

N° d'ordre : 4139

THÈSE

présentée à

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE

par **Karen Gonzalez**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : **Mathématiques Appliquées**

**CONTRIBUTION A L'ETUDE DES PROCESSUS
MARKOVIENS DETERMINISTES PAR MORCEAUX**

**Etude d'un cas-test de la sûreté de fonctionnement et
Problème d'arrêt optimal à horizon aléatoire**

Soutenue le 3 Décembre 2010 à l'Institut de Mathématiques de Bordeaux

Après avis de :

P.-E. LABEAU	Université Libre de Bruxelles, Belgique	Rapporteur
N. LIMNIOS	Université de Technologie de Compiègne, France	Rapporteur

Devant la commission d'examen composée de :

F. DUFOUR	Institut Polytechnique de Bordeaux	Directeur
B. de SAPORTA	Université Bordeaux 4	Directrice
J. SARACCO	Institut Polytechnique de Bordeaux	Examineur
T. PRIETO-RUMEAU	Facultad de Ciencias de Madrid	Examineur

Remerciements

En premier lieu, je souhaite remercier mon directeur de thèse François Dufour pour la qualité de son encadrement durant ces trois années, ainsi que pour mon mémoire de Master 1 et Master 2. Pour sa disponibilité et ses nombreux conseils, je remercie ma co-directrice Benoîte de Saporta.

Je remercie Nikolaos Limnios et Pierre-Etienne Labeau d'avoir accepté d'être mes rapporteurs, ainsi que Tomas Prieto-Rumeau et Jérôme Saracco d'avoir bien voulu faire partie de mon jury.

Je remercie chaleureusement Huilong Zhang, Thierry Colin et Yves Dutuit pour l'aide et les conseils qu'ils ont pu m'apporter durant mon doctorat.

D'une façon générale, je tiens à remercier tout le personnel, enseignant et non-enseignant, de l'université de Bordeaux 1 que j'ai pu côtoyer de ma première année de DEUG à ma dernière année de doctorat.

Merci à tous mes amis doctorants : si ces trois années se sont déroulées si vite et si bien, c'est grâce à leur bonne humeur et leur soutien. Je remercie Jade pour nos longues discussions sur tout et rien (!), Joyce, ma collègue de bureau et fournisseur officiel de gâteaux et chocolats, Cédric pour ses astuces LaTeX et Beamer, et tous les autres : Danaëlle, Johana, Aurélie, Damiano, Michele, Peng, Franck, Mohamed, Adrien, Jean-Baptiste, Anna, Frédéric, Marco, Arijit, les membres d'AquiDoc, ... (Désolée si j'en oublie!)

Je remercie ma soeur Maya d'avoir partagé ensemble nos coups de gueule, fous rire ou considérations affligées (et affligeantes?) sur la vie, la fac, et nos galères de thésardes.

Je remercie mon frère Adrien pour son aide, son soutien et sa présence lorsque j'en ai eu besoin. J'en profite pour lui souhaiter bon courage pour sa thèse!

A mes parents qui nous ont tout donné pour que nous en arrivions là, du fond du coeur : merci. Je leur en suis plus que reconnaissante et espère faire tout pour qu'ils soient fiers de moi.

Enfin et surtout, je remercie Jérôme, pour être là, tout simplement.

Table des matières

Introduction générale	7
1 Les PDMP	11
1.1 Equations différentielles ordinaires et champs de vecteurs	11
1.2 Définition	12
1.3 Exemples de PDMP	15
1.3.1 Processus de Poisson	15
1.3.2 Exemple simple	16
1.4 Espérance de fonctionnelle d'un PDMP	17
1.5 Structure des temps d'arrêt pour les PDMP	18
I Etude du réservoir chauffé	23
Introduction	25
2 Le cas-test du réservoir	27
2.1 Description du système	27
2.1.1 Principe de fonctionnement	27
2.1.2 Interactions entre les variables	29
2.1.3 Equations du système	29
2.1.3.1 Interaction température - mode	29
2.1.3.2 Interaction hauteur - mode	30
2.1.3.3 Interaction hauteur - température	31
2.2 Etude préliminaire	33
2.2.1 Les modes atteints	33
2.2.2 Résolution du Système d'Equations Différentielles	36
2.2.3 Etude qualitative du système	40
3 Caractéristiques du PDMP sous-jacent	43
3.1 Description des caractéristiques locales du PDMP	43
3.1.1 L'espace d'état E	43
3.1.2 Le flot ϕ_m	44
3.1.3 L'intensité λ_m	44
3.1.4 La mesure de transition Q_m	44
3.1.5 Le temps d'atteinte de la frontière t_m^*	45
3.2 Calcul des probabilités de défaillance	48

3.2.1	Principe de la simulation	48
3.2.2	Simulation des temps de panne des unités	49
3.2.3	Résumé de la méthode numérique développée	51
3.2.4	Calcul des probabilités de défaillance par la méthode de Monte Carlo	52
4	Approche par les EDP	61
4.1	Propriétés des PDMP et application au réservoir	61
4.2	Les domaines	62
4.2.1	Les domaines « simples »	66
4.2.1.1	Domaines E_1 , E_2 , E_7 et E_8	66
4.2.1.2	Domaines E_3 et E_9	66
4.2.1.3	Domaines E_{10} et E_{14}	66
4.2.1.4	Domaines E_{29} et E_{35}	67
4.2.1.5	Domaine E_{36}	67
4.2.2	Les domaines E_{24} et E_{27}	68
4.2.3	Le domaine E_{25}	74
4.3	Résolution numérique	78
4.3.1	Redressement des fonctions	78
4.3.2	Schéma de discrétisation	81
4.3.3	Interpolations	82
4.4	Calcul de la probabilité de surchauffe	85
II	Arrêt optimal	97
	Introduction	99
5	Etat de l'art des méthodes numériques	103
5.1	Algorithme de Costa-Davis	103
5.2	Méthode numérique pour les processus de diffusion	104
5.2.1	Démarche	104
5.2.2	Quantification	105
5.3	Spécificités des PDMP	106
6	Calcul de la fonction valeur	109
6.1	Résultats de U.S. Gugerli	109
6.2	Le problème d'arrêt optimal à horizon aléatoire	110
6.3	Notations, définitions et hypothèses	112
6.4	Interprétation probabiliste	114
6.5	Approximation de la fonction valeur	115
6.6	Résultats intermédiaires	117
6.6.1	Propriétés lipschitziennes de J et K	117
6.6.2	Propriétés lipschitziennes des fonctions valeurs	121
6.7	Estimation de l'erreur pour la fonction valeur	125
6.7.1	Premier terme	126
6.7.2	Deuxième terme	126

6.7.3	Troisième terme	127
6.7.4	Quatrième terme	131
6.7.5	Preuve du Théorème	132
7	Calcul d'un temps d'arrêt ϵ-optimal et Exemple	133
7.1	Calcul d'un temps d'arrêt ϵ -optimal	133
7.1.1	Temps d'arrêt ϵ -optimal du PAO_N	133
7.1.2	Règle d'arrêt du PAO approché	133
7.1.3	Convergence	136
7.2	Exemple et résultats numériques	138
	Conclusion générale	141
	Bilan	141
	Perspectives	143
	Table des figures	144
	Liste des tableaux	146
	Annexes	149
A	Tableaux	149
A.1	Tableau récapitulatif	150
A.2	Tableau des équations	157
A.3	Tableau des équations discrétisées	159
B	Programmes	163
B.1	Méthode de Monte Carlo	163
B.1.1	Fichier principal	164
B.1.2	Calcul du temps de saut et du nouveau mode	166
B.2	Approche par les EDP	173
B.2.1	Fichier principal	173
B.2.2	Fichier Paramètres	174
B.2.3	Fichier Equations	175
B.2.4	Fichier Interpolations	177
	Bibliographie	178

Introduction générale

Les préoccupations industrielles actuelles (maîtrise des risques, sûreté de fonctionnement, optimisation de la maintenance, de la production, etc) nécessitent l'utilisation des mathématiques appliquées, et en particulier de la théorie des probabilités, pour construire et étudier des modèles adaptés à la réalité, et prendre en compte les aléas inhérents au fonctionnement de systèmes de plus en plus complexes.

Les processus stochastiques en temps continu permettent de modéliser ces systèmes au comportement partiellement aléatoire. M.H.A. Davis ([20]) les classe en deux catégories :

- celle des processus de diffusion, dont les techniques, fondées sur la théorie des équations différentielles stochastiques et le calcul d'Itô, sont largement développées et unifiées ;
- celle des processus non-diffusifs, c'est-à-dire qui prennent en compte des trajectoires déterministes et des sauts aléatoires, définis par M.H.A. Davis comme « une collection hétérogène de modèles et de méthodologies spécifiques appliquées à des problèmes spécifiques ».

Dans le but de construire une théorie globale pour les processus non-diffusifs, à l'image de celle sur les diffusions, il introduit alors une classe générale de processus stochastiques, les processus markoviens déterministes par morceaux (*Piecewise-Deterministic Markov Processes*, PDMP en abrégé). Les PDMP sont d'une part adaptables à toutes les applications de type non-diffusion, et d'autre part peuvent être étudiés et analysés par des méthodes analogues à celles utilisées pour les diffusions.

Les PDMP sont des modèles dynamiques stochastiques hybrides non-diffusifs, définissant des trajectoires déterministes ponctuées par des sauts aléatoires. Le caractère hybride de ces processus vient du fait qu'ils prennent en compte une première variable discrète m_t caractérisant l'état (ou mode) du processus à l'instant t , correspondant au mode de fonctionnement et/ou à l'environnement ; d'autre part, l'évolution proprement dite du système étudié, décrit par une variable continue euclidienne x_t . Concrètement, si l'on considère l'exemple traité dans la Partie I d'un réservoir chauffé contenant un liquide dont la hauteur est régulée par des vannes d'alimentation et de vidange, le mode discret m_t du processus correspond à la position (ouverte ou fermée) des vannes du réservoir, et la variable continue correspond aux variables physiques du réservoir, à savoir la hauteur et la température.

Les PDMP se construisent de façon itérative : le processus suit une trajectoire, donnée en général par un système d'équations différentielles sur les variables physiques, jusqu'à un premier temps de saut T_1 . Ce premier temps de saut correspond soit à un phénomène aléatoire (panne d'une vanne par exemple, ou changement d'environnement), soit à l'atteinte d'une frontière de l'espace d'état (par exemple, dépassement d'une certaine hauteur ou température dans le cas du réservoir). Dans les deux cas, un nouveau mode est sélectionné d'une façon aléatoire, et le processus suit une nouvelle trajectoire déterministe jusqu'au prochain temps de saut T_2 , etc.

Comme nous le voyons avec l'exemple du réservoir, les PDMP permettent de modéliser le comportement physique d'un système hybride, pour par exemple, calculer des indicateurs de fiabilité (probabilités de surchauffe, d'assèchement ou de débordement dans le cas du réservoir). L'utilisation des PDMP s'étend aujourd'hui à des domaines de plus en plus variés : mécanismes de dégradation ([13]), planification de la production et de la maintenance ([8]), génétique ([77]), contrôle optimal ([35, 16, 19, 22, 25]) et impulsif ([17, 15, 26]).

De récentes recherches s'intéressent aux propriétés des PDMP. On peut citer par exemple les travaux de F. Dufour et O.L.V. Costa sur des propriétés de stabilité et d'ergodicité des PDMP ([31, 18]), ceux de J. Chiquet et N. Limnios sur une méthode numérique pour calculer la fonction de transition d'un PDMP ([12]), la représentation des PDMP par des réseaux de Petri colorés par M.H.C. Everdij et H.A.P. Blom ([34]), ou encore l'étude des densités et semi-groupes sous-stochastiques des PDMP par M. Tyran-Kaminska ([75]).

L'objectif des travaux de thèse présentés ici est double. D'une part, nous étudions un cas-test de la littérature en sûreté de fonctionnement dans le contexte des PDMP introduits par M.H.A. Davis dans [20]. Il est à noter qu'une sous-classe de ce type de processus a été introduite par J. Devooght dans [28]. Deux méthodes numériques sont proposées pour le cas spécifique du réservoir chauffé et les résultats seront comparés à ceux de la littérature. D'autre part, la partie la plus importante de la thèse a consisté à apporter une contribution théorique à l'étude des PDMP, en particulier nous proposons une méthode numérique et des résultats de convergence pour les problèmes d'arrêt optimal à horizon aléatoire (PAO_n). Ce mémoire de thèse est donc divisé en deux parties distinctes reprenant les objectifs cités ci-dessus, précédées d'un chapitre introductif sur les PDMP et leurs propriétés.

Plus précisément,

- Le *Chapitre 1* introduit la définition d'un PDMP, après quelques rappels sur les équations différentielles ordinaires et les champs de vecteurs (Section 1.1). Des résultats de M.H.A. Davis sur l'espérance de fonctionnelles d'un PDMP, nécessaires pour la méthode numérique proposée au Chapitre 4, sont donnés Section 1.4 : il est démontré que sous certaines hypothèses, la fonctionnelle d'un PDMP est solution unique d'une équation intégral-différentielle. Le chapitre se termine sur quelques résultats concernant la structure des temps d'arrêt pour les PDMP (Section 1.5) qui seront utilisés dans les Chapitres 5 et 7.

- La *Partie I* s'intéresse aux PDMP dans le contexte de la fiabilité dynamique. Nous étudions ici deux méthodes numériques pour calculer les indicateurs de fiabilité d'un réservoir, cas-test issu de la littérature en sûreté de fonctionnement et présenté dans le *Chapitre 2*.
- La première méthode numérique (*Chapitre 3*) repose sur la résolution du système différentiel régissant l'évolution des grandeurs physiques du réservoir, et l'utilisation d'un algorithme de simulation d'une variable aléatoire d'intensité variable ([14]). Le calcul des probabilités de défaillance du réservoir se fait par la méthode de Monte Carlo.
- La deuxième méthode numérique (*Chapitre 4*) utilise les résultats de M.H.A. Davis sur les fonctionnelles de PDMP, présentées au Chapitre 1, pour calculer la probabilité de surchauffe du réservoir. A partir de ces résultats, nous démontrons que l'équation intégral-différentielle introduite au Chapitre 1 s'écrit sous la forme d'un système différentiel couplé dans le cas du réservoir (Section 4.1). Puis, après avoir étudié les domaines de définition des fonctions solutions (Section 4.2), nous proposons un schéma numérique pour les calculer (Section 4.3).
- La *Partie II* s'intéresse au problème d'arrêt optimal à horizon aléatoire (PAO).
 - En premier lieu, nous donnons dans le *Chapitre 5* les principaux résultats connus dans le cadre des PDMP pour les problèmes d'arrêt optimal à horizon infini ainsi que la démarche numérique pour les processus de diffusion en temps continu (Section 5.2). En particulier, dans la Section 5.1, nous présentons l'algorithme proposé par O.L.V. Costa et M.H.A. Davis ([16]) pour le calcul numérique de la fonction valeur. Dans la Section 5.2, nous nous intéressons aux méthodes numériques développées dans le cas des processus de diffusion en temps continu, notamment au procédé de quantification de la chaîne de Markov en temps discret du processus. Dans la Section 5.3, nous regardons pourquoi la méthode numérique pour les diffusions ne s'applique pas au cas spécifique des PDMP.
 - Dans le *Chapitre 6*, nous proposons une méthode numérique pour le calcul de la fonction valeur d'un PAO à horizon aléatoire pour les PDMP (Section 6.2), basée sur les méthodes de quantification et sur les résultats théoriques de U.S. Gugerli ([42]) concernant le calcul de la fonction valeur du PAO à horizon infini par l'itération d'un opérateur que nous expliciterons ici (Section 6.1), et l'interprétation probabiliste de cet opérateur (Section 6.4).
 - Le *Chapitre 7* est dédié au calcul d'un temps d'arrêt ϵ -optimal du PAO (Section 7.1). Un exemple et des résultats numériques sont ensuite donnés dans la Section 7.2.
- Une *Annexe* regroupe divers résultats techniques sous forme de tableaux et les codes correspondant aux deux méthodes numériques de la Partie I.

Chapitre 1

Les Processus Markoviens Déterministes Par Morceaux

Ce chapitre introductif a pour but de poser les notations, les définitions et les propriétés nécessaires pour les Parties I et II. Avant de donner la définition des PDMP, nous rappelons tout d'abord quelques notations et propriétés sur les équations différentielles ordinaires et les champs de vecteurs. Nous introduirons ensuite les résultats sur le calcul d'espérance de fonctionnelle d'un PDMP. Puis, nous donnerons quelques résultats sur la structure des temps d'arrêt pour les PDMP. La plupart des énoncés de ce chapitre sont tirés de [21] et toutes les références entre parenthèses s'y rapportent.

1.1 Equations différentielles ordinaires et champs de vecteurs

Nous commençons par quelques rappels sur les champs de vecteurs. Dans toute la suite, d est un entier supérieur ou égal à 1.

Définition 1.1 Soit une fonction $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$. On dit que g est Lipschitz s'il existe une constante K telle que $|g(x) - g(y)| \leq K|x - y|$ pour tout $x, y \in \mathbb{R}^d$.

Définition 1.2 Soit $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$ une fonction Lipschitz. Alors l'équation différentielle ordinaire (EDO)

$$\frac{d}{dt}x(t) = g(x(t)), \quad x(0) = x_0 \in \mathbb{R}^d \quad (1.1.1)$$

a une unique solution $t \mapsto x(t)$ de classe C^1 , définie pour tout $t \in \mathbb{R}$.

On note cette solution $x(\cdot) = \phi(x_0, \cdot)$.

Définition 1.3 ϕ s'appelle un flot ou un champ de vecteurs.

Du fait de l'unicité de la solution, ϕ vérifie les deux propriétés suivantes :

Propriété 1.1 L'application $\phi_t : x \mapsto \phi(x, t)$ est bijective, de réciproque continue. C'est donc un homéomorphisme. En effet, $\phi^{-1}(x, t) = \phi(x, -t)$ pour tout $x \in \mathbb{R}^d$.

Propriété 1.2 La famille $(\phi_t)_{t \in \mathbb{R}}$ est un groupe, c'est-à-dire que pour tout $s, t \in \mathbb{R}$, $\phi_{t+s} = \phi_t \circ \phi_s$, ou plus explicitement, $\phi(x, t+s) = \phi(\phi(x, s), t)$, pour tout $x \in \mathbb{R}^d$.

Définition 1.4 Soit $f : \mathbb{R}^d \rightarrow \mathbb{R}$ une fonction C^1 . Alors

$$\frac{d}{dt}f(\phi(x, t)) = \sum_{i=1}^d \frac{\partial f}{\partial x_i}(\phi(x, t))g^i(\phi(x, t)),$$

où g^i est la i -ème composante de g .

Proposition 1.1 On note \mathfrak{X} l'opérateur différentiel de premier ordre

$$\mathfrak{X}f(x) = \sum_{i=1}^d \frac{\partial f}{\partial x_i}(x)g^i(x). \quad (1.1.2)$$

Alors $\phi(x, \cdot)$ satisfait (1.1.1) si et seulement si $\phi(x, \cdot)$ satisfait pour tout $f \in C^\infty(\mathbb{R}^d)$ et tout $t \in \mathbb{R}$

$$\frac{d}{dt}f(\phi(x, t)) = \mathfrak{X}f(\phi(x, t)). \quad (1.1.3)$$

1.2 Définition

Soit \mathbb{M} un ensemble dénombrable dont les éléments seront appelés modes. Pour tout $m \in \mathbb{M}$, soit E_m un sous-ensemble ouvert de \mathbb{R}^d , δE_m sa frontière et \overline{E}_m sa fermeture.

Soit

$$E = \bigcup_{m \in \mathbb{M}} E_m \times \{m\}$$

et

$$\delta E = \bigcup_{m \in \mathbb{M}} \delta E_m \times \{m\}.$$

On note $\mathcal{B}(\overline{E})$ la tribu borélienne de \overline{E} .

Définition 1.5 Si E est un ensemble dénombrable, un noyau markovien sur $E \times \mathcal{B}(E)$ est une application $Q : E \times \mathcal{B}(E) \mapsto [0, 1]$ telle que

$$\forall x, Q(x, \cdot) \text{ est une probabilité sur } \mathcal{B}(E).$$

On peut aussi définir le noyau markovien de cette façon :

Définition 1.6 Soit Q un noyau markovien sur $(E, \mathcal{B}(E))$ et $w \in \mathbb{B}(E)$. Alors

$$Qw(x) = \int_M w(y)Q(x, dy).$$

Un PDMP est déterminé par ses caractéristiques locales $(\phi_m, \lambda_m, Q_m)_{m \in \mathbb{M}}$, où

- L'application $\phi_m : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$ est un flot.
Pour tout (x, m) dans E , on note

$$t_m^*(x) = \inf\{t > 0, \phi_m(x, t) \in \delta E\}$$

avec la convention $\inf \emptyset = \infty$.

Alors t_m^* est le temps pour lequel le flot atteint la frontière de l'espace d'état.

- L'intensité de saut $\lambda_m : \overline{E} \rightarrow \mathbb{R}^+$ est une fonction mesurable, caractérisant la fréquence des sauts, et qui satisfait

$$\forall (x, m) \in E, \exists \epsilon > 0 \text{ tel que } \int_0^\epsilon \lambda_m(\phi_m(x, s)) ds < \infty.$$

- La mesure de transition du processus $Q_m : \overline{E} \times \mathcal{B}(\overline{E}) \rightarrow [0, 1]$ vérifie pour tout couple $(x, m) \in \overline{E}$

$$Q_m(x, E - \{(x, m)\}) = 1,$$

c'est-à-dire que le processus doit sauter vers un nouveau mode et/ou une nouvelle position.

Remarque 1.1 *Au moment d'une sollicitation, par exemple dans le cas du réservoir chauffé, une vanne peut se déclencher ou se bloquer. Dans les deux cas, cela correspondra à un changement de modèle, c'est-à-dire soit à un changement de mode, soit à un changement de dynamique des variables d'états définissant le système. La définition de la mesure Q telle qu'elle est donnée ici permet de prendre en compte ce genre de situations. La situation qui est exclue dans le modèle de M.H.A. Davis est celle d'un instant de saut fictif qui ne correspondrait ni à un changement de mode ou de position, ni à un changement de dynamique.*

A partir de ces caractéristiques, d'après le Théorème (25.5), il existe un espace de probabilité $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}, \{\mathbb{P}_x\}_{x \in E})$ tel que $\{X(t)\}_{t \geq 0}$, avec $X(t) = (x_t, m_t)$, est un processus de Markov fort dont les trajectoires peuvent être définies de manière itérative de la façon suivante.

Au temps $T_0 = 0$, on part du point initial $X(0) = (x_0, m_0)$ avec $(x_0, m_0) \in E$. Le premier instant de saut T_1 est déterminé, pour $t > 0$, par

$$\mathbb{P}_{(x_0, m_0)}(T_1 > t) = \begin{cases} e^{-\Lambda_{m_0}(x_0, t)} & \text{pour } t < t_{m_0}^*(x_0) \\ 0 & \text{pour } t \geq t_{m_0}^*(x_0) \end{cases}$$

où pour $(x, m) \in E$ et $t \in [0, t_m^*(x)]$

$$\Lambda_m(x, t) = \int_0^t \lambda_m(\phi_m(x, s)) ds.$$

Le premier temps de saut T_1 est donc le minimum entre un temps aléatoire donné par l'intensité λ_m et le temps déterministe d'atteinte d'une frontière t_m^* . Dans la suite, nous appellerons *sauts déterministes* les sauts qui se produisent à la frontière

et *sauts aléatoires* les autres.

Sur l'intervalle de temps $[0, T_1[$, $m_t = m_0$ et le processus x_t suit la trajectoire déterministe donnée par le flot $\phi_{m_0}(x_0, t)$ (Figure 1.1). A l'instant aléatoire T_1 , le processus subit un saut. On sélectionne alors via le noyau Q_{m_0} une variable aléatoire $Z_1 = (x_1, m_1)$ à valeurs dans E et qui a pour distribution $Q_{m_0}(\phi_{m_0}(x_0, T_1), \cdot)$.

Pour résumer, la trajectoire de $\{X(t)\}$ partant de $X(0)$, pour $t \leq T_1$, est donnée par

$$X(t) = \begin{cases} (\phi_{m_0}(x_0, t), m_0) & \text{pour } t < T_1, \\ Z_1 & \text{pour } t = T_1. \end{cases}$$

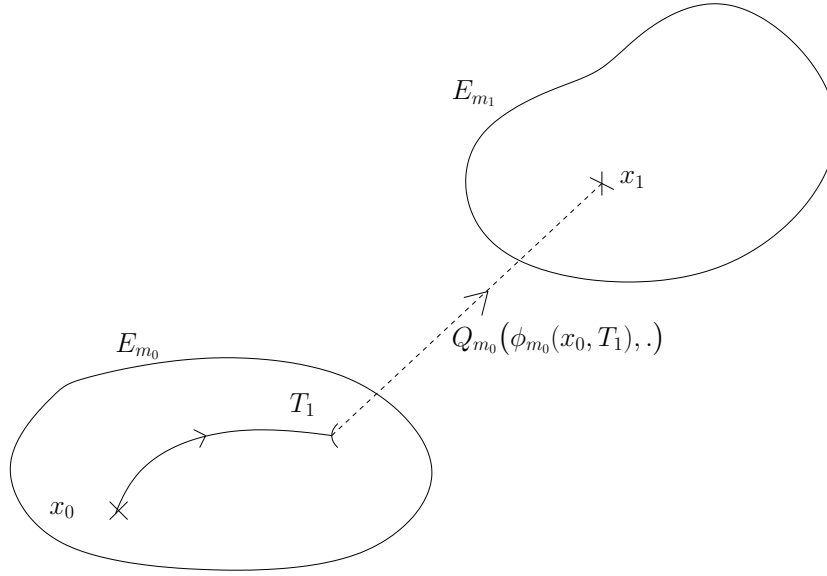


FIG. 1.1 – Trajectoire du PDMP au premier temps de saut T_1 aléatoire.

Le processus repart ensuite de $X(T_1) = Z_1$ et par la même démarche que précédemment, on construit la trajectoire de $\{X(t)\}$ sur l'intervalle $[T_1, T_2]$ où T_2 est l'instant de saut suivant, et $T_2 - T_1$ a pour loi $(\phi_{m_1}(x_1, T_2 - T_1), m_1)$. Ainsi, on a

$$X(t) = \begin{cases} (\phi_{m_1}(x_1, t - T_1), m_1) & \text{pour } t \in [T_1, T_2[, \\ Z_2 & \text{pour } t = T_2. \end{cases}$$

où $Z_2 = (x_2, m_2)$ a pour distribution $Q_{m_1}(\phi_{m_1}(x_1, T_2 - T_1), \cdot)$ (Figure 1.2). Et ainsi de suite.

Associé au processus $\{X(t)\}$, il existe un processus à temps discret $(\Theta_n)_{n \in \mathbb{N}}$ défini par $\Theta_n = (Z_n, S_n)$ avec

$$Z_n = X(T_n) \quad \text{et} \quad \begin{cases} S_0 = 0 \\ S_n = T_n - T_{n-1} & \text{pour } n \geq 1. \end{cases}$$

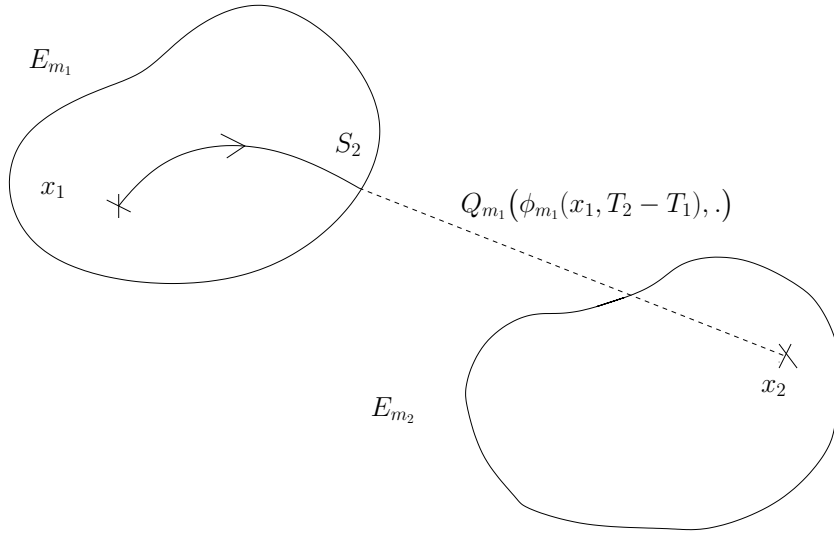


FIG. 1.2 – Trajectoire du PDMP au deuxième temps de saut T_2 déterministe.

Remarque 1.2 *La définition des PDMP donnée par M.H.A. Davis est beaucoup plus large que la définition qu'en donne par exemple J. Devooght dans [28]. En effet, la modélisation faite par M.H.A. Davis autorise des sauts sur les variables euclidiennes, ces sauts pouvant être d'origine soit déterministe, soit stochastique. Ces deux possibilités ne peuvent pas être prises en compte par la modélisation faite par J. Devooght (par exemple dans [29]).*

Proposition 1.2 *Le processus à temps discret $(\Theta_n)_{n \in \mathbb{N}}$ est une chaîne de Markov.*

On introduit ici une hypothèse standard.

Hypothèse 1.1 *Pour tout $(x, m) \in E$ et $t \in \mathbb{R}^+$, $\mathbf{E}_{(x,m)} \left[\sum_{k \in \mathbb{N}} \mathbf{1}_{\{T_k \leq t\}} \right] < \infty$.*

En particulier, cela implique que $T_k \rightarrow \infty$ quand $k \rightarrow \infty$.

Remarque 1.3 *Dans les applications pratiques, \mathbb{M} est un ensemble fini et correspond aux modes de fonctionnement du système ou à son environnement extérieur. De plus, l'espace d'états E est en général la réunion de sous-ensembles E_m de \mathbb{R}^d indexés par le mode m : $E = \cup_{m \in \mathbb{M}} E_m$.*

1.3 Exemples de PDMP

Nous allons dans cette section donner quelques exemples de PDMP.

1.3.1 Processus de Poisson

Les processus de Poisson ([76, 68]) permettent de modéliser des situations telles que les arrivées de clients dans une file d'attente, les pannes de composants électroniques, les appels à un central téléphonique ... Nous donnons ici une définition d'un processus de Poisson, issue de [76] :

Définition 1.7 Soit (T_n) une suite de variables aléatoires indépendantes et de même loi exponentielle $\mathcal{E}(\lambda)$. Posons $S_0 = 0$, et pour tout $n \geq 1$:

$$S_n = T_1 + \cdots + T_n.$$

Pour tout $t \geq 0$, définissons la variable aléatoire N_t , à valeurs dans \mathbb{N} , par :

$$N_t = n \iff S_n \leq t < S_{n+1}.$$

On appelle processus de Poisson d'intensité λ l'ensemble de variables aléatoires :

$$\{N_t ; t \geq 0\}.$$

Pour le cas de l'arrivée de clients dans une file d'attente, T_n est la durée séparant l'arrivée du $(n-1)$ -ème client de celle du n -ème client, S_n la date d'arrivée du n -ème client et N_t le nombre de clients arrivés jusqu'à l'instant t .

Un processus de Poisson est donc un PDMP d'intensité constante λ et de flot nul entre les sauts T_n . En effet, le processus saute de 1 à chaque instant aléatoire T_n , qui suivent une loi exponentielle $\mathcal{E}(\lambda)$.

Plus généralement, les processus ponctuels sont des PDMP, avec (T_n) une suite de variables aléatoires réelles telles que pour tout n , $T_n \leq T_{n+1}$. L'intensité moyenne d'un processus de comptage est donnée par

$$\lambda = \lim_{t \rightarrow \infty} \frac{\mathbb{E}[N_t]}{t}$$

quand cette limite existe.

1.3.2 Exemple simple

Nous donnons ici un exemple simple de PDMP, développé dans la Partie II.

Si l'on pose $E = [0, 1[$ l'espace d'état, $\partial E = \{1\}$ sa frontière, on définit le flot du PDMP sur $[0, 1]$ par

$$\phi(x, t) = x + ct$$

avec c une constante positive. L'intensité de saut λ est donnée sur $[0, 1]$ par

$$\lambda(x) = \beta x^\alpha$$

avec $\beta > 0$ et $\alpha \geq 1$, et on définit pour tout $x \in [0, 1]$ le noyau markovien $Q(x, \cdot)$ comme la loi uniforme sur $[0, 1/2]$.

Alors le processus avance à vitesse constante c vers 1, mais plus il s'approche de la frontière 1, plus la probabilité de revenir sur $[0, 1/2]$ est grande. La Figure 1.3 montre deux trajectoires de ce processus pour $x_0 = 0$, $c = \alpha = 1$ et $\beta = 3$ jusqu'au 10-ème saut.

Nous donnerons par la suite l'exemple plus complexe du réservoir, mais le principe des PDMP est le même. Le flot ϕ décrit l'évolution de variables physiques (température, pression, ...). Les sauts, aléatoires ou déterministes, correspondent à des pannes ou changements d'environnement : lorsque un certain seuil est atteint par une variable physique (par exemple, dépassement d'une certaine hauteur dans une cuve), le système réagit, par l'ouverture ou la fermeture de vannes par exemple. Les PDMP sont donc particulièrement adaptés à la fiabilité dynamique.

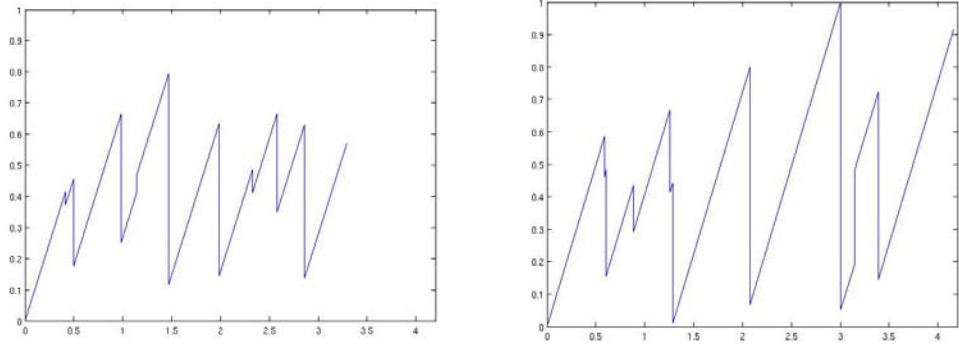


FIG. 1.3 – Deux trajectoires du PDMP (temps en abscisse).

1.4 Espérance de fonctionnelle d'un PDMP

La méthode numérique employée dans le Chapitre 4 repose sur les résultats du Chapitre 3 de [21]. En particulier, le calcul de l'espérance de fonctionnelle d'un PDMP revient à résoudre une équation intégrale-différentielle. La fonctionnelle est l'unique solution de l'équation dans une certaine classe de fonctions bien choisie, et M.H.A. Davis propose une méthode itérative générale permettant d'obtenir cette solution.

Définition 1.8 On définit pour tout $t \in \mathbb{R}$ le processus de comptage des sauts à la frontière du PDMP p_t^* par

$$p_t^* = \sum_{i=1}^{\infty} \mathbb{1}_{\{T_i \leq t\}} \mathbb{1}_{\{X(T_i^-) \in \partial E\}}.$$

On étudie alors la fonctionnelle

$$V(x, m) = \mathbb{E}_{(x, m)} \left[\int_0^{\infty} l(X(t)) dt + \int_0^{\infty} c(X(t_-)) dp_t^* \right]. \quad (1.4.1)$$

où

- $X(t) = (x_t, m_t)$ est la trajectoire du PDMP,
- $l : E \rightarrow \mathbb{R}^+$ et $c : \partial E \rightarrow \mathbb{R}^+$ sont des fonctions mesurables.

