'est rien d'autre que l'expression

$$\nabla f^\top F.$$

1.4.2 Système décrit par une équation aux dérivées partielles

Maintenant que nous avons rappelé la théorie sur l'opérateur de Koopman dans le cas d'un système décrit par une équation différentielle ordinaire, nous allons la comparer avec le cas d'un système décrit par une équation aux dérivées partielles. Concernant le flot, remarquons que la variable $u(x,t)$ dépend non seulement de la variable temporelle t et de la condition initiale u_0 mais aussi de la variable spatiale x appartenant à un ensemble compact Ω étant une partie de l'ensemble $\mathbb{R}^p$. Il est donc clair que la définition du flot doit être adaptée dans ce cas si nous voulons appliquer l'opérateur de Koopman pour l'identification d'équations différentielles partielles.

Définition 4. *Soit une application F qui est* continument différentiable *qui dépend de la trajectoire $u(t)$ ainsi que de ses gradients. Elle définit alors pour toute condition initiale u_0, une application définie sur un ensemble compact d'états spatiaux noté Ω contenu dans l'ensemble $\mathbb{R}^p$, un système dont la trajectoire u vérifie pour tout temps positif t et pour tout état spatial x dans l'ensemble compact Ω le système*

$$\begin{cases} \frac{\partial u}{\partial t}(x,t) &= F\left(u(x,t), \nabla_x u(x,t), \nabla_x^2 u(x,t), \ldots\right) \\ u(x,0) &= u_0(x) \end{cases}.$$

Afin que ce problème soit bien posé, *des conditions aux bords de l'ensemble Ω sont nécessaires au système. Dans ce cas, le flot de ce système noté φ est défini comme étant une* fonctionnelle *représentant en tout temps t positif, en tout état x dans l'ensemble $\mathbb{R}^p$ et pour toute condition initiale* fonctionnelle *u_0, la solution u du système qui sera dès lors notée $\varphi^t(u_0)(x)$.*

De la même manière, tout système dont le problème de Cauchy est donné par le système

$$\begin{cases} \frac{\partial v}{\partial t}(x,t) &= G\left(t, u(x,t), \nabla_x u(x,t), \nabla_x^2 u(x,t), \ldots\right) \\ v(x,0) &= v_0(x) \end{cases}$$

peut se réexprimer par un système de la Définition 4 avec la variable

$$u(t) = \begin{pmatrix} v(t) & t \end{pmatrix}^\top.$$

En effet, l'équation aux dérivées partielles devient

$$\begin{aligned} \frac{\partial u}{\partial t}(x,t) = \begin{pmatrix} \frac{\partial v}{\partial t}(x,t) & \frac{\partial t}{\partial t} \end{pmatrix}^\top &= \begin{pmatrix} G\left(t, u(x,t), \nabla_x u(x,t), \nabla_x^2 u(x,t), \ldots\right) & 1 \end{pmatrix}^\top \\ &=: F\left(u(x,t), \nabla_x u(x,t), \nabla_x^2 u(x,t), \ldots\right) \end{aligned}$$

et la condition initiale s'exprime comme étant

$$u(x,0) = \begin{pmatrix} v(x,0) & 0 \end{pmatrix}^\top = \begin{pmatrix} v_0(x) & 0 \end{pmatrix}^\top =: u_0(x).$$

Comme énoncé dans la sous-section précédente, la propriété 1 reste tout à fait valable dans le cas d'un système décrit par une équation aux dérivées partielles.

Propriété 5. *Soit le flot φ décrit par la Définition 4. Alors il représente un* semi-groupe, *c'est-à-dire que la fonction φ^0 est la fonction identité et que tous les scalaires positifs t et s vérifient la relation*

$$\varphi^t \circ \varphi^s = \varphi^{t+s}.$$

Démonstration.

D'une part, la condition initiale dans la Définition 4 fournit

$$\varphi^0 (u_0)(x) = u_0(x)$$

pour une condition initiale arbitraire u_0 et pour un état spatial x dans l'ensemble compact Ω, ce qui indique clairement que la fonctionnelle φ^0 est la fonction identité.

D'autre part, fixons arbitrairement deux scalaires positifs s et t. Alors, le système

$$\begin{cases} \frac{\partial u}{\partial t}(x,t) &= F\left(u(x,t), \nabla_x u(x,t), \nabla_x^2 u(x,t), \ldots\right) \\ u(x,0) &= \varphi^s (u_0)(x) \end{cases}$$

doté des mêmes conditions de bords admet comme solution $\varphi^t(\varphi^s(u_0))(x)$ par la Définition 4. Ce système admet également comme solution $\varphi^{t+s}(u_0)(x)$. En effet, par la dérivation d'une composition de fonctions,

$$\frac{\partial}{\partial t}\varphi^{t+s}(u_0)(x) = F\left(\varphi^{t+s}(u_0)(x), \nabla_x \varphi^{t+s}(u_0)(x), \nabla_x^2 \varphi^{t+s}(u_0)(x), \ldots\right)$$

dont la condition initiale s'exprime comme étant

$$\varphi^{0+s}(u_0)(x) = \varphi^s(u_0)(x).$$

Par unicité de la solution du système où nous supposons que le problème est bien posé, nous déduisons que

$$\varphi^t(\varphi^s(u_0))(x) = \varphi^{t+s}(u_0)(x).$$

Finalement, étant donné que la condition initiale u_0 est arbitraire ainsi que l'état spatial x, cela montre clairement la relation

$$\varphi^t \circ \varphi^s = \varphi^{t+s}.$$

□

Remarquons que cette propriété est très semblable aux propriétés d'une exponentielle. La notation en exposant de la variable temporelle t dans le flot φ l'indique très clairement et est donc justifiée dans le cas d'un système décrit par une équation différentielle ordinaire ou partielle.

Grâce à la nouvelle définition du flot dans le cas d'un système décrit par une équation aux dérivées partielles, la définition de l'opérateur de Koopman est tout à fait similaire.

Définition 5. *Soient un réel t positif et le flot φ d'un système décrit par une* équation différentielle partielle. *Alors l'opérateur de Koopman U^t appliqué à une* fonctionnelle f *appelée* observable *est défini pour toute condition initiale u_0 par l'équation*

$$\left(U^t f\right)(u_0) := f\left(\varphi^t(u_0)\right).$$

Afin de rendre ces définitions plus concrètes, illustrons-les à l'aide d'un exemple simple.

Exemple 3. *Soit un système décrit par une équation de la chaleur qui est une équation aux dérivées partielles qui s'exprime sous la forme*

$$\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial x^2}$$

avec la variable temporelle t qui est positive et la variable spatiale x qui appartient à l'intervalle compact $[0;1]$. À des fins de simplicité, supposons que la solution u s'exprime sous la forme $u(x,t) = X(x)\,T(t)$, de sorte que l'équation aux dérivées partielles s'exprime comme étant

$$\dot{T}(t)\,X(x) = c^2 T(t)\,X''(x).$$

Supposons également que le système est muni d'une condition initiale de la forme

$$u(x,0) = \sum_{k=0}^{n} \sigma_k \sin(k\pi x) =: u_0(x)$$

où l'ensemble de ces conditions initiales fonctionnelles sera notée $\mathcal{F}$, où la variable spatiale x appartient au même intervalle compact et où les coefficients σ_k sont réels ainsi que de conditions au bords de Dirichlet exprimées par

$$u(0,t) = u(1,t) = 0$$

quelle que soit la valeur de la variable temporelle t. Pour la résolution de l'équation, cherchons toutes les solutions non nulles sachant que la solution nulle respecte le système pour une condition initiale nulle. Par conséquent, nous pouvons diviser les deux membres de l'équation différentielle exprimée en T et en X par le produit $T(t)\,X(x)$ de sorte à obtenir la relation

$$\frac{\dot{T}(t)}{T(t)} = c^2 \frac{X''(x)}{X(x)}.$$

Étant donné que chaque membre dépend uniquement d'une variable différente, ces deux membres sont en réalité des constantes. Afin de ne pas retomber sur la solution identiquement nulle, supposons que cette constante soit strictement négative et sera donc notée $-\beta^2$. Dans ce cas, l'équation aux dérivées partielles se transforme en une double équation donnée par

$$\dot{T}(t) + \beta^2 T(t) = 0 = X''(x) + \frac{\beta^2}{c^2} X(x).$$

Ainsi, comme ces dernières équations différentielles sont ordinaires et linéaires à coefficients constants, nous pouvons directement déduire que l'inconnue admet une expression de la forme

$$u(x,t) = X(x)\,T(t) = \left(E_{\frac{\beta}{c}} \cos\left(\frac{\beta}{c}x\right) + F_{\frac{\beta}{c}} \sin\left(\frac{\beta}{c}x\right)\right) \mathrm{e}^{-\beta^2 t}.$$

Or, les conditions aux bords de Dirichlet s'expriment grâce à la forme de la solution u sous la forme

$$E_{\frac{\beta}{c}} \mathrm{e}^{-\beta^2 t} = \left(E_{\frac{\beta}{c}} \cos\left(\frac{\beta}{c}\right) + F_{\frac{\beta}{c}} \sin\left(\frac{\beta}{c}\right)\right) \mathrm{e}^{-\beta^2 t} = 0.$$

Ce qui implique que, d'une part, le coefficient $E_{\frac{\beta}{c}}$ est nul par la positivité stricte de l'exponentielle et que, d'autre part, le coefficient β s'exprime sous la forme

$$ck\pi$$

où le coefficient k est un nombre entier. Par ces constatations, l'expression de la solution u se réduit donc à

$$u(x,t) = F_{k\pi} \sin(k\pi x)\, \mathrm{e}^{-c^2k^2\pi^2 t}.$$

Nous remarquons également que toute combinaison linéaire de la solution de l'équation aux dérivées partielles munie uniquement des conditions aux bords de Dirichlet est également solution de ce même système, la solution u s'exprime donc en réalité comme étant

$$u(x,t) = \sum_{k\in\mathbb{Z}} \alpha_k \sin(k\pi x)\, \mathrm{e}^{-c^2k^2\pi^2 t}.$$

Finalement, la condition initiale implique que

$$\sum_{k\in\mathbb{Z}} \alpha_k \sin(k\pi x) = \sum_{k=0}^{n} \sigma_k \sin(k\pi x),$$

ce qui signifie que la solution finale du système de ce dernier exemple s'exprime comme étant

$$\boxed{u(x,t) = \sum_{k=0}^{n} \sigma_k \sin(k\pi x)\, \mathrm{e}^{-c^2k^2\pi^2 t}}.$$

Par conséquent, la Définition 4 implique que le flot s'exprime comme étant

$$\varphi^t(u_0)(x) = \sum_{k=0}^{n} \sigma_k \sin(k\pi x)\, \mathrm{e}^{-c^2k^2\pi^2 t}.$$

De nouveau, le flot pour un temps t fixé et pour une condition initiale fixée u_0 n'est plus un vecteur comme dans le cas d'une équation différentielle ordinaire mais bel et bien une fonction définie dans le domaine spatial du système. Enfin, pou toute fonction observable f, la Définition 5 fournit

$$\left(U^t f\right)(u_0) = f\left(\varphi^t(u_0)\right).$$

En particulier, nous pouvons appliquer l'opérateur de Koopman à la fonction moyenne ξ définie tout naturellement par

$$\begin{array}{rccc} \xi: & \mathcal{F} & \to & \mathbb{R} \\ & f & \rightsquigarrow & \int_0^1 f(x)\,\mathrm{d}x \end{array}.$$

Dans ce cas, nous obtenons

$$\left(U^t\xi\right)(u_0) = \xi\left(\varphi^t(u_0)\right) = \int_0^1 \sum_{k=0}^{n} \sigma_k \sin(k\pi x)\,\mathrm{e}^{-c^2k^2\pi^2t}\mathrm{d}x = \sum_{k=1}^{\left\lfloor\frac{n+1}{2}\right\rfloor} \frac{2\sigma_{2k-1}}{(2k-1)\,\pi}\mathrm{e}^{-c^2k^2\pi^2t}.$$

Comme énoncé dans la sous-section précédente, les propriétés 2, 3 et 4 restent tout à fait valables.

Propriété 6. *La famille d'opérateurs de Koopman pour un système de flot φ décrit par la Définition 4 est un* semi-groupe, *c'est-à-dire que l'opérateur U^0 est l'opérateur identité et que tous les scalaires t et s vérifient*

$$U^tU^s = U^{t+s}.$$

Démonstration.

D'une part, le fait que l'opérateur U^0 soit l'opérateur identité est trivial. En effet la Définition 5 indique que pour une fonctionnelle arbitraire f,

$$U^0 f = f \circ \varphi^0 = f$$

étant donné que la Propriété 5 indique que la fonctionnelle φ^0 est l'identité.

D'autre part, la Définition 5 indique que

$$U^tU^sf = U^t(f\circ\varphi^s) = f\circ\varphi^s\circ\varphi^t \tag{1.7}$$

pour des réels positifs arbitraires t et s et pour toute fonctionnelle f arbitraire. Or, la Propriété 5 donne

$$f\circ\varphi^s\circ\varphi^t = f\circ\varphi^{t+s}. \tag{1.8}$$

Enfin, la Définition 5 fournit

$$f\circ\varphi^{t+s} = U^{t+s}f. \tag{1.9}$$

Par conséquent, les équations (1.7), (1.8) et (1.9) impliquent que

$$U^tU^sf = U^{t+s}f.$$

Finalement, comme la fonctionnelle f est arbitraire, cela montre clairement que

$$U^tU^s = U^{t+s}.$$

□

De nouveau, remarquons que cette propriété est très semblable aux propriétés d'une exponentielle. La notation en exposant de la variable temporelle t dans l'opérateur de Koopman U^t l'indique très clairement et est donc justifiée. Remarquons également que cette propriété généralise la Propriété 5 étant donné que le flot n'est rien d'autre que l'opérateur de Koopman appliqué à la fonction identité. La seconde propriété est tout aussi aisée à démontrer.

Propriété 7. *Supposons que chaque opérateur de Koopman pour un système de flot φ décrit par la Définition 4 admet un domaine commun qui est l'ensemble des fonctions continues à valeur complexe de norme supremum finie et que les fonctions φ^s et $t \rightsquigarrow \varphi^t(u)$ soient continues pour tout temps réel positif s et pour toute condition initiale u. Alors, le semi-groupe d'opérateur de Koopman est ponctuellement continu, c'est-à-dire que toutes les fonctions f du domaine et toutes les conditions initiales u_0 vérifient la relation*

$$\lim_{t \downarrow 0^+} \left\| \left(U^t f\right)(u_0) - f(u_0) \right\| = 0.$$

Démonstration.

Soient une fonction f du domaine commun de la famille d'opérateurs de Koopman et une condition initiale u_0. Alors, la continuité d'une norme fournit l'équation

$$\lim_{t \downarrow 0^+} \left\| \left(U^t f\right)(u_0) - f(u_0) \right\| = \left\| \lim_{t \downarrow 0^+} \left(U^t f\right)(u_0) - f(u_0) \right\|$$

où la limite sera toujours définie dans le sens fort sauf mention contraire. Ensuite, la définition de l'opérateur de Koopman implique que

$$\left\| \lim_{t \downarrow 0^+} \left(U^t f\right)(u_0) - f(u_0) \right\| = \left\| \lim_{t \downarrow 0^+} f\left(\varphi^t(u_0)\right) - f(u_0) \right\|.$$

La prochaine étape consiste à exploiter la continuité de la fonction f pour obtenir l'égalité

$$\left\| \lim_{t \downarrow 0^+} f\left(\varphi^t(u_0)\right) - f(u_0) \right\| = \left\| f\left(\lim_{t \downarrow 0^+} \varphi^t(u_0) \right) - f(u_0) \right\|.$$

Ensuite, par hypothèse de continuité sur le flot φ imposée dans l'énoncé de la propriété, nous déduisons la relation

$$\left\| f\left(\lim_{t \downarrow 0^+} \varphi^t(u_0) \right) - f(u_0) \right\| = \left\| f\left(\varphi^0(u_0)\right) - f(u_0) \right\|.$$

Enfin, la Propriété 5 implique que

$$\left\| f\left(\varphi^0(u_0)\right) - f(u_0) \right\| = 0$$

et ces cinq équations permettent de montrer la thèse. □

Quant aux semi-groupes fortement continus $(T^t)_{t\geq 0}$, c'est-à-dire que pour toute fonction f, nous obtenons la relation

$$\lim_{t\downarrow 0^+} \left\| U^t f - f \right\| = 0$$

selon une certaine norme $\|\bullet\|$ de l'espace considéré, énormément de recherches y sont effectuées encore aujourd'hui.

Enfin, comme énoncé dans la sous-section précédente, la propriété de linéarité rend l'opérateur de Koopman très intéressante à employer dans le cadre de l'identification des équations aux dérivées partielles. En effet, cette propriété est tout à fait valable indépendamment du fait que le système soit linéaire ou non. Exploiter une linéarité dans un système qui ne l'est pas au départ donne une excellente motivation à l'emploi de l'opérateur de Koopman pour identifier des équations aux dérivées partielles.

Propriété 8. *Soient un réel t positif et le flot φ d'un système décrit par une équation différentielle partielle. Alors l'opérateur de Koopman U^t est linéaire.*

Démonstration.

Soit un système décrit par une équation différentielle partielle. Dans ce cas, nous choisissons arbitrairement un temps t positif, deux fonctionnelles f et g, deux scalaires α et β et une condition initiale u_0. Alors la Définition 5 implique que

$$U^t \left(\alpha f + \beta g\right)(u_0) = \left(\alpha f + \beta g\right)\left(\varphi^t\left(u_0\right)\right). \tag{1.10}$$

Ensuite, la définition d'une combinaison linéaire de fonctionnelles indique que

$$\left(\alpha f + \beta g\right)\left(\varphi^t\left(u_0\right)\right) = \alpha f\left(\varphi^t\left(u_0\right)\right) + \beta g\left(\varphi^t\left(u_0\right)\right). \tag{1.11}$$

La prochaine étape consiste à appliquer de nouveau la Définition 5 aux fonctionnelles f et g pour obtenir l'égalité

$$\alpha f\left(\varphi^t\left(u_0\right)\right) + \beta g\left(\varphi^t\left(u_0\right)\right) = \alpha\left(U^t f\right)(u_0) + \beta\left(U^t g\right)(u_0). \tag{1.12}$$

Enfin, la définition d'une combinaison linéaire de fonctionnelles implique que

$$\alpha\left(U^t f\right)(u_0) + \beta\left(U^t g\right)(u_0) = \left(\alpha U^t f + \beta U^t g\right)(u_0). \tag{1.13}$$

Par conséquent, les équations (1.10), (1.11), (1.12) et (1.13) fournissent la relation

$$U^t \left(\alpha f + \beta g\right)(u_0) = \left(\alpha U^t f + \beta U^t g\right)(u_0).$$

Comme la condition initiale u_0 a été fixée arbitrairement, cela signifie que

$$U^t \left(\alpha f + \beta g\right) = \alpha U^t f + \beta U^t g.$$

Cette dernière équation indique que, comme les scalaires α et β et les fonctionnelles f et g sont fixées arbitrairement, U^t est bel et bien linéaire dans le cas d'un système décrit par une équation aux dérivées partielles. □

Finalement, étant donné que la définition de l'opérateur de Koopman est tout à fait similaire selon le caractère ordinaire ou partiel de l'équation différentielle décrivant le système, il en va de même pour le générateur de Lie.

Définition 6. *Soit un système dynamique dont l'opérateur de Koopman est décrit par la Définition 5. Alors le générateur de Lie L est défini pour toute condition initiale u_0 par l'égalité*

$$(Lf)(u_0) = \lim_{t \downarrow 0^+} \frac{(U^t f)(u_0) - f(u_0)}{t}$$

pour toute fonction f du domaine de définition de l'opérateur de Koopman telle que la limite précédente soit bien définie.

L'article [11] suggère un raisonnement similaire dans le cas d'un système défini avec une équation aux dérivées partielles en faisant intervenir la dérivée de Gâteaux étant donné que la fonction observable f dépend cette fois-ci d'une variable fonctionnelle et non plus vectorielle. La dérivée de Gâteaux est introduite dans le livre [4] et se définit de la manière suivante.

Définition 7. *Soient deux espaces vectoriels U et V et soit une fonction f définie sur l'ensemble U à valeur dans l'espace V. Alors, la dérivée de Gâteaux de la fonction f en un point v de l'ensemble U, si elle existe, est définie par une fonction de l'ensemble U qui associe chaque élément u de l'ensemble U à l'élément*

$$(D_v f)(u) := \lim_{\delta \to 0} \frac{f(u + \delta v) - f(u)}{\delta}.$$

Remarquons que cette dérivée généralise la dérivée usuelle. En effet, pour une application réelle dérivable f et pour tout réel x, la Définition 7 fournit

$$\frac{\mathrm{d}}{\mathrm{d}x} f(x) = \lim_{\delta \to 0} \frac{f(x + \delta \cdot 1) - f(x)}{\delta} = (D_1 f)(x).$$

Il n'est donc pas étonnant que cette dérivée admette des propriétés analogues à celles du gradient. La première concerne la dérivée de Gâteaux d'une combinaison linéaire de fonctions.

Propriété 9. *Soit un élément v dans un espace vectoriel U. Alors, l'opérateur D_v décrit dans la Définition 7 est linéaire.*

Démonstration.

Soient deux scalaires α et β, un élément u de l'espace U et deux fonctions f et g définies sur l'ensemble U dont la dérivée de Gâteaux en l'élément v soit définie. Alors, la Définition 7 implique que

$$(D_v(\alpha f + \beta g))(u) = \lim_{\delta \to 0} \frac{(\alpha f + \beta g)(u + \delta v) - (\alpha f + \beta g)(u)}{\delta}. \tag{1.14}$$

Ensuite, la définition d'une combinaison linéaire de fonctions fournit

$$\lim_{\delta\to 0}\frac{(\alpha f+\beta g)(u+\delta v)-(\alpha f+\beta g)(u)}{\delta}=\lim_{\delta\to 0}\frac{\alpha f(u+\delta v)+\beta g(u+\delta v)-\alpha f(u)-\beta g(u)}{\delta}. \tag{1.15}$$

Or, nous obtenons grâce à la dérivabilité des fonctions f et g au sens de Gâteaux la relation

$$\lim_{\delta\to 0}\frac{\alpha f(u+\delta v)+\beta g(u+\delta v)-\alpha f(u)-\beta g(u)}{\delta}=\alpha\left(D_v f\right)(u)+\beta\left(D_v g\right)(u). \tag{1.16}$$

De plus, nous utilisons de nouveau la définition d'une combinaison linéaire de fonctions pour obtenir

$$\alpha\left(D_v f\right)(u)+\beta\left(D_v g\right)(u)=\left(\alpha D_v f+\beta D_v g\right)(u). \tag{1.17}$$

Par conséquent, les équations (1.14), (1.15), (1.16) et (1.17) donnent

$$\left(D_v\left(\alpha f+\beta g\right)\right)(u)=\left(\alpha D_v f+\beta D_v g\right)(u).$$

Finalement, comme l'élément u est arbitraire, nous déduisons l'équation

$$D_v\left(\alpha f+\beta g\right)=\alpha D_v f+\beta D_v g$$

et comme les fonctions f et g ainsi que les scalaires α et β sont arbitraires, cette dernière égalité montre bel et bien la linéarité de l'opérateur D_v. □

Une autre propriété intéressante de la dérivée de Gâteaux en commun avec le gradient est la règle de Leibniz.

Propriété 10. *Soit un élément v dans un espace vectoriel U et soient deux fonctions f et g définies dans l'ensemble U et à valeur scalaire dont la dérivée de Gâteaux est définie en l'élément v. Alors,*

$$D_v\left(f\cdot g\right)=f\cdot D_v g+g\cdot D_v f.$$

Démonstration.

Soit un élément u de l'ensemble U. Alors, la Définition 7 implique que

$$\left(D_v\left(f\cdot g\right)\right)(u)=\lim_{\delta\to 0}\frac{f(u+\delta v)\,g(u+\delta v)-f(u)\,g(u)}{\delta}. \tag{1.18}$$

Ensuite, la linéarité de la limite fournit

$$\begin{aligned}&\lim_{\delta\to 0}\tfrac{f(u+\delta v)g(u+\delta v)-f(u)g(u)}{\delta}\\ =\;&\lim_{\delta\to 0}\left(f(u+\delta v)\,\tfrac{g(u+\delta v)-g(u)}{\delta}\right)+\lim_{\delta\to 0}\tfrac{f(u+\delta v)-f(u)}{\delta}g(u).\end{aligned} \tag{1.19}$$

Or, la Définition 7 indique que la dérivabilité au sens de Gâteaux implique la continuité. Nous pouvons donc déduire la relation

$$\begin{aligned}&\lim_{\delta\to 0}\left(f(u+\delta v)\,\tfrac{g(u+\delta v)-g(u)}{\delta}\right)+\lim_{\delta\to 0}\tfrac{f(u+\delta v)-f(u)}{\delta}g(u)\\ =\;&f(u)\left(D_v g\right)(u)+\left(D_v f\right)(u)\,g(u)\end{aligned} \tag{1.20}$$

de nouveau grâce à la Définition 7. Par conséquent, les équations (1.18), (1.19) et (1.20) donnent

$$(D_v (f \cdot g))(u) = f(u)(D_v g)(u) + (D_v f)(u)\, g(u).$$

Finalement, étant donné que l'élément u est arbitraire, nous obtenons la thèse. □

Afin de rendre la définition de dérivée de Gâteaux plus concrète, illustrons-la à l'aide d'un exemple.

Exemple 4. *Considérons deux scalaires réels x et y arbitraires ainsi qu'une fonction f définie par*

$$\begin{array}{rccc} f: & \mathbb{R}^2 & \to & \mathbb{R}^2 \\ & (x,y) & \rightsquigarrow & (\mathrm{e}^{x+y}, \cos(x-y)) \end{array}.$$

Alors la Définition 7 donne

$$\left(D_{(1,1)} f\right)(x,y) = \lim_{\delta \to 0} \frac{\left(\mathrm{e}^{x+\delta\cdot 1+y+\delta\cdot 1}, \cos(x+\delta\cdot 1-(y+\delta\cdot 1))\right) - (\mathrm{e}^{x+y}, \cos(x-y))}{\delta}.$$

Ensuite, les simplifications fournissent

$$\left(D_{(1,1)} f\right)(x,y) = \lim_{\delta \to 0} \frac{\left(\mathrm{e}^{x+y+2\delta}, \cos(x-y)\right) - (\mathrm{e}^{x+y}, \cos(x-y))}{\delta}.$$

Par conséquent, la définition des opérations sur les vecteurs fournissent

$$\left(D_{(1,1)} f\right)(x,y) = \lim_{\delta \to 0} \left(\frac{\mathrm{e}^{x+y+2\delta} - \mathrm{e}^{x+y}}{\delta}, \frac{0}{\delta} \right).$$

Ainsi, la définition d'une limite dans l'espace $\mathbb{R}^2$ implique la relation

$$\left(D_{(1,1)} f\right)(x,y) = \left(\lim_{\delta \to 0} \frac{\mathrm{e}^{x+y+2\delta} - \mathrm{e}^{x+y}}{\delta}, \lim_{\delta \to 0} \frac{0}{\delta} \right).$$

Ensuite, par propriétés sur les limites, nous obtenons la relation

$$\left(D_{(1,1)} f\right)(x,y) = \left(2\lim_{\delta \to 0} \frac{\mathrm{e}^{x+y+2\delta} - \mathrm{e}^{x+y}}{2\delta}, 0 \right).$$

Comme nous reconnaissons la définition d'une dérivée usuelle, la relation devient

$$\left(D_{(1,1)} f\right)(x,y) = \left(2\exp'(x+y), 0\right).$$

Finalement, par la dérivation de fonctions usuelles,

$$\boxed{\left(D_{(1,1)} f\right)(x,y) = \left(2\exp'(x+y), 0\right) = \left(2e^{x+y}, 0\right)}.$$

Maintenant que nous sommes un peu plus familier avec la dérivée de Gâteaux, énonçons et montrons une propriété exprimant le générateur de Lie en fonction de la dérivée de Gâteaux [11].

Propriété 11. *Soit le système à identifier décrite par l'équation (1.1). Alors si une fonctionnelle f admet une dérivée de Gâteaux en l'élément*

$$\sum_{i=1}^{n} c_i W_i(u),$$

alors le générateur de Lie définie dans la Définition 6 admet comme expression analytique

$$(Lf)(u) = \left(D_{\sum_{i=1}^{n} c_i W_i(u)} f \right)(u)$$

pour toute fonction u.

Démonstration.

Le système (1.1) indique qu'il suffit de montrer la relation

$$(Lf)(u) = \left(D_{\frac{\partial}{\partial t}\varphi^t(u)} f \right)(u).$$

Étant donné que ces membres sont des limites selon une certaine norme, il suffit alors de montrer la relation

$$\lim_{t\to 0} \left\| \frac{(U^t f)(u) - f(u)}{t} - \frac{f\left(u + t\frac{\partial}{\partial t}\varphi^t(u)\right) - f(u)}{t} \right\| = 0.$$

Après simplification de l'argument de la norme du premier membre de l'égalité à montrer, ce membre peut se réexprimer comme étant

$$\lim_{t\to 0} \left\| \frac{(U^t f)(u) - f\left(u + t\frac{\partial}{\partial t}\varphi^t(u)\right)}{t} \right\|.$$

Ensuite, la Définition 5 permet de réécrire l'expression sous la forme

$$\lim_{t\to 0} \left\| \frac{f(\varphi^t(u)) - f\left(u + t\frac{\partial}{\partial t}\varphi^t(u)\right)}{t} \right\|.$$

De plus, grâce au Théorème des accroissements finis et sachant qu'un flot représente un semi-groupe, cette limite devient

$$\lim_{t\to 0} \left\| \frac{f\left(u + t\left(\frac{\partial}{\partial t}\varphi^t(u)\big|_{t=\tau}\right)\right) - f\left(u + t\frac{\partial}{\partial t}\varphi^t(u)\right)}{t} \right\|.$$

Cette limite vaut bel et bien zéro via la continuité d'une norme et des fonctions f étant donné l'existence de sa dérivée de Gâteaux. □

1.5 Méthode de Koopman

Dans cette section, nous allons décrire en détail et interpréter une méthode d'identification du système (1.1) expliquée dans l'article [11] grâce à la notion d'opérateur de Koopman dans le cas d'un système décrit par une équation aux dérivées partielles. Elle mettra en évidence les motivations de cette nouvelle méthode par rapport à d'autres méthodes existantes qui sont explicitées dans la première section.

Soit un ensemble $\{\varphi^{it_s}(u_k) : 1 \leq k \leq N_u, 0 \leq i \leq N_t\}$ de N_u trajectoires discrétisées en $N_t + 1$ pas de temps dans un espace de Hilbert $\mathcal{U}$ dont leur flot φ^t vérifie l'équation (1.1) et dont le pas de temps est constant et est un réel strictement positif t_s fixé. La première étape de l'algorithme consiste à construire les fonctions de base définies

$$\begin{array}{rccl} \forall j \in \{1,\dots,N\}, \xi_j : & \mathcal{U} & \to & \mathbb{R} \\ & u & \rightsquigarrow & \langle W_j(u); w\rangle := \int_X W_j(u)(x)\, w(x)\,\mathrm{d}x \end{array}$$

où l'espace X représente l'espace spatiale et où la fonction w est appelée *fonction de poids* et devra être choisie à l'avance dans l'algorithme. Le guide du choix d'une telle fonction sera explicité dans les chapitres suivants et constitue une contribution de ce travail.

La prochaine étape consiste à construire les matrices Ξ_1 et Ξ_2 appartenant à l'ensemble $\mathbb{R}^{(N_t N_u)\times n}$ où les composantes de la ligne $iN_u + k$ et de la colonne j sont définies par $(U^{it_s}\xi_j)(u_i) = \int_X W_j(\varphi^{it_s}(u_k))(x)\, w(x)\,\mathrm{d}x$ et $\left(U^{(i+1)t_s}\xi_j\right)(u_i)$ respectivement où $i \in \mathbb{N} \cap [0, N_t]$, $j \in \mathbb{N} \cap [1, n]$ et $k \in \mathbb{N} \cap [1, N_u]$. Remarquons au passage que si nous considérons les matrices définies Ξ_ℓ comme étant des fonctions de la fonction de poids w, alors la linéarité de l'intégration implique celle de ces fonctions matricielles.

Finalement, les coefficients c_i du système à identifier (1.1) sont approximés selon la méthode par

$$\hat{c}_i = t_s^{-1} \log\left(\Xi_1^\dagger \Xi_2\right)_{i;1}$$

où le symbole † désigne la pseudo-inverse rappelée dans la définition ci-après. [2]

Définition 8. *Soit une matrice A de taille $m \times n$ à composantes scalaires admettant une décomposition en valeurs singulières de la forme $U\Sigma V^*$ où la matrice U (respectivement la matrice V) est unitaire et est d'ordre m (respectivement d'ordre n) et où la matrice Σ est de taille $m \times n$ dont les éléments non diagonaux sont nuls et dont les éléments de la diagonale appelés valeurs singulières de la matrice A sont des réels positifs. Alors, la pseudo-inverse de Moore-Penrose de cette matrice est définie par la matrice*

$$A^\dagger := V\Sigma^\dagger U^*$$

où la matrice $\Sigma^\dagger$ s'obtient en transposant la matrice Σ et en inversant toutes les composantes non nulles.

Afin de justifier la dénomination de pseudo-inverse, remarquons que dans le cas où la matrice A est effectivement carrée et inversible, nous obtenons

$$A^\dagger = A^{-1}.$$

En effet, la matrice Σ de la décomposition en valeurs singulières de la matrice A évoquée dans la Définition 8 est inversible dont l'inverse correspond donc à la matrice Σ^{-1}. Par conséquent, par le caractère unitaire des matrices U et V de la décomposition en valeurs singulières de la matrice A, la pseudo-inverse de Moore-Penrose de cette dernière matrice devient

$$A^{\dagger} := V\Sigma^{-1}U^{*} = (U\Sigma V^{*})^{-1} = A^{-1}.$$

D'autre part, dans le cas où la matrice $\Xi_1^{\dagger}\Xi_2$ n'admet aucune valeur propre réelle négative ou nulle, son logarithme est défini comme étant l'unique matrice vérifiant l'équation

$$\mathrm{e}^{\log\left(\Xi_1^{\dagger}\Xi_2\right)} = \Xi_1^{\dagger}\Xi_2$$

et dont chaque valeur propre admet une partie imaginaire qui soit strictement comprise entre $-\pi$ et π [8].