On rappelle le résultat principal (Théorème (32.2)).

Théorème 1.1 On suppose que pour tout $(x, m) \in E$, la fonction $t \mapsto l(\phi_m(x, t), m)$ est intégrable sur tout intervalle $[0, \epsilon]$, $\epsilon > 0$ et que la fonction V définie par (1.4.1) est bornée. Alors V est absolument continue et satisfait

$$\frac{d}{dt} V(x, m) + \lambda_m(x) \left(Q_m V(x, m) - V(x, m) \right) + l(x, m) = 0, \quad (x, m) \in E, \quad (1.4.2)$$

avec la condition aux bords

$$Q_m V(x, m) - V(x, m) + c(x, m) = 0, \quad (x, m) \in \partial E. \quad (1.4.3)$$

L'unicité est donnée par le résultat suivant (Théorème (32.10)) :

Théorème 1.2 *On suppose que $f : E \rightarrow \mathbb{R}^+$ est absolument continue et satisfait*

$$\begin{aligned} \frac{d}{dt} f(x, m) + \lambda_m(x) \left(Q_m f(x, m) - f(x, m) \right) + l(x, m) &= 0, \quad (x, m) \in E \\ Q_m f(x, m) - f(x, m) + c(x, m) &= 0, \quad (x, m) \in \partial E \end{aligned}$$

On suppose de plus que $\mathbb{E}_{(x,m)}[f(x_t, m_t)] \rightarrow 0$ quand $t \rightarrow \infty$ pour tout $(x, m) \in E$ et que

$$\mathbb{E}_{(x,m)} \left[\sum_{i=1}^{\infty} \int_E |f(x_{T_i}, m_{T_i}) - f(x_{T_i^-}, m_{T_i^-})| \right] < \infty. \quad (1.4.4)$$

Alors

$$f(x, m) = \mathbb{E}_{(x,m)} \left[\int_0^\infty l(x_t, m_t) dt + \int_0^\infty c(x_{t-}, m_{t-}) dp_t^* \right]. \quad (1.4.5)$$

Nous utiliserons plus particulièrement le Corollaire (32.15) :

Corollaire 1.1 *On suppose que le PDMP $\{X(t)\}$ satisfait les conditions du Théorème 1.1 et que V définie par (1.4.1) est bornée. Alors si $\mathbb{E}_{(x,m)}[V(\phi_m(x, t))] \rightarrow 0$ quand $t \rightarrow \infty$ pour tout $(x, m) \in E$, V est l'unique solution absolument continue de (1.4.2) et (1.4.3).*

Remarque 1.4 *Pour simplifier l'écriture, nous noterons à partir de maintenant $f_m(x) = f(x, m)$ pour tout $(x, m) \in E$.*

1.5 Structure des temps d'arrêt pour les PDMP

Les calculs de la fonction valeur des Chapitres 5 et 7 nécessitent l'étude plus précise de la structure des temps d'arrêt des PDMP. Les résultats qui suivent sont dans ([23]), paru dans *Annals of Applied Probability*.

Soit τ un $\{\mathcal{F}_t\}_{t \in \mathbb{R}^+}$ -temps d'arrêt. Nous rappelons ici le résultat important donné par M.H.A Davis dans [21].

Théorème 1.3 *Il existe une suite de variables aléatoires non-négatives $(R_n)_{n \in \mathbb{N}^*}$ telle que R_n est $\mathcal{F}_{T_{n-1}}$ -mesurable et $\tau \wedge T_{n+1} = (T_n + R_{n+1}) \wedge T_{n+1}$ sur $\{\tau \geq T_n\}$.*

Lemme 1.1 *On définit $\bar{R}_1 = R_1$, et $\bar{R}_k = R_k \mathbf{1}_{\{S_{k-1} \leq \bar{R}_{k-1}\}}$. Alors on a*

$$\tau = \sum_{n=1}^{\infty} \bar{R}_n \wedge S_n.$$

Preuve : D'une façon évidente, sur $\{T_k \leq \tau < T_{k+1}\}$, on a $R_j \geq S_j$ et $R_{k+1} < S_{k+1}$ pour tout $j \leq k$. Par conséquent, par définition $\bar{R}_j = R_j$ pour tout $j \leq k+1$, d'où

$$\begin{aligned} \sum_{n=1}^{\infty} \bar{R}_n \wedge S_n &= \sum_{n=1}^k \bar{R}_n \wedge S_n + \{\bar{R}_{k+1} \wedge S_{k+1}\} + \sum_{n=k+2}^{\infty} \bar{R}_n \wedge S_n \\ &= T_k + R_{k+1} + \sum_{n=k+2}^{\infty} \bar{R}_n \wedge S_n. \end{aligned}$$

Comme $\bar{R}_{k+1} = R_{k+1} < S_{k+1}$ nous avons $\bar{R}_j = 0$ pour tout $j \geq k+2$. Ainsi, $\sum_{n=1}^{\infty} \bar{R}_n \wedge S_n = T_k + R_{k+1} = \tau$, ce qui montre le résultat. \square

Il existe une suite d'applications mesurables $(r_k)_{k \in \mathbb{N}^*}$ définies sur $E \times (\mathbb{R}^+ \times E)^{k-1}$ à valeurs dans \mathbb{R}^+ satisfaisant

$$\begin{aligned} R_1 &= r_1(Z_0), \\ R_k &= r_k(Z_0, \Gamma_{k-1}), \end{aligned}$$

car R_n est $\mathcal{F}_{T_{n-1}}$ -mesurable pour tout n , et où $\Gamma_k = (S_1, Z_1, \dots, S_k, Z_k)$.

Définition 1.9 Soit $p \in \mathbb{N}^*$. Soit $(\hat{R}_k)_{k \in \mathbb{N}^*}$ une suite d'applications définies sur $E \times (\mathbb{R}^+ \times E)^p \times \Omega$ à valeurs dans \mathbb{R}^+ et définies par

$$\hat{R}_1(y, \gamma, \omega) = r_{p+1}(y, \gamma),$$

et pour $k \geq 2$

$$\hat{R}_k(y, \gamma, \omega) = r_{p+k}(y, \gamma, \Gamma_{k-1}(\omega)) \mathbf{1}_{\{S_{k-1} \leq \hat{R}_{k-1}\}}(y, \gamma, \omega).$$

Proposition 1.3 Supposons $T_p \leq \tau \leq T_N$. Alors on a

$$\tau = T_p + \hat{\tau}(Z_0, \Gamma_p, \theta_{T_p}),$$

où $\hat{\tau}: E \times (\mathbb{R}^+ \times E)^p \times \Omega \rightarrow \mathbb{R}^+$ est définie par

$$\hat{\tau}(y, \gamma, \omega) = \sum_{n=1}^{N-p} \hat{R}_n(y, \gamma, \omega) \wedge S_n(\omega), \quad (1.5.1)$$

et θ est l'opérateur de décalage sur $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \in \mathbb{R}^+}, \mathbb{P})$.

Preuve : Tout d'abord, nous allons montrer par récurrence sur $k \in \mathbb{N}^*$ le résultat suivant :

$$\hat{R}_k(Z_0, \Gamma_p, \theta_{T_p}) = \bar{R}_{p+k}. \quad (1.5.2)$$

En effet, on a $\hat{R}_1(Z_0, \Gamma_p, \theta_{T_p}) = R_{p+1}$, et sur l'ensemble $\{\tau \geq T_p\}$, on a également $R_{p+1} = \bar{R}_{p+1}$. Par conséquent, $\hat{R}_1(Z_0, \Gamma_p) = \bar{R}_{p+1}$. Maintenant, supposons que $\hat{R}_k(Z_0, \Gamma_p, \theta_{T_p}) = \bar{R}_{p+k}$. Alors on a

$$\begin{aligned} &\hat{R}_{k+1}(Z_0(\omega), \Gamma_p(\omega), \theta_{T_p}(\omega)) \\ &= r_{p+k+1}(Z_0(\omega), \Gamma_p(\omega), \Gamma_k(\theta_{T_p}(\omega))) \mathbf{1}_{\{S_k \leq \hat{R}_k\}}(Z_0(\omega), \Gamma_p(\omega), \theta_{T_p}(\omega)). \end{aligned}$$

Par définition, on a $\Gamma_k(\theta_{T_p}(\omega)) = (S_{p+1}(\omega), Z_{p+1}(\omega), \dots, S_{p+k}(\omega), Z_{p+k}(\omega))$ et l'hypothèse de récurrence nous donne facilement

$$\mathbf{1}_{\{S_k \leq \widehat{R}_k\}}(Z_0(\omega), \Gamma_p(\omega), \theta_{T_p}(\omega)) = \mathbf{1}_{\{S_{p+k} \leq \overline{R}_{p+k}\}}(\omega).$$

Ainsi nous obtenons $\widehat{R}_{k+1}(Z_0, \Gamma_p, \theta_{T_p}) = \overline{R}_{p+k+1}$, ce qui montre (1.5.2).

En combinant les équations (1.5.1) et (1.5.2), nous avons

$$\widehat{\tau}(Z_0, \Gamma_p, \theta_{T_p}) = \sum_{n=1}^{N-n} \overline{R}_{p+n} \wedge S_{p+n}. \quad (1.5.3)$$

Cependant, comme énoncé précédemment, nous savons que sur $\{T \geq T_p\}$, on a $R_k = \overline{R}_k \geq S_k$, pour $k \leq p$. Par conséquent, en utilisant (1.5.3), nous avons

$$T_p + \widehat{\tau}(Z_0, \Gamma_p, \theta_{T_p}) = \sum_{k=1}^p S_k + \sum_{k=p+1}^N \overline{R}_k \wedge S_k = \sum_{k=1}^N \overline{R}_k \wedge S_k.$$

Comme $\tau \leq T_N$, nous obtenons du Lemme 1.1 et de sa preuve que $\tau = \sum_{n=1}^N \overline{R}_n \wedge S_n$, ce qui montre le résultat. \square

Proposition 1.4 *Soit $(U_n)_{n \in \mathbb{N}^*}$ une suite de variables aléatoires non-négatives telle que U_n est $\mathcal{F}_{T_{n-1}}$ -mesurable et $U_{n+1} = 0$ sur $\{S_n > U_n\}$, pour tout $n \in \mathbb{N}^*$. Posons*

$$U = \sum_{n=1}^{\infty} U_n \wedge S_n.$$

Alors U est un $\{\mathcal{F}_t\}_{t \in \mathbb{R}^+}$ -temps d'arrêt.

Preuve : L'hypothèse 1.1 nous donne

$$\begin{aligned} \{U \leq t\} = \bigcup_{n=0}^{\infty} \bigg[& \left(\{T_n \leq U < T_{n+1}\} \cap \{U \leq t\} \cap \{t < T_{n+1}\} \right) \\ & \cup \left(\{T_n \leq U < T_{n+1}\} \cap \{U \leq t\} \cap \{T_{n+1} \leq t\} \right) \bigg] \end{aligned} \quad (1.5.4)$$

Par définition de U_n , on a $\{U \geq T_n\} = \{U_n \geq S_n\}$, donc

$$\begin{aligned} \{T_n \leq U < T_{n+1}\} \cap \{U \leq t\} \cap \{t < T_{n+1}\} &= \{S_n \leq U_n\} \cap \{T_n + U_{n+1} \leq t\} \\ &\cap \{T_n \leq t\} \cap \{t < T_{n+1}\}. \end{aligned}$$

Le Théorème 2.10 *ii*) dans [33] nous donne

$$\{S_n \leq U_n\} \cap \{T_n + U_{n+1} \leq t\} \cap \{T_n \leq t\} \in \mathcal{F}_t,$$

nous avons donc

$$\{T_n \leq U < T_{n+1}\} \cap \{U \leq t\} \cap \{t < T_{n+1}\} \in \mathcal{F}_t. \quad (1.5.5)$$

D'autre part,

$$\{T_n \leq U < T_{n+1}\} \cap \{U \leq t\} \cap \{T_{n+1} \leq t\} = \{S_n \leq U_n\} \cap \{U_{n+1} < S_{n+1}\} \cap \{T_{n+1} \leq t\}.$$

Toujours d'après le Theorème 2.10 *ii*) dans [33],

$$\{T_n \leq U < T_{n+1}\} \cap \{U \leq t\} \cap \{T_{n+1} \leq t\} \in \mathcal{F}_t. \quad (1.5.6)$$

Ainsi, de (1.5.4), (1.5.5), et (1.5.6) nous obtenons le résultat. \square

Corollaire 1.2 *Pour tout $(y, \gamma) \in E \times (\mathbb{R}^+ \times E)^p$, $\hat{\tau}(y, \gamma, \cdot)$ est un $\{\mathcal{F}_t\}_{t \in \mathbb{R}^+}$ -temps d'arrêt satisfaisant $\hat{\tau}(y, \gamma, \cdot) \leq T_{N-p}$.*

Preuve : Il vient de la définition de \hat{R}_k que $\hat{R}_k(y, \gamma, \omega) < S_k(\omega)$ implique $\hat{R}_{k+1}(y, \gamma, \omega) = 0$ et la variable aléatoire positive $\hat{R}_k(y, \gamma, \cdot)$ est $\mathcal{F}_{T_{k-1}}$ -mesurable. Alors, par la Proposition 1.4, $\hat{\tau}(y, \gamma, \cdot)$ est un $\{\mathcal{F}_t\}_{t \in \mathbb{R}^+}$ -temps d'arrêt. Enfin, par définition de $\hat{\tau}$, équation (1.5.1), on a $\hat{\tau}(y, \gamma, \cdot) \leq \sum_{n=1}^{N-p} S_n = T_{N-p}$ ce qui montre le résultat. \square

Après cette partie préliminaire sur les PDMP, nous allons maintenant passer au sujet principal de ce mémoire.

Première partie

Etude d'un cas-test de la Sûreté de Fonctionnement par les PDMP : le réservoir chauffé

Introduction

De tout temps, l'Homme cherche à appréhender, analyser et contrôler, s'il le peut, les risques auxquels il est confronté, qu'ils soient d'ordre naturel, comme les séismes, les éruptions volcaniques et les inondations, ou technologiques. La Maîtrise des Risques est née de ce besoin d' « identifier, comprendre, évaluer, hiérarchiser les risques, puis [de] les passer sous contrôle, [de] les dominer, [de] les maîtriser, lorsque les risques s'avèrent négatifs (conduisant à une perte certaine) ou, au contraire, [de] saisir des opportunités lorsque les risques s'avèrent positifs (conduisant à un gain, un bénéfice » (A. Lannoy,[52]). Plusieurs disciplines sont issues de la Maîtrise des Risques : les cyndiniques (sciences qui étudient les risques naturels, technologiques ou domestiques, et leurs préventions), la gestion des risques ou encore la sûreté de fonctionnement.

C'est après la seconde guerre mondiale, durant la guerre froide, que la sûreté de fonctionnement se développe : la course à l'armement et aux technologies de pointe fait que l'étude de cette discipline est au départ destinée à la défense, l'aéronautique, l'aérospatial, l'électronique ou les télécommunications ([52]). Avec l'évolution des technologies dans les industries de produits « grand public », la sûreté de fonctionnement s'étend dans les années 1980-1990 aux industries des processus, de l'équipement, de l'automobile ou de l'électroménager, en cherchant à optimiser la maintenabilité, la disponibilité, la sécurité et la fiabilité des systèmes industriels. C'est à ce moment là que la définition de la sûreté de fonctionnement est clarifiée et formalisée. Avant les années 80, on amalgame, sous le mot de fiabilité, la disponibilité, la maintenance, la sécurité et donc la fiabilité. A la fin des années 80, on désigne sous l'expression de *sûreté de fonctionnement* l'étude de ces quatre propriétés. Des méthodes sont développées, telles que les arbres de défaillance, les réseaux de Petri ou les réseaux bayésiens. Mais, comme le fait remarquer P.-E. Labeau dans ([49]), ces méthodes sont basées sur des modèles booléens ou discrets, qui ne prennent pas en compte de façon intégrée le caractère continu des variables physiques (par exemple la température, la pression, ...) des systèmes étudiés.

La fiabilité dynamique, discipline de la sûreté de fonctionnement introduite dans les années 90 par Jacques Devooght ([28]), répond à cette préoccupation, en prenant en considération le caractère hybride (continu par les variables physiques et discret par le mode de fonctionnement) des systèmes industriels. Pour comprendre la difficulté de l'application de méthodes classiques de sûreté de fonctionnement sur des systèmes hybrides et la complexité de ces derniers, des exemples représentatifs des préoccupations industrielles, ont été construits ([45]). Celui qui nous intéresse est

le cas-test du réservoir chauffé, issu de la littérature (D.L. Deoss et N. Siu en 1989 [27], et T. Aldemir en 1991 [2]). Le réservoir chauffé (Figure 2.1), dont la description détaillée est donnée dans la Section 2.1, est constitué de capteurs reliés à deux vannes d'alimentation et une vanne de vidange qui permettent, par leur ouverture ou leur fermeture, de réguler le niveau et la température d'un liquide entrant par les vannes. L'objectif est de calculer les probabilités de défaillance du réservoir, à savoir l'assèchement, le débordement et la surchauffe. Ce système, à caractère hybride de par ses variables physiques continues (température et hauteur du liquide) et discrète (position des vannes), a fait l'objet de nombreuses études par différentes méthodes de fiabilité telles que les réseaux de Petri ([32, 50], l'analyse sensitive ([47]), les automates stochastiques hybrides ([64]), l'algorithme des volumes finis ([57, 51]), la méthode CCMT ([55]) et CCCMT ([74]).

Nous allons étudier dans cette partie de la thèse un cas-test de la littérature qui est le réservoir chauffé. Dans le Chapitre 2, nous en donnerons une description détaillée, de son principe de fonctionnement et des interactions entre les variables continues et discrète. Nous donnerons alors dans le Chapitre 3 la première méthode numérique permettant de calculer les probabilités de défaillance du réservoir. Dans le Chapitre 4, nous exposerons une deuxième méthode numérique, basée sur le calcul de l'espérance d'une fonctionnelle d'un PDMP en utilisant les résultats introduits dans la Section 1.4, pour calculer la probabilité de surchauffe du réservoir, en étudiant en particulier les domaines de définition et leurs frontières des fonctions solutions de l'équation intégral-différentielle (1.4.2).

Chapitre 2

Le cas-test du réservoir

Le cas test, que nous avons choisi de modéliser par les PDMP, est un exemple représentatif de l'industrie gazière ou pétrolière. Le principe, fondé sur l'ouverture et/ou la fermeture de vannes commandées par des capteurs, est de pouvoir contrôler la hauteur d'un liquide dans un réservoir chauffé afin d'en empêcher le débordement, l'assèchement ou la surchauffe. Ces trois événements, que nous qualifierons d'*événements redoutés*, conduisent à la panne totale du système. Le but est de pouvoir déterminer, à partir de l'étude des variables physiques du problème (c'est-à-dire la hauteur du liquide et la température du réservoir) et des différents modes du système (c'est-à-dire la position des vannes qui sont susceptibles de tomber en panne), la probabilité de chaque événement redouté et ainsi d'appréhender le comportement d'un tel dispositif.

Dans un premier temps, nous décrirons le fonctionnement du réservoir et l'étude préalable effectuée pour comprendre les interactions entre les différentes variables en jeu. Puis nous donnerons les deux méthodes abordées pour calculer la probabilité de défaillance du réservoir.

2.1 Description du système

Dans cette section, nous allons décrire de façon exhaustive le fonctionnement du réservoir, en nous attachant à détailler les interactions entre les variables du réservoir. Cette étude va nous permettre de modéliser le comportement du réservoir par les PDMP.

Nous précisons que nous ne discuterons pas ici de la pertinence des paramètres et du réalisme de cet exemple. Nous nous contenterons de reprendre les données proposées par B. Tombuys et T. Aldemir dans [74], ainsi que par d'autres auteurs, afin de pouvoir comparer nos résultats numériques aux leurs.

2.1.1 Principe de fonctionnement

Le réservoir (Figure 2.1) contient un liquide dont le niveau est mesuré par trois capteurs reliés chacun à une unité : les unités 1 et 2 permettent d'ajouter du liquide dans la cuve, et l'unité 3 d'en retirer. Les trois unités sont mutuellement indépendantes et non réparables. Chaque unité a quatre états possibles : ouvert

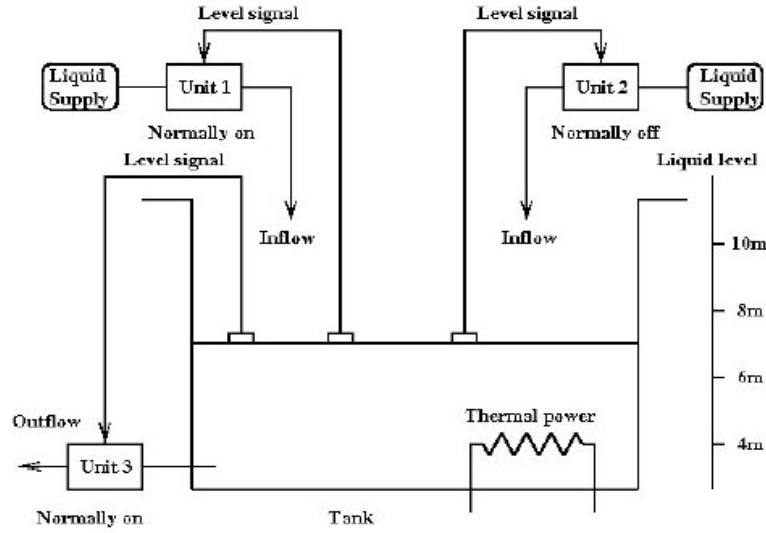


FIG. 2.1 – Schéma du réservoir thermique.

(O), fermé (F), ouvert bloqué (Ob) et fermé bloqué (Fb), les deux derniers états faisant suite à une panne de l'unité. Trois variables décrivent l'évolution du système au cours du temps : la hauteur h , la température θ et le mode $m = (a_1, a_2, a_3)$ du système, où $a_i \in \{O, Ob, F, Fb\}$ désigne l'état du composant $i = 1, 2, 3$.

On souhaite maintenir la hauteur du liquide entre 6 et 8 mètres. Pour cela, deux lois de commande vont agir sur les trois unités pour modifier le mode m et ainsi éviter deux des trois événements redoutés, c'est-à-dire l'assèchement ou le débordement de la cuve.

- La première loi de commande L^1 agit dès que le niveau du liquide est inférieur à 6 mètres. Les unités, si elles ne sont pas bloquées, se positionnent dans le mode $m = (O, O, F)$ afin de remplir le réservoir.
- La deuxième loi de commande L^2 intervient lorsque le niveau dépasse 8 mètres : les unités se placent dans le mode $m = (F, F, O)$ afin d'évacuer le surplus de liquide.

Ces deux lois de commande permettent au réservoir de maintenir la hauteur dans un intervalle acceptable en réagissant au dépassement de frontières (6m et 8m).

Cependant, en cas de panne de l'une des unités, et s'il y a franchissement de frontières, les unités bloquées n'obéissent plus aux lois de commande : il y a donc risque de débordement ou d'assèchement de la cuve. On considère que ces événements surviennent lorsque le liquide est-au-dessus des 10m ou en-dessous des 4m.

Le troisième événement redouté (la surchauffe du réservoir) survient lorsque la température de la cuve dépasse 100 ° C.

Ces trois événements conduisent à un arrêt définitif du système.

Remarque 2.1 Dans l'objectif d'une complexification du problème, mais que nous ne traiterons pas dans ce mémoire, il pourrait être envisagé, par exemple, de supposer

que les lois de commande L^1 ou L^2 qui agissent sur l'ouverture ou la fermeture d'une vanne s'appliquent avec une probabilité strictement inférieure à 1. La définition d'un PDMP, telle qu'elle est donnée dans le Chapitre 1, nous permettrait de prendre en compte ce genre de situations (voir Remarque 1.1).

2.1.2 Interactions entre les variables

L'évolution du système dépend exclusivement, comme nous l'avons dit précédemment, de la hauteur, de la température et du mode. Il existe trois types d'interactions entre ces variables.

1. Le saut d'un état non bloqué d'une unité i vers un état bloqué est régi par l'intensité de saut $2\lambda^i$, où λ^i est fonction de la température (Equation (2.1.1)). Ainsi, le passage d'un mode à un autre dans le cas d'une panne d'une unité dépend de l'évolution de la température.
2. La variation de la hauteur au cours du temps oblige le système à réagir pour se maintenir aux conditions d'équilibre. Les lois de commande L^1 et L^2 , par un changement de mode du système, permettent de modifier la trajectoire de la hauteur quand celle-ci dépasse les frontières de 6m ou 8m. Il y a donc interaction entre la hauteur et le mode.
3. La hauteur et la température sont des variables continues qui évoluent au cours du temps. Elles sont étroitement liées par le système différentiel qui les décrit (Equation (2.1.2)) : plus exactement, la hauteur est autonome, mais la température dépend de la hauteur.

2.1.3 Equations du système

Nous allons maintenant décrire en détail ces interactions.

2.1.3.1 Interaction température - mode

La Figure 2.2 représente les transitions entre les états, $2\lambda^i$ étant le taux de défaillance de l'unité $i = 1, 2, 3$. La température et λ^i sont liés par la relation suivante :

$$\lambda^i = a(\theta)\hat{\lambda}^i, \quad i = 1, 2, 3, \quad (2.1.1)$$

où la fonction a est donnée par :

$$a(\theta) = \frac{b_1 e^{b_c(\theta - \theta_{ref})} + b_2 e^{-b_d(\theta - \theta_{ref})}}{b_1 + b_2},$$

avec

b_c ($^{\circ}C^{-1}$)	b_d ($^{\circ}C^{-1}$)	b_1	b_2
0.05756	0.2301	3.0295	0.7578
$\hat{\lambda}^1$ (h^{-1})	$\hat{\lambda}^2$ (h^{-1})	$\hat{\lambda}^3$ (h^{-1})	θ_{ref} ($^{\circ}C$)
$2.2831.10^{-3}$	$2.8571.10^{-3}$	$1.5625.10^{-3}$	20

où les $\hat{\lambda}^i$, $i = 1, 2, 3$, sont les taux de transitions correspondant à la température de référence θ_{ref} . La fonction $a(\theta)$ atteint son minimum pour la température $\theta_{ref} = 20^\circ C$ (Figure 2.3). Cette fonction est une donnée du problème qui a été choisie arbitrairement de telle sorte que $a(0) \simeq 20$ et $a(100) \simeq 80$, autorisant ainsi une forte dépendance entre la température et les taux de panne.

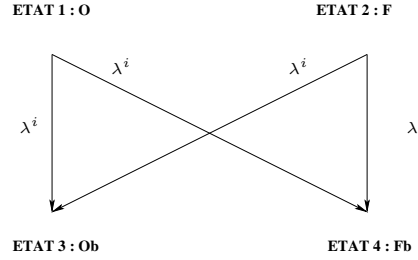


FIG. 2.2 – Diagramme des transitions.

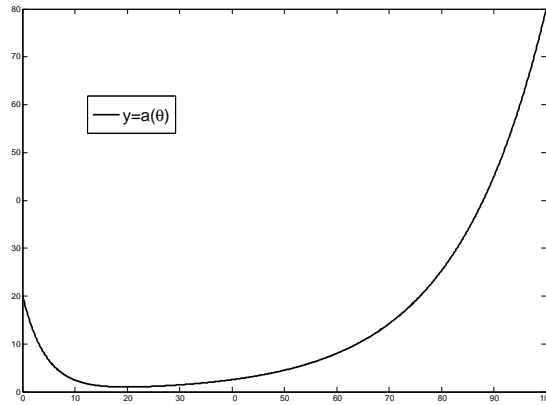


FIG. 2.3 – Fonction $a(\theta)$, avec la température θ en abscisse.

2.1.3.2 Interaction hauteur - mode

Les variations de la hauteur dépendent du mode à travers la position des unités indépendamment de leur caractère fonctionnel ou bloqué. En effet, une unité ouverte ou bloquée ouverte permet de la même façon au liquide de s'écouler. On ne s'intéresse donc pas à l'état (normal ou défaillant) de l'unité mais à la nature de son débit (écoulement ou non).

Définition 2.1 On note α le coefficient $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ défini pour tout $i = 1, 2, 3$ par

$$\alpha_i = \begin{cases} 0 & \text{si l'unité } i \text{ est fermée ou fermée bloquée} \\ 1 & \text{si l'unité } i \text{ est ouverte ou ouverte bloquée} \end{cases}$$

Les correspondances « Mode-Coefficient » sont données dans le Tableau 2.1.
La hauteur et le coefficient α sont reliés par l'équation différentielle

	mode m	coefficient α
m_1	(O, F, O)	$(1, 0, 1)$
m_2	(O, F, Ob)	$(1, 0, 1)$
m_3	(O, F, Fb)	$(1, 0, 0)$
m_4	(O, Ob, O)	$(1, 1, 1)$
m_5	(O, Ob, Ob)	$(1, 1, 1)$
m_6	(O, Ob, Fb)	$(1, 1, 0)$
m_7	(O, Fb, O)	$(1, 0, 1)$
m_8	(O, Fb, Ob)	$(1, 0, 1)$
m_9	(O, Fb, Fb)	$(1, 0, 0)$
m_{10}	(F, F, Fb)	$(0, 0, 0)$
m_{11}	(F, Ob, O)	$(0, 1, 1)$
m_{12}	(F, Ob, Ob)	$(0, 1, 1)$
m_{13}	(F, Ob, Fb)	$(0, 1, 0)$
m_{14}	(F, Fb, Fb)	$(0, 0, 0)$
m_{15}	(Ob, F, O)	$(1, 0, 1)$
m_{16}	(Ob, F, Ob)	$(1, 0, 1)$
m_{17}	(Ob, F, Fb)	$(1, 0, 0)$
m_{18}	(Ob, Ob, O)	$(1, 1, 1)$
m_{19}	(Ob, Ob, Ob)	$(1, 1, 1)$

	mode m	coefficient α
m_{20}	(Ob, Ob, Fb)	$(1, 1, 0)$
m_{21}	(Ob, Fb, O)	$(1, 0, 1)$
m_{22}	(Ob, Fb, Ob)	$(1, 0, 1)$
m_{23}	(Ob, Fb, Fb)	$(1, 0, 0)$
m_{24}	(Fb, O, F)	$(0, 1, 0)$
m_{25}	(Fb, O, Ob)	$(0, 1, 1)$
m_{26}	(Fb, O, Fb)	$(0, 1, 0)$
m_{27}	(Fb, F, O)	$(0, 0, 1)$
m_{28}	(Fb, F, Ob)	$(0, 0, 1)$
m_{29}	(Fb, F, Fb)	$(0, 0, 0)$
m_{30}	(Fb, Ob, O)	$(0, 1, 1)$
m_{31}	(Fb, Ob, F)	$(0, 1, 0)$
m_{32}	(Fb, Ob, Ob)	$(0, 1, 1)$
m_{33}	(Fb, Ob, Fb)	$(0, 1, 0)$
m_{34}	(Fb, Fb, O)	$(0, 0, 1)$
m_{35}	(Fb, Fb, F)	$(0, 0, 0)$
m_{36}	(Fb, Fb, Ob)	$(0, 0, 1)$
m_{37}	(Fb, Fb, Fb)	$(0, 0, 0)$

TAB. 2.1 – Tableau des correspondances mode-coefficient.

$$\frac{dh}{dt} = \gamma_1(\alpha),$$

où $\gamma_1(\alpha)$ est défini dans le Tableau 2.2.

2.1.3.3 Interaction hauteur - température

La température dépend de la hauteur et du coefficient α par l'équation différentielle

$$\frac{d\theta}{dt} = \frac{\gamma_2(\alpha) - \gamma_3(\alpha)\theta}{h},$$

où $\gamma_2(\alpha)$ et $\gamma_3(\alpha)$ sont définis dans le Tableau 2.2.

Ainsi, les lois de conservation de masse et d'énergie nous donnent le système d'équations différentielles suivant qui décrit le comportement global du réservoir :

$$(\mathcal{S}) \left\{ \begin{array}{l} \frac{dh}{dt} = \gamma_1(\alpha) \\ \frac{d\theta}{dt} = \frac{\gamma_2(\alpha) - \gamma_3(\alpha)\theta}{h} \end{array} \right. \quad (2.1.2)$$

sous les conditions initiales $h(0)$ et $\theta(0)$.

$\gamma_1(\alpha)$	$\gamma_2(\alpha)$	$\gamma_3(\alpha)$	q	θ_{in}	K
$(\alpha_1 + \alpha_2 - \alpha_3)q$	$(\alpha_1 + \alpha_2)q\theta_{in} + K$	$(\alpha_1 + \alpha_2)q$	1.5	15	23.88915

TAB. 2.2 – Constantes liées au système différentiel (\mathcal{S}) (avec $\gamma_1(\alpha)$, $\gamma_2(\alpha)$ et $\gamma_3(\alpha)$ en $m.h^{-1}$; q , en $m.h^{-1}$, une mesure du débit des unités; θ_{in} , en $^{\circ}C$, la température du liquide délivré par les unités 1 et 2; et K , en $m.^{\circ}C.h^{-1}$, un paramètre fixe lié aux variables physiques du réservoir.)

On pose également comme constantes :

$$\begin{aligned} - \theta_1 &= \frac{q\theta_{in} + K}{q} = 30.9261^{\circ}C \\ - \theta_2 &= \frac{2q\theta_{in} + K}{2q} = 22.96305^{\circ}C \end{aligned}$$

de sorte que nous avons $\frac{\gamma_2(\alpha)}{\gamma_3(\alpha)} = \begin{cases} \theta_1 & \text{si } \alpha_1 + \alpha_2 = 1, \\ \theta_2 & \text{si } \alpha_1 + \alpha_2 = 2. \end{cases}$

Remarque 2.2 Notons que nous avons la relation $\theta_{in} = \theta_1 - \frac{K}{q}$.

Au temps initial $t = 0$, nous avons les conditions initiales (\mathcal{S}_0) suivantes :

$$(\mathcal{S}_0) \begin{cases} h(0) = 7 \text{ m}, \\ \theta(0) = \theta_1 = 30.9261^{\circ}C, \\ m(0) = (O, F, O). \end{cases}$$

Lemme 2.1 Aux conditions initiales (\mathcal{S}_0) , le système est en équilibre.

Preuve : Au temps $t = 0$, et en utilisant la Remarque 2.2, le système différentiel (\mathcal{S}) s'écrit

$$(\mathcal{S}) \begin{cases} \frac{dh}{dt} = 0, \\ \frac{d\theta}{dt} = 0. \end{cases}$$

La hauteur et la température sont donc constantes et prennent respectivement comme valeur $h(0) = 7m$ et $\theta(0) = 30.9261^{\circ}C$. \square

Remarque 2.3 Comme la température initiale $\theta(0)$ est égale à $30.9261^{\circ}C$, la température entrante $\theta_{in} = 15^{\circ}C$ compense exactement le chauffage du liquide dans le réservoir.

Le but de notre étude est de calculer, à partir des données et du système d'équations différentielles dont on dispose, la probabilité d'assèchement avant t , $p_1(t)$, de débordement avant t , $p_2(t)$ et de surchauffe avant t , $p_3(t)$, en utilisant les PDMP.

2.2 Etude préliminaire

L'objectif de cette étude préliminaire dans le cas particulier du réservoir est d'utiliser les spécificités du système afin de comprendre et de déterminer avec précision le comportement de celui-ci. Dans un premier temps, nous nous intéressons aux modes atteints par le réservoir pour pouvoir décrire l'ensemble des scénarios réalisables. Nous allons pour cela étudier les modes du réservoir et montrer que nous pouvons nous ramener à un système de 37 modes et non de 64, regroupables selon les coefficients α . L'analyse et l'étude de méthodes générales pour la simplification de modes ne fait pas l'objet de ce travail de thèse. Dans un deuxième temps, nous étudions le système différentiel (\mathcal{S}).