Afin de comprendre ce que représentent les différents éléments de la méthode, commençons par déterminer la matrice notée $\tilde{U}^{t_s}$ comme étant la meilleure approximation de la représentation matricielle de l'opérateur de Koopman réduit au domaine étant le sous-espace engendré par les fonctions ξ_j. Ce qui signifie que la composante de la i-ème ligne et de la j-ème colonne de la matrice $\tilde{U}^{t_s}$ représente la i-ème composante de la fonction $U^{t_s}\xi_j$ selon la base des fonctions ξ_j. Par conséquent, nous souhaiterions que pour chaque fonction ξ_j, l'expression

$$\sum_{k=1}^{n} \tilde{U}_{k;j}^{t_s}\xi_k$$

s'ajuste au mieux à la fonction $U^{t_s}\xi_j$. En pratique, nous allons nous baser sur les données connues de la méthode et nous souhaiterions donc en particulier que pour chaque fonction ξ_j et pour chaque condition initiale u_i, l'expression

$$\sum_{k=1}^{n} \tilde{U}_{k;j}^{t_s}\xi_k\left(u_i\right)$$

s'ajuste au mieux à la donnée $\left(U^{t_s}\xi_j\right)\left(u_i\right)$, c'est-à-dire à l'élément $\left(\Xi_2\right)_{i;j}$. Selon la définition de la matrice Ξ_1, cette dernière somme peut se réexprimer comme étant

$$\sum_{k=1}^{n} \tilde{U}_{k;j}^{t_s}\left(\Xi_1\right)_{k;j}$$

ou plus simplement l'élément

$$\left(\Xi_1\tilde{U}^{t_s}\right)_{i;j}$$

selon la définition du produit matriciel. Étant donné que nous souhaiterions que ce dernier élément correspondent au mieux à l'élément $\left(\Xi_2\right)_{i;j}$, il est alors équivalent d'énoncer que nous souhaiterions que la matrice $\Xi_1\tilde{U}^{t_s}$ s'ajuste au mieux à la matrice Ξ_2. Ce procédé est possible en déterminant la matrice $\Xi_1\tilde{U}^{t_s}$ minimise l'écart en norme au carré et est connu

sous le nom de *problème des moindres carrés*. La solution de cette minimisation est la matrice $\Xi_1^\dagger \Xi_2$. Ce raisonnement montre ainsi que cette dernière matrice représente l'approximation de l'opérateur de Koopman de domaine réduit correspondant à l'espace vectoriel engendré par les fonctions ξ_j. Cette méthode d'approximation de l'opérateur de Koopman s'appelle la méthode étendue de décomposition en modes dynamique (extended dynamic mode decomposition - EDMD). Cette méthode se base principalement sur la méthode EDMD développée dans l'article [16].

Poursuivons l'interprétation des différents éléments de l'algorithme en fournissant une approximation du générateur de Lie notée $\tilde{L}^t$ sur base de la définition 6 impliquant notamment la relation

$$\left(LU^t \sum_{k=1}^{n} v_k \xi_k\right)(u_i) = \lim_{\tau \to 0} \frac{\left(U^{t+\tau} \sum_{k=1}^{n} v_k \xi_k\right)(u_i) - \left(U^t \sum_{k=1}^{n} v_k \xi_k\right)(u_i)}{\tau}$$

pour tout vecteur $v = \begin{pmatrix} v_1 & \dots & v_n \end{pmatrix}^\top$, pour tout temps positif t et pour toute condition initiale u_i. Pour ce faire, remarquons que la linéarité de l'opérateur de Koopman nous fournit la relation

$$U^t \sum_{k=1}^{n} v_k \xi_k = \sum_{k=1}^{n} v_k U^t \xi_k.$$

Ce qui justifie le fait que la fonction $U^t \sum_{k=1}^{n} v_k \xi_k$ peut être approximée par le vecteur $\tilde{U}^t v$ où la i-ème composante représente l'approximation appliquée à la condition initiale u_i. Nous pouvons donc raisonnablement définir la matrice d'approximation $\tilde{L}^t$ à l'aide de la relation

$$\tilde{L}^t \tilde{U}^t v = \lim_{\tau \to 0} \frac{\tilde{U}^{t+\tau} v - \tilde{U}^t v}{\tau}.$$

Cette approximation nous permet alors de reconnaitre la définition de la dérivée et de déduire l'équation différentielle

$$\tilde{L}^t \tilde{U}^t v = \frac{\mathrm{d}}{\mathrm{d}t} \tilde{U}^t v.$$

Ceci nous permet alors de fournir une autre expression de l'approximation de l'opérateur de Koopman $\tilde{U}^t$. En effet, la résolution de l'équation différentielle nous permet d'une part de déduire la relation

$$\tilde{U}^t v = e^{\tilde{L}^t t} v.$$

D'autre part, étant donné que nous pouvons obtenir la j-ème colonne d'une matrice en la post-multipliant par le vecteur e_j représentant le j-ème vecteur de la base canonique, alors la particularisation du vecteur v à chacune des vecteurs de base nous permet de déduire que la j-ème colonne de la matrice $\tilde{U}^t$ vaut la j-ème colonne de la matrice $e^{\tilde{L}^t t}$. Ainsi, la matrice d'approximation $\tilde{U}^t$ peut également se réexprimer comme étant $e^{\tilde{L}^t t}$. Par conséquent, l'approximation du générateur de Lie $\tilde{L}^{t_s}$ s'exprime comme étant

$$\frac{\log\left(\tilde{U}^{t_s}\right)}{t}$$

où sa première colonne est utilisée pour identifier le système (1.1).

Afin de justifier cette méthode d'identification, appliquons la relation fournie dans la propriété 11 à la fonction $f = \xi_1$. Ainsi, nous obtenons pour toute fonction u la relation

$$\left(L\xi_1\right)(u) = \left(D_{\sum_{i=1}^{n} c_i W_i(u)} \xi_1\right)(u).$$

Or, par définition de la dérivée de Gâteaux et de la fonction ξ_1, la relation peut se réécrire comme étant

$$\left(L\xi_1\right)(u) = \lim_{t\to 0} \frac{\int_X \left(\varphi^t(u) + t\sum_{i=1}^{n} c_i W_i(u)\right)(x)\, w(x)\,\mathrm{d}x - \int_X \varphi^t(u)(x)\, w(x)\,\mathrm{d}x}{t}.$$

Ensuite, grâce à la linéarité de l'intégration, le deuxième membre se simplifie et la relation devient alors

$$\left(L\xi_1\right)(u) = \sum_{i=1}^{n} c_i \int_X W_i(u)(x)\, w(x)\,\mathrm{d}x.$$

Enfin, la définition des fonctions ξ_i permet de réécrire de nouveau le deuxième membre pour obtenir la relation

$$\left(L\xi_1\right)(u) = \sum_{i=1}^{n} c_i \xi_i(u).$$

Par conséquent, comme cette relation est vérifiée pour toute fonction u, alors le générateur de Lie admet comme relation

$$L\xi_1 = \sum_{i=1}^{n} c_i \xi_i.$$

Ainsi, si nous considérons une approximation du générateur de Lie de domaine réduit à l'espace vectoriel générée par les fonctions ξ_j (qui ne correspond pas tout à fait à la matrice $\tilde{L}^{t_s}$), alors cette dernière relation implique que les éléments de la première colonne et de la i-ème ligne correspond au coefficient à identifier c_i. Ce qui justifie la méthode d'identification de cette section.

Remarque 2. *Dans le cas où la matrice A est carrée et inversible, il est très fortement déconseillé de calculer le produit matriciel $A^{-1}B$ pour une matrice quelconque B en inversant explicitement préalablement la matrice A, surtout si l'ordre de cette dernière est beaucoup plus grand que le nombre de colonnes de la matrice B. Ceci est dû au fait que la complexité de l'algorithme d'inversion matricielle est tout à fait déraisonnable. Certains langages de programmation tels que* `MATLAB` *ou* `Julia` *utilisent la commande* `A\B` *pour calculer beaucoup plus efficacement le produit matriciel $A^{-1}B$. Il est donc clair que ce constat est tout à fait similaire pour une matrice A qui ne soit ni nécessairement inversible, ni nécessairement carrée pour le calcul de la matrice $A^{\dagger}B$. Cette même commande* `A\B` *calcule cette même matrice de manière beaucoup plus efficace en* `MATLAB` *et en* `Julia`*.*

Concernant les résultats de la convergence, l'article [11] énonce que si la matrice Ξ_1 est de plein rang, alors les vraies valeurs des coefficients c_i s'expriment comme étant

$$\lim_{t \downarrow 0} \hat{c}_i$$

et ce, malgré les deux approximations différentes du générateur de Lie utilisés pour justifier la méthode d'identification de Koopman.

Nous avons vu dans la sous-section précédente qu'un très gros avantage de cette méthode consiste à exploiter une linéarité dans le système qui ne l'est pas au départ. La description de l'algorithme met en évidence un autre avantage tout aussi considérable par rapport aux autres méthodes existantes décrites dans la première section et qui a déjà été évoqué dans la section précédente : elle ne nécessite pas l'approximation de dérivées temporelles pour approximer les coefficients c_i.

Cependant, le choix de la fonction de poids w est crucial. Nous allons illustrer ce fait à l'aide d'un exemple numérique tiré de l'article [11] exprimé sous la forme

$$\partial_t u = -2u - 0.5(1+u)\partial_x u + (1 - 0.2u)\partial_x^2 u + 0.1\partial_x^3 u$$

où la variable temporelle t est positive et où la variable spatiale x parcourt l'intervalle

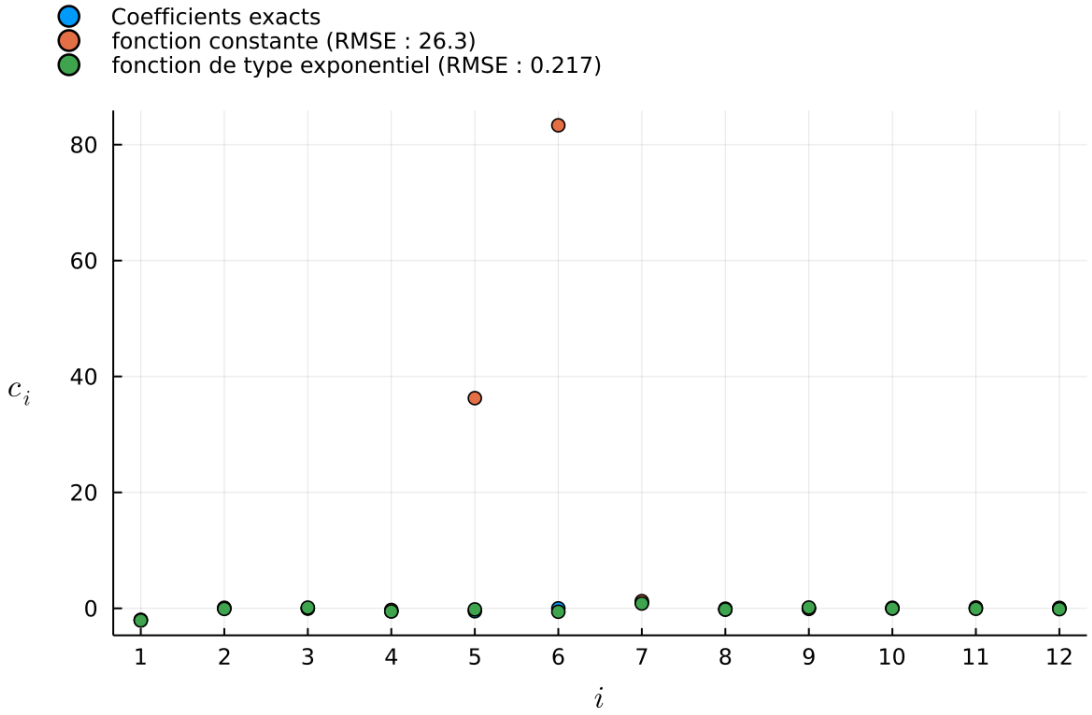


FIGURE 1.2 – Identification du système de l'article [11] à l'aide de deux fonctions de poids différentes, réalisé à l'aide du langage de programmation `Julia`.

compact $[0;5]$. De nouveau, les conditions aux bords de Dirichlet sont exprimées par

$$u(0,t) = u(5,t) = 0$$

quelle que soit la valeur de la variable temporelle t. Cette première simulation se basera sur vingt-cinq trajectoires dont chaque condition initiale s'exprimera sous la forme

$$u_k(x,0) = x(x-5)\cos\left(\pi\left(\frac{\sigma_{1;k}x}{5} + \sigma_{2;k}\right)\right)$$

pour tout état spatial x où les coefficients $\sigma_{i;j}$ sont choisis aléatoirement dans l'intervalle compact $[0;1]$ de manière uniforme. Concernant les autres paramètres, le temps t_s a été fixé à 0,3, le paramètre N_t a été fixé à deux, de sorte que les temps considérés sont 0.0, 0.3 et 0.6 et l'intervalle compact $[0;5]$ a été discrétisé uniformément en $N_x = 52$ points. De plus, les dérivées spatiales sont approximées à l'aide de différences finies centrées d'ordre trois et le calcul du produit scalaire dans l'espace L^2 sera déterminé à l'aide d'une intégration numérique.

La figure 1.2 illustre les résultats pour deux fonctions de poids en fournissant le RMSE définie comme étant

$$\frac{\|\hat{c} - c\|_2}{n}$$

où le vecteur $\hat{c}$ contient les coefficients estimés et où le vecteur c contient les véritables coefficients. Les fonctions de poids en question sont la fonction de poids identiquement égale à un et la fonction de poids w définie par

$$\forall x \in X, w(x) = \exp\left(\frac{1}{0.04x^2 - 1}\right).$$

Clairement, la qualité de l'identification est très différente car avec ces premiers essais, le RMSE peut-être soixante fois plus important notamment à cause des coefficients c_5 et c_6. C'est pourquoi il est important de bien choisir la fonction de poids la plus adéquate et cette étude fera l'objet des chapitres suivants.

Chapitre 2

Premiers résultats sur la méthode de Koopman

À partir de ce chapitre, nous allons commencer la description de ma contribution personnelle approfondissant la méthode de Koopman décrite dans la section 1.5. Plus spécifiquement, nous allons commencer ce chapitre par l'analyse théorique de l'effet de la multiplication d'une fonction de poids par un scalaire sur l'identification des coefficients. Nous allons ensuite dans une seconde section tester la méthode de Koopman avec plusieurs fonctions de poids, calculer l'estimation moyenne et comparer les résultats avec les autres méthodes décrites dans le chapitre précédent.

2.1 Multiplication d'une fonction de poids par un scalaire

Débutons l'analyse de la méthode de Koopman par une propriété remarquable.

Propriété 12.
Soit l'équation aux dérivées partielles (1.1) à identifier et notons $\tilde{c}(w)$ l'estimation du vecteur de coefficients $\hat{c} = \begin{pmatrix} \hat{c}_1 & \dots & \hat{c}_n \end{pmatrix}^\top$ à estimer à l'aide de la méthode de Koopman avec la fonction de poids w. Alors, pour tout scalaire réel non nul α,

$$\tilde{c}(\alpha w) = \tilde{c}(w).$$

Démonstration.

Soient les fonctions matricielles $\tilde{\Xi}_\ell$ pour l'indice ℓ parcourant l'ensemble $\{1;2\}$ associant chaque fonction de poids à la matrice Ξ_ℓ définie dans la section 1.5. Alors, la définition de la fonction vectorielle $\tilde{c}$ ainsi que des fonctions matricielles $\tilde{\Xi}_\ell$ dont la linéarité a été établie dans la description de l'opérateur de Koopman impliquent la relation

$$\tilde{c}(\alpha w) = t_s^{-1} \log\left(\left(\alpha\tilde{\Xi}_1(w)\right)^\dagger \alpha\tilde{\Xi}_2(w)\right). \tag{2.1}$$

Afin de poursuivre les calculs, définissons également le produit $U_\ell \Sigma_\ell V_\ell^*$ comme étant la décomposition en valeurs singulières de la matrice $\tilde{\Xi}_\ell(w)$ correspondante. Ainsi, la matrice $\alpha\tilde{\Xi}_\ell$ admet comme décomposition en valeurs singulières $U_\ell(\alpha\Sigma_\ell)V_\ell^*$. Par conséquent, la définition 8 nous donne la relation

$$\tilde{c}(\alpha w) = t_s^{-1} \log\left(V_1 \alpha^\dagger \Sigma_1^\dagger U_1^* U_2 \alpha \Sigma_2 V_2^*\right). \tag{2.2}$$

Or, rappelons-nous que le scalaire réel α est supposé non-nul. Nous pouvons donc à nouveau utiliser la définition 8 pour obtenir la relation

$$\tilde{c}(\alpha w) = t_s^{-1} \log\left((U_1 \Sigma_1 V_1^* U_2)^\dagger \Sigma_2 V_2^*\right). \tag{2.3}$$

Ce qui conclut la démonstration par la définition des décompositions singulières, des matrices Ξ_ℓ et de la fonction vectorielle $\tilde{c}$. □

Toutefois, si le scalaire α de la propriété que nous venons de démontrer était nul, alors la démonstration indique qu'étant donné que l'argument du logarithme est une matrice nulle et donc n'admettant que des valeurs propres nulles, alors le logarithme matriciel ne serait pas défini. Cette constatation nous permet donc de conclure que le cas où le scalaire α est

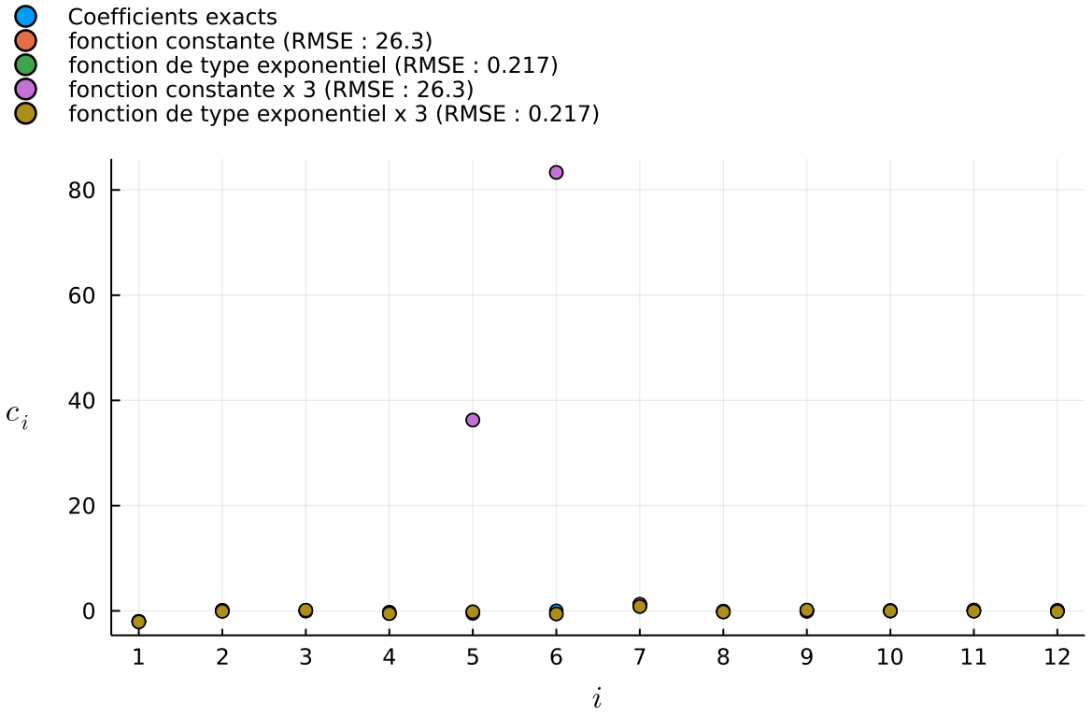


FIGURE 2.1 – Identification du système de l'article [11] à l'aide de plusieurs fonctions de poids différentes, réalisé à l'aide du langage de programmation `Julia`.

nul n'est pas intéressant et que, par extension, *la fonction de poids nulle n'est jamais intéressante à choisir*. De manière plus générale, cette propriété nous informe sur le fait que *deux fonctions non nulles linéairement dépendantes fournissent exactement la même estimation des coefficients à identifier*. Ce constat est illustré dans la figure 2.1 qui emploie les mêmes fonctions de poids que pour la figure 1.2. De plus, nous effectuons la même identification pour chaque fonction de poids multipliée par trois. Nous constatons dès lors une identification identique par paire de fonctions de poids linéairement dépendantes. Comme ces résultats sont en concordance avec la propriété que nous venons de démontrer, ils valident donc l'implémentation de la méthode d'identification de l'article [11].

Remarquons également que nous n'avons aucune propriété sur l'effet de la somme de fonctions de poids sur l'estimation des coefficients par la méthode de Koopman puisque nous n'en avons pas non plus sur la pseudo-inverse d'une somme de matrices.

2.2 Estimation moyenne des coefficients

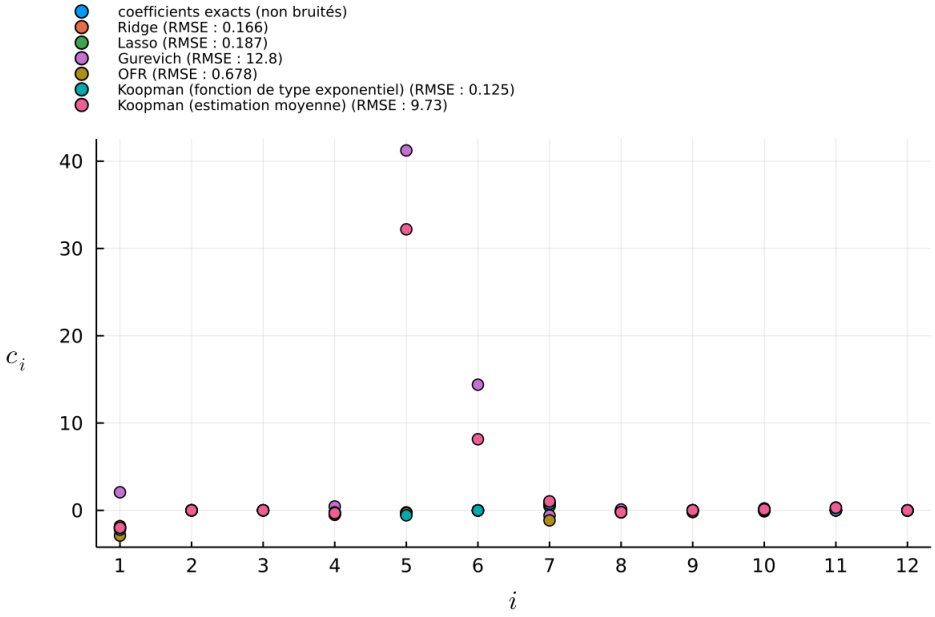
Dans cette section, nous allons tester la méthode de Koopman avec plusieurs fonctions de poids et nous allons déterminer l'estimation moyenne de ces coefficients. Nous appellerons ce procédé une *estimation moyenne*. Nous allons également comparer cette estimation moyenne avec les autres méthodes du chapitre précédent. La fonction de poids employée pour la méthode de l'article [7] est la même que celle employée dans l'article [11].

Pour ce faire, nous allons considérer deux ensembles de données caractérisés dans la table 2.1. Pour chacun de ces ensembles de données, nous considérerons également le cas où un bruit normal d'écart type σ valant 10^{-2} a été ajouté.

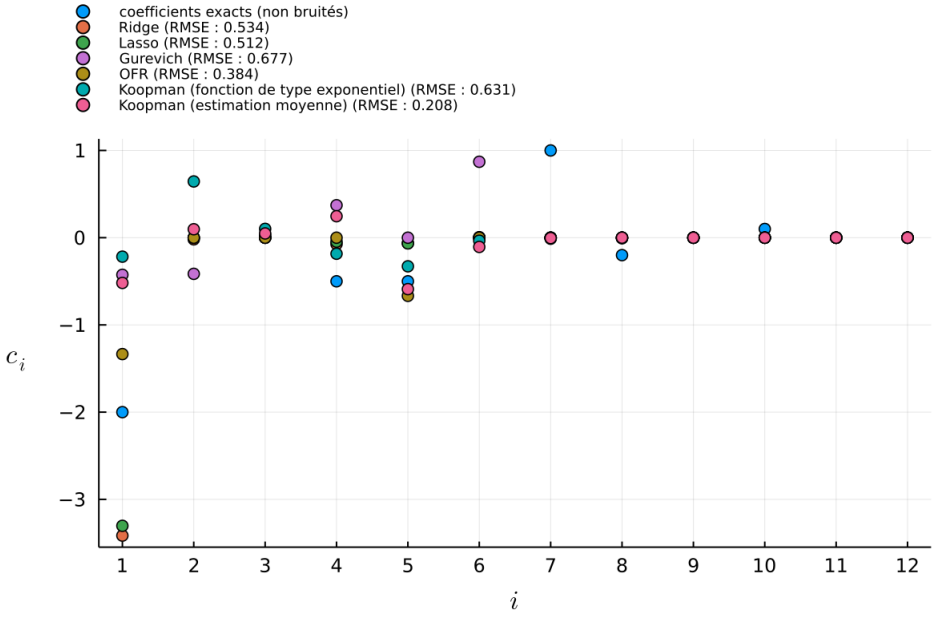
Concernant l'estimation moyenne, les fonctions de poids ont été choisies grâce au package `ExprRules` de `Julia` qui se base sur des fonctions initiales, des compositions et opérations sur celles-ci ainsi qu'un nombre maximal de ces deux procédés. Dans ce cas, les

	Sur base de l'article [11]	**Équation de Burgers [15]**
$\partial_t u$	$-2u$ $-0.5(1+u)\partial_x u$ $+(1-0.2u)\partial_x^2 u$ $+0.1\partial_x^3 u$	$0.01\partial_x^2 u - u\partial_x u$
$u_k(x,0)$	$x(x-5)\cos\left(\pi\left(\frac{ax}{5}+b\right)\right)$	$x(x-1)\cos(\pi(ax))$
Conditions au bord	$u_k(0,t) = u_k(5,t) = 0$	$u_k(0,t) = u_k(1,t) = 0$
N_u	25	25
X discrétisé (pas spatial constant)	$[0;5]$	$[0;1]$
N_x	52	52
t_s	0.3	0.3
N_t	2	10

TABLE 2.1 – Caractérisation des données considérées dans cette section.



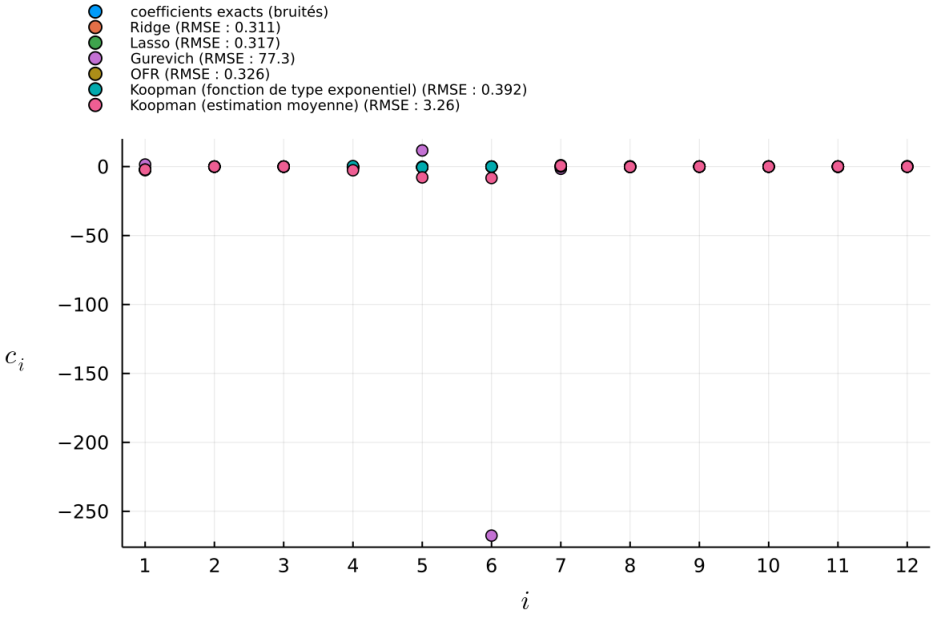
(a) Données sur base de l'article [11]



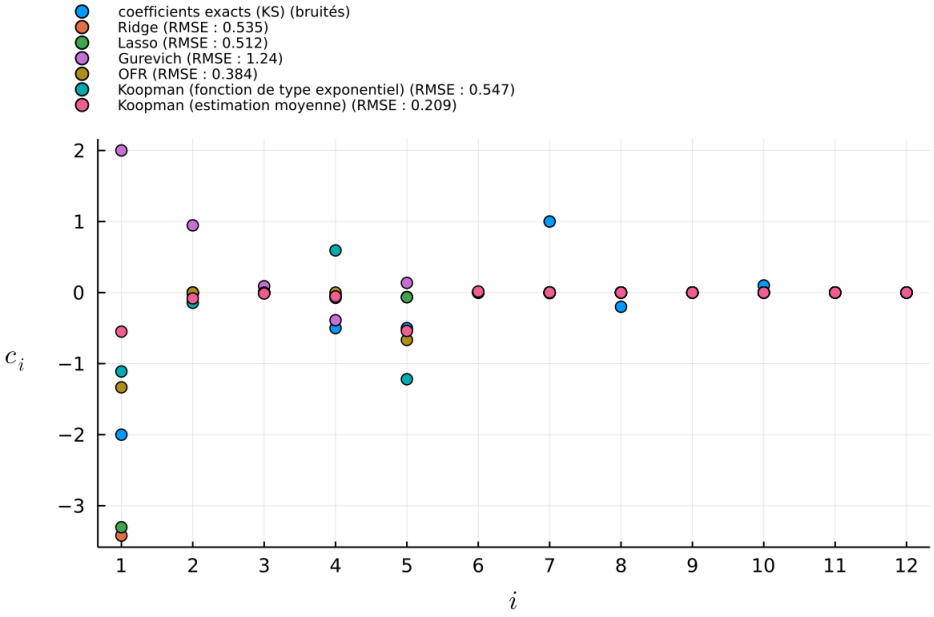
(b) Données sur base de l'article [15]

FIGURE 2.2 – Comparaison des méthodes pour le cas non bruité, réalisée à l'aide du langage de programmation `Julia`.

fonctions de base sont la fonction identité et les fonctions constantes appartenant à l'ensemble $\{1; 2; e; 3; 4; 5; 6; 7; 8; 9; 10\}$. Les compositions et opérations sont la somme, le produit, l'exponentiation et les fonctions trigonométriques sin et cos. Le nombre maximal de com-



(a) Données sur base de l'article [11]



(b) Données sur base de l'article [15]

FIGURE 2.3 – Comparaison des méthodes pour le cas bruité, réalisée à l'aide du langage de programmation `Julia`.

positions et opérations a été fixé à un. Par exemple, la fonction f associant la variable x à la valeur $\sin(x) + 2$ fait partie des fonctions de poids considérées. En effet, les fonctions initiales sont la fonction identité et la fonction valant constamment deux, l'opération choisie

est la somme et la composition choisie est la composition entre la fonction sin et la fonction identité. En vertu de la propriété de la section précédente, les fonctions identiques à une autre à une constante près sont ignorées.

Dans la figure 2.2 traitant du cas non bruité, nous pouvons remarquer notamment dans la première sous-figure traitant des données sur base de l'article [11] que l'estimation moyenne n'est pas adaptée. En effet, le RMSE est presque 80 fois plus grand que le RMSE associée à la méthode de Koopman avec une fonction de poids adaptée choisie dans l'article [11]. De plus, cette estimation moyenne présente un RMSE nettement moins bon que les autres méthodes nécessitant l'approximation des dérivées temporelles et spatiales (mis à part la méthode de l'article [7]).

Cette tendance se confirme dans la figure 2.3 traitant du cas bruité. Cependant, sur base des deux ensembles de données, nous pouvons noter que le RMSE ne semble pas pas augmenter significativement quelle que soit la méthode employée.

Ces résultats nous permettent de conclure la nécessité de choisir de manière automatique une fonction de poids pour identifier le système (1.1). Le chapitre suivant poursuivra la description de ma contribution personnelle en explicitant la recherche d'une telle méthode automatique qui se focalisera sur une méthode d'optimisation.

Chapitre 3

Méthodes d'optimisation

Nous poursuivons la description de ma contribution personnelle dans ce chapitre ayant pour objectif de déterminer une méthode adéquate d'optimisation permettant d'identifier le système (1.1). À cette fin, nous allons débuter par une première section décrivant la méthode heuristique d'optimisation utilisée par défaut dans le package `Optim` de `Julia` qui est la méthode Nelder-Mead [14]. Ensuite, nous allons détailler les méthodes d'optimisation qui seront considérées et comparées tout au long du chapitre. La prochaine section se portera sur la notion de gradient stochastique et commencera par la motivation d'une telle notion. Enfin, une dernière section sera consacrée à la comparaison des méthodes d'optimisation.