2.2.1 Les modes atteints

Comme nous l'avons vu précédemment, chaque unité a quatre états possibles : O , F , Ob et Fb . Le réservoir a donc potentiellement 64 modes possibles. Nous avons cherché, avant de nous lancer dans la modélisation, à simplifier le système en éliminant, si possible, les modes non réalisables. Nous avons décliné par étape en partant de l'état initial $m(0) = (O, F, O)$, tous les modes découlant de $m(0)$, et nous avons ainsi pu nous ramener à un système de 37 états.

– 1ère étape

De $m(0)$, seule la panne d'une unité peut modifier l'état du système : le réservoir étant à l'équilibre, la hauteur du liquide est constante (les trois unités ayant même débit). Il n'y a donc pas d'augmentation ni de baisse du niveau, donc aucune raison pour que les lois de commande agissent.

Ainsi, de $m(0) = (O, F, O)$, on ne peut aller que vers un mode où l'une des trois unités est en panne, donc bloquée : on atteint les 6 modes suivants : (Ob, F, O) , (O, Fb, O) , (O, F, Ob) , (Fb, F, O) , (O, Ob, O) , (O, F, Fb) .

On élimine ainsi les 7 autres modes sans panne : (O, O, O) , (O, O, F) , (O, F, F) , (F, O, O) , (F, O, F) , (F, F, O) , (F, F, F) .

– 2ème étape

En raison de la configuration des trois modes (Fb, F, O) , (O, Ob, O) , (O, F, Fb) issus de $m(0)$, et qui peuvent conduire à un dépassement de frontières, on peut atteindre, par l'action des lois de commande, les 3 modes (Fb, O, F) , (F, Ob, O) et (F, F, Fb) .

On élimine les 15 modes issus directement et uniquement des 7 modes non atteints précédents, en excluant tous ceux qui sont issus de $m(0)$ en première ou deuxième étape.

– 3ème étape

Depuis les 9 modes issus de $m(0)$ en première ou deuxième étape, on en déduit les 19 modes suivants :

(Ob, F, Ob) , (Ob, F, Fb) , (Ob, Ob, O) , (Ob, Fb, O) , (O, Fb, Ob) , (O, Fb, Fb) ,
 (Fb, Fb, O) , (O, Ob, Ob) , (Fb, F, Ob) , (Fb, F, Fb) , (Fb, Ob, O) , (O, Ob, Fb) ,

(Fb, O, Ob) , (Fb, O, Fb) , (Fb, Ob, F) , (Fb, Fb, F) , (F, Ob, Ob) , (F, Ob, Fb) et (F, Fb, Fb) .

On écarte ensuite les modes issus directement et uniquement des modes éliminés lors de la deuxième étape et qui ne peuvent être atteints depuis les 19 modes précédents. On en élimine ainsi 5 :

(F, Fb, Ob) , (Ob, O, Ob) , (Ob, O, Fb) , (Ob, Ob, F) et (Ob, Fb, F) .

– 4ème étape

Des 19 modes précédents, on peut atteindre les 8 modes où toutes les unités sont en panne :

(Ob, Ob, Ob) , (Ob, Ob, Fb) , (Ob, Fb, Ob) , (Ob, Fb, Fb) , (Fb, Ob, Ob) , (Fb, Ob, Fb) , (Fb, Fb, Ob) et (Fb, Fb, Fb) .

– 5ème étape

Aux 8 modes précédents, on associe trois modes « cimetières » correspondant aux trois événements redoutés :

- 1 : Assèchement
- 2 : Débordement
- 3 : Surchauffe

Remarque 2.4 *Les trois modes cimetières permettent d'arrêter le processus dès qu'un événement redouté est atteint, empêchant ainsi le système de poursuivre son évolution.*

Les 37 modes qui sont atteints par le système sont résumés et numérotés dans le tableau en Annexe A.1 et le graphe des 37 modes et des transitions possibles entre eux est donné par la Figure 2.4.

Remarque 2.5 *L'étude qui suit prouvera le Graphe 2.4, et en particulier le fait que 4 modes « tout bloqué » sur 8 conduisent à l'événement 2 (débordement) : cet événement est donc prépondérant par rapport aux deux autres. Cela s'explique par le fait que le réservoir est composé de deux unités qui alimentent en liquide et d'une seule unité qui évacue du liquide, favorisant ainsi le débordement du réservoir.*

D'autre part, deux modes ($m_{22} = (Ob, Fb, Ob)$ et $m_{32} = (Fb, Ob, Ob)$), ne conduisent à aucun événement redouté. En effet, pour ces deux modes, la hauteur est constante et la température n'atteint jamais 100 °C (Tableau 2.5). Ils ne conduisent donc à aucun événement redouté.

Remarque 2.6 *Les huit configurations possibles du coefficient α : $(0, 1, 0)$, $(1, 0, 0)$, $(1, 1, 0)$, $(1, 1, 1)$, $(0, 0, 0)$, $(0, 1, 1)$, $(1, 0, 1)$ et $(0, 0, 1)$, sont toutes atteignables par l'un ou plusieurs des 37 modes (Tableau 2.3).*

Les trois variables, la hauteur h , la température θ et le mode m ou le coefficient α , vont donc nous permettre de caractériser l'état du système à chaque instant. Il est donc important d'étudier les variations de la température et de la hauteur pour pouvoir par la suite définir les différents changements d'états.

Nous allons étudier dans la partie suivante les solutions analytiques du système différentiel (\mathcal{S}).

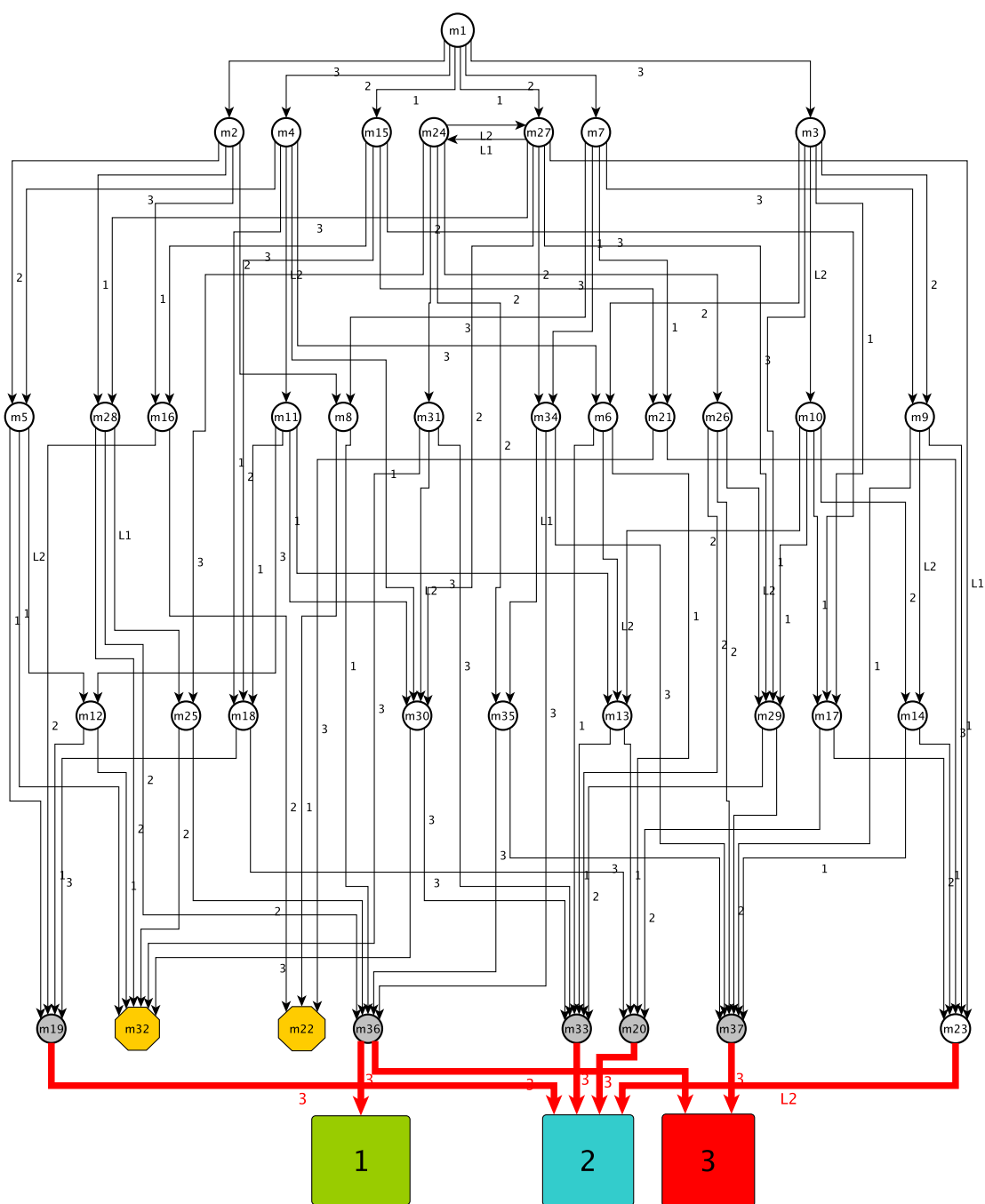


FIG. 2.4 – Graphe des 37 modes.

CAS	coefficient α	mode m
Cas 1.1	$(0, 1, 0)$	$m_{13}, m_{24}, m_{26}, m_{31}, m_{33}$
	$(1, 0, 0)$	m_3, m_9, m_{17}, m_{23}
Cas 1.2	$(1, 1, 0)$	m_6, m_{20}
Cas 1.3	$(1, 1, 1)$	m_4, m_5, m_{18}, m_{19}
Cas 2	$(0, 0, 0)$	$m_{10}, m_{14}, m_{29}, m_{35}, m_{37}$
Cas 3	$(0, 1, 1)$	$m_{11}, m_{12}, m_{25}, m_{30}, m_{32}$
	$(1, 0, 1)$	$m_1, m_2, m_7, m_8, m_{15}, m_{16}, m_{21}, m_{22}$
Cas 4	$(0, 0, 1)$	$m_{27}, m_{28}, m_{34}, m_{36}$

TAB. 2.3 – Tableau des correspondances coefficient-mode.

CAS	$\alpha = (\alpha_1, \alpha_2, \alpha_3)$	$\gamma_1(\alpha)$	$\gamma_2(\alpha)$	$\gamma_3(\alpha)$	$\frac{\gamma_2}{\gamma_3}$	$\frac{\gamma_3}{\gamma_1}$
1.1	$\alpha = (0, 1, 0)$ et $(1, 0, 0)$	q	$q\theta_{in} + K$	q	θ_1	1
1.2	$\alpha = (1, 1, 0)$	$2q$	$2q\theta_{in} + K$	$2q$	θ_2	1
1.3	$\alpha = (1, 1, 1)$	q	$2q\theta_{in} + K$	$2q$	θ_2	2
2	$\alpha = (0, 0, 0)$	0	K	0	n.d.	n.d.
3	$\alpha = (0, 1, 1)$ et $(1, 0, 1)$	0	$q\theta_{in} + K$	q	θ_1	n.d.
4	$\alpha = (0, 0, 1)$	$-q$	K	0	n.d.	0

TAB. 2.4 – Configurations en fonction du coefficient α .

2.2.2 Résolution du Système d'Equations Différentielles

Pour pouvoir comparer nos résultats numériques avec ceux proposés par B. Tombuys et T. Aldemir dans [74], nous considérons, comme nous l'avons fait depuis le début et contrairement à M. Marseguerra et E. Zio dans [56], que le débit q est identique pour les trois unités, c'est-à-dire $q = 1.5m.h^{-1}$. De plus, qu'une unité soit ouverte (resp. fermée) ou ouverte bloquée (resp. fermée bloquée), la variation du débit dans le réservoir est la même. Nous proposons donc une classification, non pas en fonction du mode m , mais selon les configurations du coefficient α , en fonction de la nature de la variation de la hauteur liée au régime :

- **CAS 1** : quatre états conduisent à une augmentation du niveau : $(0, 1, 0)$, $(1, 0, 0)$, $(1, 1, 0)$ et $(1, 1, 1)$
- **CAS 2** : un état est sans débit : $(0, 0, 0)$
- **CAS 3** : deux états donnent un débit constant : $(0, 1, 1)$ et $(1, 0, 1)$
- **CAS 4** : un état entraîne une diminution du niveau : $(0, 0, 1)$

En étudiant les coefficients $\gamma_1(\alpha)$, $\gamma_2(\alpha)$ et $\gamma_3(\alpha)$ des huit configurations, on remarque que l'on peut distinguer pour le CAS 1, trois sous-cas (Tableau 2.4), en fonction des coefficients $\gamma_1(\alpha)$, $\gamma_2(\alpha)$ et $\gamma_3(\alpha)$.

Le CAS 1.1 regroupe ainsi les configurations $(0, 1, 0)$ et $(1, 0, 0)$ pour lesquelles $\frac{\gamma_2}{\gamma_3} = \theta_1$ et $\frac{\gamma_3}{\gamma_1} = 1$; le CAS 1.2 correspond à la configuration $(1, 1, 0)$ pour la-

quelle $\frac{\gamma_2}{\gamma_3} = \theta_2$ et $\frac{\gamma_3}{\gamma_1} = 1$ et le CAS 1.3 à $(1, 1, 1)$ pour laquelle $\frac{\gamma_2}{\gamma_3} = \theta_2$ et $\frac{\gamma_3}{\gamma_1} = 2$. Les correspondances « Coefficient-Mode » sont données dans le Tableau 2.3.

Proposition 2.1 *Soit $(h(0), \theta(0), m(0))$ les conditions initiales du réservoir. Alors, l'ensemble des solutions de (\mathcal{S}) sont :*

1.1 Si $m(0)$ correspond au coefficient $\alpha = (0, 1, 0)$ ou $(1, 0, 0)$, alors $\forall t \in \mathbb{R}^+$

$$\begin{cases} h(t) = qt + h(0), \\ \theta(t) = \frac{h(0)(\theta(0) - \theta_1)}{h(t)} + \theta_1. \end{cases}$$

1.2 Si $m(0)$ correspond au coefficient $\alpha = (1, 1, 0)$, alors $\forall t \in \mathbb{R}^+$

$$\begin{cases} h(t) = 2qt + h(0), \\ \theta(t) = \frac{h(0)(\theta(0) - \theta_2)}{h(t)} + \theta_2. \end{cases}$$

1.3 Si $m(0)$ correspond au coefficient $\alpha = (1, 1, 1)$, alors $\forall t \in \mathbb{R}^+$

$$\begin{cases} h(t) = qt + h(0), \\ \theta(t) = \frac{h^2(0)(\theta(0) - \theta_2)}{h^2(t)} + \theta_2. \end{cases}$$

2. Si $m(0)$ correspond au coefficient $\alpha = (0, 0, 0)$, alors $\forall t \in \mathbb{R}^+$

$$\begin{cases} h(t) = h(0), \\ \theta(t) = \frac{K}{h(0)}t + \theta(0). \end{cases}$$

3. Si $m(0)$ correspond au coefficient $\alpha = (0, 1, 1)$ ou $(1, 0, 1)$, alors $\forall t \in \mathbb{R}^+$

$$\begin{cases} h(t) = h(0), \\ \theta(t) = (\theta(0) - \theta_1)e^{-\frac{qt}{h(0)}} + \theta_1. \end{cases}$$

4. Si $m(0)$ correspond au coefficient $\alpha = (0, 0, 1)$, alors $\forall t \in \mathbb{R}^+$

$$\begin{cases} h(t) = -qt + h(0), \\ \theta(t) = \theta(0) - \frac{K}{q} \ln \left(\frac{h(t)}{h(0)} \right). \end{cases}$$

Preuve : Comme l'équation de la température dépend de la hauteur et que l'équation de la hauteur est une équation linéaire de premier ordre, autonome car ne dépendant que de α , nous commençons par résoudre celle-ci, puis dans un deuxième temps celle de θ .

Nous avons directement la solution : $h(t) = \gamma_1(\alpha)t + h(0) \forall t \in \mathbb{R}^+$.

Nous allons maintenant résoudre la deuxième équation différentielle en fonction des différents cas.

CAS 1 On remarque que dans le cas 1, $\gamma_1(\alpha)$ et $\gamma_3(\alpha)$ sont non nuls. On pose $\beta = \frac{\gamma_3(\alpha)}{\gamma_1(\alpha)}$. On cherche la solution de $\frac{d\theta}{dt} = \frac{\gamma_2(\alpha) - \gamma_3(\alpha)\theta}{h}$.

Ecrivons cette équation sous une forme normalisée : $\forall t \in \mathbb{R}^+, \theta'(t) = A(t)\theta(t) + B(t)$ où

$$\begin{cases} A : t \mapsto -\frac{\gamma_3(\alpha)}{h(t)}, \\ B : t \mapsto \frac{\gamma_2(\alpha)}{h(t)}. \end{cases}$$

Comme $t \mapsto h(t)$ ne s'annule pas sur \mathbb{R}^+ , A et B sont des fonctions continues sur \mathbb{R}^+ . Il existe donc une solution de l'équation différentielle sur \mathbb{R}^+ . La solution générale est donnée par :

$$\forall t \in \mathbb{R}^+, \theta(t) = \theta(0)e^{\int_0^t A(x)dx} + \int_0^t e^{\int_x^t A(s)ds} B(x)dx.$$

Calculons le premier terme de cette expression. On a, pour tout $t \in \mathbb{R}^+$,

$$\int_0^t A(x)dx = \int_0^t -\frac{\gamma_3(\alpha)}{h(x)}dx = -\frac{\gamma_3(\alpha)}{\gamma_1(\alpha)} [\ln(h(x))]_0^t,$$

car $\gamma_1(\alpha)$ est non nul.

Avec $\beta = \frac{\gamma_3(\alpha)}{\gamma_1(\alpha)}$, on a donc pour le premier terme, $\forall t \in \mathbb{R}^+$:

$$\theta(0)e^{\int_0^t A(x)dx} = \theta(0)e^{-\beta \ln\left(\frac{h(t)}{h(0)}\right)} = \frac{\theta(0)h^\beta(0)}{h^\beta(t)}.$$

En reprenant le calcul précédent, nous avons pour le deuxième terme de l'expression

$\forall t \in \mathbb{R}^+$:

$$\begin{aligned} \int_0^t e^{\int_x^t A(s)ds} B(x)dx &= \int_0^t e^{-\beta \ln\left(\frac{h(t)}{h(x)}\right)} \times \frac{\gamma_2(\alpha)}{h(x)}dx \\ &= \frac{\gamma_2(\alpha)}{h^\beta(t)} \int_0^t h^{\beta-1}(x)dx. \end{aligned}$$

Or, comme $\beta - 1 = \frac{\alpha_3}{\alpha_1 + \alpha_2 - \alpha_3} \in \{0, 1\}$, on connaît la primitive de $x \mapsto h^{\beta-1}(x) = (\gamma_1(\alpha)x + h(0))^{\beta-1}$. Donc

$$\begin{aligned} \int_0^t e^{\int_x^t A(s)ds} B(x)dx &= \frac{\gamma_2(\alpha)}{h^\beta(t)} \left[\frac{1}{\beta\gamma_1(\alpha)} (\gamma_1(\alpha)x + h(0))^\beta \right]_0^t \\ &= \frac{\gamma_2(\alpha)}{\gamma_3(\alpha)} \left(1 - \frac{h^\beta(0)}{h^\beta(t)} \right). \end{aligned}$$

On a donc, dans le cas 1, $\forall t \in \mathbb{R}^+$:

$$\begin{aligned} \theta(t) &= \frac{\theta(0)h^\beta(0)}{h^\beta(t)} + \frac{\gamma_2(\alpha)}{\gamma_3(\alpha)} \left(1 - \frac{h^\beta(0)}{h^\beta(t)} \right) \\ &= \frac{h^\beta(0)}{h^\beta(t)} \left(\theta(0) - \frac{\gamma_2(\alpha)}{\gamma_3(\alpha)} \right) + \frac{\gamma_2(\alpha)}{\gamma_3(\alpha)}. \end{aligned}$$

En remplaçant β et $\frac{\gamma_2(\alpha)}{\gamma_3(\alpha)}$ par leur valeur correspondante aux trois sous-cas (Tableau 2.4), on a les solutions 1.1, 1.2 et 1.3 de la proposition.

CAS 2 On remarque que dans le cas 2, $\gamma_1(\alpha) = \gamma_3(\alpha) = 0$ et $\gamma_2(\alpha) = K$. Le système différentiel (\mathcal{S}) devient pour tout $t \in \mathbb{R}^+$:

$$\begin{cases} \frac{dh}{dt} = 0, \\ \frac{d\theta}{dt} = \frac{K}{h}. \end{cases}$$

Pour la première équation, on a :

$$\forall t \in \mathbb{R}^+, h(t) = h(0).$$

Pour l'équation de θ , nous avons assez facilement :

$$\forall t \in \mathbb{R}^+, \theta(t) = \theta(0) + \frac{K}{h(0)}t.$$

CAS 3 Dans le cas 3, $\gamma_1(\alpha) = 0$ et $\gamma_2(\alpha)$ et $\gamma_3(\alpha)$ sont non nuls. Le système (\mathcal{S}) devient, pour tout $t \in \mathbb{R}^+$:

$$\begin{cases} \frac{dh}{dt} = 0, \\ \frac{d\theta}{dt} = \frac{q(\theta_{in} - \theta) + K}{h}. \end{cases}$$

Comme dans le premier cas, $h(t) = h(0)$, pour $t \in \mathbb{R}^+$. Une solution générale de la seconde équation est donnée par :

$$\forall t \in \mathbb{R}^+, \theta(t) = \theta(0)e^{\int_0^t -\frac{q}{h(0)}dx} + \int_0^t e^{\int_x^t -\frac{q}{h(0)}ds} \times \frac{q(\theta_{in} - \theta) + K}{h(0)}dx.$$

On a donc, pour tout $t \in \mathbb{R}^+$,

$$\begin{aligned} \theta(t) &= \theta(0)e^{-\frac{qt}{h(0)}} + \frac{q(\theta_{in} - \theta) + K}{h(0)} \int_0^t e^{-\frac{q}{h(0)}(t-x)}dx \\ &= \theta(0)e^{-\frac{qt}{h(0)}} + \frac{q(\theta_{in} - \theta) + K}{h(0)} e^{-\frac{q}{h(0)}t} \left[\frac{h(0)}{q} e^{\frac{q}{h(0)}x} \right]_0^t. \end{aligned}$$

En utilisant la Remarque 2.2, on a donc, pour tout $t \in \mathbb{R}^+$,

$$\theta(t) = (\theta(0) - \theta_1)e^{-\frac{qt}{h(0)}} + \theta_1.$$

CAS 4 Dans le cas 4, $\gamma_1(\alpha)$ et $\gamma_2(\alpha)$ sont non nuls, et $\gamma_3(\alpha) = 0$. Le système différentiel (\mathcal{S}) devient alors, pour tout $t \in \mathbb{R}^+$:

$$\begin{cases} \frac{dh}{dt} = -q, \\ \frac{d\theta}{dt} = \frac{K}{h}. \end{cases}$$

Nous déduisons facilement l'expression de $h : \forall t \in \mathbb{R}^+, h(t) = -qt + h(0)$.
Ainsi, l'équation de la température devient, pour tout $t \in \mathbb{R}^+$

$$\frac{d\theta}{dt} = \frac{K}{-qt + h(0)} = -\frac{K}{q} \frac{h'(t)}{h(t)}.$$

Une solution générale de cette équation est donnée par :
 $\forall t \in \mathbb{R}^+,$

$$\begin{aligned}\theta(t) &= \theta(0) - \frac{K}{q} \int_0^t \frac{h'(x)}{h(x)} dx \\ &= \theta(0) - \frac{K}{q} \left[\ln(h(x)) \right]_0^t.\end{aligned}$$

On a ainsi, pour tout $t \in \mathbb{R}^+,$

$$\theta(t) = \theta(0) - \frac{K}{q} \ln\left(\frac{h(t)}{h(0)}\right),$$

d'où le résultat. □

Nous avons ainsi une solution analytique du système différentiel (\mathcal{S}) correspondant à chaque mode du réservoir. Nous pouvons déterminer à tout instant t la hauteur et la température dans le réservoir en fonction de la position des unités. Pour la méthode numérique présentée dans le Chapitre 3, la résolution du système (\mathcal{S}) est une étape importante dans le processus de modélisation du problème par les PDMP pour le cas spécifique du réservoir : ce travail préalable nous permet d'établir l'évolution déterministe du réservoir, et ainsi de ne pas avoir à discrétiser le flot du processus. En effet, la hauteur et la température représentent le flot ϕ du PDMP décrivant le comportement du réservoir (Section 3.1).

2.2.3 Etude qualitative du système

Par une étude assez simple des fonctions h et θ , on peut maintenant déterminer le comportement du système en fonction du cas dans lequel il se trouve. Par exemple, pour le CAS 1.1, on a pour tout $t \in \mathbb{R}^+, h(t) = qt + h(0)$. Donc h est croissante : la hauteur augmente linéairement au cours du temps. Elle peut donc dépasser les frontières.

Pour la température, on a pour tout $t \in \mathbb{R}^+, \theta(t) = \frac{h(0)(\theta(0) - \theta_1)}{h(t)} + \theta_1$. Alors si $\theta(0) \leq \theta_1$, θ est croissante, et si $\theta(0) \geq \theta_1$, θ est décroissante. De plus, θ tend vers θ_1 donc la température ne peut dépasser 100°C . Les CAS 1.2 et 1.3 sont assez semblables : la température tend vers θ_2 et la hauteur est croissante. Nous résumons dans le Tableau 2.5 le comportement de la hauteur et de la température pour chaque cas.

Remarque 2.7 *D'après la Remarque 2.3 concernant la température θ_{in} délivrée par les unités 1 et 2, la température dans les Cas 1.1, 1.2, 1.3 et 3 ne peut physiquement dépasser 100°C du fait de la position ouverte d'au moins une des unités d'alimentation.*

CAS	Variations de h	Variations de θ	Événement redouté potentiel
1.1	$h(0) \nearrow 10$	si $\theta(0) \leq \theta_1$ $\theta(0) \nearrow \theta_1$	Débordement
		si $\theta(0) \geq \theta_1$ $\theta(0) \searrow \theta_1$	
1.2 et 1.3	$h(0) \nearrow 10$	si $\theta(0) \leq \theta_2$ $\theta(0) \nearrow \theta_2$	Débordement
		si $\theta(0) \geq \theta_2$ $\theta(0) \searrow \theta_2$	
2	$h(t) = h(0)$	$\theta(0) \nearrow 100$	Surchauffe
3	$h(t) = h(0)$	si $\theta(0) \leq \theta_1$ $\theta(0) \nearrow \theta_1$	Aucun
		si $\theta(0) \geq \theta_1$ $\theta(0) \searrow \theta_1$	
4	$h(0) \searrow 4$	$\theta(0) \nearrow 100$	Assèchement et Surchauffe

TAB. 2.5 – Evolution de la hauteur et de la température en fonction du cas.

Cette étude qualitative nous a permis de construire le graphe des transitions (Figure 2.4). Cette étape est indispensable pour la compréhension du fonctionnement du réservoir, et pour la modélisation du problème par les PDMP. Nous allons détailler ce dernier point dans le chapitre suivant.

Chapitre 3

Caractéristiques du PDMP sous-jacent

Dans ce chapitre, une première méthode numérique est donnée pour calculer les probabilités de défaillance du réservoir. Nous préciserons tout d'abord les caractéristiques locales du PDMP dans le cas du réservoir, puis nous proposerons une méthode de simulation pour calculer la probabilité d'assèchement, de débordement ou de surchauffe du réservoir.

3.1 Description des caractéristiques locales du PDMP

Nous allons décrire ici les caractéristiques locales du PDMP associé au réservoir, c'est-à-dire l'espace d'état, le flot du processus, l'intensité des sauts et la mesure de transition du processus (voir Section 1.2).

3.1.1 L'espace d'état E

La hauteur, initialement à 7 m, doit évoluer dans l'intervalle $[6, 8]$. Au-delà de cet intervalle, le système réagit, grâce aux lois de commande. Il en résulte un changement d'état. On considère donc que les hauteurs de 6 m et 8 m sont des frontières de l'espace d'états. Lorsque la hauteur se situe en dessous de 6 m (resp. au-dessus de 8 m), il se peut qu'elle franchisse la hauteur critique de 4 m (resp. 10 m), ce qui constitue un événement redouté conduisant au dysfonctionnement du système et à un état cimetière pour le PDMP.

On prendra donc en compte les frontières des lois de commande L^1 et L^2 , ainsi que les frontières d'événements redoutés « 4 m » et « 10 m ».

La température, quant à elle, doit évoluer entre $0^\circ C$ et $100^\circ C$. Cependant, on montrera dans le chapitre suivant que la température ne peut descendre physiquement en dessous de $\theta_2 = 22.96305^\circ C$.

On peut donc poser comme ensemble de définition

$$E = \left([4, 6] \times [\theta_2, 100]\right) \cup \left([6, 8] \times [\theta_2, 100]\right) \cup \left([8, 10] \times [\theta_2, 100]\right),$$

et sa frontière

$$\begin{aligned}\partial E = & \left(\{4\} \times [\theta_2, 100] \right) \cup \left(\{6\} \times [\theta_2, 100] \right) \cup \left(\{8\} \times [\theta_2, 100] \right) \cup \left(\{10\} \times [\theta_2, 100] \right) \\ & \cup \left([4, 6] \times \{100\} \right) \cup \left([6, 8] \times \{100\} \right) \cup \left([8, 10] \times \{100\} \right).\end{aligned}$$

Remarque 3.1 *Il est possible de raffiner les domaines de définition en fonction du mode dans lequel on se trouve. Ce travail sera fait dans le Chapitre 4.*

3.1.2 Le flot ϕ_m

Le flot associé au mode m est défini pour tout $t \in \mathbb{R}$ par $\phi_m(h, \theta, t) = (h(t), \theta(t))$ où $h(t)$ et $\theta(t)$ sont les solutions de (\mathcal{S}) avec les conditions initiales $h(0) = h$, $\theta(0) = \theta$ et $m(0) = m$ (Proposition 2.1).

3.1.3 L'intensité λ_m

L'intensité de saut λ_m pour un mode m donné est la somme des taux de défaillance $2\lambda^i$ des unités. Par exemple, d'après le Tableau des transitions de l'Annexe A.1, pour le mode m_1 , on peut aller vers m_{15} et m_{27} avec l'intensité λ^1 , vers m_4 et m_5 avec l'intensité λ^2 , et vers m_2 et m_3 avec l'intensité λ^3 . Donc $\lambda_1(\theta) = 2(\lambda^1(\theta) + \lambda^2(\theta) + \lambda^3(\theta))$. Nous résumons les intensités de saut des 37 modes dans le Tableau 3.1 avec les notations suivantes :

$$\begin{aligned}\ell_1(\theta) &= 2\lambda^1(\theta) \\ \ell_2(\theta) &= 2\lambda^2(\theta) \\ \ell_3(\theta) &= 2\lambda^3(\theta) \\ \ell_{1,2}(\theta) &= 2(\lambda^1(\theta) + \lambda^2(\theta)) \\ \ell_{1,3}(\theta) &= 2(\lambda^1(\theta) + \lambda^3(\theta)) \\ \ell_{2,3}(\theta) &= 2(\lambda^2(\theta) + \lambda^3(\theta)) \\ \ell_{1,2,3}(\theta) &= 2(\lambda^1(\theta) + \lambda^2(\theta) + \lambda^3(\theta))\end{aligned}$$

3.1.4 La mesure de transition Q_m

La hauteur et la température sont des grandeurs continues. Le noyau de saut Q n'agit donc que sur le mode du processus :

$$\forall A \in E, Q((m, h, \theta); (m', A)) = \begin{cases} 0 & \text{si } \{(h, \theta) \notin A\} \text{ ou } \{(h, \theta) \in A \text{ et } m = m'\}, \\ p_{m,m'} & \text{sinon .} \end{cases}$$

Le noyau Q se résume donc aux probabilités de transition $p_{m,m'}$ d'un mode m à un autre mode m' . Ces probabilités sont données dans le Tableau 3.2.

mode m	intensité de saut λ_m
m_1	$\ell_{1,2,3}(\theta)$
m_2	$\ell_{1,2}(\theta)$
m_3	$\ell_{1,2}(\theta)$
m_4	$\ell_{1,3}(\theta)$
m_5	$\ell_1(\theta)$
m_6	$\ell_1(\theta)$
m_7	$\ell_{1,3}(\theta)$
m_8	$\ell_1(\theta)$
m_9	$\ell_1(\theta)$
m_{10}	$\ell_{1,2}(\theta)$
m_{11}	$\ell_{1,3}(\theta)$
m_{12}	$\ell_1(\theta)$
m_{13}	$\ell_1(\theta)$
m_{14}	$\ell_1(\theta)$
m_{15}	$\ell_{2,3}(\theta)$
m_{16}	$\ell_2(\theta)$
m_{17}	$\ell_2(\theta)$
m_{18}	$\ell_3(\theta)$
m_{19}	0

mode m	intensité de saut λ_m
m_{20}	0
m_{21}	$\ell_3(\theta)$
m_{22}	0
m_{23}	0
m_{24}	$\ell_{2,3}(\theta)$
m_{25}	$\ell_2(\theta)$
m_{26}	$\ell_2(\theta)$
m_{27}	$\ell_{2,3}(\theta)$
m_{28}	$\ell_2(\theta)$
m_{29}	$\ell_2(\theta)$
m_{30}	$\ell_3(\theta)$
m_{31}	$\ell_3(\theta)$
m_{32}	0
m_{33}	0
m_{34}	$\ell_3(\theta)$
m_{35}	$\ell_3(\theta)$
m_{36}	0
m_{37}	0

TAB. 3.1 – Intensité de saut λ_m en fonction du mode m .

3.1.5 Le temps d'atteinte de la frontière t_m^*

Le temps t_m^* (Section 1.2) est le plus petit temps pour lequel il y a atteinte de frontière. Il dépend du mode dans lequel on se trouve et de l'évolution des grandeurs physiques. Par exemple, si $h(0) > 8$ (c'est-à-dire que la loi de commande L^2 n'a pas été appliquée car au moins une des unités est en panne), et qu'on est dans le cas 1.1, la hauteur augmente et la température converge vers θ_1 (Tableau 2.5). La hauteur peut donc dépasser 10 m, mais la température ne peut dépasser 100 °C. Il y a ainsi franchissement de frontière quand $h(t) = 10$, donc $t_m^*(h, \theta) = \frac{10 - h}{q}$. Nous résumons tous les cas dans le Tableau 3.3.

Remarque 3.2 Si $h(0) < 6$, cela veut dire que la loi de commande L^1 (positionnant le mode en (O, O, F)) n'a pas fonctionné, donc il est impossible de se trouver dans le cas 1.2 ($\alpha = (1, 1, 0)$).

De même, si $h(0) > 8$, la loi de commande L^2 (positionnant le mode en (F, F, O)) n'a pas fonctionné, donc il est impossible de se trouver dans le cas 4 ($\alpha = (0, 0, 1)$).

Partant de la description et de la modélisation du problème du cas test, nous proposons une première méthode numérique pour déterminer les probabilités cumulées de défaillance du réservoir : la méthode de Monte Carlo appliquée dans le cadre des PDMP.

Mode m initial	Modes atteignables	Probabilités de transition en (h, θ)
m_1	m_{15}, m_{27} m_4, m_7 m_2, m_3	$\lambda^1(\theta)/\ell_{1,2,3}(\theta)$ $\lambda^2(\theta)/\ell_{1,2,3}(\theta)$ $\lambda^3(\theta)/\ell_{1,2,3}(\theta)$
m_2	m_{16}, m_{28} m_5, m_8	$\lambda^1(\theta)/\ell_{1,2}(\theta)$ $\lambda^2(\theta)/\ell_{1,2}(\theta)$
m_3	m_{10} m_{17}, m_{29} m_6, m_9	$\mathbb{1}_{\{h=8\}}$ $\lambda^1(\theta)/\ell_{1,2}(\theta) \mathbb{1}_{\{h \neq 8\}}$ $\lambda^2(\theta)/\ell_{1,2}(\theta) \mathbb{1}_{\{h \neq 8\}}$
m_4	m_{11} m_{18}, m_{30} m_5, m_6	$\mathbb{1}_{\{h=8\}}$ $\lambda^1(\theta)/\ell_{1,3}(\theta) \mathbb{1}_{\{h \neq 8\}}$ $\lambda^3(\theta)/\ell_{1,3}(\theta) \mathbb{1}_{\{h \neq 8\}}$
m_5	m_{12} m_{19}, m_{32}	$\mathbb{1}_{\{h=8\}}$ $1/2 \mathbb{1}_{\{h \neq 8\}}$
m_6	m_{13} m_{20}, m_{33}	$\mathbb{1}_{\{h=8\}}$ $1/2 \mathbb{1}_{\{h \neq 8\}}$
m_7	m_{21}, m_{34} m_8, m_9	$\lambda^1(\theta)/\ell_{1,3}(\theta)$ $\lambda^3(\theta)/\ell_{1,3}(\theta)$
m_8	m_{22}, m_{36}	$1/2$
m_9	m_{14} m_{23}, m_{37}	$\mathbb{1}_{\{h=8\}}$ $1/2 \mathbb{1}_{\{h \neq 8\}}$
m_{10}	m_{17}, m_{29} m_{13}, m_{14}	$\lambda^1(\theta)/\ell_{1,2}(\theta)$ $\lambda^2(\theta)/\ell_{1,2}(\theta)$
m_{11}	m_{18}, m_{30} m_{12}, m_{13}	$\lambda^1(\theta)/\ell_{1,3}(\theta)$ $\lambda^3(\theta)/\ell_{1,3}(\theta)$
m_{12}	m_{19}, m_{32}	$1/2$
m_{13}	m_{20}, m_{33}	$1/2$
m_{14}	m_{23}, m_{37}	$1/2$
m_{15}	m_{18}, m_{21} m_{16}, m_{17}	$\lambda^2(\theta)/\ell_{2,3}(\theta)$ $\lambda^3(\theta)/\ell_{2,3}(\theta)$
m_{16}	m_{19}, m_{22}	$1/2$
m_{17}	m_{20}, m_{23}	$1/2$
m_{18}	m_{19}, m_{20}	$1/2$
m_{19}	2	$\mathbb{1}_{\{h=10\}}$
m_{20}	2	$\mathbb{1}_{\{h=10\}}$
m_{21}	m_{22}, m_{23}	$1/2$
m_{22}	\times	0
m_{23}	2	$\mathbb{1}_{\{h=10\}}$
m_{24}	m_{27} m_{31}, m_{35} m_{25}, m_{26}	$\mathbb{1}_{\{h=8\}}$ $\lambda^2(\theta)/\ell_{2,3}(\theta) \mathbb{1}_{\{h \neq 8\}}$ $\lambda^3(\theta)/\ell_{2,3}(\theta) \mathbb{1}_{\{h \neq 8\}}$

Mode m initial	Modes atteignables	Probabilités de transition en (h, θ)
m_{25}	m_{32}, m_{36}	$1/2$
m_{26}	m_{29} m_{33}, m_{37}	$\mathbb{1}_{\{h=8\}}$ $1/2 \mathbb{1}_{\{h \neq 8\}}$
m_{27}	m_{24} m_{30}, m_{34} m_{28}, m_{29}	$\mathbb{1}_{\{h=6\}}$ $\lambda^2(\theta)/\ell_{2,3}(\theta) \mathbb{1}_{\{h \neq 6\}}$ $\lambda^3(\theta)/\ell_{2,3}(\theta) \mathbb{1}_{\{h \neq 6\}}$
m_{28}	m_{25} m_{32}, m_{36}	$\mathbb{1}_{\{h=6\}}$ $1/2 \mathbb{1}_{\{h \neq 6\}}$
m_{29}	m_{33}, m_{37}	$1/2$
m_{30}	m_{32}, m_{33}	$1/2$
m_{31}	m_{30} m_{32}, m_{33}	$\mathbb{1}_{\{h=8\}}$ $1/2 \mathbb{1}_{\{h \neq 8\}}$
m_{32}	\times	0
m_{33}	2	$\mathbb{1}_{\{h=10\}}$
m_{34}	m_{35} m_{36}, m_{37}	$\mathbb{1}_{\{h=6\}}$ $1/2 \mathbb{1}_{\{h \neq 6\}}$
m_{35}	m_{36}, m_{37}	$1/2$
m_{36}	1 3	$\mathbb{1}_{\{h=4\}}$ $\mathbb{1}_{\{\theta=100\}}$
m_{37}	3	$\mathbb{1}_{\{\theta=100\}}$

TAB. 3.2 – Noyau de saut Q en fonction du mode m .

CAS	$h(0)$	type de frontière	$t_m^*(x)$
1.1	si $h(0) < 6$	6m	$(6 - h(0))/q$
	si $6 < h(0) < 8$	8m	$(8 - h(0))/q$
	si $h(0) > 8$	10m	$(10 - h(0))/q$
1.2	si $6 < h(0) < 8$	8m	$(8 - h(0))/2q$
	si $h(0) > 8$	10m	$(10 - h(0))/2q$
1.3	si $h(0) < 6$	6m	$(6 - h(0))/q$
	si $6 < h(0) < 8$	8m	$(8 - h(0))/q$
	si $h(0) > 8$	10m	$(10 - h(0))/q$
2		100 °C	$h(0)(100 - \theta(0))/K$
3		pas de frontière	∞
4	si $h(0) < 6$	4m	$\min\left((h(0) - 4)/q, \left(1 - e^{\frac{(\theta(0)-100)q}{K}}\right) \frac{h(0)}{q}\right)$
	si $6 < h(0) < 8$	6m	$\min\left((h(0) - 6)/q, \left(1 - e^{\frac{(\theta(0)-100)q}{K}}\right) \frac{h(0)}{q}\right)$

TAB. 3.3 – Tableau des temps de franchissement de frontière $t_m^*(x)$.

3.2 Calcul des probabilités de défaillance

Nous voulons évaluer les probabilités de défaillance totale du réservoir à l'aide de la méthode de Monte Carlo basée sur des simulations de la trajectoire du PDMP associé au système. Pour simuler la trajectoire du PDMP, on utilise un algorithme générique adapté aux PDMP, présenté par H. Zhang et *al.* dans [78]. La simulation d'une trajectoire se décompose en plusieurs étapes que nous allons expliciter dans la partie suivante.

3.2.1 Principe de la simulation

Pour simuler $X(t)$, $0 \leq t \leq T_F$, c'est-à-dire la trajectoire du processus jusqu'à l'horizon T_F , nous procédons en plusieurs étapes.

Etape 1 : Calcul des temps de saut du processus Le premier temps de saut du processus est le minimum entre le temps de saut déterministe et le temps de saut aléatoire. Pour chaque unité i en position non bloquée, on calcule le temps de saut aléatoire τ_i qui a une intensité $2\lambda^i$.

Remarque 3.3 *La principale difficulté de la simulation se situe à cette étape. En effet, λ^i dépendant de la position du processus, on ne peut pas utiliser les algorithmes classiques de simulation d'une loi de Poisson d'intensité constante (voir par exemple [30] et [9]). Nous utilisons un résultat de C. Coccozza-Thivent dans [Proposition 1.21, p.17][14] que nous développerons dans la Section 3.2.2.*

Puis on calcule t_m^* le temps d'atteinte de la frontière de l'espace d'états. Pour cela, on utilise tout simplement les résultats décrits dans le Tableau 3.3. Alors

$$T_1 = \min_i \{\tau_i, t_m^*\}.$$

Etape 2 : Mise à jour du flot à l'instant T_n On actualise la valeur du flot (donc de la température et de la hauteur) au temps T_n pour obtenir la nouvelle trajectoire $\phi_{m_{n-1}}(x_{n-1}, T_n)$. Pour cela, on utilise la résolution analytique du système différentiel (\mathcal{S}) en fonction du mode dans lequel on se trouve.

Etape 3 : Sélection du nouveau mode après le saut Le noyau markovien nous permet de déterminer le mode après le saut (Tableau 3.2). Si le saut est déterministe, alors c'est qu'une loi de commande a été appliquée. En fonction du mode dans lequel on se trouve et de l'évolution de la hauteur, on peut alors sélectionner le nouveau mode en ouvrant et/ou en fermant les unités selon la loi de commande actionnée. Si $S_n = \tau_i$, c'est-à-dire l'unité i est tombée en panne, pour sélectionner le nouveau mode, on simule une loi de Bernoulli de paramètre $\frac{1}{2}$ pour déterminer si l'unité i devient *Ob* ou *Fb*.

On recommence ensuite à partir de l'étape 1 en posant

$$X(T_n) = (\phi_{m_{n-1}}(x_{n-1, T_n}), m_n)$$

comme nouvelles conditions initiales, car h et θ sont continues.

Nous avons ainsi $\{X(T_1), X(T_2), \dots, X(T_F)\}$ jusqu'au temps final fixé T_F . En simulant un grand nombre de trajectoires ($N = 10^3, 10^4, 10^5$ et 10^7), nous obtenons les probabilités de défaillance p_1, p_2 et p_3 par la méthode de Monte Carlo. Avant de donner les résultats obtenus par cette méthode, nous revenons dans la section suivante sur le calcul du temps de saut aléatoire du PDMP grâce à l'algorithme de C. Coccozza-Thivent, point central de notre démarche.

3.2.2 Simulation des temps de panne des unités

Pour calculer le temps de saut du processus, il faut :

1. Simuler le temps de panne de chaque unité non bloquée,
2. Calculer le temps de saut déterministe t_m^* ,
3. Prendre le minimum de ces deux temps pour obtenir le temps de saut du processus.

Comme nous l'avons noté dans la Remarque 3.3, le point difficile est la simulation des temps de saut aléatoire. Nous utilisons un résultat de C. Coccozza-Thivent ([Proposition 1.21, p.17][14]) qui propose un algorithme de simulation d'une variable aléatoire d'intensité variable λ donnée :

Soit λ une fonction positive définie sur \mathbb{R}^+ et bornée sur tout compact. L'algorithme suivant donne la construction d'une variable aléatoire τ d'intensité λ .

1. Poser $a = 0$, choisir $A > 0$.
2. Poser $a = a + A$ et $M \geq \sup_{a \leq x < a+A} \lambda(x)$.
3. Choisir un entier n suivant la loi de Poisson de paramètre MA .
4. Si $n = 0$, retourner en 2.
5. Si $n \neq 0$,
 - (i) indépendamment, répéter n fois la démarche suivante :
 - tirer un nombre aléatoire u_k suivant la loi uniforme sur $[a, a + A]$,
 - tirer un nombre aléatoire v_k suivant la loi uniforme sur $[0, M]$.
 - Si $v_k \leq \lambda(u_k)$, mémoriser la valeur de u_k .
 - (ii) S'il y a des valeurs de u_k mémorisées, alors $\tau = \min_k \{u_k, v_k \leq \lambda(u_k)\}$
 - (iii) Sinon, retourner en 2.

Cet algorithme est du type « algorithme de rejet » sur des intervalles de longueur A qui balayent la courbe de l'intensité λ selon la température : si le couple (u_k, v_k) se situe au-dessus de la courbe, on rejette la valeur de u_k , sinon on la mémorise. La plus petite valeur de u_k mémorisée est alors égale à la variable aléatoire τ .

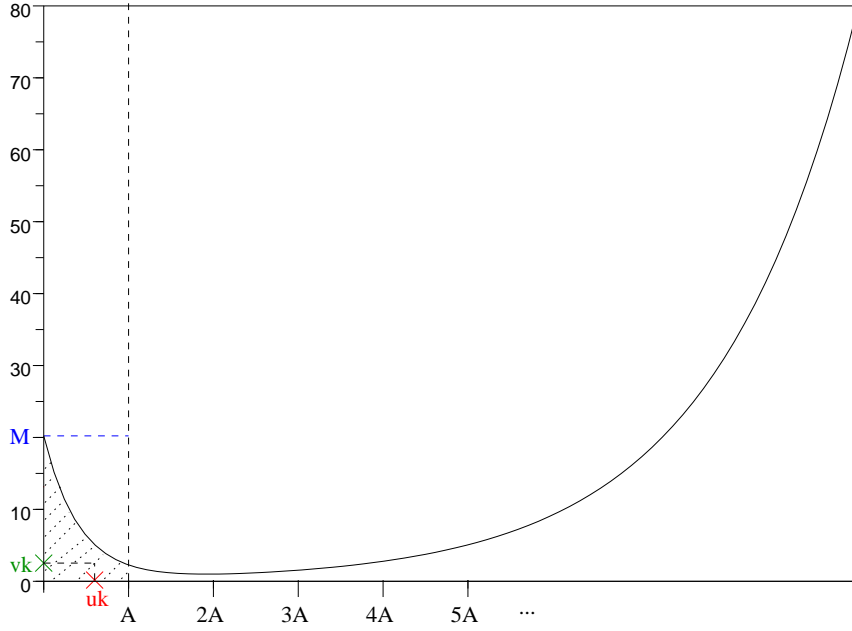


FIG. 3.1 – Algorithme de rejet (C. Coccozza-Thivent) sur la courbe d’une intensité variable λ .

m	$h(0)$	$\theta(0)$	τ_1	τ_2	τ_3	t^*	T
O, F, O	7	θ_1	282.9	72.8	19.1	$+\infty$	19.1
O, F, Fb	7	θ_1	$> T_F$	$> T_F$	\times	0.7	19.8
F, F, Fb	8	θ_1	5.9	19.9	\times	23.1	25.7
Ob, F, Fb	8	48.75	\times	$> T_F$	\times	1.3	27.0

TAB. 3.4 – Temps de saut d’une trajectoire calculés par l’algorithme de C. Coccozza-Thivent, avec m le mode, h_0 et θ_0 la hauteur et la température initiale correspondant au mode m , τ_1 , τ_2 , et τ_3 les temps de panne des unités 1, 2 et 3, t^* le temps de saut déterministe et T le temps de saut du processus.

Dans le cas du réservoir, on choisit arbitrairement $A = 1$. On détermine M comme le maximum entre $\lambda^i(a)$ et $\lambda^i(a + A)$ sur l’intervalle $[a, a + A]$. En effet, comme la fonction λ^i est strictement monotone (strictement décroissante pour $\theta \in [0, 15]$, strictement croissante pour $\theta \in]15, 100]$), la borne supérieure de λ^i sur l’intervalle $[a, a + A]$ est la plus grande valeur entre $\lambda^i(a)$ et $\lambda^i(a + A)$.

Exemple

- En partant du mode initial (O, F, O) et des conditions initiales $(7, \theta_1)$ (Figure 3.2), l’algorithme calcule, lors de la première itération (ligne 1 du Tableau 3.4), les temps de panne τ_1 , τ_2 et τ_3 des unités 1, 2 et 3, et le temps d’atteinte d’une frontière t^* grâce au Tableau 3.3. Le minimum des quatre temps, et donc le premier temps de saut T_1 du processus, est ici $\tau_3 = 19.1$ h, c’est-à-dire

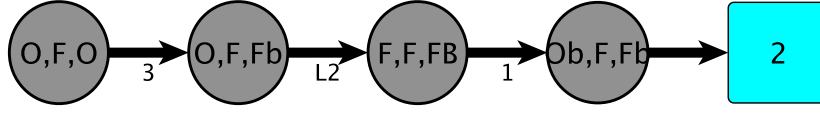


FIG. 3.2 – Graphe des transitions de l'exemple.

que l'unité 3 tombe en panne. L'algorithme sélectionne alors le nouvel état de l'unité 3 par une loi de Bernoulli de paramètre $\frac{1}{2}$ et met à jour la valeur du flot. Le nouveau mode sélectionné est, dans cet exemple, (O, F, Fb) avec comme valeur du flot $(7, \theta_1)$ (ligne 2 du Tableau 3.4).

- Lors de la deuxième itération, comme l'unité 3 est en panne, l'algorithme ne recalcule pas son temps de panne (croix dans le Tableau 3.4). De plus, les temps calculés τ_1 et τ_2 sont supérieurs au temps de mission fixé $T_F = 1000$ h. Le minimum est donc $t^* = 0.7$ h, c'est-à-dire qu'il y a franchissement d'une frontière (ici, il s'agit de la frontière des 8m). Le deuxième temps de saut T_2 du processus est alors égal à $T_1 + t^* = 19.8$ h. La loi de commande L^2 s'applique et l'algorithme sélectionne le nouveau mode et remet à jour le flot, etc.
- A la quatrième itération de l'algorithme, le mode sélectionné est (Ob, F, Fb) et le flot vaut $(8, 48.75)$. Le minimum des temps de saut est encore t^* : il y a donc franchissement d'une frontière. Comme le mode (Ob, F, Fb) correspond au cas 1.1, d'après le Tableau 2.5, l'événement redouté atteint est le débordement du réservoir ($h = 10$) pour un temps $T = 27.0$.

3.2.3 Résumé de la méthode numérique développée

Dans le cas du réservoir chauffé, nous avons cherché à exploiter au maximum les particularités du réservoir pour proposer une méthode numérique spécifique à celui-ci. Nous donnons ici une description de la méthodologie suivie. Les détails des différentes étapes sont développés dans les parties ou chapitres référencés.

1. Analyse des scénarios réalisables

Dans la mesure où il est possible de le faire pour le réservoir, nous cherchons à réduire le système initial de 64 modes. Pour cela, nous étudions l'ensemble des scénarios réalisables en partant du mode initial (O, F, O) . Ce point est développé dans la partie 2.2.1.

2. Résolution du système d'équations différentielles et étude qualitative

L'expression du flot donnée par le système (\mathcal{S}) peut être déterminée explicitement dans le cas du réservoir. Nous le résolvons donc analytiquement (partie 2.2.2). A partir des solutions, nous donnons alors le comportement du réservoir en fonction des cas 1.1 à 4 (partie 2.2.3).

3. Construction du graphe des transitions

A partir des étapes 1 et 2, nous pouvons construire le graphe des transitions

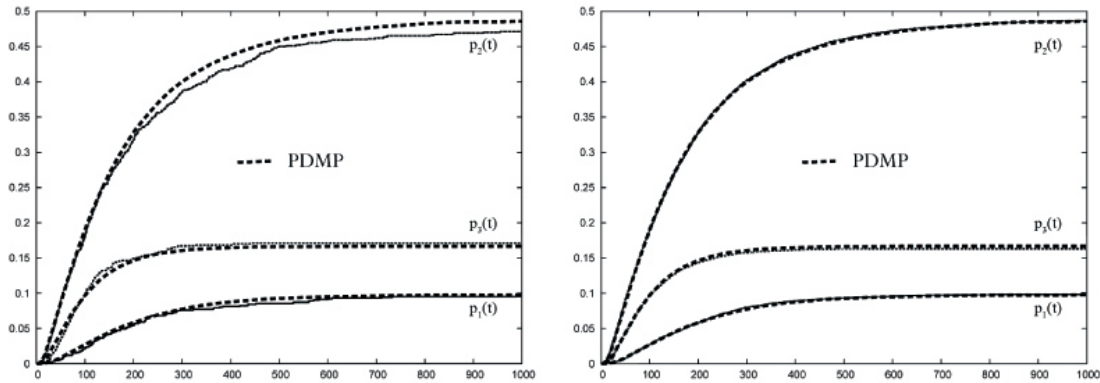


FIG. 3.3 – Probabilités de défaillance pour la méthode PDMP pour $N = 10^3$ et $N = 10^4$ comparés à la courbe de référence ($N = 10^7$), avec le temps (heure) en abscisse.

pour le réservoir (Figure 2.4).

4. Description des caractéristiques locales du PDMP sous-jacent

Nous déterminons l'espace d'état E , le flot ϕ_m et le temps d'atteinte d'une frontière t_m^* grâce à l'étape 2, la mesure de transition Q_m grâce à l'étape 3.

5. Simulation d'une trajectoire et Méthode de Monte Carlo

Grâce aux étapes précédentes, nous pouvons calculer les probabilités de défaillance, comme expliqué dans le début de la partie 3.2.

- (i) Calcul des temps de saut du processus en utilisant l'algorithme de C. Cocozza-Thivent pour calculer les temps de panne des unités.
- (ii) Mise à jour du flot.
- (iii) Sélection du nouveau mode après le saut.

A partir de ce simulateur de trajectoires, on utilise la méthode de Monte Carlo pour déterminer les probabilités d'événements redoutés.

3.2.4 Calcul des probabilités de défaillance par la méthode de Monte Carlo

Pour obtenir des résultats significatifs concernant les probabilités $p_1(t)$, $p_2(t)$ et $p_3(t)$, nous avons simulé un nombre important de trajectoires et nous avons compté le nombre de trajectoires conduisant à des modes cimetières. Les deux graphiques de la Figure 3.3 montrent les courbes obtenues pour $N = 10^3$ et 10^4 comparées à la courbe de référence $N = 10^7$ en pointillés et la Figure 3.4 est un exemple de trajectoire obtenue par notre algorithme. Le code correspondant se trouve en Annexe B.1.

L'observation de N histoires nous permet donc de dire qu'au cours du temps, $p_2(t) \geq p_3(t) \geq p_1(t)$. Il est donc fortement plus probable que le système s'arrête à

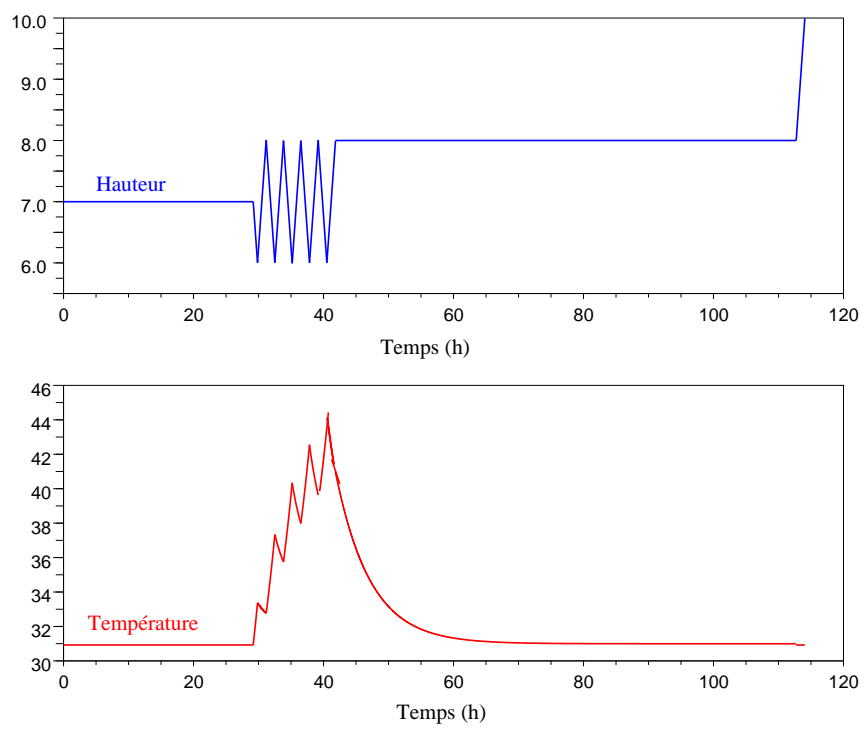


FIG. 3.4 – Exemple d’une trajectoire du PDMP jusqu’au 14-ème temps de saut, avec la hauteur en mètre sur le graphique du haut, et la température en degré sur le graphique du bas.

Temps (heures)	Assèchement	Débordement	Surchauffe
100	0.028463	0.205090	0.094659
200	0.059185	0.336688	0.143356
300	0.077217	0.405502	0.157475
400	0.086785	0.441659	0.161640
500	0.091610	0.461626	0.162823
600	0.093929	0.473156	0.163193
700	0.095065	0.479960	0.163312
800	0.095633	0.484020	0.163342
900	0.095908	0.486432	0.163363
1000	0.096032	0.487753	0.163368

TAB. 3.5 – Probabilités d’événements redoutés pour $N = 10^4$ histoires.

cause du débordement du réservoir. Très peu de scenarios (Tableau 3.5) conduisent à l’assèchement ($p_1(1000) = 0.096032$) et à la surchauffe ($p_3(1000) = 0.163368$). L’étude réalisée nous permet donc de prévoir le type d’événements susceptibles de se produire pour le système proposé ainsi que leur probabilité. Il serait ainsi intéressant d’utiliser ces résultats pour améliorer la fiabilité de ce système en tenant compte de la prépondérance de l’événement $\{h(t) > 10\}$.

Pour connaître l’efficacité de notre méthode de simulation, nous l’avons comparée avec les résultats d’une publication antérieure à nos travaux ([79] dans *Journal of Risk and Reliability*) : la méthode CCCMT (Continuous Cell-to-Cell Mapping Technique) présentée par B. Tombuys et T. Aldemir dans [74] ; et deux publications postérieures : la méthode ASH (Automates Stochastiques Hybrides) de G. Perez Castaneda dans [64] et la méthode des volumes finis pour les PDMP de W. Lair et *al.* dans [51].

Remarque 3.4 *Dans ce qui suit, les temps de simulation donnés pour le calcul des probabilités d’événements redoutés pour chacune des approches ne tiennent pas compte du travail préliminaire nécessaire à l’élaboration de toute méthode numérique. Nous ne donnons que les temps de compilation effective. Il ne s’agit pas ici bien sûr de faire un comparatif des temps de calcul de méthodes plus anciennes, n’ayant pas à disposition d’ordinateurs aussi puissants qu’aujourd’hui.*

Comparaison avec la méthode CCCMT La méthode CCCMT est la version continue de l’approche markovienne discrète CCMT (Cell-to-Cell Mapping Technique) proposée par T. Aldemir en 1987 ([1]). L’idée de la méthode CCMT telle qu’elle est exposée par T. Aldemir est de partitionner l’espace d’états des variables physiques en cellules et de considérer les probabilités de transitions du système dynamique entre ces cellules pendant un intervalle de temps $k\sigma \leq t \leq (k+1)\sigma$ avec $k = 0, 1, 2, \dots$. Le choix de σ s’avère difficile car il doit être assez petit pour éviter le franchissement de plus d’une frontière de la cellule dans un même intervalle de temps. Et un σ trop petit par rapport à la taille de la cellule empêcherait la dy-

Nb d'histoires N	Approche Monte Carlo Estimation de l'erreur relative sur $p_1(1000)$	PDMP CPU Time (2.2 GHz)	PDMP Estimation de l'erreur relative
10^3	4%	0.98s	1.75%
10^4	1.5%	10.03s	1.5%
10^5	0.4%	1mn37s	0.9%
10^6	0.15%	16mn37s	0.13%

TAB. 3.6 – Temps de simulation pour la méthode PDMP et erreur relative de l'approche Monte Carlo et de la méthode PDMP.

namique du système de sortir de la cellule pendant l'intervalle de temps considéré, faussant ainsi sa représentation. Une solution serait de réduire la taille des cellules, mais cela affecterait très fortement l'efficacité de la méthode. Une alternative est d'utiliser une extension continue de la méthode CCMT, la méthode CCCMT. Les équations décrivant l'évolution du système sont alors réduites à l'équation de Chapman-Kolmogorov ([28]), intégrée sur la cellule. Dans leur article, B. Tombuyses et T. Aldemir comparent la méthode CCCMT à une autre approche Monte Carlo, décrite dans la Section 4.2 de [74] et dont la différence avec notre démarche réside dans la méthode de simulation des instants de saut, et à la méthode CCMT dans le cas du réservoir chauffé. Nous nous intéressons plus particulièrement aux résultats donnés par la méthode CCCMT décrite précédemment, et l'approche Monte Carlo qui donne la solution de référence en utilisant notamment l'algorithme de Newton-Raphson pour calculer le temps de sortie t^* et un schéma adapté Runge-Kutta d'ordre 4 pour déterminer l'évolution du système entre des temps de transition aléatoires. La Figure 3.5 superpose les résultats des trois méthodes (PDMP, Monte Carlo et CCCMT).

L'avantage de notre méthode numérique dans le cas spécifique du réservoir chauffé est de donner une méthode de résolution exacte : en effet, contrairement à l'approche Monte Carlo qui utilise un schéma de Runge-Kutta d'ordre 4 pour déterminer l'évolution du système entre deux sauts, et la méthode CCCMT qui utilise une version intégrée de l'équation de Chapman-Kolmogorov sur les cellules de l'espace d'états partitionné, nous avons calculé une solution analytique du système pour chaque cas α . Nous obtenons des résultats cohérents avec ces méthodes (Figure 3.5) pour des erreurs relatives comparables à celle de l'approche Monte Carlo (Tableau 3.6) et CCCMT (Tableau 3.7) (à partir de $n_h = 3$ et $n_\theta = 16$, où n_h est le nombre de divisions de la hauteur et n_θ est le nombre de divisions pour la température pour la partition de l'espace d'états des variables physiques), et des temps de compilation raisonnables (Tableau 3.6) : 10.03s pour une erreur relative de 1.5% par exemple.

Comparaison avec la méthode ASH La méthode ASH, développée dans la thèse de G. Perez Castaneda [64], utilise les automates stochastiques hybrides (ASH) implémentés par la boîte à outils Scicos de Scilab. Un ASH est un automate composé

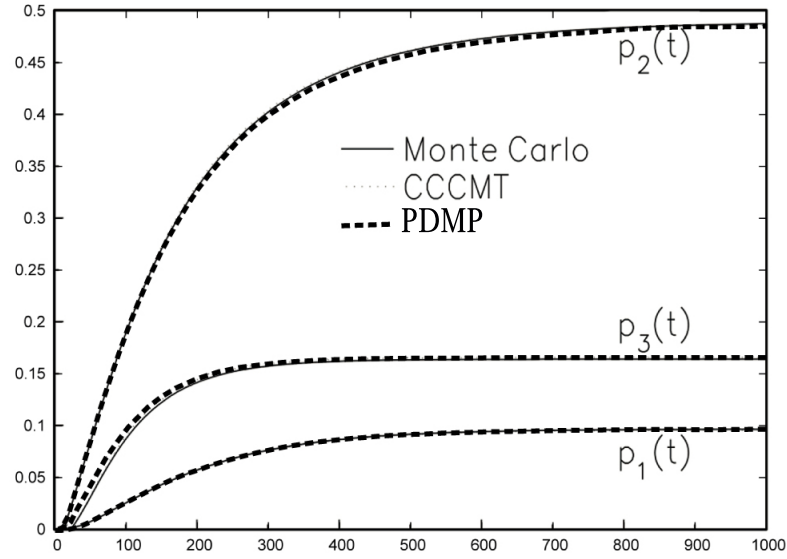


FIG. 3.5 – Probabilités de défaillance pour la méthode PDMP pour $N = 10^4$ comparé aux probabilités de défaillance calculées avec l’approche Monte Carlo (solution référence) et la méthode CCCMT (avec $n_h = 30$ et $n_\theta = 60$), avec le temps (en heure) en abscisse.

n_h	n_θ	Estimation de de l’erreur relative $p_1(1000)$
3	1	15%
6	1	14%
9	1	13%
3	4	8%
3	16	5%
9	16	2.5%
30	60	0.7%

TAB. 3.7 – Erreur relative pour la méthode CCCMT.

d'états discrets, chacun défini par un système d'équations continues et par des sauts de transition déterministes. Chaque état discret est un sous-ensemble de variables aléatoires et de sauts de transition aléatoires. G. Perez Castaneda donne la définition suivante d'un ASH dans [64] :

Un ASH est défini comme un 11-uplet

$$ASH = \left(\mathcal{X}, \mathcal{E}, \mathcal{A}, X, A, \mathcal{H}, \mathcal{F}, p, \mathcal{X}_0, x_0, p_0 \right) \quad (3.2.1)$$

dans lequel \mathcal{X} est un ensemble fini d'états discrets, \mathcal{E} un ensemble fini d'événements déterministes ou stochastiques, X un ensemble fini de variables réelles évoluant dans le temps, \mathcal{A} un ensemble fini d'arcs, $A : \mathcal{X} \times X \rightarrow (\mathbb{R}^{n^+} \rightarrow \mathbb{R})$ est une fonction des « activités », \mathcal{H} un ensemble fini d'horloges défini sur \mathbb{R} , $\mathcal{F} : \mathcal{H} \rightarrow (\mathbb{R} \rightarrow [0.1])$ une application qui associe à chaque horloge une fonction de répartition de probabilité, p une distribution de probabilité de transition, \mathcal{X}_0 l'état discret initial, x_0 la valeur initiale du vecteur d'état continu dans l'état initial discret et p_0 la distribution initiale de probabilité de transition dans l'état initial discret.

Dans le cas du réservoir chauffé, G. Perez Castaneda construit six automates à états finis élémentaires (trois pour les composants, un pour le réservoir, un pour la commande et un pour la détection de dépassement de température) dont la composition synchronisée par le logiciel DESUMA permet d'obtenir l'automate global. Celui-ci est composé de 1746 états regroupables en fonction de l'événement redouté atteint en 84 états. Il utilise ensuite la méthode de Monte Carlo pour quantifier les probabilités d'événements redoutés.

Nous avons vu qu'au lieu d'étudier un système à 64 modes, l'étude préalable et exhaustive du réservoir nous a permis de montrer que seuls 37 modes étaient atteignables à partir du mode initial (O, F, O) , réduisant ainsi la complexité du problème et le temps de simulation, car, par les ASH, le temps de compilation pour calculer les probabilités d'événements redoutés pour par exemple 10^4 histoires et un horizon de 1000h, est de l'ordre de 5h et seulement de 10.03s pour notre méthode.

De plus, les résultats sont sensiblement différents au niveau des probabilités d'assèchement et de surchauffe (Figure 3.6 issue de [64]). Ceci s'explique par le fait que G. Perez Castaneda considère que l'occurrence d'un événement redouté n'empêche pas l'atteinte d'un autre événement redouté : l'assèchement du réservoir n'exclut pas la surchauffe de celui-ci. En effet, d'après l'étude réalisée au Chapitre 2, on observe que l'augmentation de la température dans le réservoir est liée à la baisse de la hauteur du liquide.

Cependant, nous avons choisi d'introduire des modes cimetières correspondant aux événements redoutés afin de « tuer » le processus une fois l'assèchement, le débordement ou la surchauffe atteints : dans l'optique d'une application industrielle réelle et pour des préoccupations liées à la sûreté de fonctionnement (en particulier la sécurité), il nous a semblé plus réaliste d'arrêter l'évolution du réservoir lorsque celui-ci atteint un événement redouté.

De ce point de vue, nous pouvons comparer nos résultats avec ceux de W. Lair et *al.* dans [51] qui ont également pris en compte l'existence de ces trois modes cimetières. Leur méthode numérique, basée sur les PDMP, utilise l'algorithme de volumes finis.

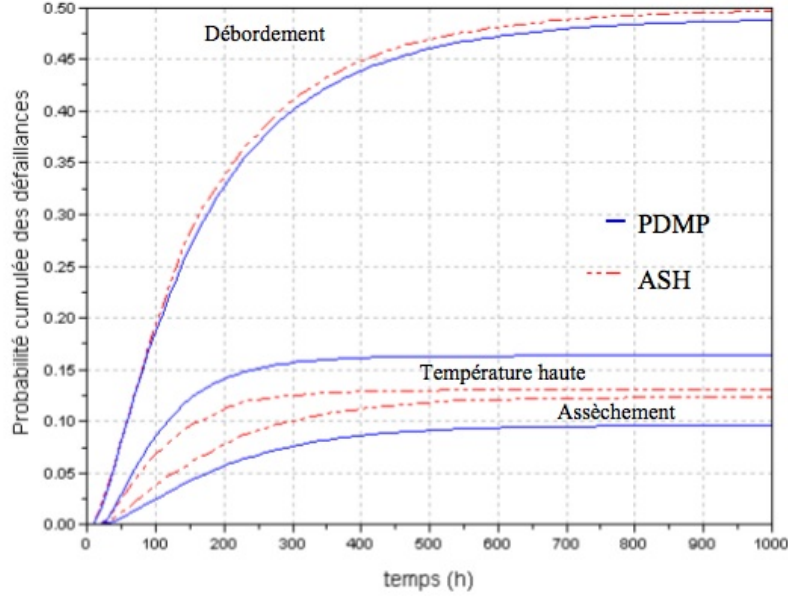


FIG. 3.6 – Probabilités de défaillance pour la méthode PDMP pour $N = 10^4$ comparé aux probabilités de défaillance calculées avec la méthode ASH, avec le temps (en heure) en abscisse.