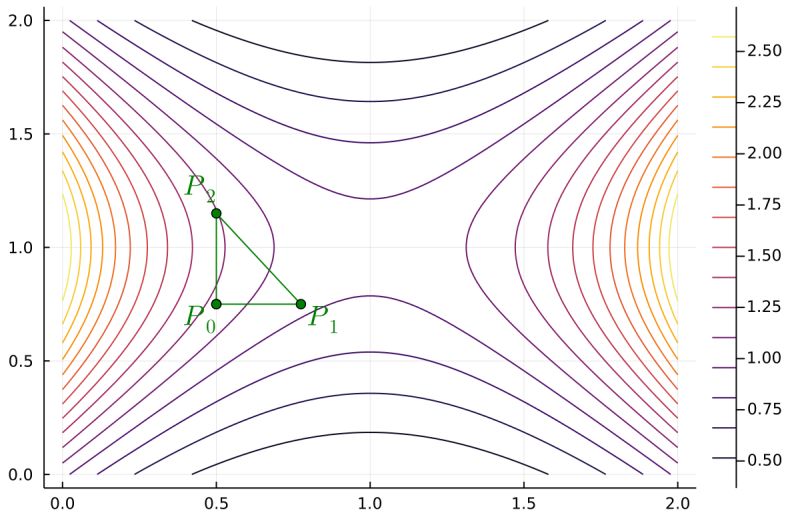


FIGURE 3.1 – Exemple de simplexe pour une fonction à deux variables à minimiser, réalisé à l'aide du langage de programmation `Julia`.

3.1 Méthode de Nelder-Mead

L'objectif de la méthode itérative décrite dans l'article [14] est de minimiser sans contrainte une fonction $f : \mathbb{R}^p \to \mathbb{R}$. Cette méthode ne nécessite pas la connaissance ou l'estimation du hessien ni même du gradient. Toutefois, elle est heuristique, c'est-à-dire que nous n'avons aucune garantie de convergence vers un optimum, même local.

Avant la première itération, la première étape de la méthode Nelder-Mead consiste à définir un ensemble de $p+1$ points $\{P_k \in \mathbb{R}^p : k \in \mathbb{N}, k \leq p\}$ appelé simplexe. Dans le package `Optim` de `Julia`, le point P_0 est choisi par l'utilisateur tandis que les autres points sont définis comme étant

$$P_k = P_0 + \left(0.5 e_k^\top P_0 + 0.025\right) e_k$$

de sorte que le simplexe représente un hypercube découpé sur l'hyper-diagonale principale ne passant pas par le sommet P_0. La figure 3.1 illustre cette interprétation lorsque $p = 2$. Dans ce dernier cas, le demi-hypercube devient un demi-carré et donc un triangle rectangle dont la grande diagonale ne passe pas par le sommet P_0.

Ensuite, pour chaque itération, nous mettons à jour le simplexe. Pour ce faire, nous commençons par trouver les indices ℓ et h vérifiant

$$\forall k \in \mathbb{N} \cap [0, p], f(P_\ell) \leq f(P_k) \leq f(P_h)$$

de sorte que les indices ℓ et h représentent respectivement le point minimisant et maximisant la fonction f sur le simplexe courant. Ensuite, nous définissons le centroïde courant $\bar{P}$ par

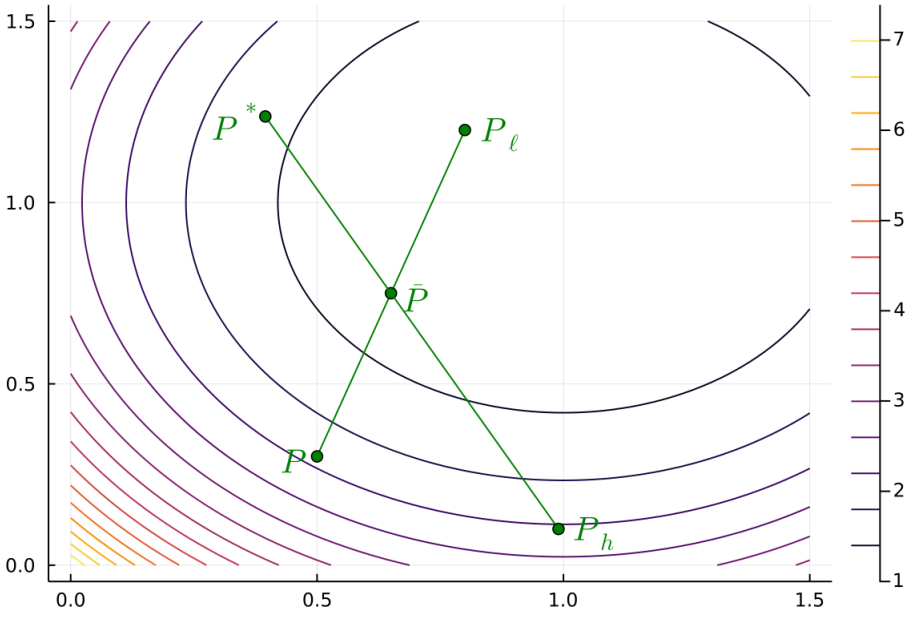


FIGURE 3.2 – Illustration du point de réflexion pour une fonction à deux variables à minimiser, réalisée à l'aide du langage de programmation `Julia`.

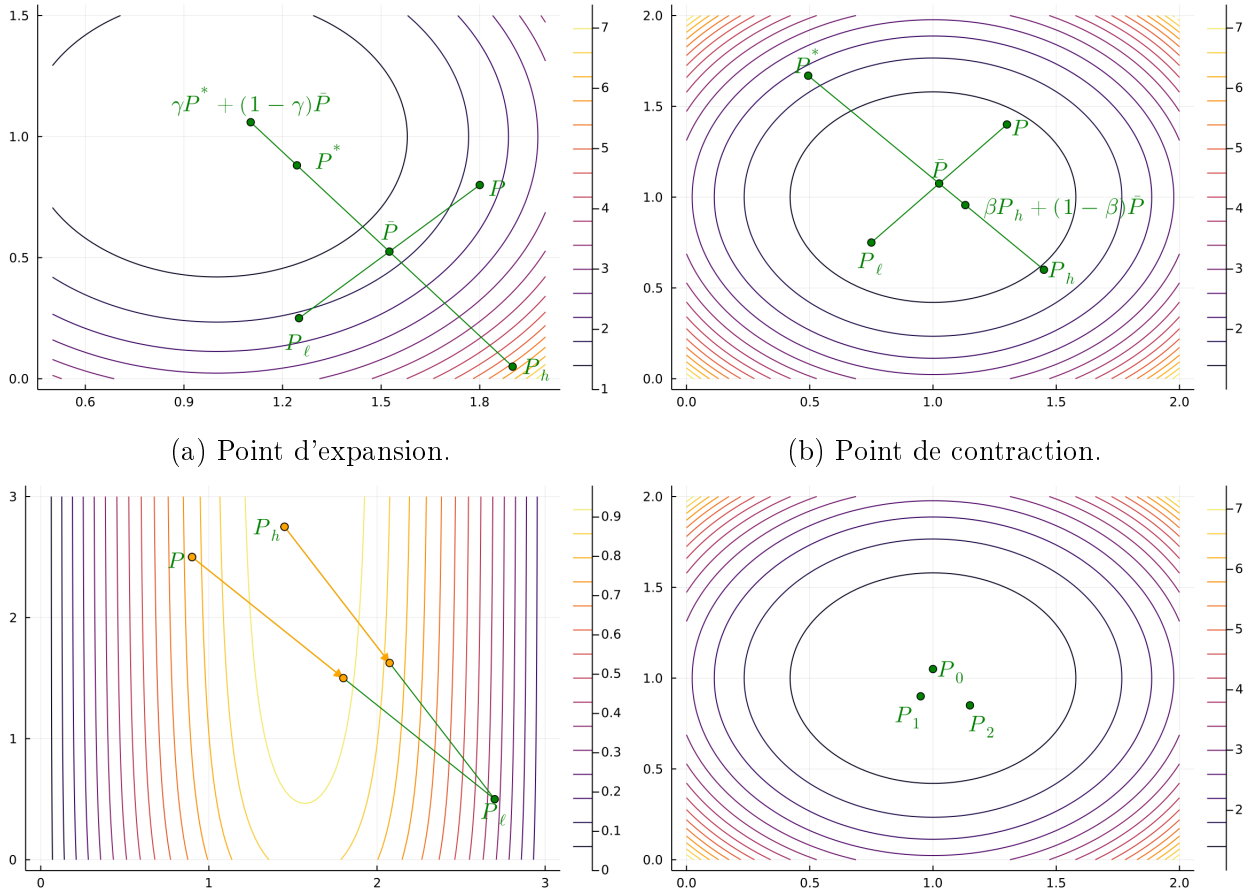
l'expression

$$\frac{1}{p}\sum_{\substack{k=0\\k\neq h}}^{p} P_k$$

de sorte que ce centroïde représente le point moyen du simplexe en ne tenant pas compte du point P_h qui est le moins intéressant. Nous définissons également un point dit *de réflexion* noté P^* définie par la relation

$$(1+\alpha)\,\bar{P} - \alpha P_h$$

où α est un scalaire positif choisi par l'utilisateur. Ce point représente un élément de la droite contenant les points $\bar{P}$ et P_h notée $\overline{\bar{P}P_h}$ mais ne se trouve pas dans le segment d'extrémités $\bar{P}$ et P_h noté $\left[\bar{P}, P_h\right]$ mais bien sur la demi-droite d'extrémité $\bar{P}$. De cette manière, le point de réflexion est *réfléchie* selon le centroïde de sorte à s'éloigner du point peu intéressant P_h. La figure 3.2 en dimension deux montre bien un intérêt d'un telle sommet : espérer que ce



(a) Point d'expansion.

(b) Point de contraction.

(c) Rapprochement des sommets.

(d) Condition d'arrêt

FIGURE 3.3 – Illustration des points particuliers pour une fonction à deux variables à minimiser, réalisée à l'aide du langage de programmation `Julia`.

dernier point améliore le point P_h en s'éloignant de celui-ci. Dans le package `Optim` de `Julia`, le scalaire α vaut par défaut l'unité, de sorte que la réflexion soit parfaite dans le sens où les segments $\left[\bar{P}, P_h\right]$ et $\left[\bar{P}, P^*\right]$ sont de même longueur. Dans l'éventualité où ce point de réflexion améliore le point P_h dans le sens où $f(P_\ell) > f(P^*)$, nous pourrions nous demander si nous ne pouvons pas aller encore plus loin dans la direction suivi dans la construction du point de réflexion en construisant un point dit *d'expansion* définie par l'expression

$$\gamma P^* + (1-\gamma)\,\bar{P}$$

où le scalaire γ choisi par l'utilisateur est strictement supérieur à l'unité. Le package `Optim` de `Julia` définit ce scalaire par défaut par $1 + \frac{2}{p}$. Dans le cas de la figure 3.3a, le point d'expansion améliore nettement l'optimisation de la fonction f, ce qui justifie le procédé dans ce cas. En revanche, si le remplacement du point le moins intéressant P_h par le point de réflexion préserve le statut du point le moins intéressant du simplexe, alors un point dit *de contraction* du segment $\left[\bar{P}, P_h\right]$ est défini à l'aide de l'expression

$$\beta P_h + (1-\beta)\,\bar{P}$$

où le scalaire β choisi par l'utilisateur et définie par le package `Optim` par défaut par $1 - \frac{1}{2p}$, est compris entre zéro et un. Ce point est construit dans ce cas puisque la direction prise pour construire le point de réflexion n'améliore pas le simplexe. Nous sommes en quelque sorte partis trop loin avec le point de réflexion comme l'indique la figure 3.3b où le point de contraction est plus intéressant dans ce cas. Une fois tous ces points définis, le simplexe courant est mis à jour en remplaçant le point le moins intéressant P_h par un des points parmi ce même point, le point de réflexion et éventuellement le point d'expansion ou de contraction si l'itération courante le construit. Dans le cas où le point P_h reste inchangé, les points du simplexe P_k différents du point intéressant P_ℓ sont remplacés par le point défini par

$$\delta P_\ell + (1-\delta)\,P_k$$

où le scalaire δ choisi par l'utilisateur, vaut $\frac{1}{2}$ dans l'article [14] et $1 - \frac{1}{p}$ dans le package `Optim`. Ce nouveau point se trouve dans le segment $[P_k, P_\ell]$ défini avant remplacement de sorte à rapprocher les points non intéressants du simplexe du point intéressant P_ℓ. La figure 3.3c illustre l'intérêt de cette construction. En effet, le point P_h devient beaucoup plus intéressant car la valeur $f(P_h)$ diminue. La condition d'arrêt de l'algorithme itératif borne la variance des valeurs $f(P_k)$ pour tous les sommets du simplexe à une certaine valeur égale à 10^{-8} dans l'article [14] et dans le package `Optim`. Ce cas signifie que les écarts des valeurs de la fonction f évaluée au simplexe est très petite et que donc la fonction f est approximativement identique sur le simplexe. Dans ce cas, la fonction est plus difficile à minimiser davantage et nous pourrions donc nous attendre à ce qu'un minimum local soit en approche comme le montre la figure 3.3d où l'élément du domaine minimisant la fonction semble se situer autour du point $(1; 1)$. Le package `Optim` définit également un nombre maximum d'itération de mille par défaut.

Une fois que les itérations sont terminées, l'algorithme retourne le point du simplexe accompagné du centroïde minimisant la fonction f.

3.2 Présentation des méthodes d'optimisation

Dans ce mémoire, deux méthodes principales d'optimisation seront considérées. Ces méthodes optimiseront sur un espace vectoriel $\mathcal{W}$ de fonctions de poids. Par conséquent, nous allons commencer par une première sous-section dédiée au choix de ces espaces vectoriels avant d'entrer au cœur du sujet dans une deuxième sous-section consacrée aux méthodes d'optimisations envisagées dans ce mémoire.

3.2.1 Choix du sous-espace de fonctions de base

Commençons par remarquer qu'étant donné que, numériquement, les fonctions de poids w de cet espace vectoriel sont discrétisées sous la forme d'un vecteur $\begin{pmatrix} w(x_1) & \cdots & w(x_{N_x}) \end{pmatrix}$, il est alors inutile de considérer un espace vectoriel $\mathcal{W}$ de dimension strictement supérieure à l'entier naturel N_x. Durant le reste du mémoire, trois types d'espaces vectoriels fonctionnels seront considérés. La première concerne l'espace engendré par les fonctions définies dans le chapitre précédent utilisant le package `ExprRules` de `Julia`. Ensuite, le second espace vectoriel est un espace de Fourier généré par des fonctions de base dont la première a pour

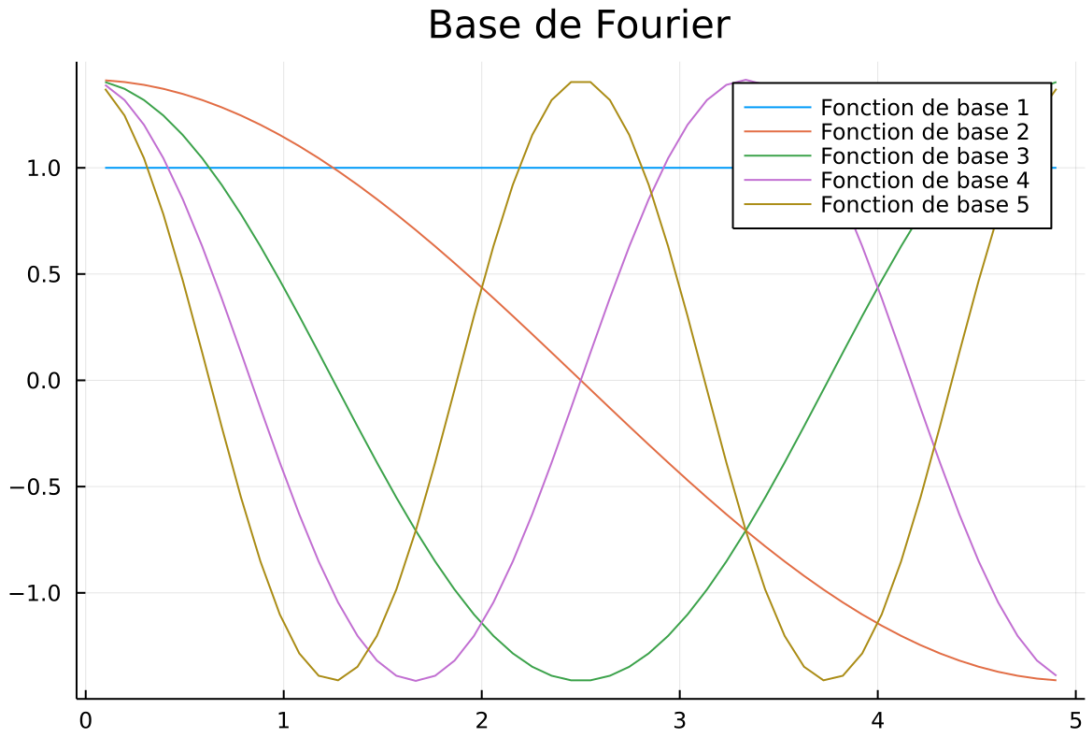


FIGURE 3.4 – Illustration des premières fonctions de base de la base de Fourier, réalisée à l'aide du langage de programmation `Julia`.

expression en tout élément réel x

$$\frac{1}{\sqrt{\max X - \min X}}$$

et dont les autres fonctions de bases indicées par l'indice k supérieure ou égale à deux sont définies en tout réel x par

$$\sqrt{\frac{2}{\max X - \min X}} \cos\left((k-1)\,\pi \frac{x - \min X}{\max X - \min X}\right).$$