Comparaison avec la méthode des volumes finis La méthode proposée dans [51] est basée sur la description des probabilités d'événements redoutés en fonction des distributions marginales du PDMP modélisant le réservoir. Elle utilise le fait que les lois marginales du PDMP sont l'unique solution des équations de Chapman-Kolmogorov, équations qui décrivent un bilan de flux de probabilités. Pour donner une estimation des lois marginales, l'algorithme utilisé est le schéma des volumes finis explicite. Il est basé sur la discrétisation du temps et des variables physiques. Une partition de l'espace des variables physiques est réalisée, d'une façon analogue à celle de la méthode CCMT. Le schéma des volumes finis calcule une approximation des lois marginales dont la densité est donnée par la mesure de Lebesgue, qui est constante pour chaque pas de temps et chaque cellule de la partition. Les résultats trouvés par W. Lair et *al.* sont comparables aux nôtres (Figure 3.7). Le Tableau 3.8 donne le temps de calcul et l'erreur relative en fonction de différents pas de discrétisation pour le schéma des volumes finis. Ainsi, nous voyons que pour des erreurs relatives comparables, les temps de calcul sont plus intéressants avec notre méthode numérique.

Conclusion Dans le cadre spécifique du réservoir, nous avons proposé une méthode de simulation exacte basée sur la résolution analytique des équations d'état du système (système différentiel (\mathcal{S})). Il est à noter que les méthodes CCCMT et ASH ne discutent pas de la convergence de leur algorithme. Sans chercher à obtenir un algorithme performant au niveau de la complexité, nous obtenons des résultats cohérents avec des temps de calcul intéressants et une précision comparable aux

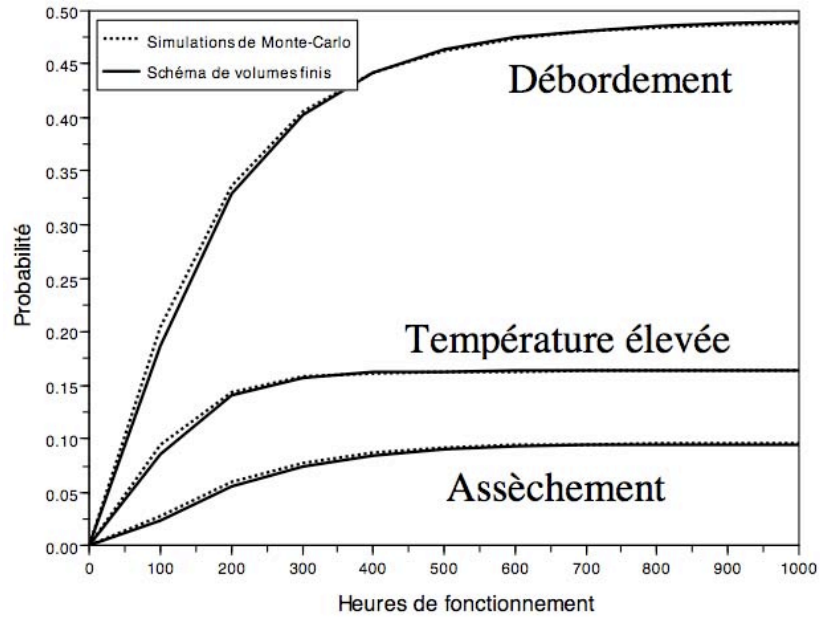


FIG. 3.7 – Probabilités de défaillance pour la méthode PDMP pour $N = 10^4$ comparé aux probabilités de défaillance calculées avec la méthode des volumes finis avec $\delta h = 0.5$, $\delta \theta = 2$, $\delta t = 0.03$.

Pas de discrétisation			Temps de calcul	Erreur relative
δh	$\delta \theta$	δt	1mn	4.23%
1	10	0.111		
δh	$\delta \theta$	δt	2mn48s	2.87%
1	4	0.53		
δh	$\delta \theta$	δt	9mn12s	1.21%
0.5	2	0.03		
δh	$\delta \theta$	δt	13mn14s	0.71%
0.25	2	0.31		
δh	$\delta \theta$	δt	49mn	0.56%
0.25	1	0.16		

TAB. 3.8 – Temps de calcul et erreur relative d'estimation de $p_1(1000)$ associés à différents pas de discrétisation pour le schéma de volumes finis.

autres méthodes. Les résultats obtenus paraissent prometteurs même s'ils mettent en évidence la difficulté de généraliser cette approche à des systèmes plus complexes et plus réalistes. Même si l'étude complète des modes réalisée dans la partie 2.2.1 n'est pas nécessaire (notre simulateur de trajectoires est capable de détecter les modes non atteignables (Annexe B.1)), nous avons besoin de connaître les solutions analytiques du système (\mathcal{S}) pour déterminer l'évolution déterministe du réservoir. Celles-ci ne sont généralement pas calculables pour des systèmes plus complexes. D'autre part, notre méthode numérique, ainsi que certaines des méthodes présentées, utilisent le fait que le cas-test étudié ici est un problème de petite dimension, régi par des lois de commande déterministes avec des composants non réparables et une dépendance entre les variables simples, mais qui apporte déjà son lot de difficultés. Si la modélisation par les PDMP permet de prendre en compte des lois de commande aléatoires (Remarque 2.1) ainsi que la réparabilité des unités (ajout d'arcs sur le graphe des transitions (Figure 2.4)), la dimension du système considéré reste un problème majeur en vue d'une application plus complexe.

Chapitre 4

Approche par les EDP

Dans ce chapitre, nous proposons une deuxième méthode numérique, basée sur le calcul de la fonctionnelle d'un PDMP. Après quelques rappels sur les résultats de M.H.A. Davis et leur application sur l'exemple du réservoir, nous explicitons les domaines de définition des fonctions solutions de l'équation intégral-différentielle. Nous donnons ensuite la démarche de notre résolution numérique et les résultats obtenus.

4.1 Propriétés des PDMP et application au réservoir

Nous allons appliquer les propriétés énoncées dans la Section 1.4 dans le cas du réservoir. En particulier, nous souhaitons calculer la probabilité de surchauffe de la cuve, en utilisant le Théorème 1.1. Dans ce cas, $l = 0$, $c(x, m) = \mathbb{1}_{\{\theta=100\}}(x)$ et $x = (h, \theta)$, car grâce à la présence des cimetières, le passage $\theta = 100$ correspond bien à un saut à la frontière. De plus, en utilisant la Proposition 1.1.2, nous avons

$$\frac{d}{dt}f_m(x) = \mathfrak{X}f_m(x) = g^1(x, m)\frac{\partial f_m(x)}{\partial h} + g^2(x, m)\frac{\partial f_m(x)}{\partial \theta}(x, m),$$

avec

$$\begin{aligned} g^1(x, m) &= \frac{dh}{dt} = \gamma_1(\alpha) \\ g^2(x, m) &= \frac{d\theta}{dt} = \frac{\gamma_2(\alpha) - \gamma_3(\alpha)\theta}{h}. \end{aligned}$$

L'équation intégral-différentielle (1.4.2) devient donc pour tout $(x, m) \in E$,

$$\gamma_1(\alpha)\frac{\partial}{\partial h}f_m(x) + \frac{\gamma_2(\alpha) - \gamma_3(\alpha)\theta}{h}\frac{\partial}{\partial \theta}f_m(x) + \lambda_m(x)(Q_m f_m(x) - f_m(x)) = 0 \quad (4.1.1)$$

et la condition au bord (1.4.3)

$$f_m(x) = c(x, m) + Q_m f_m(x), \quad (x, m) \in \delta E. \quad (4.1.2)$$

La présence du noyau markovien Q dans l'équation (4.1.1) rend plus compliquée qu'il n'y paraît la résolution de celle-ci. Il ne s'agit pas ici de résoudre l'équation (4.1.1)

mode par mode car les modes, et donc les équations, sont couplés par le noyau Q . Par exemple, en partant du mode initial m_1 , avec $x = (7, \theta_1)$, on peut atteindre m_2 , m_3 , m_4 , m_7 , m_{15} et m_{27} (Figure 4.1). L'équation satisfaite par f_1 devient alors

$$f_1(x) = \frac{1}{2(\lambda^1(x) + \lambda^2(x) + \lambda^3(x))} \left[\lambda^1(x)(f_{15}(x) + f_{27}(x)) + \lambda^2(x)(f_4(x) + f_7(x)) + \lambda^3(x)(f_2(x) + f_3(x)) \right].$$

Puis, par exemple, de m_{27} , on peut atteindre m_{24} , m_{28} , m_{29} , m_{30} et m_{34} (Figure 4.1). On obtient en fait un système de 37 équations couplées à résoudre.

Cependant, les modes pour lesquels il n'est pas possible d'atteindre $\theta = 100$ n'apportent pas de contribution. En effet, dans ce cas, la fonction solution de l'équation rattachée au mode est nulle. Par exemple, pour le mode $m_{19} = (Ob, Ob, Ob)$ (correspondant au cas 1.3), toutes les unités sont bloquées, sans possibilité pour la température d'atteindre $100^\circ C$ (Tableau 2.5). On trouve $f_{19} = 0$.

Il en est de même pour les fonctions f_{20} , f_{22} , f_{23} , f_{32} , f_{33} et f_{37} . Par répercussion, d'autres fonctions n'apportent pas de contribution. Par exemple, pour le mode m_{13} , on a l'équation correspondante pour tout $x \in E_{13}$:

$$q \frac{\partial}{\partial h} f_{13}(x) + \frac{q(\theta_1 - \theta)}{h} \frac{\partial}{\partial \theta} f_{13}(x) + \lambda^1(x)(f_{20}(x) + f_{33}(x)) - 2\lambda^1(x)f_{13}(x) = 0$$

et

$$f_{13}(x) = 0 \text{ au bord.}$$

Or, $f_{20} = f_{33} = 0$, donc on a $f_{13} = 0$. Ainsi, f_4 , f_5 , f_6 , f_{11} , f_{12} , f_{13} , f_{15} , f_{16} , f_{17} , f_{18} , f_{21} , f_{30} et f_{31} sont également nulles. On a donc un système de 17 équations couplées non triviales, dont la solution est donnée par

$$\{f_1, f_2, f_3, f_7, f_8, f_9, f_{10}, f_{14}, f_{24}, f_{25}, f_{26}, f_{27}, f_{28}, f_{29}, f_{34}, f_{35}, f_{36}\}.$$

La Figure 4.1 représente le graphe des transitions entre les 17 modes liés aux 17 fonctions précédentes. On prendra donc

$$\mathbb{M} = \{1, 2, 3, 7, 8, 9, 10, 14, 24, 25, 26, 27, 28, 29, 34, 35, 36\}.$$

Il nous faut maintenant décrire les domaines E_m et leurs frontières précises pour expliciter les conditions aux bords et donc les équations précises à résoudre mode par mode.

4.2 Les domaines

Pour déterminer les différents domaines de définition de chaque équation, nous avons utilisé les résultats de la Section 2.2.2, notamment l'étude de l'évolution de la hauteur et de la température en fonction des cas 1.1 à 4, en partant du mode initial m_1 . Dans une première partie, nous détaillons les domaines faciles à déterminer. Nous étudierons à part les domaines E_{24} et E_{27} , ainsi que leurs descendants, et le domaine E_{25} dans une dernière partie.

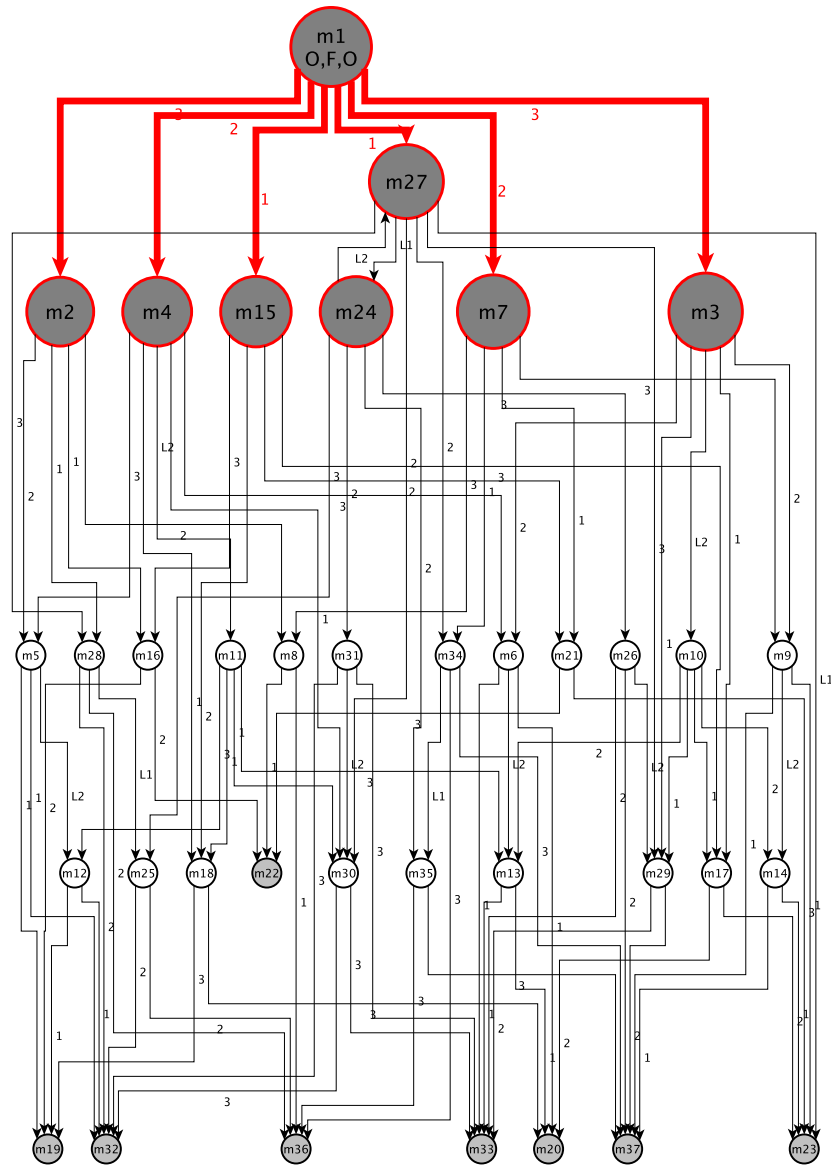


FIG. 4.1 – Graphe des transitions en partant de m_1 .

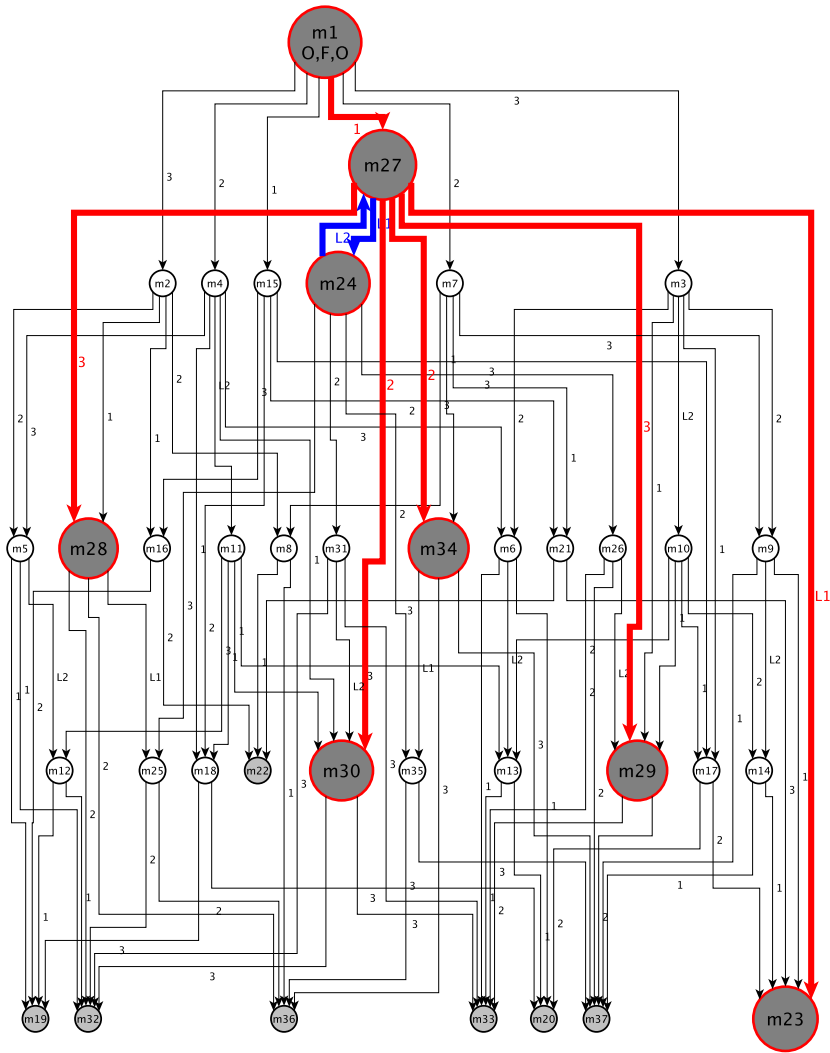


FIG. 4.2 – Graphe des transitions en partant de m_{27} .

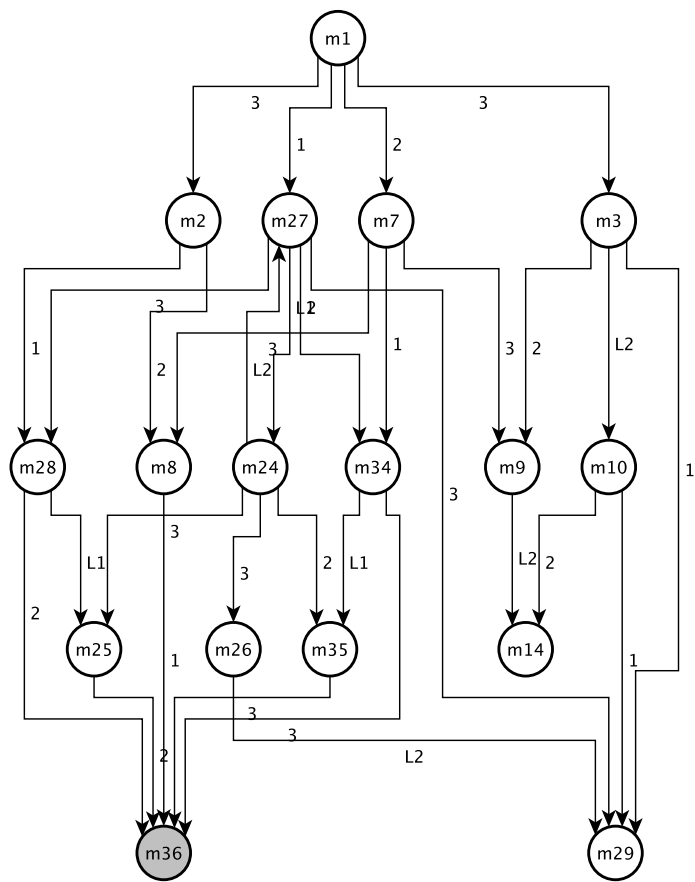


FIG. 4.3 – Graphe des 17 modes.

4.2.1 Les domaines « simples »

Les domaines suivants se déterminent à partir de l'étude de la hauteur et de la température (Tableau 2.5) en fonction du mode dans lequel on se trouve. Nous pouvons distinguer 5 groupes de domaines différents.

4.2.1.1 Domaines E_1 , E_2 , E_7 et E_8

Partons du mode initial $m_1 = (O, F, O)$. La hauteur est à 7 m et la température vaut θ_1 °C : le système est en équilibre, donc

$$E_1 = \{(7, \theta_1)\} \text{ et } \partial E_1 = \emptyset.$$

D'après la Figure 4.1 et le Tableau 2.5, comme les modes m_2 , m_7 et m_8 sont dans le même cas 3, et atteignables par aucun autre mode, on en déduit que

$$E_1 = E_2 = E_7 = E_8 = \{(7, \theta_1)\}.$$

4.2.1.2 Domaines E_3 et E_9

On passe du mode m_1 au mode m_3 par la panne de l'unité 3 en position Fb . On est donc dans le cas 1.1, avec comme conditions initiales $\{(7, \theta_1)\}$. D'après le Tableau 2.5, dans le cas 1.1, la hauteur augmente et la température vaut toujours θ_1 . Le domaine de définition est donc :

$$E_3 = [7, 8[\times \{\theta_1\}.$$

Remarque 4.1 *La hauteur ne peut atteindre 8 m car la loi de commande L^2 est appliquée.*

D'après la Figure 4.1, de m_3 , on accède uniquement à m_9 par la panne de l'unité 2. On est dans le même cas que m_3 : le domaine de définition pour m_9 est le même que celui de m_3 .

$$E_3 = E_9 = [7, 8[\times \{\theta_1\}.$$

Et la frontière du domaine est donnée par

$$\partial E_3 = \{(8, \theta_1)\}.$$

De la même façon que précédemment, en étudiant les variations de la hauteur et de la température, et les transitions entre les modes, on peut déterminer les domaines de définition suivants.

4.2.1.3 Domaines E_{10} et E_{14}

En remarquant que m_{10} est issu du mode m_3 par commande, que m_{14} est issu de m_{10} , et que ces deux modes sont sans débit (cas 2), on trouve

$$E_{10} = E_{14} = \{8\} \times [\theta_1, 100[$$

et

$$\partial E_{10} = \{(8, 100)\}.$$

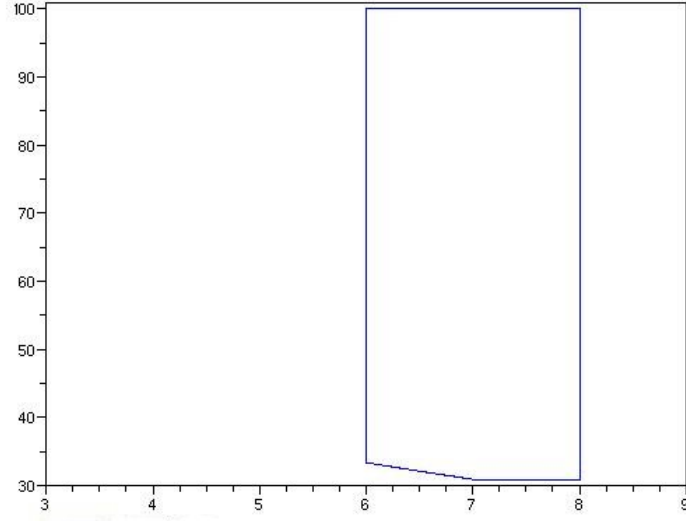


FIG. 4.4 – Domaine E_{29} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.

4.2.1.4 Domaines E_{29} et E_{35}

En remarquant que m_{29} est issu du mode m_{10} et que m_{35} vient du mode m_{34} (Remarque 4.5, Section 4.2.2), on a

$$E_{29} = E_{35} =]6, 8] \times [\theta_1, 100[$$

et

$$\partial E_{29} = [6, 8[\times \{100\}.$$

4.2.1.5 Domaine E_{36}

Comme m_{36} est issu du mode m_{35} , on en déduit

$$E_{36} =]4, 8] \times [\theta_1, 100[$$

et

$$\partial E_{36} = \{4\} \times [\theta_1, 100[\cup]4, 8[\times \{100\}.$$

Pour ces domaines simples, on peut alors écrire les équations correspondantes avec les conditions au bord associées pour chaque mode m (Tableau 4.1).

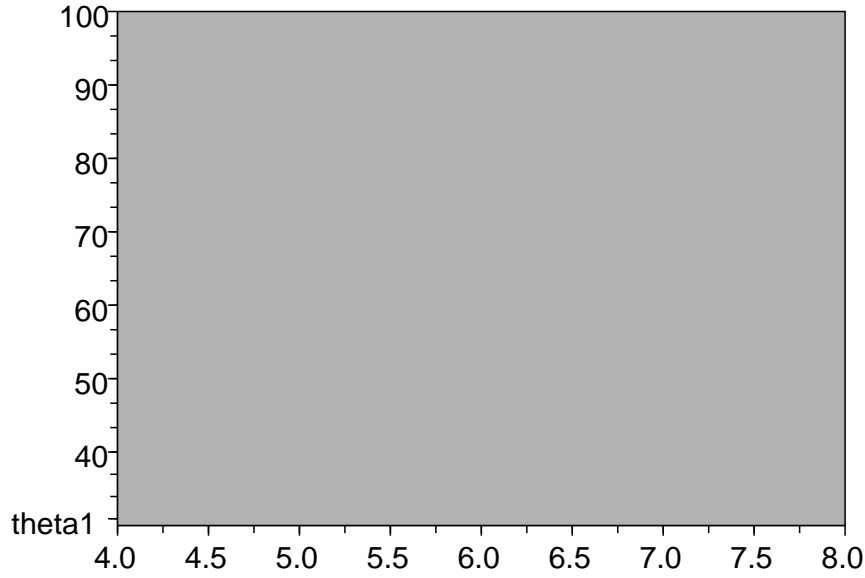


FIG. 4.5 – Domaine E_{36} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.

4.2.2 Les domaines E_{24} et E_{27}

Les modes m_{24} et m_{27} sont deux modes qui bouclent entre eux. Nous remarquons que nous pouvons définir ces courbes par récurrence. Pour cela, nous regardons le comportement de la trajectoire du processus qui part de m_1 , va en m_{27} , arrive ensuite en m_{24} , retourne en m_{27} , repart en m_{24} , ainsi de suite.

1. Nous partons des conditions initiales $(7, \theta_1, m_{27})$. D'après les équations de h et θ pour le cas 3, la hauteur diminue pour atteindre 6 m, et la température augmente jusqu'à $u_1 = \theta_1 - \frac{K}{q} \ln\left(\frac{6}{7}\right)$.
2. Nous sautons dans le mode m_{24} par la loi de commande L^1 avec pour conditions initiales $(6, u_1, m_{24})$. La hauteur augmente jusqu'à 8 m et la température diminue jusqu'à $v_1 = \frac{3}{4}(u_1 - \theta_1) + \theta_1$.
3. Nous retournons dans le mode m_{27} par la loi de commande L^2 avec pour conditions initiales $(8, v_1, m_{27})$. La hauteur diminue jusqu'à 6 m et la température augmente jusqu'à $u_2 = v_1 - \frac{K}{q} \ln\left(\frac{3}{4}\right)$.
4. etc

On peut ainsi définir les suites u_n et v_n :

$$\forall n \in \mathbb{N}^*, \begin{cases} u_{n+1} = v_n - \frac{K}{q} \ln\left(\frac{3}{4}\right) \\ u_1 = \theta_1 - \frac{K}{q} \ln\left(\frac{6}{7}\right) \end{cases} \text{ et } \forall n \in \mathbb{N}, v_{n+1} = \frac{3}{4}(u_{n+1} - \theta_1) + \theta_1.$$

Théorème 4.1 Pour tout $n \geq 2$, on a

MODE	CAS	DOMAINE	FRONTIERE
$m_1 = (O, F, O)$	3	E_1	∂E_1
$f_1(7, \theta_1) = \frac{1}{\ell_{1,2,3}(\theta_1)} [\lambda^1(\theta_1)f_{27}(7, \theta_1) + \lambda^2(\theta_1)f_7(7, \theta_1) + \lambda^3(\theta_1)(f_2(7, \theta_1) + f_3(7, \theta_1))]$			
$m_2 = (O, F, Ob)$	3	E_1	∂E_1
$f_2(7, \theta_1) = \frac{1}{\ell_{1,2}(\theta_1)} [\lambda^1(\theta_1)f_{28}(7, \theta_1) + \lambda^2(\theta_1)f_8(7, \theta_1)]$			
$m_3 = (O, F, Fb)$	1.1	E_3	∂E_3
$q \frac{\partial}{\partial h} f_3(h, \theta_1) + \lambda^1(\theta_1)f_{29}(h, \theta_1) + \lambda^2(\theta_1)f_9(h, \theta_1) - \ell_{1,2}(\theta_1)f_3(h, \theta_1) = 0$ $f_3(8, \theta_1) = f_{10}(8, \theta_1)$			
$m_7 = (O, Fb, O)$	3	E_1	∂E_1
$f_7(7, \theta_1) = \frac{1}{\ell_{1,3}(\theta_1)} [\lambda^1(\theta_1)f_{34}(7, \theta_1) + \lambda^3(\theta_1)(f_8(7, \theta_1) + f_9(7, \theta_1))]$			
$m_8 = (O, Fb, Ob)$	3	E_1	∂E_1
$f_8(7, \theta_1) = \frac{1}{2}f_{36}(7, \theta_1)$			
$m_9 = (O, Fb, Fb)$	1.1	E_3	∂E_3
$q \frac{\partial}{\partial h} f_9(h, \theta_1) + \lambda^1(\theta_1) - \ell_1(\theta_1)f_9(h, \theta_1) = 0$ $f_9(8, \theta_1) = f_{14}(8, \theta_1)$			
$m_{10} = (F, F, Fb)$	2	E_{10}	∂E_{10}
$\frac{K}{8} \frac{\partial}{\partial \theta} f_{10}(8, \theta) + \lambda^1(\theta)f_{29}(8, \theta) + \lambda^2(\theta)f_{14}(8, \theta) - \ell_{1,2}(\theta)f_{10}(8, \theta) = 0$ $f_{10}(8, 100) = 1$			
$m_{14} = (F, Fb, Fb)$	2	E_{10}	∂E_{10}
$\frac{K}{8} \frac{\partial}{\partial \theta} f_{14}(8, \theta) + \lambda^1(\theta) - \ell_1(\theta)f_{14}(8, \theta) = 0$ $f_{14}(8, 100) = 1$			
$m_{29} = (Fb, F, Fb)$	2	E_{29}	∂E_{29}
$\frac{K}{h} \frac{\partial}{\partial \theta} f_{29}(h, \theta) + \lambda^2(\theta) - \ell_2(\theta)f_{29}(h, \theta) = 0$ $f_{29}(h, 100) = 1$			
$m_{35} = (Fb, Fb, F)$	2	E_{29}	∂E_{29}
$\frac{K}{h} \frac{\partial}{\partial \theta} f_{35}(h, \theta) + \lambda^3(\theta)(f_{36}(h, \theta) + 1) - \ell_3(\theta)f_{35}(h, \theta) = 0$ $f_{35}(h, 100) = 1$			
$m_{36} = (Fb, Fb, Ob)$	4	E_{36}	∂E_{36}
$-q \frac{\partial}{\partial h} f_{36}(h, \theta) + \frac{K}{h} \frac{\partial}{\partial \theta} f_{36}(h, \theta) = 0$ $f_{36}(4, \theta) = 0$ avec $\theta \in [\theta_1, 100[$ $f_{36}(h, 100) = 1$ avec $h \in [4, 8]$			

TAB. 4.1 – Equations pour les domaines simples.

$$u_n = C \left(\frac{3}{4} \right)^{n-1} \frac{K}{q} + \frac{4K}{q} \ln \left(\frac{4}{3} \right) + \theta_1,$$

et pour tout $n \geq 1$,

$$v_n = C \left(\frac{3}{4} \right)^n \frac{K}{q} + \frac{3K}{q} \ln \left(\frac{4}{3} \right) + \theta_1,$$

où $C = \ln \left(\frac{7}{6} \right) + 4 \ln \left(\frac{3}{4} \right)$. De plus, si on note u_∞ la limite de u_n quand n tend vers l'infini et v_∞ la limite de v_n quand n tend vers l'infini, alors

$$u_\infty = \frac{4K}{q} \ln \left(\frac{4}{3} \right) + \theta_1 \text{ et } v_\infty = \frac{3K}{q} \ln \left(\frac{4}{3} \right) + \theta_1.$$

Remarque 4.2 On a $u_\infty = 49.252714^\circ\text{C}$ et $v_\infty = 44.6710^\circ\text{C}$. Ces deux valeurs sont inférieures à 100°C . La frontière $\theta = 100$ n'est donc pas une frontière de E_{24} et E_{27} .

Preuve : Nous allons exprimer u_n et v_n en fonction de n . Ecrivons tout d'abord u_{n+1} en fonction de u_n . Pour tout $n \in \mathbb{N}^*$,

$$u_{n+1} = \frac{3}{4}u_n + \frac{1}{4}\theta_1 + \frac{K}{q} \ln \left(\frac{4}{3} \right).$$

Nous allons utiliser la méthode du point fixe. Notons B la partie constante de u_{n+1} :

$$B = \frac{1}{4}\theta_1 + \frac{K}{q} \ln \left(\frac{4}{3} \right).$$

Soit $r \in \mathbb{R}^*$ qui vérifie $r = \frac{3}{4}r + B$. Alors $r = 4B = \theta_1 + \frac{4K}{q} \ln \left(\frac{4}{3} \right)$. On pose alors pour tout $n \in \mathbb{N}^*$,

$$\bar{u}_n = u_n - r.$$

$(\bar{u}_n)_{n \in \mathbb{N}^*}$ est une suite géométrique dont la raison est $\frac{3}{4}$. On peut donc écrire, pour tout $n \in \mathbb{N}^*$

$$\bar{u}_n = \left(\frac{3}{4} \right)^{n-1} \bar{u}_1 \text{ où } \bar{u}_1 = u_1 - r.$$

D'où, en utilisant le fait que $u_n = \bar{u}_n + r$, on a pour tout $n \geq 2$

$$u_n = C \left(\frac{3}{4} \right)^{n-1} \frac{K}{q} + \frac{4K}{q} \ln \left(\frac{4}{3} \right) + \theta_1, \text{ où } C = \ln \left(\frac{7}{6} \right) + 4 \ln \left(\frac{3}{4} \right).$$

Nous en déduisons alors l'écriture de v_n pour tout $n \geq 1$:

$$v_n = C \left(\frac{3}{4} \right)^n \frac{K}{q} + \frac{3K}{q} \ln \left(\frac{4}{3} \right) + \theta_1, \text{ où } C = \ln \left(\frac{7}{6} \right) + 4 \ln \left(\frac{3}{4} \right).$$

En prenant la limite de ces deux expressions, on a bien le résultat voulu. \square

Ainsi, nous pouvons définir les domaines E_{24} et E_{27} (Figure 4.6, avec E_{24} en pointillés et E_{27} en gras) et leurs frontières respectives ∂E_{24} et ∂E_{27} :

$$E_{24} = \bigcup_{n=1}^{\infty} \left\{ h \in [6, 8[, \theta = \frac{6}{h}(u_n - \theta_1) + \theta_1 \right\},$$

$$\partial E_{24} = \bigcup_{n=1}^{\infty} \{(8, v_n)\}.$$

et

$$E_{27} = \left\{ h \in]6, 7], \theta = \theta_1 - \frac{K}{q} \ln \left(\frac{h}{7} \right) \right\} \cup \bigcup_{n=1}^{\infty} \left\{ h \in]6, 8], \theta = v_n - \frac{K}{q} \ln \left(\frac{h}{8} \right) \right\},$$

$$\partial E_{27} = \bigcup_{n=1}^{\infty} \{(6, u_n)\}.$$

Remarque 4.3 *Pour toute itération n , chaque morceau de courbe $[v_n, u_{n+1}]$ représente le domaine de définition E_{27} lorsque le système arrive en m_{27} à l'itération n . De même, pour tout n , chaque morceau de courbe $[u_n, v_n]$ représente le domaine de définition E_{24} lorsque le système arrive en m_{24} à l'itération n . Les Figures 4.2.2 et 4.2.2 représentent donc l'ensemble exact des domaines de définition pour le mode m_{24} et m_{27} . Par abus de langage, nous parlons des domaines de définition E_{24} et E_{27} au lieu d'ensemble des domaines de définition E_{24} et E_{27} .*

Et on a les équations de f_{24} et de f_{27} :

$$q \frac{\partial}{\partial dh} f_{24}(x) + \frac{q(\theta_1 - \theta)}{h} \frac{\partial}{\partial d\theta} f_{24}(x) + \lambda^2(x) f_{35}(x) + \lambda^3(x) (f_{25}(x) + f_{26}(x)) - 2(\lambda^2(x) + \lambda^3(x)) f_{24}(x) = 0, x \in E_{24}$$

avec la condition aux bords

$$f_{24}(8, \theta) = f_{27}(8, v_n),$$

et

$$-q \frac{\partial}{\partial dh} f_{27}(x) + \frac{K}{h} \frac{\partial}{\partial d\theta} f_{27}(x) + \lambda^2(x) f_{34}(x) + \lambda^3(x) (f_{28}(x) + f_{29}(x)) - 2(\lambda^2(x) + \lambda^3(x)) f_{27}(x) = 0, x \in E_{27}$$

avec la condition au bord

$$f_{27}(6, \theta) = f_{27}(6, u_n).$$

Comme m_{26} a pour antécédent m_{24} et est dans le même cas 1.1, on a $E_{24} = E_{26}$ et $\partial E_{24} = \partial E_{26}$. De même, comme m_{28} et m_{34} ont pour antécédent m_{27} et sont dans le même cas 4, on a $E_{27} = E_{28} = E_{34}$ et $\partial E_{27} = \partial E_{28} = \partial E_{34}$.