Ces fonctions de base ainsi définies et représentées dans la figure 3.4 sont orthogonales par rapport au produit scalaire

$$\langle f, g\rangle_F = \int_X f(x)\, g(x)\, \mathrm{d}x = \int_0^1 (\max X - \min X)\, f(\varphi_F(x))\, g(\varphi_F(x))\, \mathrm{d}x$$

avec le changement de variable

$$\varphi_F(x) = (\max X - \min X)\, x + \min X.$$

En effet, la première fonction de base est normée puisque

$$\int_0^1 \mathrm{d}x = 1$$

et est orthogonale aux autres fonctions de base étant donné que

$$\int_0^1 \sqrt{2}\cos((k-1)\,\pi x)\,\mathrm{d}x = \int_0^1 \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\sqrt{2}}{(k-1)\,\pi}\sin((k-1)\,\pi x)\right)\mathrm{d}x = 0.$$

Ces autres fonctions de base sont normées puisque

$$\int_0^1 2\cos((k-1)\,\pi x)^2\,\mathrm{d}x = \int_0^1 \frac{\mathrm{d}}{\mathrm{d}x}\left(x + \frac{\sin(2\,(k-1)\,\pi x)}{2\,(k-1)\,\pi}\right)\mathrm{d}x = 1$$

et elles sont orthogonales entre elles puisque lorsque l'indice k est différent de l'indice ℓ,

$$\begin{aligned}\int_0^1 2\cos((k-1)\,\pi x)\cos((\ell-1)\,\pi x)\,\mathrm{d}x &= \int_0^1 \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\sin((k+\ell)\,\pi x)}{(k+\ell)\,\pi} + \frac{\sin((k-\ell)\,\pi x)}{(k-\ell)\,\pi}\right)\mathrm{d}x \\ &= 0.\end{aligned}$$

Enfin, le troisième espace vectoriel considéré est un espace de Tchebycheff généré par des fonctions de base dont la première a pour expression en tout élément réel x

$$\sqrt{\frac{2}{\pi\,(\max X - \min X)}}$$

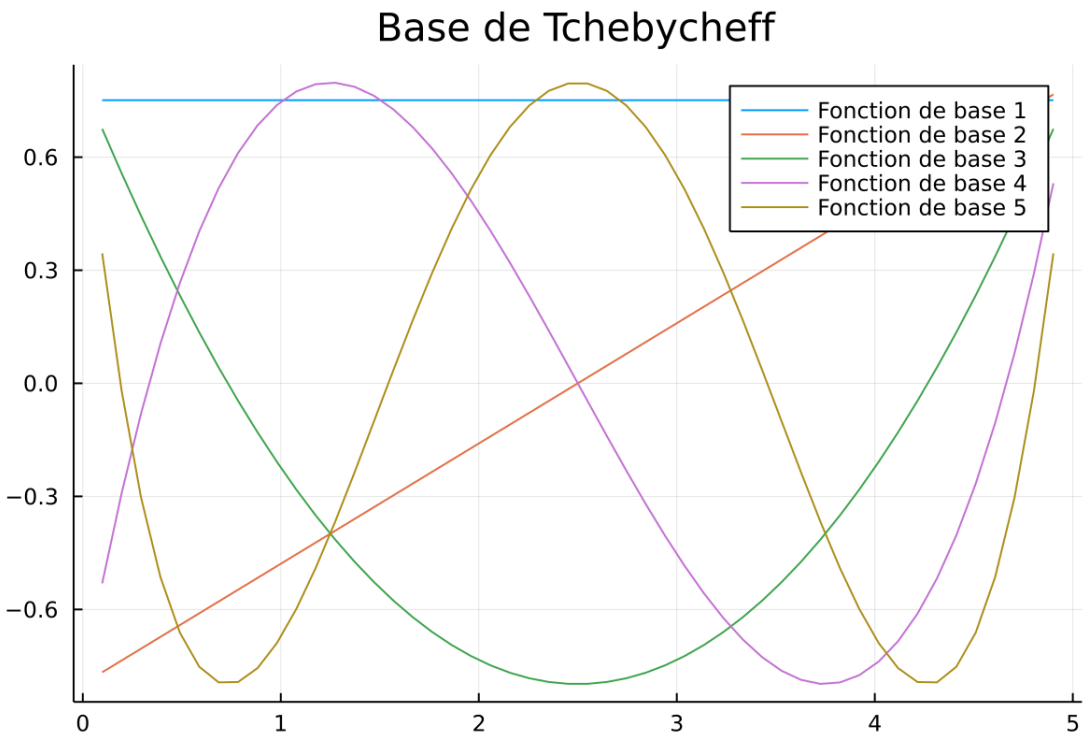


FIGURE 3.5 – Illustration des premières fonctions de base de la base de Fourier, réalisée à l'aide du langage de programmation `Julia`.

et dont les autres fonctions de bases indicées par l'indice k supérieure ou égale à deux sont définies en tout réel x par

$$\frac{2}{\sqrt{\pi\left(\max X-\min X\right)}}\cos\left(\left(k-1\right)\arccos\left(2\frac{x-\min X}{\max X-\min X}-1\right)\right).$$

De nouveau, ces fonctions de bases représentées dans la figure 3.5 sont orthogonales mais par rapport à un autre produit scalaire définie par la relation

$$\left\langle f,g\right\rangle_T=\int_X\frac{f\left(x\right)g\left(x\right)}{\sqrt{1-\left(2\frac{x-\min X}{\Delta X}-1\right)^2}}\mathrm{d}x=\int_0^1\frac{\pi\left(\max X-\min X\right)}{2}f\left(\varphi_T\left(x\right)\right)g\left(\varphi_T\left(x\right)\right)\mathrm{d}x$$

avec le changement de variable

$$\varphi_T\left(x\right)=\min X+\frac{\left(\max X-\min X\right)\left(1-\cos\left(\pi x\right)\right)}{2}.$$

En effet, la première fonction de base est normée puisque de nouveau,

$$\int_0^1\mathrm{d}x=1$$

et cette fonction est orthogonale aux autres fonctions de base étant donné que

$$\int_0^1 \sqrt{2}\cos\left((1-k)\,\pi\,(x-1)\right)\mathrm{d}x = \int_0^1 \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\sqrt{2}}{(1-k)\,\pi}\sin\left((1-k)\,\pi\,(x-1)\right)\right)\mathrm{d}x = 0.$$

Ces autres fonctions de base sont normées puisque

$$\int_0^1 2\cos\left((1-k)\,\pi\,(x-1)\right)^2\mathrm{d}x = \int_0^1 \frac{\mathrm{d}}{\mathrm{d}x}\left(x + \frac{\sin\left(2\,(1-k)\,\pi\,(x-1)\right)}{2\,(k-1)\,\pi}\right)\mathrm{d}x = 1$$

et elles sont orthogonales entre elles puisque lorsque l'indice k est différent de l'indice ℓ,

$$\begin{aligned}
&\int_0^1 2\cos\left((1-k)\,\pi\,(x-1)\right)\cos\left((1-\ell)\,\pi\,(x-1)\right)\mathrm{d}x \\
&= \int_0^1 \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\sin\left((k+\ell)\,\pi\,(x-1)\right)}{(k+\ell)\,\pi} + \frac{\sin\left((k-\ell)\,\pi\,(x-1)\right)}{(k-\ell)\,\pi}\right)\mathrm{d}x \\
&= 0.
\end{aligned}$$

3.2.2 Choix de la fonction objectif

Concernant les méthodes d'optimisations, la première méthode principale consiste à minimiser sur un espace vectoriel choisi de fonctions de poids une fonctionnelle C définie dans ce paragraphe. Afin de définir en chaque fonction de poids w l'expression $C\,(w)$, la première étape consiste à répartir l'ensemble des conditions initiales $\{u_k : k \in \{1,\ldots,N_u\}\}$ en deux sous-ensembles $\mathcal{C}$ et $\mathcal{D}$ formant ainsi une partition de cet ensemble de conditions initiales. Ensuite, pour chaque sous-ensemble $\mathcal{C}$ ou $\mathcal{D}$, nous déterminons le vecteur de coefficients estimé à l'aide de la méthode de Koopman, pour une fonction de poids w et nous noterons cette estimation $\tilde{c}_{\mathcal{C}}\,(w)$ ou $\tilde{c}_{\mathcal{D}}\,(w)$, respectivement. Ainsi, la fonctionnelle C est définie pour toute fonction de poids w par la relation

$$C\,(w) = \left\|\tilde{c}_{\mathcal{C}}\,(w) - \tilde{c}_{\mathcal{D}}\,(w)\right\|_2.$$

En d'autres termes, la fonctionnelle C représente l'écart d'estimation des *coefficients*. La minimisation de cette fonctionnelle implique l'hypothèse qu'une fonction de poids adéquate identifie le système (1.1) de manière satisfaisante indépendamment du choix de conditions initiales.

La seconde méthode quant à elle, minimise une autre fonctionnelle notée T ayant le même domaine de définition que la fonctionnelle définie au paragraphe précédent. De nouveau, définissons la fonctionnelle T en sélectionnant une fonction de poids w qui est arbitraire. Pour ce faire, nous allons commencer par estimer le vecteur de coefficients $\tilde{c}_{(-\ell)}\,(w) = \left(\tilde{c}_{(-\ell)}\,(w)_1 \quad \ldots \quad \tilde{c}_{(-\ell)}\,(w)_n\right)^\top$ défini comme étant l'estimation des coefficients c_i du système (1.1) à l'aide de la méthode de Koopman avec la fonction de poids w et en considérant l'ensemble des conditions initiales $\{u_k : k \in \mathbb{N} \cap [1; N_u]\} \setminus \{u_\ell\}$. Ensuite, nous allons estimer $\tilde{u}_\ell$

vérifiant le système (1.1) où chaque coefficient c_i correspond au coefficient $\tilde{c}_{(-\ell)}(w)_i$ préalablement calculé. Autrement dit, cette seconde méthode représente une validation croisée de type "leave one out", c'est-à-dire que nous estimons les coefficients c_i en tenant compte de toutes les conditions initiales excepté une qui servira à tester la concordance entre les données et la prédiction sur base des autres conditions initiales. À l'aide de ces calculs préliminaires, nous pouvons définir la fonctionnelle T en chaque fonction de poids w par l'expression

$$T(w) = \|u_\ell - \tilde{u}_\ell\| = \sqrt{\sum_{i=1}^{N_t}\sum_{j=1}^{N_x}\left(u_\ell(x_j, it_s) - \tilde{u}_\ell(x_j, it_s)\right)^2}.$$

À la différence de la première méthode, la fonctionnelle T représente l'écart d'estimation entre les trajectoires futures estimées et les véritables trajectoires. Nous émettons alors comme hypothèse qu'une fonction de poids adéquate identifie le système (1.1) de sorte que sa résolution numérique concorde avec les vraies données. Il est intéressant de noter qu'aucune condition initiale n'intervient à la fois dans l'estimation des coefficients et dans l'estimation de la trajectoire.

La section suivante précisera la manière de choisir la partition des conditions initiales en tenant compte de la variabilité des données.

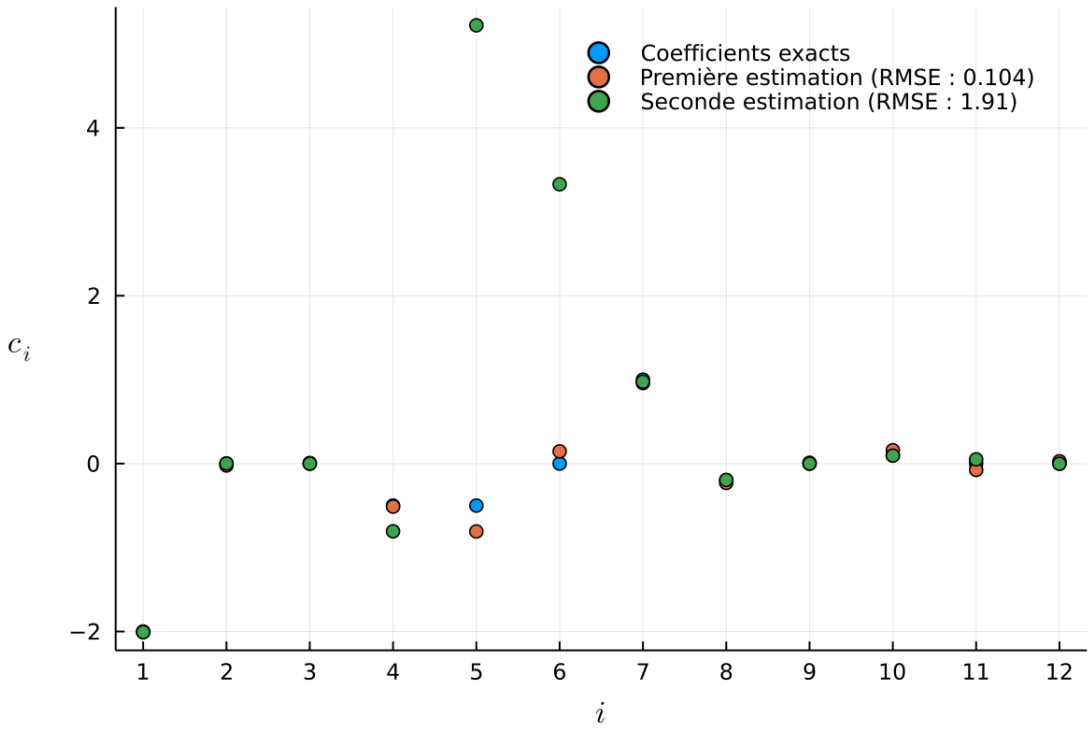


FIGURE 3.6 – Illustration du point de réflexion pour une fonction à deux variables à minimiser, réalisée à l'aide du langage de programmation `Julia`.

3.3 Gradient stochastique

Afin de motiver la nécessité de la notion du gradient stochastique, identifions le système donné par la première colonne de la table 2.1 en minimisant la fonction C décrite lors de la section précédente sur l'espace vectoriel de Fourier de dimension cinq définie dans la section précédente. La figure 3.6 illustre les résultats de la méthode d'optimisation en changeant simplement la partition des conditions initiales formée par les ensembles $\mathcal{C}$ et $\mathcal{D}$. Malgré ce changement, la qualité de l'identification est très différente. Afin de tenir compte de la variabilité des données illustrées dans la figure 3.6 dans les méthodes d'optimisation, l'idée est de changer de partition des conditions initiales. Ce changement de partition peut se faire à chaque itération de la méthode itérative d'optimisation soit par itération, soit par appel de fonction à minimiser.

L'article [3] énonce que le gradient stochastique a pour objectif de minimiser une fonction dite *fonction moyennée* sous la forme

$$\sum_{k=1}^{n} f_k$$

avec des fonctions f_k (ou ∇f_k qui sont connues ou estimées) qui sont potentiellement coûteuses en évaluation. Ainsi, pour accélérer les calculs, le principe consiste à effectuer la méthode itérative utilisée pour l'optimisation en ne considérant cette fois-ci qu'une seule fonction particulière f_k différente selon les itérations et choisie aléatoirement.

Dans le cas des fonctions qui nous intéressent dans le cadre du mémoire, nous allons d'une part pour la minimisation de l'écart d'estimation définir les fonctions C_k jouant le rôle de la fonction correspondante f_k comme étant la fonction C où l'ensemble $\mathcal{C}$ est posé comme étant

$$\left\{ u_i, i \in \left\{ k, \cdots, k-1+\left\lfloor \frac{N_u}{2} \right\rfloor \right\} \right\}$$

où pour chaque indice strictement positif i, nous posons comme convention $u_{i+N_u} = u_i$ et où l'entier N_u représente le nombre de conditions initiales dans les données. Concernant l'autre fonction décrite dans la section précédente, nous allons poser T_k jouant le rôle de f_k comme étant la fonction T où l'indice k représente l'indice ℓ représentant la condition initiale qui sera utilisée seulement pour résoudre uniquement l'équation aux dérivées partielles après estimation préliminaire des coefficients.

3.4 Comparaison des méthodes d'optimisation

Terminons ce chapitre par la comparaison des méthodes d'optimisation à l'aide des données décrites dans la table 2.1.