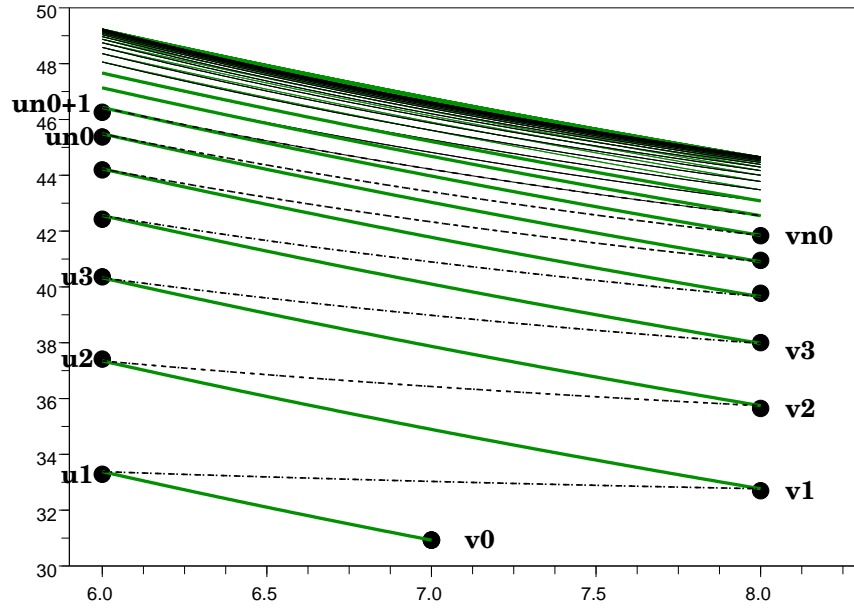


FIG. 4.6 – Domaines E_{24} et E_{27} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.

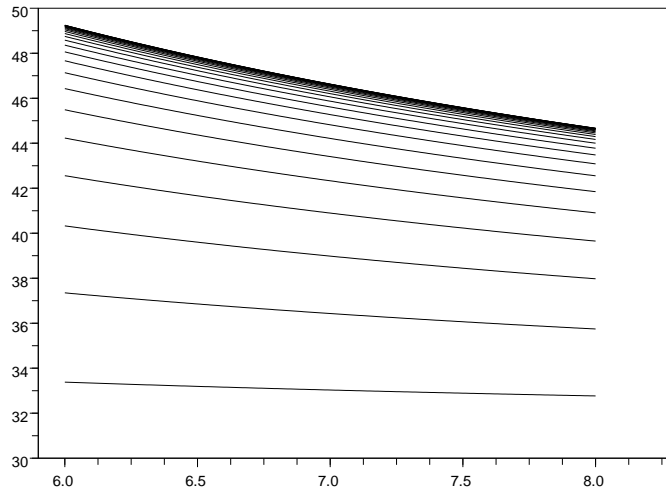


FIG. 4.7 – Domaine E_{24} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.

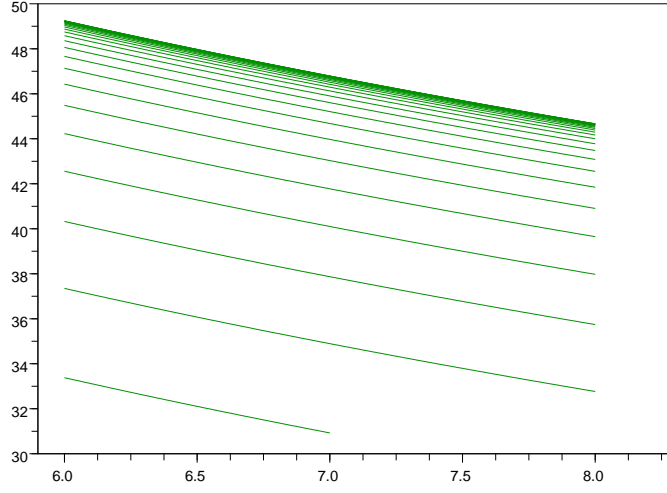


FIG. 4.8 – Domaine E_{27} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.

MODE	CAS	DOMAINE	FRONTIERE
$m_{24} = (Fb, O, F)$	1.1	E_{24}	∂E_{24}
$q \frac{\partial}{\partial h} f_{24}(h, \theta) + \frac{q(\theta_1 - \theta)}{h} \frac{\partial}{\partial \theta} f_{24}(h, \theta) + \lambda^2(\theta) f_{35}(h, \theta) + \lambda^3(\theta) (f_{25}(h, \theta) + f_{26}(h, \theta)) - \ell_{2,3}(\theta) f_{24}(h, \theta) = 0$ $f_{24}(8, \theta) = f_{27}(8, \theta)$			
$m_{26} = (Fb, O, Fb)$	1.1	E_{24}	∂E_{24}
$q \frac{\partial}{\partial h} f_{26}(h, \theta) + \frac{q(\theta_1 - \theta)}{h} \frac{\partial}{\partial \theta} f_{26}(h, \theta) + \lambda^2(\theta) - \ell_2(\theta) f_{26}(h, \theta) = 0$ $f_{26}(8, \theta) = f_{29}(8, \theta)$			
$m_{27} = (Fb, F, F)$	4	E_{27}	∂E_{27}
$-q \frac{\partial}{\partial h} f_{27}(h, \theta) + \frac{K}{h} \frac{\partial}{\partial \theta} f_{27}(h, \theta) + \lambda^2(\theta) f_{34}(h, \theta) + \lambda^3(\theta) (f_{28}(h, \theta) + f_{29}(h, \theta)) - \ell_{2,3}(\theta) f_{27}(h, \theta) = 0$ $f_{27}(6, \theta) = f_{24}(6, \theta)$			
$m_{28} = (Fb, F, Ob)$	4	E_{27}	∂E_{27}
$-q \frac{\partial}{\partial h} f_{28}(h, \theta) + \frac{K}{h} \frac{\partial}{\partial \theta} f_{28}(h, \theta) + \lambda^2(\theta) f_{36}(h, \theta) - \ell_2(\theta) f_{28}(h, \theta) = 0$ $f_{28}(6, \theta) = f_{25}(6, \theta)$			
$m_{34} = (Fb, Fb, O)$	4	E_{27}	∂E_{27}
$-q \frac{\partial}{\partial h} f_{34}(h, \theta) + \frac{K}{h} \frac{\partial}{\partial \theta} f_{34}(h, \theta) + \lambda^3(\theta) (f_{36}(h, \theta) + 1) - \ell_3(\theta) f_{34}(h, \theta) = 0$ $f_{34}(6, \theta) = f_{35}(6, \theta)$			

TAB. 4.2 – Equations associées aux modes m_{24} , m_{26} , m_{27} , m_{28} et m_{34} .

Remarque 4.4 Les fonctions f_{24} , f_{26} , f_{27} , f_{28} et f_{34} sont définies le long d'union de courbes sur $[6, 8]$ et qui s'accroissent vers une courbe limite.

On peut alors écrire les équations correspondantes à ces modes, avec les conditions au bord associées (Tableau 4.2).

Remarque 4.5 Comme m_{35} est issu de m_{34} par la loi de commande L^2 , on peut maintenant en déduire le domaine E_{35} donné précédemment dans le paragraphe 4.2.1.4.

4.2.3 Le domaine E_{25}

Le mode m_{25} est issu du mode m_{24} . On a donc

$$E_{25} = [6, 8[\times]\theta_1, u_\infty].$$

On remarque que pour ce mode, le domaine n'a pas de frontière, donc pas de condition au bord car les valeurs $h = 8$ ou $\theta = \theta_1$ ne peuvent être atteintes (Tableau 2.5). L'équation du mode m_{25} est donnée par

$$\frac{q(\theta_1 - \theta)}{h} \frac{\partial}{\partial \theta} f(h, \theta) + \lambda^2(\theta) f_{36}(h, \theta) - \ell_2(\theta) f(h, \theta) = 0. \quad (4.2.1)$$

On constate qu'il n'y a pas unicité de la solution. En utilisant le Corollaire 1.1 du Chapitre 1, on a la proposition suivante

Proposition 4.1 L'unique solution bornée f_{25} de l'équation (4.2.1), définie le long du flot, et qui vérifie $\mathbb{E}_x[f(\phi(x, t))] \rightarrow 0$ quand $t \rightarrow \infty$ pour tout $x \in E_{25}$ et tout $t \in \mathbb{R}^+$ est donnée par

$$f_{25}(\phi(x, t)) = \int_t^\infty e^{\Lambda(x, t) - \Lambda(x, s)} \lambda^2(\phi(x, s)) f_{36}(\phi(x, s)) ds. \quad (4.2.2)$$

Preuve : Toutes les solutions sont de la forme

$$f(\phi(x, t)) = e^{\Lambda(x, t)} \left(\gamma - \int_0^t e^{-\Lambda(x, s)} \lambda^2(\phi(x, s)) f_{36}(\phi(x, s)) ds \right),$$

avec $\gamma \in \mathbb{R}$. Pour obtenir une solution bornée le long du flot et donc la solution unique d'après le Corollaire 1.1 du Chapitre 1, γ doit être égal à

$$\gamma = \int_0^\infty e^{-\Lambda(x, s)} \lambda^2(\phi(x, s)) f_{36}(\phi(x, s)) ds.$$

ce qui nous donne le résultat. \square

Corollaire 4.1 Pour tout $(h, \theta) \in E_{25}$, l'unique solution bornée f_{25} de l'équation (4.2.1) est donnée par

$$f_{25}(h, \theta) = \int_\theta^{\theta_1} \exp\left(-\int_\theta^{\theta''} \frac{2\lambda^2(\theta')h}{q(\theta_1 - \theta')} d\theta'\right) \frac{2\lambda^2(\theta'')h}{q(\theta_1 - \theta'')} \times \frac{1}{2} f_{36}(h, \theta'') d\theta''. \quad (4.2.3)$$

De plus, on a

$$\lim_{\theta \rightarrow \theta_1} f_{25}(h, \theta) = \frac{1}{2} f_{36}(h, \theta_1). \quad (4.2.4)$$

Preuve : Comme m_{25} correspond au cas 3, l'expression (4.2.3) se retrouve à partir de (4.2.2) par le changement de variable suivant :

$$\theta(t) = (\theta(0) - \theta_1)e^{-\frac{q}{h}t} + \theta_1.$$

Pour le calcul de la limite, on pose $F(\theta) = \frac{2\lambda^2(\theta)h}{q(\theta_1 - \theta)}$. Alors, (4.2.3) s'écrit

$$\begin{aligned} f_{25}(h, \theta) &= \lim_{t \rightarrow \theta_1} \int_{\theta}^t \exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) F(\theta'') \times \frac{1}{2} f_{36}(h, \theta'') d\theta'' \\ &= \lim_{t \rightarrow \theta_1} \int_{\theta}^t \exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) F(\theta'') \times \frac{1}{2} f_{36}(h, \theta_1) d\theta'' \\ &\quad + \lim_{t \rightarrow \theta_1} \int_{\theta}^t \exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) F(\theta'') \times \frac{1}{2} (f_{36}(h, \theta'') - f_{36}(h, \theta_1)) d\theta'' \end{aligned} \quad (4.2.5)$$

$$(4.2.6)$$

Calculons tout d'abord la limite du premier terme (4.2.5).

On a

$$\frac{\partial}{\partial \theta''} \left(\exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) \right) = -F(\theta'') \exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right).$$

Donc

$$\begin{aligned} &\frac{1}{2} f_{36}(h, \theta_1) \int_{\theta}^t \exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) F(\theta'') d\theta'' \\ &= \frac{1}{2} f_{36}(h, \theta_1) \left(-\exp\left(-\int_{\theta}^t F(\theta')d\theta'\right) + \exp\left(-\int_{\theta}^{\theta} F(\theta')d\theta'\right) \right) \\ &= \frac{1}{2} f_{36}(h, \theta_1) \left(-\exp\left(-\int_{\theta}^t F(\theta')d\theta'\right) + 1 \right). \end{aligned}$$

On a

$$\begin{aligned} \int_{\theta}^t F(\theta')d\theta' &= \int_{\theta}^t \frac{2h\lambda^2(\theta')}{q(\theta_1 - \theta')} d\theta' \\ &= \frac{2h\lambda^2(\theta_1)}{q} \int_{\theta}^t \frac{1}{\theta_1 - \theta'} d\theta' + \frac{2h}{q} \int_{\theta}^t \frac{(\lambda^2(\theta') - \lambda^2(\theta_1))}{\theta_1 - \theta'} d\theta' \\ &= \frac{2h\lambda^2(\theta_1)}{q} \ln\left(\frac{\theta_1 - \theta}{\theta_1 - t}\right) + \frac{2h}{q} \int_{\theta}^t \frac{(\lambda^2(\theta') - \lambda^2(\theta_1))}{\theta_1 - \theta'} d\theta' \end{aligned}$$

Or le théorème des accroissements finis appliqué à la fonction λ^2 , continue sur l'intervalle $[\theta_1, \theta'] \subset [\theta_1, \theta]$ nous donne

$$\lambda^2(\theta') - \lambda^2(\theta_1) = (\theta' - \theta_1)\lambda'^2(\theta_c) \text{ avec } \theta_c \in [\theta_1, \theta'].$$

Ainsi,

$$\exp\left(-\int_{\theta}^t F(\theta')d\theta'\right) = \exp\left(-\frac{2h\lambda^2(\theta_1)}{q} \ln\left(\frac{\theta_1 - \theta}{\theta_1 - t}\right) + \frac{2h}{q}\lambda'^2(\theta_c)(\theta' - \theta_1)\right)$$

et donc

$$\lim_{t \rightarrow \theta_1} \exp\left(-\int_{\theta}^t F(\theta')d\theta'\right) = 0.$$

La limite du premier terme (4.2.5) vaut

$$\lim_{\theta \rightarrow \theta_1} \lim_{t \rightarrow \theta_1} \frac{1}{2} f_{36}(h, \theta_1) \int_{\theta}^t \exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) F(\theta'')d\theta'' = \frac{1}{2} f_{36}(h, \theta_1). \quad (4.2.7)$$

Passons maintenant au calcul du deuxième terme de (4.2.6).

Le théorème des accroissements finis appliqué à la fonction $f_{36}(h, \cdot)$ sur l'intervalle $[\theta_1, \theta''] \subset [\theta_1, \theta]$, nous donne

$$f_{36}(h, \theta'') - f_{36}(h, \theta_1) = (\theta'' - \theta_1) f'_{36}(h, \theta_c) \text{ avec } \theta_c \in [\theta'', \theta_1].$$

On a ainsi

$$\begin{aligned} (4.2.6) &= \int_{\theta}^{\theta_1} \exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) F(\theta'') \times \frac{1}{2}(\theta'' - \theta_1) f'_{36}(h, \theta_c) d\theta'' \\ &= \frac{1}{2} f'_{36}(h, \theta_c) \int_{\theta}^{\theta_1} \exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) F(\theta'') \times (\theta'' - \theta_1) d\theta''. \end{aligned}$$

Pour tout $\theta'' \in [\theta, \theta_1]$, on a

$$\begin{aligned} F(\theta'') &\geq 0, \\ \exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) &\geq 0, \\ \theta'' - \theta_1 &\leq 0. \end{aligned}$$

Donc

$$\int_{\theta}^{\theta_1} \exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) F(\theta'')(\theta'' - \theta_1) d\theta'' \leq 0.$$

De plus,

$$\int_{\theta}^{\theta''} F(\theta')d\theta' \geq 0,$$

donc

$$\exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) \leq 1.$$

Ainsi

$$0 \geq \int_{\theta}^{\theta_1} \exp\left(-\int_{\theta}^{\theta''} F(\theta')d\theta'\right) (\theta'' - \theta_1) F(\theta'') d\theta'' \geq \int_{\theta}^{\theta_1} \frac{-2h\lambda^2(\theta'')}{q} d\theta''.$$

Comme $\lim_{\theta \rightarrow \theta_1} \int_{\theta}^{\theta_1} \frac{-2h\lambda^2(\theta'')}{q} d\theta'' = \lim_{\theta \rightarrow \theta_1} \frac{-2h\lambda^2(\theta_1)}{q} (\theta_1 - \theta) = 0$, on a par le théorème des gendarmes que la limite du deuxième terme (4.2.6) tend vers 0 quand θ tend vers θ_1 . Ainsi, avec (4.2.7), on a bien $\lim_{\theta \rightarrow \theta_1} f_{25}(h, \theta) = \frac{1}{2} f_{36}(h, \theta_1)$. \square

D'après le Corollaire 4.1, on peut alors poser comme condition au bord :

$$f_{25}(h, \theta_1) = \frac{1}{2} f_{36}(h, \theta_1).$$

Remarque 4.6 *Pour pouvoir appliquer le théorème des accroissements finis à la fonction f_{36} , il faut s'assurer de la régularité de celle-ci. Or nous savons déterminer f_{36} sur tout son domaine de définition E_{36} . En effet*

$$\begin{cases} f_{36}(4, \theta) = 0 & \text{pour } \theta \in [\theta_1, 100[, \\ f_{36}(h, 100) = 1 & \text{pour } h \in [4, 8]. \end{cases}$$

La Figure 4.9 représente le domaine E_{36} . La courbe en gras délimite la partie du domaine où f_{36} a pour valeur 0 et celle où elle vaut 1. Pour déterminer l'équation de cette courbe frontière, on montre qu'il existe $t \in \mathbb{R}^+$ tel que

$$S_{36} \begin{cases} h(t) = 4, \\ \theta(t) = 100. \end{cases}$$

Les expressions de h et θ sont données par la Proposition 2.1 dans le cas 4 (correspondant au mode m_{36}). On a

$$S_{36} \Leftrightarrow \begin{cases} -qt + h(0) = 4, \\ \theta(0) - \frac{K}{q} \ln\left(\frac{4}{h(0)}\right) = 100. \end{cases}$$

et donc

$$S_{36} \Leftrightarrow \begin{cases} t = \frac{h(0) - 4}{q}, \\ \theta(0) = 100 + \frac{K}{q} \ln\left(\frac{4}{h(0)}\right). \end{cases}$$

L'équation de la courbe frontière en fonction de h est donc de la forme

$$\theta = 100 + \frac{K}{q} \ln\left(\frac{4}{h}\right).$$

Ainsi, comme on peut le voir sur la Figure 4.9, on a que pour tout couple (h, θ) situé en-dessous de la courbe, $f_{36}(h, \theta) = 0$ et pour tout (h, θ) situé au-dessus de la courbe, $f_{36}(h, \theta) = 1$. En particulier, tout point du domaine E_{25} se situe sous la courbe frontière. Donc f_{36} est bien régulière sur E_{25} .

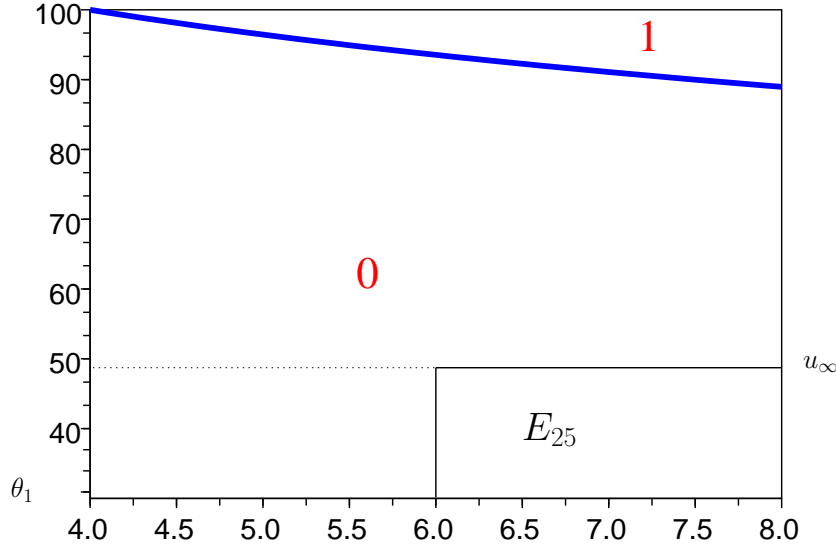


FIG. 4.9 – Domaines E_{36} et E_{25} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.

4.3 Résolution numérique

Pour calculer les fonctions solutions du système couplé, nous allons procéder en plusieurs étapes. Dans un premier temps, il faut redresser les fonctions définies sur les domaines du type E_{24} ou E_{27} pour les calculer sur des grilles cartésiennes. Une interpolation des fonctions sera ensuite nécessaire pour connaître les valeurs de celles-ci sur toutes les grilles. Nous présenterons ensuite les résultats du calcul de la probabilité de surchauffe.

Remarque 4.7 *Remarquons que chacune des équations différentielles du système couplé peut se résoudre explicitement. En effet, l'équation de f_{36} est autonome et sa solution permet de résoudre l'équation de f_{35} , qui ne dépend elle-même que de f_{35} et f_{36} , etc. Cependant, nous ne chercherons pas à résoudre analytiquement le système car nous souhaitons développer une méthode numérique pour les solutions du système couplé qui ne peuvent pas être forcément calculées explicitement.*

4.3.1 Redressement des fonctions

Les domaines de définition étant assez complexes, nous allons définir de nouveaux domaines sur lesquels les courbes seront « redressées », grâce à un changement de variable. Prenons l'exemple du mode m_{24} pour illustrer notre démarche. L'équation correspondant au mode m_{24} est donnée par :

$$q \frac{\partial}{\partial h} f_{24}(h, \theta_n) + \frac{q(\theta_1 - \theta_n)}{h} \frac{\partial}{\partial \theta} f_{24}(h, \theta_n) + \lambda^2(\theta_n) f_{35}(h, \theta_n) + \lambda^3(\theta_n) (f_{25}(h, \theta_n) + f_{26}(h, \theta_n)) - \ell_{2,3}(\theta_n) f_{24}(h, \theta_n) = 0.$$

Par un changement de variable décrit ci-après et pour simplifier l'implémentation, nous allons définir un nouveau domaine $D_{24} = [6, 8] \times \mathbb{N}^*$, l'idée étant de décrire f_{24}

sur chaque morceau de courbe.

En remarquant que pour le mode m_{24} ,

$$\frac{\partial \theta}{\partial h} = \frac{\partial \theta}{\partial t} \frac{\partial t}{\partial h} = \frac{q(\theta_1 - \theta)}{h} \times \frac{1}{q} = \frac{\theta_1 - \theta}{h},$$

on peut récrire l'équation :

$$q \left(\frac{\partial}{\partial h} f_{24}(h, \theta_n) + \frac{\partial \theta}{\partial h} \frac{\partial}{\partial \theta} f_{24}(h, \theta_n) \right) + \lambda^2(\theta_n) f_{35}(h, \theta_n) + \lambda^3(\theta_n) (f_{25}(h, \theta_n) + f_{26}(h, \theta_n)) - \ell_{2,3}(\theta_n) f_{24}(h, \theta) = 0.$$

Nous procédons maintenant à un changement de variable. Soit

$$\begin{aligned} \phi : D_{24} &\rightarrow E_{24} \\ (h, n) &\mapsto (h, \theta_n) \end{aligned} \quad \text{et} \quad \begin{aligned} f_{24} : E_{24} &\rightarrow \mathbb{R}^2 \\ (h, \theta) &\mapsto f_{24}(h, \theta) \end{aligned}$$

où $\theta_n = \frac{6}{h}(u_n - \theta_1) + \theta_1$ et ϕ est un \mathcal{C}^1 -difféomorphisme. On pose $g_{24} = f_{24} \circ \phi : D_{24} \rightarrow \mathbb{R}^2$.

Comme

$$\begin{aligned} \frac{\partial \phi}{\partial h} &= (1, -\frac{6}{h^2}(u_n - \theta_1)) \\ &= (1, \frac{\partial \theta}{\partial h}(h, \theta_n)), \end{aligned}$$

la dérivée de g_{24} fonction composée, est :

$$\begin{aligned} \frac{\partial g_{24}}{\partial h}(h, n) &= \left(\frac{\partial f_{24}}{\partial h} \circ \phi \right) \times 1 + \left(\frac{\partial f_{24}}{\partial \theta} \circ \phi \right) \times \frac{\partial \theta}{\partial h}(h, \theta_n) \\ &= \frac{\partial f_{24}}{\partial h}(h, \theta_n) + \frac{\partial \theta}{\partial h}(h, \theta_n) \frac{\partial f_{24}}{\partial \theta}(h, \theta_n). \end{aligned}$$

Avec ce changement de variables, nous pouvons donner une nouvelle description du mode m_{24} . Le domaine de définition est donné par $D_{24} = [6, 8[\times \mathbb{N}^*$, la frontière par $\partial D_{24} = \{8\} \times \mathbb{N}^*$, et avec $g_{24}(h, n) = f_{24}(h, \theta_n)$, on a l'équation

$$q \frac{\partial}{\partial h} g_{24}(h, n) + \lambda^2(\theta_n) f_{35}(h, \theta_n) + \lambda^3(\theta_n) (f_{25}(h, \theta_n) + f_{26}(h, \theta_n)) - \ell_{2,3}(\theta_n) g_{24}(h, n) = 0,$$

et la condition au bord devient

$$g_{24}(8, n) = f_{27}(8, v_n).$$

On procède de la même façon pour m_{26} . Pour les modes m_{27} , m_{28} et m_{34} , en prenant $\phi'_1(h, n) = 1$ et $\phi'_2(h, n) = \frac{-K}{qh} = \frac{\partial \theta}{\partial h}$, nous obtenons également les équations sur les nouveaux domaines redressés (Tableau 4.3).

MODE	CAS	DOMAINE	FRONTIERE
$m_{24} = (Fb, O, F)$	1.1	D_{24}	∂D_{24}
$q \frac{\partial}{\partial h} g_{24}(h, n) + \lambda^2(\theta_n) f_{35}(h, \theta_n) + \lambda^3(\theta_n) (f_{25}(h, \theta_n) + f_{26}(h, \theta_n)) - \ell_{2,3}(\theta_n) g_{24}(h, n) = 0$ $g_{24}(8, n) = f_{27}(8, v_n)$			
$m_{26} = (Fb, O, Fb)$	1.1	D_{24}	∂D_{24}
$q \frac{\partial}{\partial h} g_{26}(h, n) + \lambda^2(\theta_n) - \ell_2(\theta_n) g_{26}(h, n) = 0$ $g_{26}(8, n) = f_{29}(8, v_n)$			
$m_{27} = (Fb, F, F)$	4	D_{27}	∂D_{27}
$-q \frac{\partial}{\partial h} g_{27}(h, n) + \lambda^2(\theta_n) f_{34}(h, \theta_n) + \lambda^3(\theta_n) (f_{28}(h, \theta_n) + f_{29}(h, \theta_n)) - \ell_{2,3}(\theta_n) g_{27}(h, n) = 0$ $g_{27}(6, n) = f_{24}(6, u_{n+1})$			
$m_{28} = (Fb, F, Ob)$	4	D_{27}	∂D_{27}
$-q \frac{\partial}{\partial h} g_{28}(h, n) + \lambda^2(\theta_n) f_{36}(h, \theta_n) - \ell_2(\theta_n) g_{28}(h, n) = 0$ $g_{28}(6, n) = f_{25}(6, u_{n+1})$			
$m_{34} = (Fb, Fb, O)$	4	D_{27}	∂D_{27}
$-q \frac{\partial}{\partial h} g_{34}(h, n) + \lambda^3(\theta_n) (f_{36}(h, \theta_n) + 1) - \ell_3(\theta_n) g_{34}(h, n) = 0$ $g_{34}(6, n) = f_{35}(6, u_{n+1})$			

TAB. 4.3 – Equations des fonctions redressées associées aux modes m_{24} , m_{26} , m_{27} , m_{28} et m_{34} .

Pas $\delta\theta$	Pas δh	N_θ	N_h
1	0.2386	69	16
0.1	0.02386	690	167
0.01	0.002386	6907	1676

TAB. 4.4 – Tableau des différents pas de discrétisation.

4.3.2 Schéma de discrétisation

Les domaines étant définis, on établit sur le plus grand d'entre eux $[4, 8] \times [\theta_1, 100]$ une grille de discrétisation cartésienne en θ et h . On note :

- $\delta\theta$ le pas en θ ,
- δh le pas en h ,

liés par une relation standard de calcul numérique :

$$\delta h = \frac{4q \times \delta\theta}{K} \times 0.95.$$

On définit ainsi pour $i = 1, \dots, N_h + 1$ et $j = 1, \dots, N_\theta + 1$ le couple (h_i, θ_j) par

$$(h_i, \theta_j) = (4 + (i - 1)\delta h, \theta_1 + (j - 1)\delta\theta),$$

où $N_h = \frac{4}{\delta h}$ et $N_\theta = \frac{100 - \theta_1}{\delta\theta}$ sont les nombres de points dans la grille. Le Tableau 4.4 donne les valeurs de N_θ et N_h en fonction des pas de discrétisation $\delta\theta$ et δh .

Et pour $m \in \mathbb{M}$, on note

$$f_m(i, j) = f_m(h_i, \theta_j).$$

Remarque 4.8 Dans le cas où l'on étudie l'événement $\{\theta > 100\}$, la hauteur du réservoir ne peut dépasser physiquement 8 m (on le voit grâce à l'étude de l'équation de la hauteur dans les modes correspondant à cette situation).

Pour chaque équation, on part de la condition au bord, qui nous donne sur la grille les points de la « couche » d'où l'on part, et à partir de ces points, on calcule les points de la couche suivante. Et on avance ainsi pas à pas. Expliquons notre démarche par un exemple. On regarde le cas de l'équation de f_{35} . Le domaine de définition $E_{35} =]6, 8] \times [\theta_1, 100[$ ne recouvre pas la totalité de la grille initiale. On peut donc restreindre les calculs sur la moitié de cette grille. La frontière de l'espace d'état est $\partial E_{35} = [6, 8] \times \{100\}$. La condition au bord est $f(i, N_\theta + 1) = 1$ pour tout $1 \leq i \leq N_h + 1$ fixé. On connaît donc la valeur de f_{35} sur $\{1, \dots, N_h + 1\} \times \{N_\theta + 1\}$. On va calculer la valeur de f_{35} en N_θ pour tout $1 \leq i \leq N_h + 1$ grâce à la condition au bord. Et ainsi de suite. L'équation discrétisée est :

$$\frac{K}{h_i} \frac{f_{35}(i, j + 1) - f_{35}(i, j)}{\delta\theta} + \lambda^3(\theta_j)(f_{35}(i, j) + 1) - 2\lambda^3(\theta_j)f_{35}(i, j) = 0$$

pour $j = N_\theta$ à 1.

On va donc écrire les $f_{35}(i, j)$ en fonction des $f_{35}(i, j + 1)$ connus car calculés au pas précédent.

Le Tableau A.3 en Annexes donne les 17 équations discrétisées pour les 17 modes.

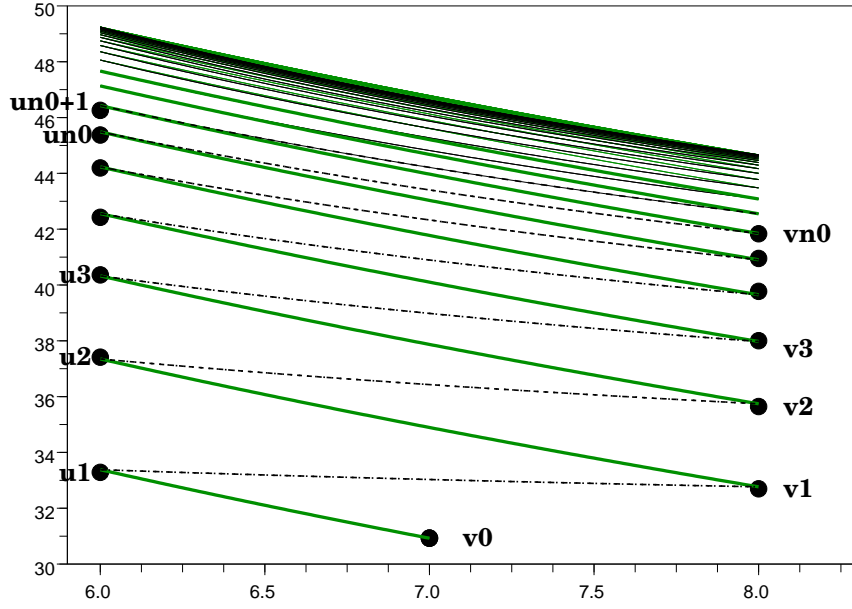


FIG. 4.10 – Domaines E_{24} et E_{27} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.

4.3.3 Interpolations

Comme nous l'avons expliqué précédemment, pour les fonctions du type f_{24} ou f_{27} , nous faisons un changement de variables qui nous permet de calculer les fonctions sur une grille plus adéquate. Nous avons vu que les nouvelles fonctions sont définies sur un espace du type $[4, 8] \times \mathbb{N}$. Pour la résolution numérique, nous avons donc besoin de déterminer sur la grille jusqu'à quel rang $n_0 \in \mathbb{N}$ les fonctions vont être calculées. C'est ce que nous ferons dans un premier temps, puis nous interpolerons sur cette grille les fonctions intervenant dans les équations des modes du type m_{24} ou m_{27} et définies sur des domaines différents.

Détermination de n_0 Nous avons défini précédemment deux suites $(u_n)_n$ et $(v_n)_n$ qui nous permettent de décrire les domaines de définition E_{24} et E_{27} (Figure 4.10). L'indice n_0 est choisi de telle sorte que l'écart entre $f_{27}(h, n_0)$ et $f_{27}(h, n_0 + 1)$ d'une part et $f_{24}(h, n_0)$ et $f_{24}(h, n_0 + 1)$ d'autre part soit suffisamment petit. Pour trouver n_0 , il faut donc définir un critère d'arrêt qui détermine l'instant où $v_{n_0+1} - v_{n_0} \leq \delta\theta$.

Interpolations Les équations des fonctions du type f_{24} ou f_{27} (Tableau 4.2) font intervenir des fonctions définies sur des grilles cartésiennes différentes de celles de f_{24} ou f_{27} (par exemple D_{27} et E_{36} sur la Figure 4.11). D'après le Tableau 4.2, on voit que pour calculer f_{28} par exemple, il faut connaître f_{36} sur D_{27} . Or, on ne connaît les valeurs de f_{36} que sur la grille cartésienne $(h_i, \theta_j)_{1 \leq i \leq N_h+1, 1 \leq j \leq N_\theta+1}$ (c'est-à-dire les croix sur la Figure 4.11). Nous devons donc interpoler f_{36} sur D_{27} .

Remarque 4.9 Nous noterons G_{24} la grille de discrétisation du domaine D_{24} et

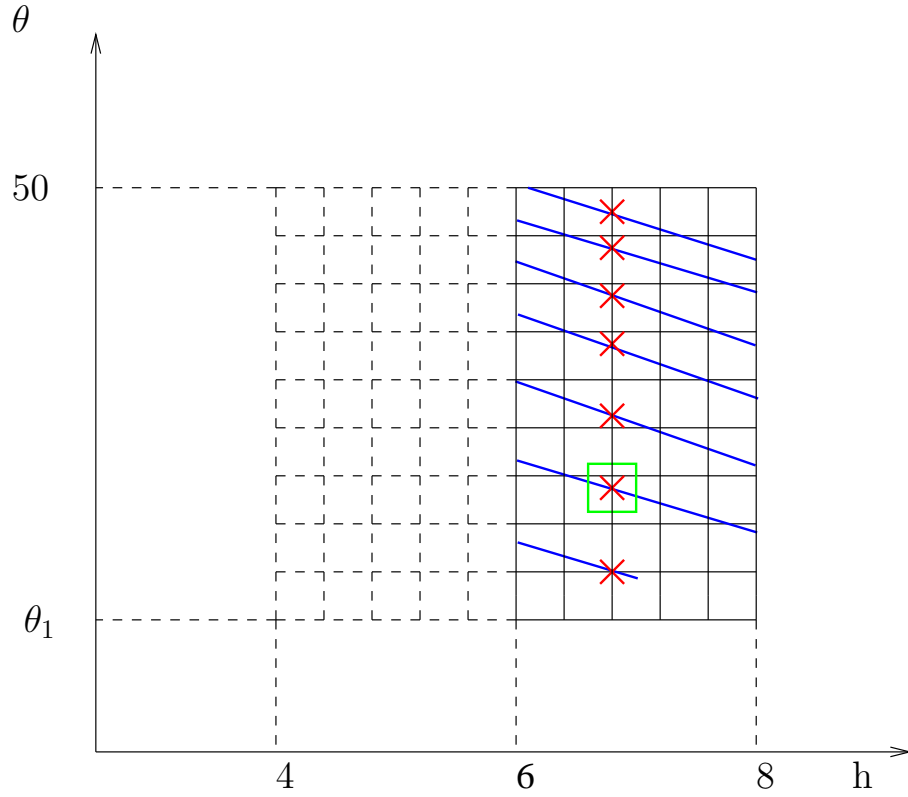


FIG. 4.11 – Grilles cartésiennes superposées D_{27} et E_{36} .

G_{27} la grille de discrétisation du domaine D_{27} .

Avant cela, nous rappelons le changement de variable effectué pour calculer les fonctions du type f_{27} :

$$g_{27}(h, n) = f_{27}(h, w_n) \text{ pour tout } h \in]6, 8] \times \{0, 1, \dots, n_0\}$$

avec la suite (w_n) définie par

$$\begin{cases} w_n = v_n - \frac{K}{q} \ln \frac{h}{8} & h \in [6, 8] \ \forall n \in \{0, 1, \dots, n_0\}, \\ w_0 = v_0 - \frac{K}{q} \ln \frac{h}{8} & h \in [6, 7]. \end{cases}$$

Sur la grille discrétisée G_{27} , on a alors

$$g_{27}(i, n) = f_{27}(i, w_{i,n}) \text{ pour tout } N_h/2 + 1 \leq i \leq N_h + 1 \text{ et } n \in \{0, 1, \dots, n_0\}$$

avec la suite $(w_{i,n})$ définie pour tout $N_h/2 + 1 \leq i \leq N_h + 1$ et $n \in \{0, 1, \dots, n_0\}$ par

$$\begin{cases} w_{i,n} = v_n - \frac{K}{q} \ln \frac{h_i}{8}, \\ w_{i,0} = v_0 - \frac{K}{q} \ln \frac{h_i}{8}. \end{cases}$$

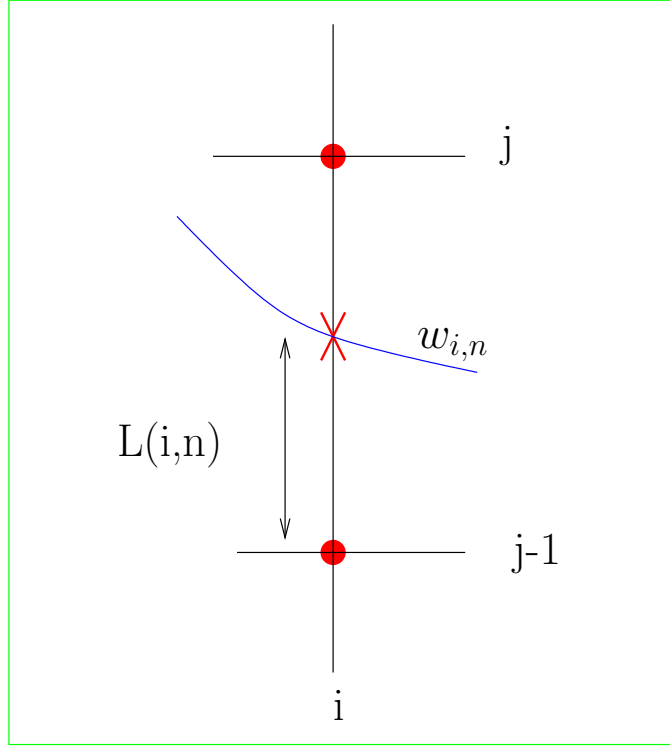


FIG. 4.12 – Interpolation d'un point de E_{36} sur D_{27} .

Remarque 4.10 Le point $w_{i,n}$ est donc le point d'intersection entre la droite $h = h_i$ et $(D_{27} \times \{n\})$. (Figure 4.12).

Pour interpoler f_{36} sur G_{27} , nous choisissons de définir $f_{36}(i, w_{i,n})$, où $(i, w_{i,n})$ est le point de la grille G_{27} pour lequel nous ne connaissons pas f_{36} , comme le barycentre de $f_{36}(i, j-1)$ et $f_{36}(i, j)$, points pour lesquels nous connaissons f_{36} . Il faut donc tout d'abord déterminer les coordonnées de $(i, j-1)$ et (i, j) sur la grille G_{27} , puis déterminer le poids de $f_{36}(i, j-1)$ et $f_{36}(i, j)$.

1. Comme nous gardons la même hauteur en abscisse, nous cherchons uniquement $j-1$ et j dans la grille G_{27} . Nous avons assez facilement que

$$j-1 = \left\lfloor \frac{w_{i,n} - \theta_1}{\delta\theta} \right\rfloor,$$

avec pour tout réel x , la partie entière $[x]$ définie comme le plus grand entier relatif inférieur ou égal à x . Donc, si l'on note ent_w le vecteur des indices j dans la grille G_{27} , on a

$$ent_w(i, n) = \left\lfloor \frac{w_{i,n} - \theta_1}{\delta\theta} \right\rfloor + 1.$$

2. Le poids de $f_{36}(i, w_{i,n})$ est donné par la distance $L_w(i, n)$ entre l'indice $ent_w(i, n)$ et $w_{i,n}$. On a

$$L_w(i, n) = \frac{w_{i,n} - \theta_1}{\delta\theta} - ent_w(i, n).$$

De même, pour la partie de courbe de D_{27} comprise entre 6 m et 7 m, on a

$$ent_{w0}(i) = E \left[\frac{w_{i,0} - \theta_1}{\delta\theta} \right] + 1$$

et

$$L_{w0}(i) = \frac{w_{i,0} - \theta_1}{\delta\theta} - ent_{w0}(i).$$

Ainsi, pour calculer l'interpolée \tilde{f}_{36} de la fonction f_{36} sur la grille G_{27} , on a

$$\tilde{f}_{36}(i, n) = L_w(i, n)f_{36}(i, ent_w(i, n)) + (1 - L_w(i, n))f_{36}(i, ent_w(i, n) + 1).$$

De la même façon, en considérant le changement de variable

$$g_{24}(h, n) = f_{24}\left(h, \frac{6}{h}(u_n - \theta_1) + \theta_1\right) \text{ pour tout } h \in [6, 8[\times \{0, 1, \dots, n_0\}$$

on définit la suite (z_n) pour tout $h \in [6, 8[$ et tout $n \in \{0, 1, \dots, n_0\}$ par

$$z_n = \frac{6}{h}(u_n - \theta_1) + \theta_1.$$

La suite discrétisée $(z_{i,n})$ est définie pour tout $N_h/2 + 1 \leq i \leq N_h + 1$ et tout $n \in \{0, 1, \dots, n_0\}$ par

$$z_{i,n} = \frac{6}{h_i}(u_n - \theta_1) + \theta_1.$$

Sur la grille discrétisée G_{24} , on a alors

$$g_{24}(i, n) = f_{24}(i, z_{i,n}) \text{ pour tout } N_h/2 + 1 \leq i \leq N_h + 1 \text{ et } n \in \{0, 1, \dots, n_0\}.$$

Alors on a

$$ent_z(i, n) = \left[\frac{z_{i,n} - \theta_1}{\delta\theta} \right] + 1$$

et

$$L_z(i, n) = \frac{z_{i,n} - \theta_1}{\delta\theta} - ent_z(i, n).$$

Et l'interpolée \tilde{f} de toute fonction f sur la grille G_{24} est donnée par

$$\tilde{f}(i, n) = L_z(i, n)f(i, ent_z(i, n)) + (1 - L_z(i, n))f(i, ent_z(i, n) + 1).$$

4.4 Calcul de la probabilité de surchauffe

A partir de la résolution numérique des équations différentielles discrétisées dans l'ordre donné par le graphique de la Figure 4.13, nous pouvons calculer les 17 fonctions sur leur grille de discrétisation, à des pas différents pour $\delta\theta$, dans l'ordre d'implémentation permettant de calculer f_1 en $(7, \theta_1)$. Pour cela, nous avons programmé en Scilab, avec différents pas de discrétisation. Le code se trouve en Annexe B.2.

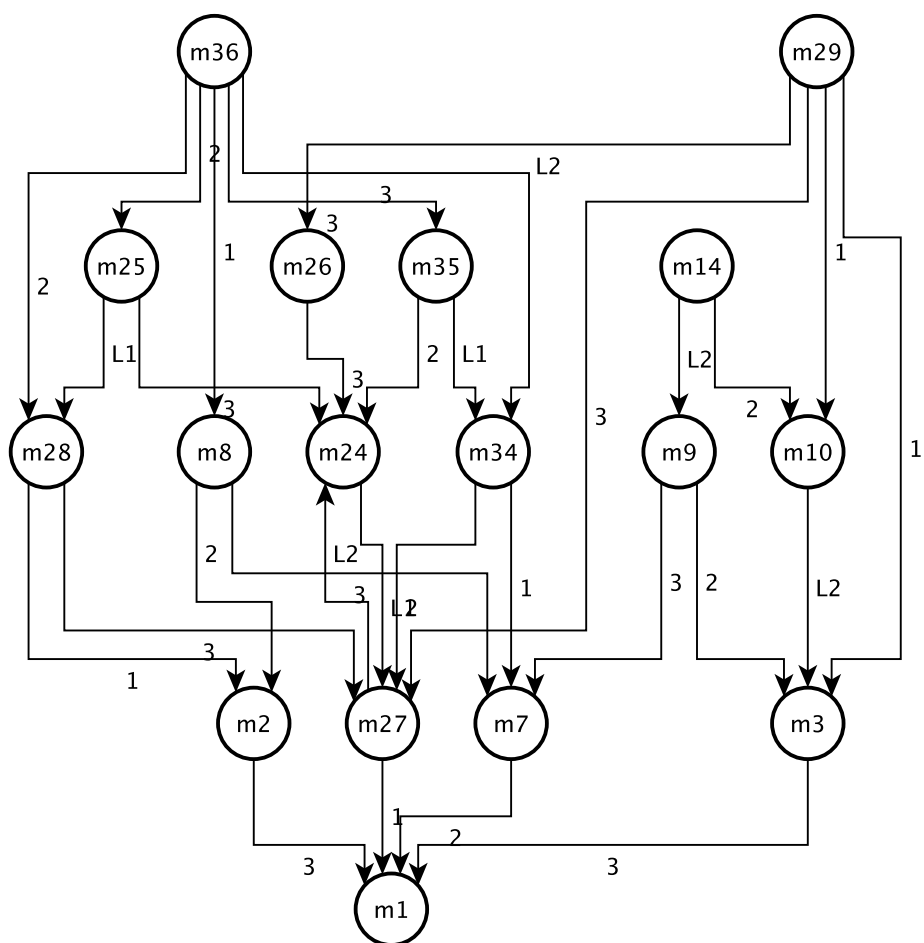


FIG. 4.13 – Graphe des 17 modes dans l'ordre de calcul.

Pas de discrétisation ($\delta\theta$)	1	0.1	0.01
$f_{36}(7, \theta_1)$	0	0	0
$f_{35}(7, \theta_1)$	0.749	0.741	0.739
$g_{34}(7, \theta_1)$	0.998	0.998	0.998
$f_{29}(7, \theta_1)$	0.561	0.552	0.550
$f_{25}(7, \theta_1)$	0	0	0
$g_{28}(7, \theta_1)$	0.993	0.992	0.992
$g_{26}(7, \theta_1)$	0.545	0.538	0.536

TAB. 4.5 – Les fonctions f_{36} , f_{35} , g_{34} , f_{29} , f_{25} , g_{28} et g_{26} en $(7, \theta_1)$ pour différents pas de discrétisation.

Pas de discrétisation ($\delta\theta$)	1	0.1	0.01
$f_{14}(\theta_1)$	0.571	0.564	0.562
$f_{10}(\theta_1)$	0.311	0.303	0.302

TAB. 4.6 – Les fonctions f_{14} et f_{10} en θ_1 pour différents pas de discrétisation.

Fonctions f_{36} , f_{35} , g_{34} , f_{29} , f_{25} , g_{28} et g_{26} La Figure 4.14 représente la fonction f_{36} avec un pas $\delta\theta = 1$. Elle prend comme valeurs, par définition, 1 ou 0 (Figure 4.9) avec une frontière entre ces valeurs définie par

$$\theta = 100 + \frac{K}{q} \ln\left(\frac{4}{h}\right).$$

Le calcul numérique de f_{36} est cohérent avec la fonction théorique d’après la Figure 4.14. Pour connaître la valeur de f_1 en $(7, \theta_1)$, nous avons besoin de connaître f_{36} sur tous les points de sa grille. De la même façon, nous avons f_{35} , g_{34} , f_{29} , f_{25} , g_{28} et g_{26}

Dans le Tableau 4.5 nous donnons uniquement les valeurs de ces fonctions en $(7, \theta_1)$ pour différents pas $\delta\theta$. Les Figures 4.14-4.20 donnent le tracé des fonctions.

Fonction f_{14} et f_{10} Les fonctions f_{14} et f_{10} ne dépendent que de la température. Pour calculer f_1 en $(7, \theta_1)$, nous avons donc besoin de la valeur de f_{14} et f_{10} en θ_1 . Les Figures 4.21 et 4.22 nous donnent le tracé de f_{14} et f_{10} et le Tableau 4.6 leurs valeurs en θ_1 .

Fonction f_9 et f_3 Les fonctions f_9 et f_3 ne dépendent que de la hauteur. On a donc besoin de leur valeur pour $h = 7$. Les Figures 4.23 et 4.24 nous donnent le tracé de f_9 et f_3 , et le Tableau 4.7 leurs valeurs en θ_1 .

Fonction f_8 , f_7 et f_2 Les fonctions f_8 , f_7 et f_2 sont définies en un seul point $(7, \theta_1)$. Le Tableau 4.8 donne leurs valeurs.

Pas de discrétisation ($\delta\theta$)	1	0.1	0.01
$f_9(7)$	0.570	0.563	0.562
$f_3(7)$	0.311	0.303	0.301

TAB. 4.7 – Les fonctions f_9 et f_3 en $h = 7$ pour différents pas de discrétisation.

Pas de discrétisation ($\delta\theta$)	1	0.1	0.01
f_8	0	0	0
f_7	0.338	0.335	0.334
f_2	0	0	0

TAB. 4.8 – Les fonctions f_8 , f_7 et f_2 pour différents pas de discrétisation.

Fonctions g_{24} et g_{27} Pour calculer f_1 en $(7, \theta_1)$, nous n'avons besoin de connaître g_{24} et g_{27} qu'en $(7, \theta_1)$. Or, d'après le graphe des transitions (Figure 4.1) et le domaine de définition couplé de ces deux fonctions (Section 4.2.2), nous voyons qu'il suffit de connaître la valeur de g_{27} uniquement au point initial $(7, \theta_1)$. Notre programme en Annexe B.2 retourne donc la valeur de g_{27} en $(7, \theta_1)$ après itérations de g_{24} et g_{27} . Le Tableau 4.9 nous donne ces valeurs.

Fonction f_1 La fonction f_1 définie en $(7, \theta_1)$, correspond à la probabilité de surchauffe du réservoir. Comme expliqué précédemment, elle se calcule à partir des fonctions précédentes au point $(7, \theta_1)$. Le Tableau 4.10 donne ses valeurs aux différents pas de discrétisation.

Pas de discrétisation ($\delta\theta$)	1	0.1	0.01
$f_1(7, \theta_1)$	0.187	0.185	0.185

TAB. 4.10 – La fonction f_1 en $(7, \theta_1)$ pour différents pas de discrétisation.

Le Tableau 4.11 donne les temps de compilation en fonction du pas de discrétisation pour le calcul de la fonction f_1 , sur un ordinateur de bureau standard.

Pas de discrétisation ($\delta\theta$)	1	0.1	0.01
Temps de calcul	0.15s	2.49s	2mn41s

TAB. 4.11 – Temps de calcul pour différents pas de discrétisation.

Pas de discrétisation ($\delta\theta$)	1	0.1	0.01
$g_{27}(7, \theta_1)$	0.463	0.463	0.463

TAB. 4.9 – La fonction g_{27} en $(7, \theta_1)$ pour différents pas de discrétisation.

Nous rappelons que la première méthode numérique développée au Chapitre 3 nous donnait $p_3(1000) = 0.163368$ pour $N = 10^4$ histoires, et nous obtenons $p_3(\infty) = 0.185$ pour un pas de discrétisation en θ de 0.01.

L'ordre de grandeur des deux résultats est similaire, ce qui illustre la cohérence de ceux-ci. La différence entre ces deux valeurs vient en partie du fait que nous calculons ici la probabilité de surchauffe à *horizon infini*, et non pas à horizon de 1000h, comme dans le premier cas. Il est donc logique d'obtenir une valeur légèrement supérieure. De plus, nous avons été assez rapidement confronté à un problème de taille mémoire dû aux capacités limitées dans ce domaine de Scilab. En effet, à partir du pas $\delta\theta = 0.005$, Scilab ne peut plus stocker toutes les valeurs. Le pas de discrétisation étant moins fin que pour la première méthode numérique, il est logique d'obtenir une valeur supérieure. Les temps de simulation pour cette méthode sont raisonnables (Tableau 4.11) comparés à la première méthode numérique (10.03s pour $N = 10^4$) et autres méthodes exposées précédemment. Les équations différentielles peuvent se résoudre une à une assez facilement, sans passer par un algorithme de discrétisation partant de la fonction f_{36} , mais nous avons appliqué cette méthode afin de tester la faisabilité de notre démarche.

Les deux méthodes numériques développées dans le Chapitre 3 et 4 nous permettent de mettre en évidence l'intérêt d'une modélisation par les PDMP. La similarité des résultats avec les autres méthodes de fiabilité dynamique montre bien la cohérence de notre démarche. L'étude, qualitative et quantitative, du comportement du réservoir nous a particulièrement aidé à comprendre les interactions entre la hauteur, la température et le mode. Nous avons ainsi pu calculer les probabilités d'assèchement, de débordement et de surchauffe pour la première méthode, et uniquement la probabilité de surchauffe pour la deuxième méthode, mais un travail analogue à celui exposé ici conduirait au calcul des probabilités des deux autres événements redoutés. Il pourrait être également intéressant d'envisager une complexification du problème du cas-test, en modifiant les lois de commande déterministes en lois de commande aléatoires et/ou en considérant que les unités sont réparables. Dans le premier cas, nous avons vu que la définition du PDMP, telle que nous la donnons, prend en compte cette situation. En particulier, l'expression de la mesure de transition se trouverait modifiée, sans que cela n'affecte fondamentalement la construction des deux méthodes numériques. Dans le deuxième cas (unités réparables), le graphe des transitions se trouverait modifié (ajout d'arcs entre les modes). Cette situation ne changerait pas essentiellement le déroulement de la première méthode. Dans le cas de la deuxième méthode, nous nous trouverions avec un système d'équations couplées de la même façon que les équations 24 et 27. Il faudrait alors utiliser une résolution itérative sur le même modèle que celui de la résolution de ces deux équations, mais opérée cette fois-ci sur l'ensemble des équations, ce qui peut être assez complexe, mais réalisable.

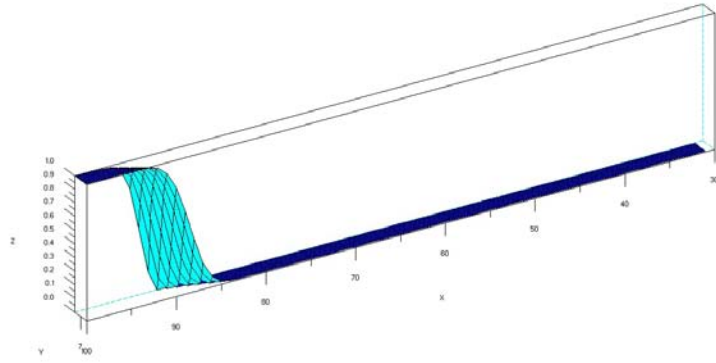


FIG. 4.14 – Fonction f_{36} , avec la température en X , la hauteur en Y et la probabilité en Z .

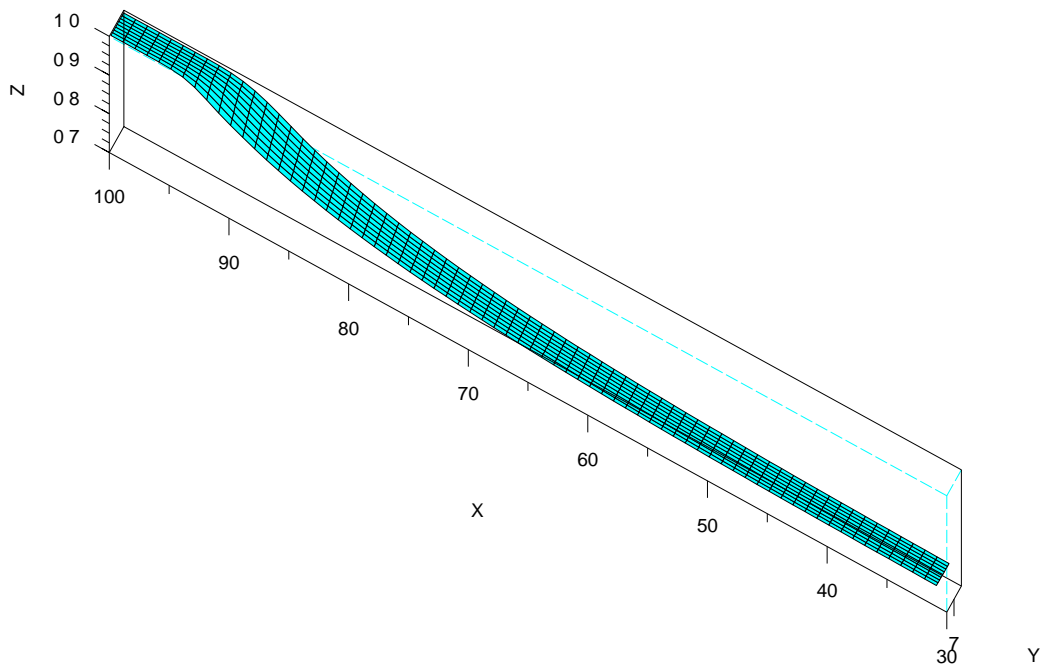


FIG. 4.15 – Fonction f_{35} , avec la température en X , la hauteur en Y et la probabilité en Z .

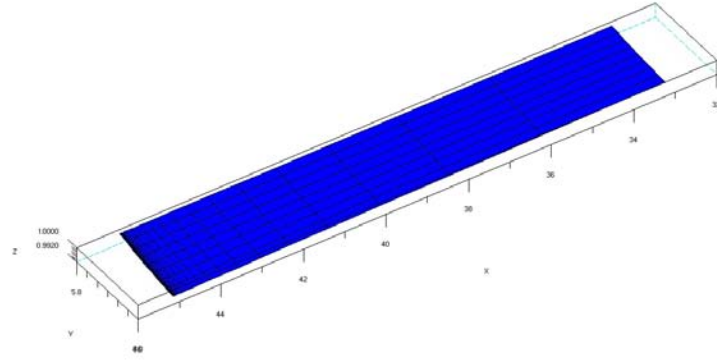


FIG. 4.16 – Fonction g_{34} , avec la température en X , la hauteur en Y et la probabilité en Z .

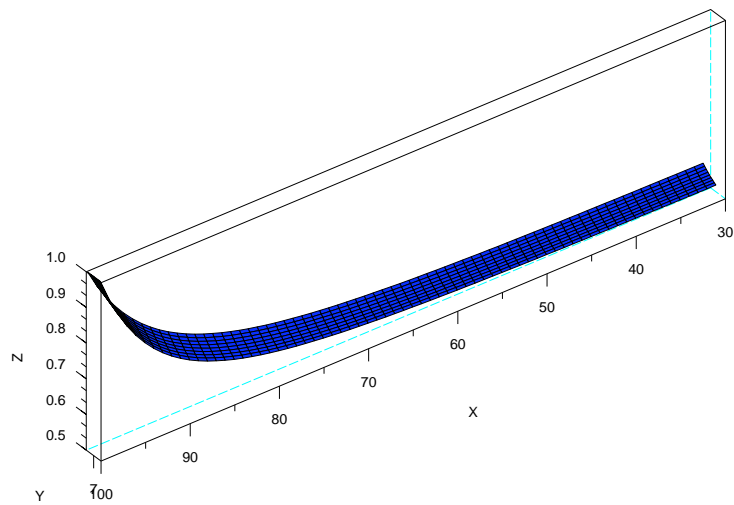


FIG. 4.17 – Fonction f_{29} , avec la température en X , la hauteur en Y et la probabilité en Z .

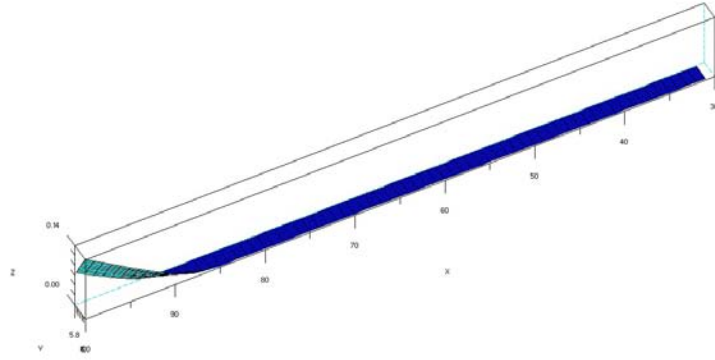


FIG. 4.18 – Fonction f_{25} , avec la température en X , la hauteur en Y et la probabilité en Z .

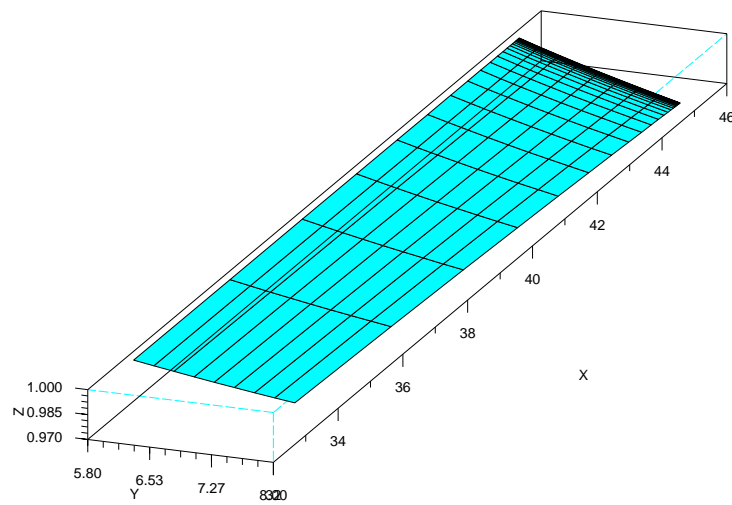


FIG. 4.19 – Fonction g_{28} , avec la température en X , la hauteur en Y et la probabilité en Z .

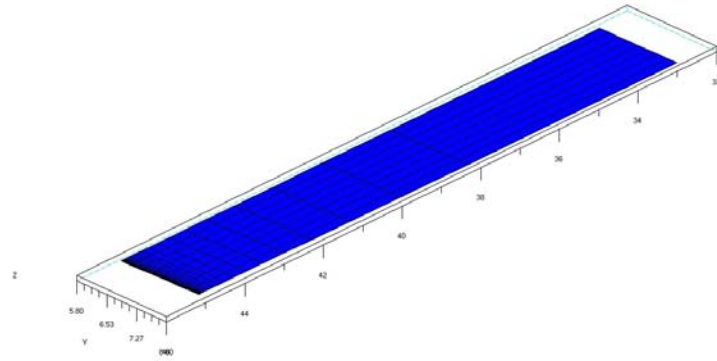


FIG. 4.20 – Fonction g_{26} , avec la température en X, la hauteur en Y et la probabilité en Z.

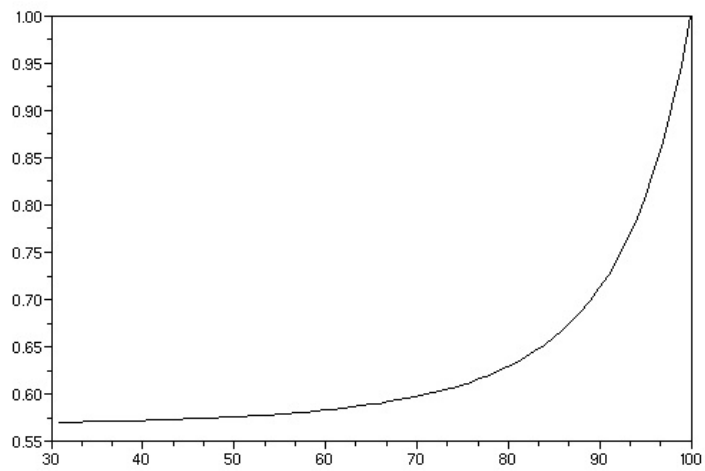


FIG. 4.21 – Fonction f_{14} , avec la température en abscisse et la probabilité en ordonnée.

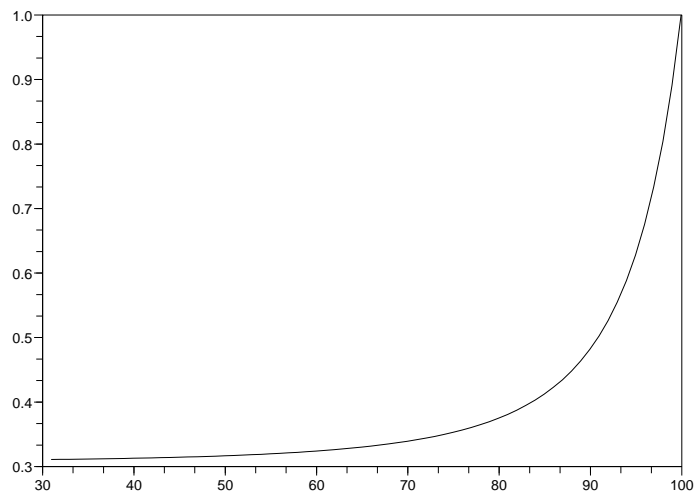


FIG. 4.22 – Fonction f_{10} , avec la température en abscisse et la probabilité en ordonnée.

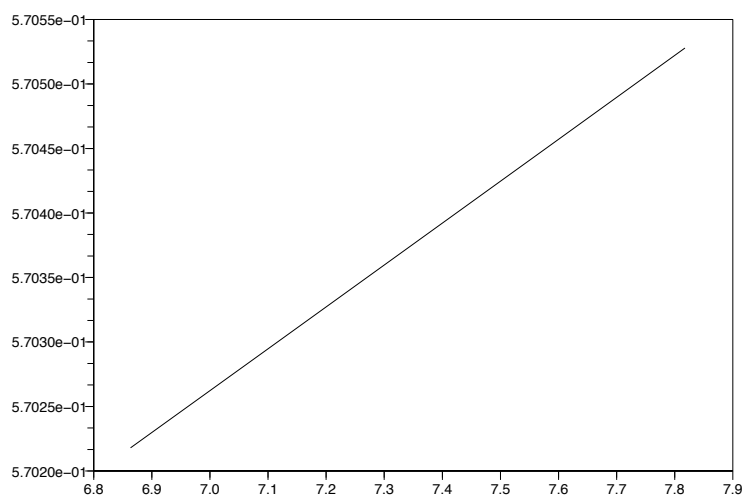


FIG. 4.23 – Fonction f_9 , avec la hauteur en abscisse et la probabilité en ordonnée.

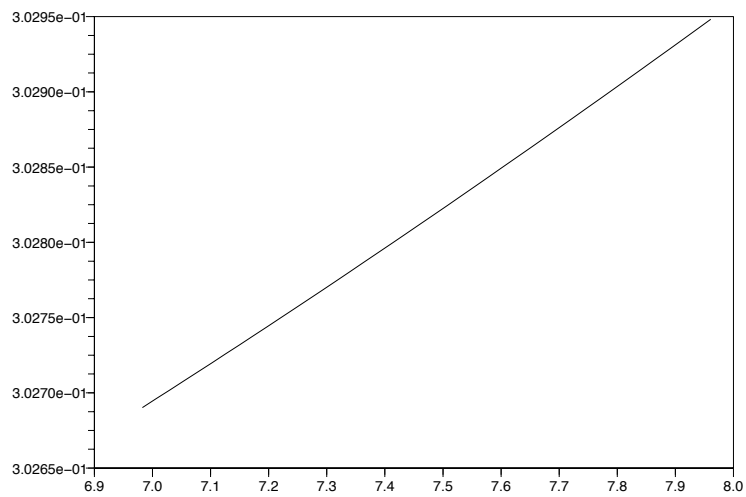


FIG. 4.24 – Fonction f_3 , avec la hauteur en abscisse et la probabilité en ordonnée.

Deuxième partie

Arrêt optimal

Introduction

Dans de nombreux domaines d'application, tels que la santé, le militaire, la maintenance ou la finance, il est nécessaire de savoir décider du moment opportun pour agir ou ne plus agir en fonction de l'objectif à atteindre. Par exemple, il est important de connaître l'instant optimal d'arrêt d'un traitement médical afin de maximiser l'efficacité de celui-ci tout en minimisant les effets secondaires (voir par exemple [70, 5, 71, 67, 44]), ou encore, dans le domaine du nucléaire, de déterminer le temps optimal pour l'enfouissement de déchets nucléaires en minimisant les risques d'éventuelles fuites et les coûts en cas de radiation (voir par exemple [54, 37, 46]). Dans le domaine de la finance, les résultats de Black et Scholes ou de Cox, Ross et Rubinstein permettent de déterminer le temps optimal d'exercice d'une option pour maximiser les gains (voir par exemple [65, 6, 58]).