Remarque 3. *Afin de correspondre au mieux à la fonction à optimiser* $\sum_{k=1}^{n} f_k$ *à l'itération courante* i *considérant un élément* x_i *dans l'espace de recherche d'optimisation, les conditions*

d'arrêts associées seront portées non pas sur $f_k(x_i)$ *mais sur* $\sum_{j=i+1-n}^{i} f_{k_j}(x_j)$ *dont chacun des termes a déjà été préalablement calculé.*

Dans le changement de partition de conditions initiales par appel de fonctions évoqué dans le premier paragraphe de cette section, comme le nombre d'appels à la fonction par itération est supposé inconnu, il peut donc être supposé aléatoire. Ce qui revient à choisir aléatoirement la fonction C_k *ou* T_k *comme dans la méthode de gradient stochastique.*

Quant au changement de partition par itération, cela revient à choisir la fonction C_k *ou* T_k *suivante et donc de manière non-aléatoire, ce qui constitue la seule différence avec le gradient stochastique choisissant ces fonctions aléatoirement. Cependant, les résultats numériques ont montré que dans le cas non bruité, les cas où l'identification est significativement meilleure lors du changement de partition par itération sont en légère majorité alors que dans le cas bruité, c'est-à-dire avec un bruit normal d'amplitude* $\sigma = 10^{-2}$ *ajouté aux données, aucune conclusion ne peut vraiment être établie. En effet, à l'itération courante, les critères d'arrêts se porteront sur l'élément* $\sum_{k=i+1-N_u}^{i} f_k(x_k)$ *représentant la fonction moyennée où la fonction* f_k *représente la fonction* C_k *ou* T_k *et où nous utilisons les mêmes conventions pour les indices qui dépassent le nombre de fonctions définies tandis que pour le changement de partition de conditions initiales par appels de fonctions, ce critère d'arrêt s'exprime sous la forme* $\sum_{k=i+1-N_u}^{i} f_{k_j}(x_k)$ *représentant la fonction moyennée où les indices* k_j *sont supposés être choisis aléatoirement. Étant donné que l'ensemble* $\{k_j : j \in \{i+1-N_u, \cdots, i\}\}$ *ne peut être représentée que de temps à autre l'ensemble* $\{1, \cdots, N_u\}$ *dû au choix aléatoire des fonc-*

Dimension du domaine d'optimisation		**Fourier**		**Tchebycheff**		`ExprRules`	
		[11]	[15]	[11]	[15]	[11]	[15]
5	$\min C(w)$	0.0987	0.2	0.28	0.204	1.1	0.365
	$\min T(w)$	0.378	0.208	0.587	0.208	1.13	0.371
15	$\min C(w)$	0.0813	0.193	0.91	0.126	0.113	0.326
	$\min T(w)$	0.0881	0.213	0.235	0.205	0.267	0.303
25	$\min C(w)$	0.0837	0.181	0.0844	0.183	0.0729	0.373
	$\min T(w)$	0.551	0.212	0.084	0.204	131	0.314
36	$\min C(w)$	0.0933	0.183	1.54	0.172	0.129	0.347
	$\min T(w)$	0.43	0.365	0.207	0.199	1.34	2.12
50	$\min C(w)$			0.249	5.33		
	$\min T(w)$			0.0956	115		

TABLE 3.1 – Tableau de matrices de RMSE où les colonnes représentent les systèmes de la table 2.1, où la première ligne représente la minimisation de la fonction C et où la seconde ligne représente la minimisation de la fonction T : cas non bruité

Dimension du domaine d'optimisation		Fourier		Tchebycheff		ExprRules	
		[11]	[15]	[11]	[15]	[11]	[15]
5	$\min C(w)$	24.8	0.211	**0.21**	0.211	0.942	0.371
	$\min T(w)$	**1.02**	0.209	14.7	0.21	0.929	**0.213**
15	$\min C(w)$	**11.8**	**0.191**	18.7	0.208	**0.591**	**0.287**
	$\min T(w)$	44.7	25.9	**0.208**	0.211	0.901	0.559
25	$\min C(w)$	**6.65**	0.212	122	0.211	**0.131**	**0.34**
	$\min T(w)$	10.9	0.205	**2.39**	0.206	2.37	2.98
36	$\min C(w)$	24.7	0.282	**12.8**	0.209	**0.258**	**0.31**
	$\min T(w)$	**2.39**	**0.206**	16.7	0.209	0.285	2.94
50	$\min C(w)$			0.561	**176**		
	$\min T(w)$			**0.416**	315		

TABLE 3.2 – Tableau de matrices de RMSE où les colonnes représentent les systèmes de la table 2.1, où la première ligne représente la minimisation de la fonction C et où la seconde ligne représente la minimisation de la fonction T : cas bruité de paramètre $\sigma = 10^{-2}$

tions et que donc l'ensemble $\left\{f_{k_j} : j \in \{i+1-N_u, \cdots, i\}\right\}$ *ne peut représenter que de temps à autre l'ensemble* $\{f_k : k \in \{1, \cdots, N_u\}\}$ *qui représente au mieux la fonction à minimiser* $\sum_{k=1}^{N_u} f_k$ *comme expliqué au premier paragraphe de cette remarque, cela explique cette tendance observée. Par conséquent, la méthode de changement de partition employée par la suite dans le mémoire se réalisera* par itération.

Dans le cas non bruité, la table 3.1 montre clairement que dans une grande majorité des cas, la méthode optimisant les écarts d'estimation des coefficients est la meilleure méthode d'optimisation. Cela peut s'expliquer par le fait que la mesure de qualité de l'identification des coefficients se porte sur les coefficients eux-mêmes et non sur la différence entre les trajectoires estimées après identification et les véritables trajectoires.

Cette constatation se confirme dans la table 3.2 même si ce message est un peu moins marqué compte tenu du bruit des données. De nouveau, même dans le cas bruité, la figure 3.7 illustre bien le fait que la méthode d'optimisation de la fonction C fournit une très bonne qualité d'identification des coefficients.

Ces illustrations permettent de conclure que la méthode d'optimisation minimisant les écarts d'estimations des coefficients est la plus adéquate pour identifier des équations aux dérivées partielles à fonctions de base connues. Notons également un constat très satisfaisant dans les figures représentées dans cette section : dans la plupart des cas, les courbes de la fonction à minimiser décroissent très rapidement dès les premières itérations, ce qui signifie que les méthodes d'optimisation *améliorent significativement la méthode de l'article [11].*

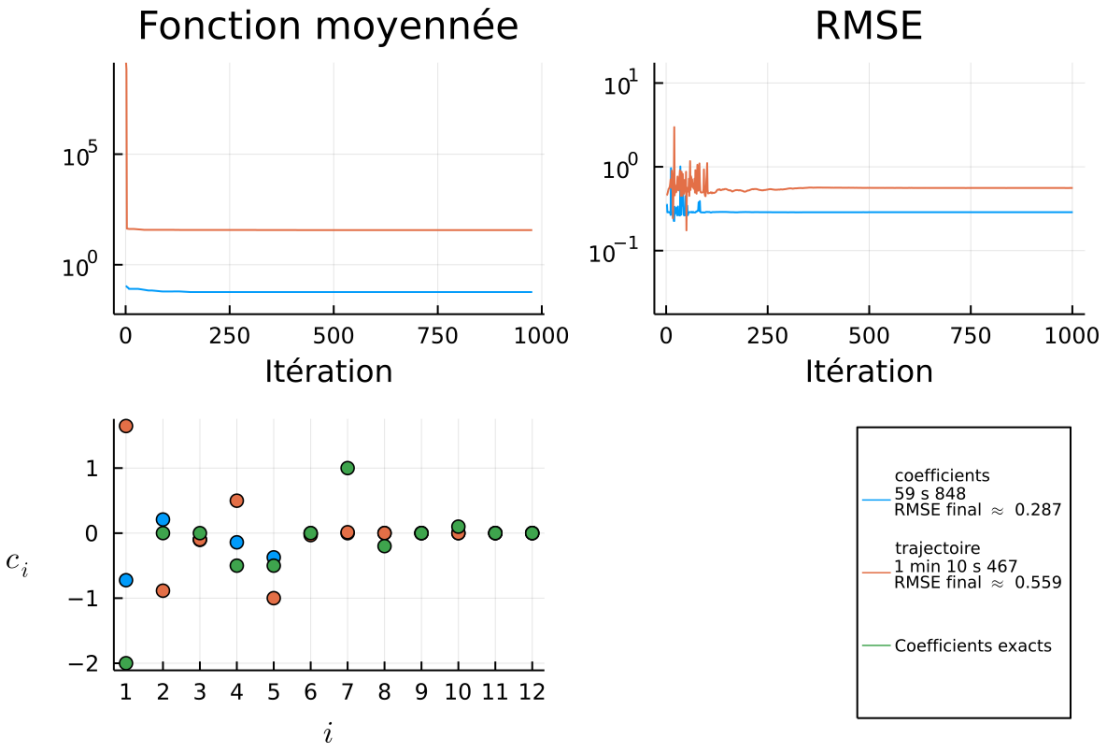


FIGURE 3.7 – Illustration de la comparaison des méthodes d'optimisation pour l'équation de l'article [15] sur l'espace vectoriel de recherche engendré par 15 fonctions définies dans le package `ExprRules` du langage de programmation `Julia`.

Remarque 4. *En analysant la variance intervenant dans la condition d'arrêt, celle-ci semble rarement descendre en dessous de* 0.1. *Par conséquent, un allègement de la condition d'arrêt sur la borne de la variance n'est pas envisageable.*

Chapitre 4

Amélioration des performances d'optimisation

Dans ce chapitre, nous allons tenter d'optimiser la méthode établie au terme du chapitre précédent. Plus particulièrement, nous allons nous concentrer sur une section traitant d'un calcul plus rapide d'approximation du générateur de Lie à l'aide d'une interpolation de Lagrange.

4.1 Estimation du générateur

Pour commencer cette section, rappelons que l'estimation du générateur de Lie se réalise via un calcul de logarithme matriciel. Seulement, ce procédé est relativement lent. En effet, comme expliqué dans l'article [1], le calcul du logarithme matriciel se base sur le constat que pour toute matrice unitaire Q, pour toute matrice triangulaire supérieure M et pour tout naturel k, nous obtenons

$$e^{kQ\log(M)Q^*} = e^{Q\log(M)kQ^*} = Qe^{\log(M)k}Q^* = QM^kQ^* = e^{\log\left(QM^kQ^*\right)}.$$

Ainsi, la première étape consiste à effectuer une décomposition de Schur sous la forme

$$QTQ^*$$

où Q est une matrice unitaire et où T représente une matrice triangulaire supérieure. Cette étape permet de simplifier le calcul du logarithme avec des transformations avec un bon nombre de conditionnement tel que les matrices unitaires. La décomposition de Schur commence par la détermination d'une matrice semblable par des transformations unitaires à une matrice de Hessenberg, c'est-à-dire une matrice triangulaire supérieure avec une diagonale supplémentaire juste en dessous de la diagonale principale. Cette matrice de Hessenberg se transformera ensuite en une matrice triangulaire supérieure semblable par transformations unitaires à l'aide d'une méthode itérative. Ensuite, la deuxième étape consiste à appliquer un certain nombre de fois s la racine carrée matricielle de la matrice triangulaire supérieure

pour obtenir une matrice M vérifiant la relation

$$M = T^{2^{-s}}.$$

Ainsi, la matrice initiale s'écrit sous la forme

$$QM^{2^s}Q^*$$

et le logarithme recherché s'exprime alors sous la forme

$$2^s Q \log(M) Q^*$$

selon la relation sur laquelle se base la méthode numérique de calcul du logarithme matriciel. Enfin, le logarithme de la matrice M s'approxime à l'aide d'une approximation dite de Padé définie comme étant

$$\sum_{i=1}^{m} \alpha_j^{(m)} \left(I + \beta_j^{(m)} (M - I)\right)^{-1} (M - I)$$

où le paramètre m est choisi de manière à rendre l'approximation la plus optimale possible, où le calcul de l'expression $\left(I + \beta_j^{(m)} (M - I)\right)^{-1} (M - I)$ se réalise selon la remarque 2 et où les coefficients $\alpha_j^{(m)}$ et $\beta_j^{(m)}$ représentent respectivement les poids et les nœuds de la quadrature de Gauss-Legendre sur l'intervalle $[0; 1]$, c'est-à-dire vérifiant la propriété

$$\forall k \in \{0, \cdots, 2m-1\}, \sum_{j=1}^{m} \alpha_j^{(m)} \left(\beta_j^{(m)}\right)^k = \int_0^1 x^k \mathrm{d}x = \frac{1}{k+1}.$$

Ces coefficients sont déterminés selon l'article [5] se basant sur une décomposition spectrale d'une matrice tridiagonale et donc de Hessenberg nécessitant donc une décomposition de Schur à l'aide de la méthode itérative évoquée dans cette section.

Comme nous venons de le voir, l'approximation du générateur de Lie $\tilde{L}$ à l'aide du logarithme matriciel est relativement coûteuse. Rappelons que cette matrice doit vérifier la relation

$$\tilde{U}^t = e^{\tilde{L}t}.$$

Si nous dérivons l'approximation de l'opérateur de Koopman et que nous l'évaluons pour $t = 0$, alors nous obtenons

$$\frac{\mathrm{d}}{\mathrm{d}t} \tilde{U}^t \Big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t} \mathrm{e}^{\tilde{L}t} \Big|_{t=0} = \tilde{L}\mathrm{e}^{\tilde{L}0} = \tilde{L}.$$

Ainsi, une autre alternative serait d'approximer la dérivée de l'approximation de l'opérateur de Koopman $\tilde{U}^t$ évaluée en $t = 0$ à l'aide de l'approximation de Lagrange de l'opérateur approximé $\tilde{U}^t$. Ainsi,

$$\tilde{L} = \left(\frac{\mathrm{d}}{\mathrm{d}t} \sum_{k=0}^{N_t} \tilde{U}^{kt_s} \prod_{\substack{j=0 \\ j \neq k}}^{N_t} \frac{t - jt_s}{kt_s - jt_s} \right) \Bigg|_{t=0}.$$

Or, la linéarité de la dérivation et la règle de Leibniz nous fournissent

$$\tilde{L} = \left(\sum_{k=0}^{N_t} \tilde{U}^{kt_s} \sum_{\substack{\ell=0 \\ \ell \neq k}}^{N_t} \frac{\mathrm{d}}{\mathrm{d}t} \frac{t - \ell t_x}{kt_s - \ell t_s} \prod_{\substack{j=0 \\ j \neq k \\ j \neq \ell}}^{N_t} \frac{t - jt_s}{kt_s - jt_s} \right) \Bigg|_{t=0},$$

de sorte à ne devoir dériver qu'une simple fonction usuelle pour obtenir

$$\tilde{L} = \left(\sum_{k=0}^{N_t} \tilde{U}^{kt_s} \sum_{\substack{\ell=0 \\ \ell \neq k}}^{N_t} \frac{1}{kt_s - \ell t_s} \prod_{\substack{j=0 \\ j \neq k \\ j \neq \ell}}^{N_t} \frac{t - jt_s}{kt_s - jt_s} \right) \Bigg|_{t=0}.$$

De plus, l'évaluation de cette expression en $t = 0$ nous fournit une autre expression encore plus simple. En effet, cela nous donne

$$\tilde{L} = \sum_{k=0}^{N_t} \tilde{U}^{kt_s} \sum_{\substack{\ell=0 \\ \ell \neq k}}^{N_t} \frac{1}{kt_s - \ell t_s} \prod_{\substack{j=0 \\ j \neq k \\ j \neq \ell}}^{N_t} \frac{j}{j - k}.$$

Nous pouvons encore simplifier davantage ce résultat. En effet, en isolant le terme indicé par $k = 0$ des autres termes et en isolant pour les autres termes le sous-terme indicé par $\ell = 0$ des autres sous-termes, l'expression de l'approximation du générateur de Lie devient

$$\tilde{L} = \sum_{k=1}^{N_t} \tilde{U}^{kt_s} \frac{1}{kt_s} \prod_{\substack{j=1 \\ j \neq k}}^{N_t} \frac{j}{j - k} + \sum_{k=1}^{N_t} \tilde{U}^{kt_s} \sum_{\substack{\ell=1 \\ \ell \neq k}}^{N_t} \frac{1}{kt_s - \ell t_s} \frac{0}{0 - k} \prod_{\substack{j=1 \\ j \neq k \\ j \neq \ell}}^{N_t} \frac{j}{j - k} - \tilde{U}^0 \sum_{\ell=1}^{N_t} \frac{1}{\ell t_s}.$$

Ainsi, après simplifications, nous obtenons

$$\tilde{L} = \sum_{k=1}^{N_t} \frac{1}{kt_s} \left(\tilde{U}^{kt_s} \prod_{\substack{j=1 \\ j \neq k}}^{N_t} \frac{j}{j - k} - \tilde{U}^0 \right).$$

Terminons le calcul en montrant que le produit indicé par l'indice j peut être réécrit à l'aide de coefficients binomiaux. En effet le développement de ce produit nous fournit

$$\tilde{L} = \sum_{k=1}^{N_t} \frac{1}{kt_s} \left(\tilde{U}^{kt_s} \frac{\prod_{j=1}^{k-1} j \prod_{j=k+1}^{N_t} j}{\prod_{j=1}^{k-1} (j - k) \prod_{j=k+1}^{N_t} (j - k)} - \tilde{U}^0 \right)$$

ou encore, en adaptant les indices,

$$\tilde{L} = \sum_{k=1}^{N_t} \frac{1}{kt_s} \left(\tilde{U}^{kt_s} \frac{\prod_{j=1}^{k-1} j \prod_{j=k+1}^{N_t} j}{\prod_{j=1-k}^{-1} j \prod_{j=1}^{N_t-k} j} - \tilde{U}^0 \right).$$

Ensuite, la division des $k-1$ premiers facteurs du numérateur par les $k-1$ premiers facteurs du dénominateur permet de déduire la relation

$$\tilde{L} = \sum_{k=1}^{N_t} \frac{1}{kt_s} \left(\tilde{U}^{kt_s} (-1)^{k-1} \frac{\prod_{j=k+1}^{N_t} j}{\prod_{j=1}^{N_t-k} j} - \tilde{U}^0 \right)$$

tandis que la division restante permet de réexprimer l'approximation du générateur de Lie sous la forme

$$\tilde{L} = \sum_{k=1}^{N_t} \frac{1}{kt_s} \left(\tilde{U}^{kt_s} (-1)^{k-1} \frac{N_t!}{k!\,(N_t-k)!} - \tilde{U}^0 \right).$$

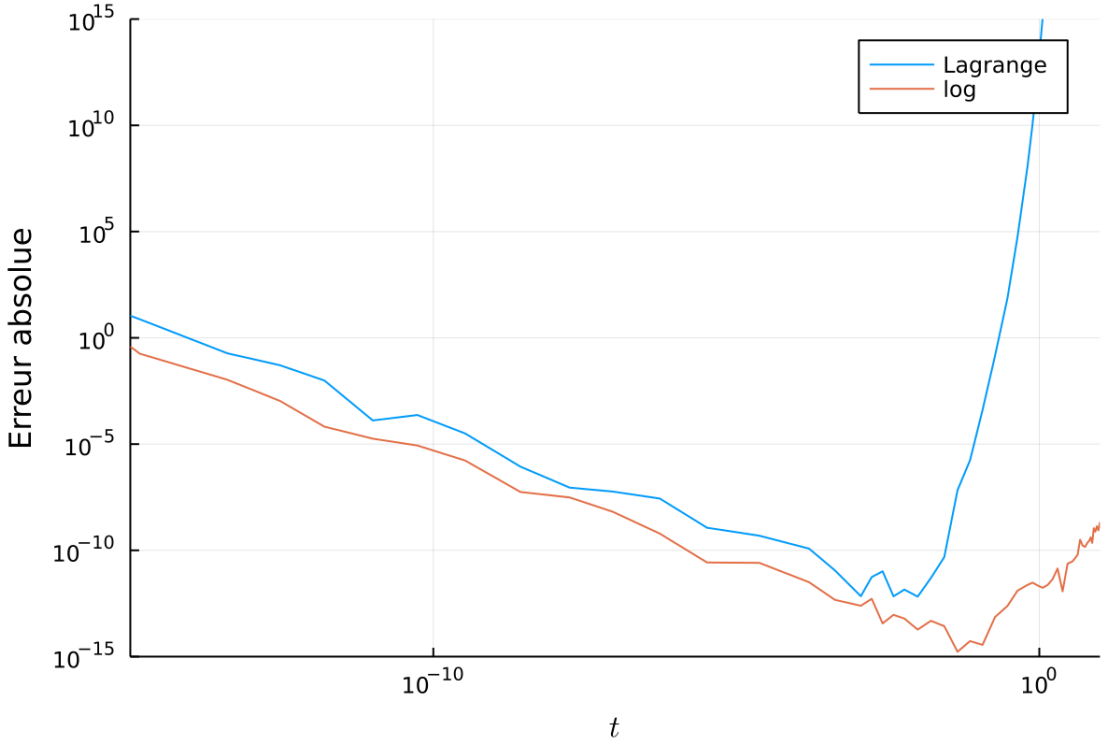


FIGURE 4.1 – Validation de la méthode de Lagrange, réalisée à l'aide du langage de programmation `Julia`.

Ainsi, l'expression finale du générateur de Lie devient

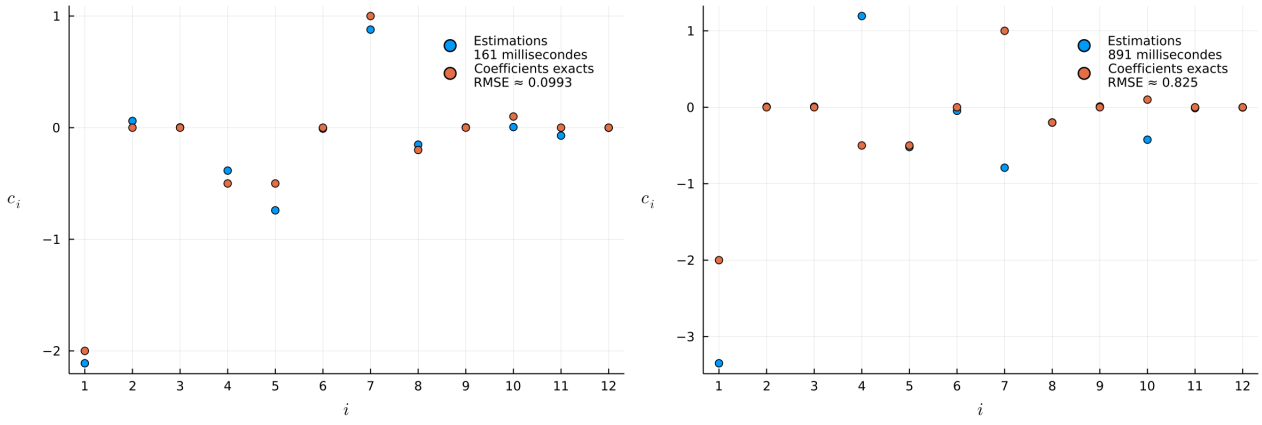
$$\boxed{\tilde{L} = \sum_{k=1}^{N_t} \frac{1}{kt_s} \left((-1)^{k-1} C_{N_t}^k \tilde{U}^{kt_s} - \tilde{U}^0 \right).}$$

La figure 4.1 approxime la première colonne du générateur de Lie exprimée par

$$L = \begin{pmatrix} 1 & 2 \\ -2 & 5 \end{pmatrix}$$

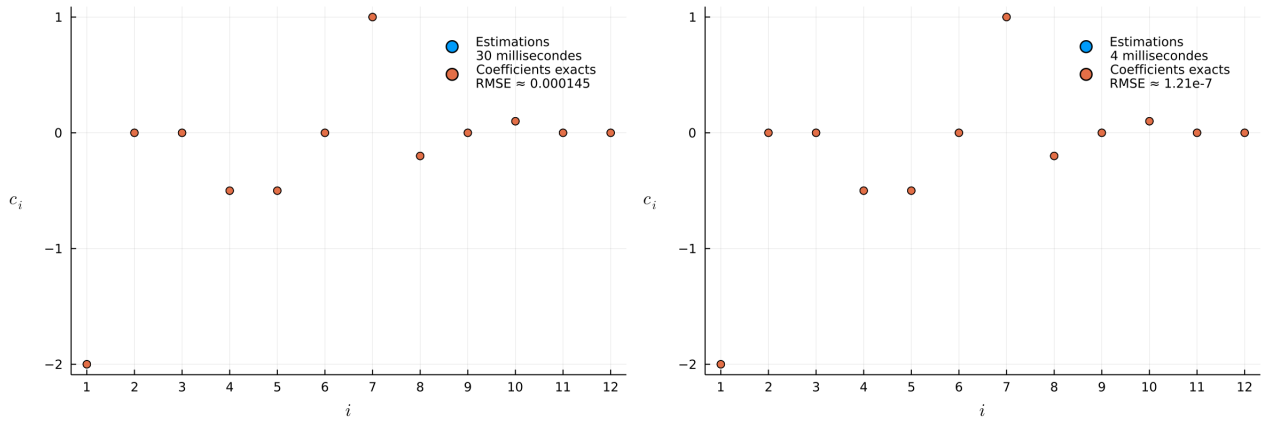
à l'aide du logarithme matriciel et de l'interpolation de Lagrange en fonction du pas de temps positif t. Lors de cette validation, nous avons opté pour $N_t = 10$. Bien que l'erreur absolue commise par l'interpolation de Lagrange est toujours supérieure comparée à celle commise par le logarithme matriciel, ces deux erreurs suivent globalement la même allure et l'erreur commise par la méthode de Lagrange reste tout à fait acceptable étant donné que l'erreur minimale commise est de l'ordre de 10^{-12}. Pour les deux méthodes, l'évolution de l'erreur s'explique de la même manière que pour les différences finies, à savoir une décroissance expliquée par la diminution des erreurs d'arrondis avant une croissance expliquée par l'augmentation de l'erreur de troncature sur la variable t. Ainsi, ces interprétations permettent de valider l'implémentation de l'approximation du générateur de Lie par interpolation de Lagrange.

Testons à présent l'influence de la méthode d'approximation du générateur de Lie sur l'identification des équations aux dérivées partielles. Commençons par fixer la fonction de poids selon l'article [11] et utilisons les données de l'article [11] décrites dans la table 2.1. Lorsque nous fixons les paramètres $N_t = 10$ et $N_u = 50$, les figures 4.2a et 4.2b indiquent que l'approximation du générateur de Lie par le logarithme matriciel fournit de meilleurs résultats en termes d'identification d'équations aux dérivées partielles même si cette approximation nécessite environ six fois plus de temps. Cependant, bien que le RMSE est plus de huit fois



(a) Méthode du logarithme matriciel (b) Méthode d'interpolation de Lagrange

FIGURE 4.2 – Estimation des coefficients en fonction de l'estimation du générateur de Lie, réalisée à l'aide du langage de programmation `Julia`.



(a) Méthode du logarithme matriciel (b) Méthode d'interpolation de Lagrange

FIGURE 4.3 – Estimation des coefficients en fonction de l'estimation du générateur de Lie, réalisée à l'aide du langage de programmation `Julia`.

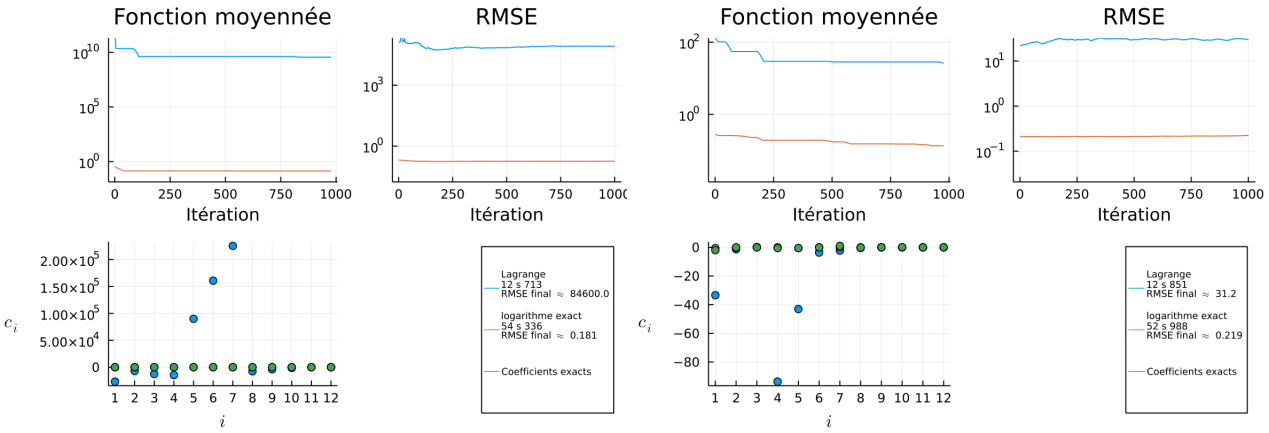
plus grand dans le cas de l'emploi de l'interpolation de Lagrange, cette dernière méthode reste acceptable puisque le RMSE qui vaut 0.825 est strictement inférieur à l'unité.

Si en plus nous fixons le pas de temps $t_s = 10^{-3}$, alors les figures 4.3a et 4.3b indiquent que les deux méthodes d'approximation fournissent d'excellents résultats sur l'identification d'équations aux dérivées partielles. Cela confirme le résultat de convergence énoncé dans l'article [11]

$$\lim_{t \downarrow 0} \hat{c}_i = c_i.$$

Or, dans le cas des figures 4.3a et 4.3b, la troncature sur t est très petite. Ce qui explique ces bons résultats. Ces résultats nous laissent penser que bien que la méthode de Lagrange soit bien plus rapide, l'approximation du générateur de Lie par logarithme matriciel reste à privilégier.

Vérifions cette hypothèse avec la méthode d'optimisation sélectionnée dans le chapitre précédent avec les données de la table 2.1 sans modifier les paramètres contrairement à ce qui a été réalisé au paragraphe précédent et les bases de recherche de fonctions de poids décrites au chapitre précédent. Dans le cas non bruité, la figure 4.4a illustre un exemple où la méthode d'interpolation de Lagrange fournit de très mauvais résultats. En effet, le RMSE est de l'ordre de 10^4. Ce constat est surtout observé dans le cas de l'identification de l'équation de Burgers décrite dans l'article [15]. Ce cas semble se confirmer dans la figure 4.4b dans une moindre mesure. Ces observations permettent de confirmer le fait que l'emploi du logarithme matriciel pour approximer le générateur de Lie reste le plus adéquat pour identifier des systèmes décrites par l'équation aux dérivées partielles (1.1).



(a) Cas non bruité avec la base de Fourier de dimension 25.

(b) Cas bruité de paramètre $\sigma = 10^{-2}$ avec la base de Tchebycheff de dimension 36.

FIGURE 4.4 – Comparaison de l'effet de la méthode d'approximation sur l'identification de l'équation de l'article [15] des données de la table 2.1, réalisée à l'aide du langage de programmation `Julia`.

Conclusion

Dans ce travail, nous avons commencé par un premier chapitre comparant les méthodes existantes identifiant les équations aux dérivées partielles avec la méthode de l'article [11] utilisant l'opérateur de Koopman. Nous nous sommes rendus compte que le choix de la fonction de poids était crucial pour l'efficacité de la méthode. Afin d'améliorer cette dernière, nous avons commencé dans le second chapitre par étudier d'un peu plus près cette méthode et nous avons réalisé plusieurs tests préliminaires. Nous arrivons alors à la conclusion qu'une méthode d'optimisation sur les fonctions de poids est nécessaire et nous avons creusé cette piste lors d'un troisième chapitre où nous avons envisagé plusieurs problèmes d'optimisation possibles ainsi que des manières différentes de considérer la variabilité des données. Une fois que nous avons obtenu une méthode d'optimisation satisfaisante, un dernier chapitre a permis de tenter d'améliorer davantage la méthode par des calculs plus rapides. Nous avons conclu que ces tentatives n'améliorent pas significativement la méthode.

Ce travail a dès lors permis de montrer que l'identification d'équations aux dérivées partielles à l'aide de la méthode utilisant l'opérateur de Koopman peut se montrer très efficace lorsque nous utilisons une méthode d'optimisation sur les fonctions de poids. Nous avons également envisagé de tenir compte de la variabilité des résultats.

Cependant, nous ne pouvons pas employer la méthode de l'article [11] dans le cas des équations différentielles ordinaires. En effet, l'absence de variables spatiales empêche de définir la notion de fonction de poids établie dans l'article [11].

Une idée de perspective pour approfondir le sujet consisterait à identifier une équation aux dérivées partielles où les fonctions de base sont inconnues. Il faudrait alors optimiser directement sur des fonctions de base possibles. Nous pourrions ainsi appliquer ce procédé à des équations différentielles ordinaires à identifier. Si nous restons dans le cadre des équations aux dérivées partielles à identifier, nous pourrions envisager de faire varier d'autres paramètres des méthodes d'optimisation évoquées dans ce travail tels que l'influence de l'ordre de dérivation numérique spatiale, le choix d'une base de recherche de fonctions de poids ou le choix d'une autre méthode d'optimisation, qu'elle soit heuristique ou non.

Bibliographie

[1] Awad H Al-Mohy, Nicholas J Higham, and Samuel D Relton. Computing the fréchet derivative of the matrix logarithm and estimating the condition number. *SIAM Journal on Scientific Computing*, 35 :C394–C410, 2013.

[2] Ake Bjorck. *Numerical Methods for Least Squares Problems*, volume 51. SIAM, Philadelphia, United States, 1996.

[3] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38 :367–378, 2002.

[4] René Gateaux. Sur les fonctionnelles continues et les fonctionnelles analytiques. *CR Acad. Sci. Paris*, 157 :65, 1913.

[5] Gene H Golub and John H Welsch. Calculation of gauss quadrature rules. *Mathematics of computation*, 23 :221–230, 1969.

[6] LZ Guo, Stephen A Billings, and Daniel Coca. Identification of partial differential equation models for a class of multiscale spatio-temporal dynamical systems. *International Journal of Control*, 83 :40–48, 2010.

[7] Daniel R Gurevich, Patrick AK Reinbold, and Roman O Grigoriev. Robust and optimal sparse regression for nonlinear pde models. *Chaos : An Interdisciplinary Journal of Nonlinear Science*, 29 :103113, 2019.

[8] Nicholas J Higham. *Functions of matrices : theory and computation*. SIAM, Philadelphia, United States, 2008.

[9] BO Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17 :315–318, 1931.

[10] Xiuting Li, Liang Li, Zuogong Yue, Xiaoquan Tang, Henning U Voss, Jürgen Kurths, and Ye Yuan. Sparse learning of partial differential equations with structured dictionary matrix. *Chaos : An Interdisciplinary Journal of Nonlinear Science*, 29 :043130, 2019.

[11] Alexandre Mauroy. Koopman operator framework for spectral analysis and identification of infinite-dimensional systems. *Mathematics*, 9 :2495, 2021.

[12] Alexandre Mauroy and Jorge Goncalves. Koopman-based lifting techniques for nonlinear systems identification. *IEEE Transactions on Automatic Control*, 65 :2550–2565, 2019.

[13] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41 :309–325, 2005.

[14] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7 :308–313, 1965.

[15] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3 :e1602614, 2017.

[16] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator : Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25 :1307–1346, 2015.