Un exemple de Problème d'Arrêt Optimal (PAO) classique est le problème de la princesse (voir par exemple [36, 43]), encore appelé problème classique de la secrétaire. Une princesse doit choisir un mari parmi cent prétendants qu'elle auditionne un par un. Elle peut décider soit de choisir le candidat qui se présente devant elle, et dans ce cas elle ne pourra pas voir les suivants, soit de passer au candidat suivant, sans avoir la possibilité de revenir ensuite sur son choix. Si après avoir rencontré le 99-ème prétendant, son choix n'est pas arrêté, elle devra obligatoirement prendre le centième candidat pour époux. Elle doit donc déterminer le temps d'arrêt optimal qui lui permette de choisir le « meilleur mari ».

Les PAO ont également été étudiés dans le cas des modèles de chocs (voir par exemple [67, 80, 11]). Un système subit des chocs aléatoires et il faut déterminer le temps d'arrêt qui permette de réparer/remplacer le système avant que les dommages soient trop importants. On retrouve des applications dans le domaine de la fiabilité et de la maintenance préventive (voir par exemple [73, 72]), mais également dans celui de la santé. Ainsi, la radiation de cellules cancéreuses peut se modéliser par un modèle de chocs (voir par exemple [71, 67]). La radiation pouvant également tuer des cellules normales, il est important de savoir quand arrêter le traitement pour éviter de tuer des cellules saines.

Pour revenir au cas-test présenté dans la Partie I, le PAO à étudier dans le cadre d'une maintenance préventive (sous l'hypothèse que les unités sont réparables) serait de déterminer le temps d'arrêt optimal de remplacement ou de réparation d'une unité défaillante pour maximiser la disponibilité et le rendement du réservoir tout en minimisant les coûts de réparation.

Les PAO ont été étudiés pour les PDMP dans [16, 19, 21, 38, 42, 53], à *horizon infini* (noté PAO_∞) lorsque l'ensemble des \mathcal{F}_t -temps d'arrêt τ , noté \mathcal{M} , vérifie $\mathbb{P}_x(\tau < \infty) = 1$ pour tout $x \in E$, ou à *horizon aléatoire* (noté PAO_n), lorsque l'ensemble des \mathcal{F}_t -temps d'arrêt τ est dominé par le n -ème temps de saut T_n pour $n \in \mathbb{N}$, avec \mathcal{F}_t la filtration naturelle du PDMP (voir Chapitre 1). A partir de la définition qu'en donne M.H.A. Davis, on peut définir le PAO_∞ comme un triplet $(\mathcal{M}_\infty, g, v)$ où

1. $\mathcal{M}_\infty = \{\tau \in \mathcal{M}, \mathbb{P}_x(\tau < \infty) = 1 \text{ pour tout } x \in E\}$,
2. la fonction g , mesurable continue et bornée, détermine le gain $g(x)$ obtenu lorsque le processus est arrêté au point $x \in E$.
3. la fonction valeur v est définie pour tout $x \in E$ par

$$v(x) = \sup_{\tau \in \mathcal{M}_\infty} \mathbb{E}_{x_0}[g(X_\tau)],$$

et le PAO_n comme un triplet (\mathcal{M}_n, g, v) où

1. $\mathcal{M}_n = \{\tau \in \mathcal{M}, \tau \leq T_n \text{ p.s.}\}$,
2. la fonction g , mesurable continue et bornée, détermine le gain $g(x)$ obtenu lorsque le processus est arrêté au point $x \in E$.
3. la fonction valeur v est définie pour tout $x \in E$ par

$$v(x) = \sup_{\tau \in \mathcal{M}_n} \mathbb{E}_{x_0}[g(X_\tau)].$$

Alors, résoudre le PAO_∞ (resp. le PAO_n) revient à déterminer, s'il existe, le temps d'arrêt optimal $\tau^* \in \mathcal{M}_\infty$ (resp. $\tau^* \in \mathcal{M}_n$) tel que $v(x) = \mathbb{E}_{x_0}[g(X_{\tau^*})]$ pour tout $x \in E$.

Remarque 4.11 *Un temps d'arrêt τ^* est dit ϵ -optimal s'il vérifie*

$$\mathbb{E}_{x_0}[g(X_{\tau^*})] \geq v(x) - \epsilon.$$

En général, il n'existe pas de temps d'arrêt optimal d'un PAO, mais il existe toujours des temps d'arrêt ϵ -optimaux du PAO.

Dans [42], l'auteur définit un opérateur relatif au premier temps de saut du processus, et montre que la fonction valeur du PAO à horizon infini est un point fixe pour cet opérateur. L'hypothèse de base dans ce cas est que la fonction gain est continue le long des trajectoires, et il est montré que la fonction valeur a la même propriété. Dans [38, 53], les auteurs posent des hypothèses fortes de régularité pour montrer que la fonction valeur du PAO satisfait des inégalités variationnelles, relatives à des équations intégral-différentielles. Dans [21], M.H.A. Davis suppose que la fonction valeur est bornée et localement lipschitzienne le long des trajectoires pour montrer que les inégalités variationnelles sont nécessaires et suffisantes pour caractériser la fonction valeur du PAO. Dans [19], les auteurs considèrent des hypothèses de régularité plus faibles que [21, 38, 53]. Nous nous intéressons particulièrement aux résultats

présentés par O.L.V. Costa et M.H.A. Davis dans [16], qui sont les seuls à proposer une méthode numérique pour résoudre le PAO pour un PDMP, fondée sur la discrétisation de l'espace d'états similaire à celle développée par H.J. Kushner dans [48]. En particulier, les auteurs de [16] trouvent un résultat de convergence pour le schéma d'approximation, sans toutefois pouvoir calculer une vitesse de convergence.

Notre but est d'utiliser les résultats de [42] et [16] pour les adapter au cas de l'horizon aléatoire, ainsi que des méthodes numériques basées sur la quantification (voir par exemple [3, 4, 41, 60, 61, 63, 62]) et développées dans le cadre des processus de diffusion, pour écrire un schéma de discrétisation adapté aux PDMP. En effet, la structure particulière des PDMP ne permet pas d'appliquer directement les méthodes existantes, notamment celles développées pour les diffusions. Nous proposons donc ici une méthode, prenant en compte les particularités des PDMP, qui nous permet d'obtenir, en plus de la convergence, l'estimation de l'erreur commise et le calcul d'un temps d'arrêt ϵ -optimal du PAO pour les PDMP.

Pour cela, dans le Chapitre 5, nous commencerons par donner un rapide état de l'art des méthodes numériques, en rappelant les résultats de O.L.V. Costa et M.H.A. Davis ([16]) pour les PAO à horizon infini (Section 5.1). Dans la Section 5.2, nous expliquerons la méthode numérique appliquée dans le cadre des processus de diffusion et montrer dans quelles mesures on peut les appliquer au cas spécifique des PDMP.

Dans le Chapitre 6, nous donnerons une méthode numérique, basée sur les résultats de U.S. Gugerli (Section 6.1), permettant de calculer la fonction valeur (Section 6.5) en donnant des résultats de convergence et la vitesse de convergence de notre algorithme (Section 6.7).

Et dans le Chapitre 7, nous calculerons un temps d'arrêt ϵ -optimal du PAO pour les PDMP (Section 7.1), et nous donnerons nos résultats numériques sur un exemple simple (Section 7.2).

Chapitre 5

Etat de l'art des méthodes numériques

Nous allons ici préciser le contexte de notre étude ainsi que les problématiques soulevées par la spécificité des PDMP, en nous intéressant en particulier aux travaux réalisés par O.L.V. Costa et M.H.A. Davis sur les PAO à horizon infini. Nous regarderons ensuite la méthode numérique proposée par, entre autres, V. Bally et G. Pagès, pour les processus de diffusion pour l'adapter au cas spécifique des PDMP.

5.1 Algorithme de Costa-Davis

Dans l'article [16], O.L.V. Costa et M.H.A. Davis étudient le PAO_∞ d'un PDMP

$$v(x) = \inf_{\tau \in \mathcal{M}_\infty} \mathbb{E}[g(X_\tau)],$$

pour une certaine classe de fonctions, afin d'étudier les problèmes de contrôle impulsif ([17, 15]). De plus, ils proposent une construction des temps d'arrêt ϵ -optimaux et prouvent la convergence de la fonction valeur discrétisée pour le problème d'arrêt optimal avec facteur de pénalité $\alpha > 0$

$$v(x) = \inf_{\tau \in \mathcal{M}_\infty} \mathbb{E}[\exp^{-\alpha\tau} g(X_\tau)].$$

En utilisant des résultats de U.S. Gugerli (voir Section 6.1), ils construisent une suite de fonctions (v_m) qui converge vers la fonction valeur v du problème d'arrêt optimal.

La méthode numérique utilisée se fait en deux étapes :

1. la discrétisation de l'espace d'état avec des grilles cartésiennes fixes, identiques pour chaque saut,
2. la discrétisation de la position du processus après saut (S_n, Z_n) en une suite (S_n^N, Z_n^N) sur chaque grille, avec N le nombre de points de la grille.

L'algorithme de résolution consiste alors à effectuer N minimisations à une dimension et obtenir une approximation de la fonction valeur discrétisée v^N pour chaque point de la grille de discrétisation.

O.L.V. Costa et M.H.A. Davis donnent également des résultats de convergence uniforme de la fonction valeur discrétisée vers la fonction valeur du PAO_∞ basée sur la convergence en loi, qui ne permet donc pas d'estimer la vitesse de convergence, et une construction des temps d'arrêt ϵ -optimaux à partir de la fonction valeur discrétisée v^N .

Nous souhaitons pour notre part obtenir une vitesse de convergence. Pour cela, nous allons nous intéresser à la méthode numérique pour les processus de diffusion, basée sur une discrétisation en espace qui, d'une part, est non-régulière, c'est-à-dire adaptée aux lois des variables aléatoires sous-jacentes, et d'autre part est différente à chaque saut.

5.2 Méthode numérique pour les processus de diffusion

Dans cette section, nous allons décrire la méthode numérique développée dans le cas des processus de diffusion, sur la base des résultats de V. Bally et G. Pagès ([3]), V. Bally, G. Pagès et J. Printemps ([4]) et G. Pagès, H. Phàm et J. Printemps ([63]).

5.2.1 Démarche

Les approches numériques pour la résolution des problèmes d'arrêt optimal dans le cadre des diffusions utilisent en général des méthodes analytiques qui sont cependant difficilement applicables pour de grandes dimensions. L'interprétation probabiliste se révèle nécessaire pour surmonter cette difficulté. La méthode probabiliste proposée dans [3, 4, 63] pour étudier le problème d'arrêt optimal pour un processus de diffusion en temps continu $\{Y(t)\}$ se fait en trois temps :

1. la discrétisation en temps du processus de diffusion pour obtenir une chaîne de Markov en temps discret $\{\bar{Y}_k\}$ adaptée à la trajectoire du processus. Il est montré que le problème original d'optimisation en temps continu peut être converti en un problème d'arrêt optimal auxiliaire associé à la chaîne de Markov en temps discret $\{\bar{Y}_k\}$.
2. l'approximation de la chaîne de Markov en temps discret par un processus à espace d'état fini. Il existe de nombreuses références dans la littérature au sujet des méthodes de quantification des variables aléatoires et des processus. Nous n'allons pas présenter ces méthodes de façon exhaustive, mais le lecteur intéressé pourra par exemple consulter les travaux suivants [41, 60, 63] ainsi que leurs références.
3. l'approximation des espérances conditionnelles de la formule de programmation dynamique rétrograde par des sommes finies. En effet, l'enveloppe de Snell en temps discret $\{\bar{U}_k\}$ de la chaîne de Markov discrétisée $\{\bar{Y}_k\}$ satisfait le principe de la programmation dynamique suivant :

$$\begin{aligned}\bar{U}_N &= h(\bar{Y}_N), \\ \bar{U}_k &= \max \left(h(\bar{Y}_k), \mathbb{E}[\bar{U}_{k+1} | \bar{Y}_k] \right) \quad \text{pour } 0 \leq k \leq N-1,\end{aligned}$$

où h est la fonction gain du PAO.

Cette étape consiste donc à remplacer les espérances conditionnelles par des sommes finies, ce qui se fait naturellement en remplaçant le processus à valeurs continues \bar{Y}_k par son approximation à valeurs dans un espace fini.

Alors, sous certaines hypothèses, en particulier que la fonction gain soit Lipschitz, on sait calculer une approximation de la fonction valeur et déterminer la vitesse de convergence de la fonction valeur discrétisée vers la vraie fonction valeur du problème d'arrêt optimal initial.

5.2.2 Quantification

La deuxième étape de la démarche pour les processus de diffusion utilise la quantification. La quantification est issue de la théorie du traitement du signal et de l'information. Elle permet d'approcher une variable aléatoire à valeurs dans un espace continu Y par une variable aléatoire à valeurs dans un ensemble fini \hat{Y} , de telle sorte que $\|Y - \hat{Y}\|_p$ soit minimale en norme L^p .

Ce procédé est utilisé dans de nombreux domaines scientifiques tels que la théorie de l'information (compression de signal) ([39]), l'intégration numérique ([60]), les modèles mathématiques en économie ([7]) ou encore les processus stochastiques ([10]).

Les méthodes de quantification ont été développées récemment dans le cadre des probabilités numériques ([40], des filtrations non linéaires ou du contrôle optimal stochastique avec des applications en finance [3, 4, 60, 61, 62, 63, 66, 69]. Plus spécifiquement, des méthodes intéressantes et efficaces ont été développées dans [3, 4, 63] pour calculer l'enveloppe de Snell associée à des chaînes de Markov à temps discret et des processus de diffusion.

En particulier, nous nous intéressons à une méthode numérique de discrétisation en espace, la *quantification en arbre*, développée par V. Bally et G. Pagès dans [3] pour les processus de diffusion. Le principe est de projeter, pour chaque pas de temps k , $\{\bar{Y}_k\}$, la chaîne de Markov approchée en temps discret du processus de diffusion en temps continu $\{Y(t)\}$, sur une grille $\Gamma_k = \{x_1^k, \dots, x_{N_k}^k\}$ suivant la règle du plus proche voisin. On obtient alors la suite quantifiée $\{\hat{Y}_k\}$ de la chaîne de Markov $\{\bar{Y}_k\}$ définie par

$$\hat{Y}_k = \sum_{1 \leq i \leq N_k} x_i^k \mathbb{1}_{\{\bar{Y}_k \in C_i^k\}},$$

où $(C_i^k)_{1 \leq i \leq N_k}$, appelée *diagramme de Voronoï* ([40, 59]), est définie par

$$C_i^k = \{y ; \|y - x_i^k\| = \min_{j=1, \dots, N_k} \|y - x_j^k\|\}.$$

Alors, V. Bally et G. Pagès assurent que pour un choix spécifique des grilles Γ_k , on obtient une estimation de l'erreur commise en norme L^p en approchant la chaîne de Markov en temps continu $\{Y\}$ par la suite quantifiée $\{\hat{Y}_k\}$ ([Théorème 2.1][63]) :

Théorème 5.1 *Pour $p \geq 1$, supposons que $\mathbb{E}|Y|^{p+\epsilon} < +\infty$ pour tout $\epsilon > 0$. Alors*

$$\min_{1 \leq k \leq N_k} \|Y - \hat{Y}_k\|_p \sim C \times N^{-\frac{1}{d}},$$

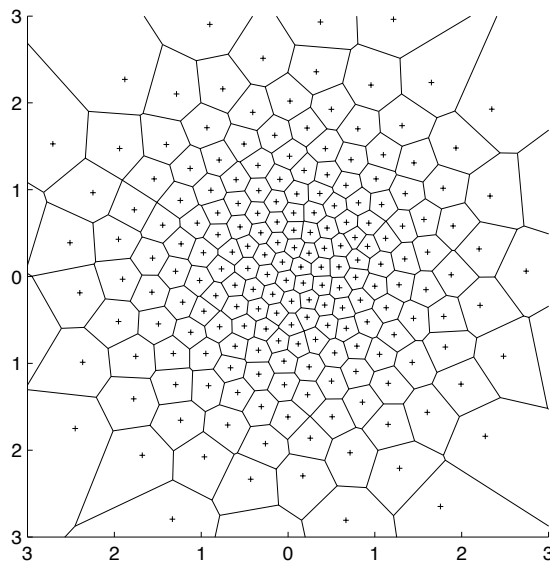


FIG. 5.1 – Diagramme de Voronoï de la loi gaussienne centrée réduite, pour 200 points et 100 000 itérations.

où N_k est la taille de la grille Γ_k , d la dimension de l'espace d'état et C une constante qui dépend de p , d et de la distribution de Y .

Les algorithmes de quantification sont basés sur du Monte Carlo et un simulateur de trajectoires. Les temps de calcul sont donc assez longs. Cependant, l'avantage de la quantification est de pouvoir calculer en avance et stocker les N grilles de discrétisation. L'un des plus utilisés est l'algorithme des gradients stochastiques (voir par exemple [60]), pour lequel cependant aucune preuve de convergence n'a été donnée, hormis pour le cas L^2 .

La quantification de la chaîne de Markov nous donne d'une part une suite de grilles, le poids des cellules de chaque grille, la matrice de transitions d'une grille à l'autre ainsi qu'une estimation de l'erreur de quantification en norme L^p . Il nous semble donc intéressant d'appliquer la démarche pour les processus de diffusion au cas spécifique des PDMP.

5.3 Spécificités des PDMP

Pour le cas des PDMP, il a été montré dans [42] que la fonction valeur du problème d'arrêt optimal peut être calculée en itérant un opérateur fonctionnel L , défini dans la Section 6.1, relatif à la maximisation en temps continu et à la formule de programmation dynamique en temps discret. Ainsi, pour approcher la fonction valeur du problème d'arrêt optimal d'un PDMP $\{X(t)\}$, une approche naturelle serait de suivre la même démarche que dans [3] et [4] et [63], en commençant par

discrétiser le processus en temps. Cependant, leur méthode ne peut être appliquée directement à notre problème pour deux raisons principales relatives à la spécificité des PDMP.

Comme nous l'avons dit, les PDMP sont par nature discontinus à des temps aléatoires. C'est pourquoi, comme le souligne U.S. Gugerli dans [42], il serait problématique de convertir le problème original d'optimisation en un problème d'arrêt optimal associé à une discrétisation en temps de $\{X(t)\}$ avec des propriétés de convergence satisfaisantes. En particulier, il semble inadapté de proposer comme dans [60] un schéma de discrétisation en temps à pas fixe $\{X(k\Delta)\}$ du processus originel $\{X(t)\}$. De plus, un autre problème vient du semi-groupe des transitions $\{P_t\}_{t \in \mathbb{R}^+}$ de $\{X(t)\}$. D'une part, on ne peut le calculer explicitement à partir des caractéristiques locales (ϕ, λ, Q) du PDMP ([18] et [31]). En conséquence, il serait compliqué d'expliciter le noyau markovien P_Δ associé à la chaîne de Markov $\{X(k\Delta)\}$. D'autre part, la chaîne de Markov $\{X(k\Delta)\}$ n'est pas en général une chaîne de Feller ([21, p 76-77]), c'est pourquoi il serait difficile d'assurer qu'elle est K-Lipschitz ([3, Définition 1]).

La deuxième différence fondamentale vient du fait que la fonction apparaissant dans la formule de programmation dynamique rétrograde associée à L (Equation (6.1.1)) et à la fonction gain g n'est pas continue (fonction indicatrice) même si l'on pose des hypothèses fortes de régularité sur g . Ainsi, l'approche développée en [3], [4] et [63] doit être affinée puisque elle ne concerne que des espérances conditionnelles de fonctions Lipschitz.

Cependant, en utilisant la structure spécifique des PDMP, nous arrivons à surmonter ces deux difficultés, comme nous allons le voir dans la suite, sans pour autant surmonter le problème de dimension exposé dans le cas des diffusions. Pour cela, on utilise $\{\Theta_k\} = \{(Z_k, S_k)\}$ la chaîne de Markov en temps discret sous-jacente ainsi qu'une maximisation en temps continu pour expliciter l'opérateur principal L . La démarche que nous allons appliquer consiste dans un premier temps à convertir la maximisation en temps continu de l'opérateur L en une maximisation en temps discret en utilisant un schéma de discrétisation en temps adapté aux trajectoires du processus. Ceci nous permet d'approcher la fonction valeur par la solution d'une équation de programmation dynamique rétrograde en temps discret induisant une espérance conditionnelle de sa chaîne de Markov $\{\Theta_k\}$. Alors une approximation naturelle de ce problème d'optimisation est obtenue en remplaçant $\{\Theta_k\}$ par son approximation quantifiée.

Chapitre 6

Calcul de la fonction valeur

Dans ce chapitre, nous donnerons en détails les résultats théoriques proposés par U.S. Gugerli dans [42] sur les PAO_∞ , et leurs applications dans le cas des PAO_n . Puis, après avoir posé les définitions, hypothèses et notations nécessaires pour la suite, nous donnerons une interprétation probabiliste des opérateurs définis par U.S. Gugerli pour approcher la fonction valeur du PAO_n . Grâce aux propriétés lipschitziennes de ces opérateurs, nous serons alors en mesure de donner une estimation de l'erreur pour la fonction valeur.

6.1 Résultats de U.S. Gugerli

Pour résoudre le PAO_∞ défini par

$$\omega(x) = \sup_{\tau \in \mathcal{M}_\infty} \mathbb{E}_{x_0} [g(X_\tau)],$$

dans le cas des PDMP et contourner le problème posé par leur construction particulière (c'est-à-dire la continuité par morceaux), et donc tenir compte des sauts aléatoires du processus, U.S. Gugerli, dans [42], propose de construire un opérateur, noté L , satisfaisant les caractéristiques du PDMP et associé à la fonction gain g du PAO_∞ . L'itération successive de l'opérateur L détermine une suite croissante (ω_n) de fonctions dominant g . La fonction ω_n au n -ème temps de saut est alors la fonction valeur du PAO_n et sa limite, fonction valeur pour le PAO_∞ , est un point fixe pour l'opérateur L . De plus, sous ces conditions, la fonction coût est continue le long des trajectoires, et la fonction valeur a la même propriété. Nous donnons tout d'abord la définition de l'opérateur L , avant de donner la construction itérative de la fonction valeur pour le PAO_n .

Définition de l'opérateur L Soit E un espace métrique. On note $\mathbb{B}(E)$ l'ensemble des fonctions réelles, bornées et mesurables de E . On rappelle que $\mathcal{B}(E)$ est la tribu borélienne de E . Pour tout $n \in \mathbb{N}$, on rappelle que \mathcal{M}_n est la famille de tous les $\{\mathcal{F}_t\}$ -temps d'arrêt dominés par T_n . Pour tout $n < p$, on pose $\mathcal{M}_{n,p}$ la famille de tous les $\{\mathcal{F}_t\}$ -temps d'arrêt ν satisfaisant $T_n \leq \nu \leq T_p$. On note \mathbb{B}^c l'ensemble de toutes les fonctions w définies sur \overline{E} , bornées et mesurables, à valeurs réelles, et continues le long des trajectoires jusqu'à l'horizon de temps de saut :

Pour tout $x \in E$, $w(\phi(x, \cdot))$ est continue sur $[0, t^*(x)]$.

U.S. Gugerli construit donc l'opérateur $L : \mathbb{B}(E) \times \mathbb{B}^c \rightarrow \mathbb{B}^c$ à partir des opérateurs $H : \mathbb{B}(\bar{E}) \rightarrow \mathbb{B}(E \times \mathbb{R}^+)$, $I : \mathbb{B}(E) \rightarrow \mathbb{B}(E \times \mathbb{R}^+)$, $J : \mathbb{B}(E) \times \mathbb{B}(\bar{E}) \rightarrow \mathbb{B}(E \times \mathbb{R}^+)$ et $K : \mathbb{B}(E) \rightarrow \mathbb{B}(E)$ introduits dans [42, section 2] et de l'horizon de temps de saut s^* défini pour tout $x \in E$ par $t^*(x) \wedge \inf\{t \geq 0, e^{-\Lambda(x,t)} = 0\}$:

$$\begin{aligned} Hf(x, t) &= \mathbb{1}_{[0, s^*(x)]}(t) f(\phi(x, t)) e^{-\Lambda(x, t)} + \mathbb{1}_{[s^*(x), \infty]}(t) f(\phi(x, t^*(x))) e^{-\Lambda(x, t^*(x))}, \\ Iw(x, t) &= \int_0^{t \wedge s^*(x)} \lambda Qw(\phi(x, s)) e^{-\Lambda(x, s)} ds, \\ J(w, f)(x, t) &= Iw(x, t) + Hf(x, t), \\ Kw(x) &= \int_0^{s^*(x)} \lambda Qw(\phi(x, s)) e^{-\Lambda(x, s)} ds + Qw(\phi(x, t^*(x))) e^{-\Lambda(x, t^*(x))}, \\ L(w, h)(x) &= \sup_{t \geq 0} J(w, h)(x, t) \vee Kw(x). \end{aligned} \tag{6.1.1}$$

Remarque 6.1 *L'opérateur K représente en fait la stratégie consistant à ne pas s'arrêter avant le prochain saut, c'est-à-dire qu'il est optimal de ne rien faire avant le saut. L'opérateur J représente la stratégie de s'arrêter avant le saut ou à l'instant de saut t^* . En prenant la borne sup de J , on choisit le meilleur temps d'arrêt sur $[0, t^*]$. Ainsi, l'opérateur L représente la stratégie optimale entre ces deux stratégies.*

Construction itérative de la fonction valeur pour le PAO_n U.S. Gugerli montre alors (Théorème 1) que pour une fonction gain $g \in \mathbb{B}^c$, la fonction valeur ω_n du PAO_n peut être construite récursivement par la procédure suivante sur l'opérateur L : pour tout $x \in E$, $\epsilon > 0$ et $n \in \mathbb{N}^*$,

$$\omega_n(x) = \sup_{\tau \in \mathcal{M}_n} \mathbb{E}_x[g(X(\tau))]$$

avec pour $n \in \mathbb{N}$

$$\begin{cases} \omega_0 = g, \\ \omega_{n+1} = L(\omega_n, g). \end{cases}$$

et on a

$$\omega = \lim_{n \rightarrow \infty} \omega_n.$$

De plus, ce résultat donne une construction d'une famille des temps d'arrêt ϵ -optimaux du PAO_∞ .

Dans notre cas, nous voulons calculer numériquement ω_N , que nous renommons v_0 .

6.2 Le problème d'arrêt optimal à horizon aléatoire

Nous allons utiliser une démarche similaire à celle proposée par U.S. Gugerli pour la résolution numérique du problème d'arrêt optimal par les PDMP. En particulier, nous allons préciser les résultats de U.S. Gugerli, en donnant une autre

interprétation des fonctions ω_n . A partir de maintenant, nous fixons N et nous regardons le problème d'arrêt optimal à horizon le temps de saut T_N (PAO_N). Nous changeons la notation de U.S. Gugerli en récrivant (6.1.2) de façon rétrograde :

$$\begin{cases} v_N = g (= \omega_0), \\ v_n = L(v_{n+1}, g) (= \omega_{N-n}) \quad \text{pour } n \leq N-1. \end{cases}$$

Et nous voulons calculer numériquement v_0 (ω_N avec les notations de U.S. Gugerli).

Définition 6.1 *On définit les variables aléatoires $(V_n)_{n \in \{0, \dots, N\}}$ par*

$$V_n = v_n(Z_n),$$

ou d'une façon équivalente

$$\begin{aligned} V_n &= \sup_{u \leq t^*(Z_n)} \left\{ \mathbb{E} \left[v_{n+1}(Z_{n+1}) \mathbf{1}_{\{S_{n+1} < u\}} + g(\phi(Z_n, u)) \mathbf{1}_{\{S_{n+1} \geq u\}} \middle| Z_n \right] \right\} \\ &\vee \mathbb{E} \left[v_{n+1}(Z_{n+1}) \middle| Z_n \right]. \end{aligned} \quad (6.2.1)$$

Avec cette écriture, nous pouvons obtenir un résultat plus précis sur la construction de la fonction valeur. On sait que

$$v_n = \sup_{\tau \in \mathcal{M}_{N-n}} \mathbb{E}[g(X(\tau))].$$

Alors le théorème suivant montre que la suite $(V_n)_{n \in \{0, \dots, N\}}$ correspond à une *quasi* enveloppe de Snell associée au processus de gain $\{g(X(t))\}_{t \in \mathbb{R}^+}$ où le temps d'horizon est aléatoire et donné par les temps de saut $(T_n)_{n \in \{0, \dots, N\}}$ du processus $\{X(t)\}_{t \in \mathbb{R}^+}$:

Théorème 6.1 *On considère un entier $n < N$. Alors*

$$V_n = \sup_{\nu \in \mathcal{M}_{n,N}} \mathbb{E}_{x_0}[g(X(\nu)) | \mathcal{F}_{T_n}].$$

Preuve : Soit $\nu \in \mathcal{M}_{n,N}$. D'après la Proposition 1.3 et le Corollaire 1.2, il existe $\hat{\nu}: E \times (\mathbb{R}^+ \times E)^n \times \Omega \rightarrow \mathbb{R}^+$ tel que pour tout $(z_0, \gamma) \in E \times (\mathbb{R}^+ \times E)^n$ l'application $\hat{\nu}(z_0, \gamma): \Omega \rightarrow \mathbb{R}_+$ est un $\{\mathcal{F}_t\}_{t \in \mathbb{R}^+}$ -temps d'arrêt satisfaisant $\hat{\nu}(z_0, \gamma) \leq T_{N-n}$, et $\nu = T_n + \hat{\nu}(Z_0, \Gamma_n, \theta_{T_n})$, où $\Gamma_n = (S_1, Z_1, \dots, S_n, Z_n)$ et θ est l'opérateur de décalage. Pour $(z_0, \gamma) \in E \times (\mathbb{R}^+ \times E)^n$, on définit $\mathcal{W}: E \times (\mathbb{R}^+ \times E)^n \rightarrow \mathbb{R}$ par

$$\mathcal{W}(z_0, \gamma) = \mathbb{E}_{z_n}[g(X(\hat{\nu}(z_0, \gamma)))] \leq \sup_{\tau \in \mathcal{M}_{N-n}} \mathbb{E}_{z_n}[g(X(\tau))]$$

où $\gamma = (s_1, z_1, \dots, s_n, z_n)$.

Alors, par la propriété forte de Markov appliquée au processus $\{X(t)\}$, on a

$$\mathbb{E}_{x_0}[g(X(\nu)) | \mathcal{F}_{T_n}] = \mathbb{E}_{x_0}[g(X(T_n + \hat{\nu}(Z_0, \Gamma_n, \theta_{T_n}))) | \mathcal{F}_{T_n}] = \mathcal{W}(Z_0, \Gamma_n).$$

Par conséquent, on a

$$\mathbb{E}_{x_0}[g(X(\nu)) | \mathcal{F}_{T_n}] = \mathcal{W}(Z_0, \Gamma_n) \leq \sup_{\tau \in \mathcal{M}_{N-n}} \mathbb{E}_{z_n}[g(X(\tau))],$$

et ainsi on a

$$\sup_{\nu \in \mathcal{M}_{n,N}} \mathbb{E}_{x_0}[g(X(\nu)) | \mathcal{F}_{T_n}] \leq \sup_{\tau \in \mathcal{M}_{N-n}} \mathbb{E}_{Z_n}[g(X(\tau))]. \quad (6.2.2)$$

Inversement, on considère $\tau \in \mathcal{M}_{N-n}$. Il est facile de montrer que $T_n + \tau \circ \theta_{T_n} \in \mathcal{M}_{n,N}$.

La propriété de Markov du processus $\{X(t)\}$ nous donne encore

$$\mathbb{E}_{Z_n}[g(X(\tau))] = \mathbb{E}_{x_0}[g(X(T_n + \tau \circ \theta_{T_n})) | \mathcal{F}_{T_n}] \leq \sup_{\nu \in \mathcal{M}_{n,N}} \mathbb{E}_{x_0}[g(X(\nu)) | \mathcal{F}_{T_n}].$$

Alors nous obtenons

$$\sup_{\tau \in \mathcal{M}_{N-n}} \mathbb{E}_{Z_n}[g(X(\tau))] \leq \sup_{\nu \in \mathcal{M}_{n,N}} \mathbb{E}_{x_0}[g(X(\nu)) | \mathcal{F}_{T_n}]. \quad (6.2.3)$$

En combinant (6.2.2) et (6.2.3), on a

$$\sup_{\tau \in \mathcal{M}_{N-n}} \mathbb{E}_{Z_n}[g(X(\tau))] = \sup_{\nu \in \mathcal{M}_{n,N}} \mathbb{E}_{x_0}[g(X(\nu)) | \mathcal{F}_{T_n}].$$

Enfin, comme $v_n(x) = \sup_{\tau \in \mathcal{M}_{N-n}} \mathbb{E}_x[g(X(\tau))]$, on a

$$V_n = \sup_{\tau \in \mathcal{M}_{N-n}} \mathbb{E}_{Z_n}[g(X(\tau))],$$

ce qui montre le résultat. \square

Nous allons maintenant poser les définitions et les hypothèses que nous utiliserons par la suite.

6.3 Notations, définitions et hypothèses

Nous allons dans cette partie introduire les notations utilisées tout au long de ce chapitre, et donner les définitions et les hypothèses nécessaires à notre démarche.

Avec les notations du Chapitre 1, on définit un PDMP par ses caractéristiques locales (ϕ, λ, Q) et la chaîne de Markov associée $(\Theta_n)_{n \in \mathbb{N}} = (Z_n, S_n)_{n \in \mathbb{N}}$, correspondant aux positionnements après sauts et au temps entre deux sauts.

Remarque 6.2 *D'après la Remarque (24.9) dans [21], nous pouvons occulter le mode dans les notations, en l'intégrant à la variable euclidienne x , pour une écriture plus simple. C'est ce que nous ferons à partir de maintenant.*

Notation 6.1 *Pour tout $(a, b) \in \mathbb{R}^2$, on a*

$$a \wedge b = \min(a, b) \text{ et } a \vee b = \max(a, b).$$

On note \mathbb{L}^c l'ensemble des fonctions w définies sur \overline{E} , bornées et mesurables, à valeurs réelles et Lipschitz le long des trajectoires :

1. il existe $[w]_1 \in \mathbb{R}^+$ tel que pour tout $(x, y) \in E^2$, $u \in [0, t^*(x) \wedge t^*(y)]$, on a

$$|w(\phi(x, u)) - w(\phi(y, u))| \leq [w]_1 |x - y|,$$

2. il existe $[w]_2 \in \mathbb{R}^+$ tel que pour tout $x \in E$, et $(t, s) \in [0, t^*(x)]^2$, on a

$$|w(\phi(x, t)) - w(\phi(x, s))| \leq [w]_2 |t - s|,$$

3. il existe $[w]_* \in \mathbb{R}^+$ tel que pour tout $(x, y) \in E^2$, on a

$$|w(\phi(x, t^*(x))) - w(\phi(y, t^*(y)))| \leq [w]_* |x - y|.$$

Définition 6.2 Pour toute fonction f dans \mathbb{B}^c , on note C_f sa borne, définie par :

$$C_f = \sup_{x \in E} |f(x)|.$$

Définition 6.3 Pour toute fonction f Lipschitz dans $\mathbb{B}(E)$ ou $\mathbb{B}(\overline{E})$, on note $[f]$ sa constante de Lipschitz, définie par :

$$[f] = \sup_{x \neq y \in E} \frac{|f(x) - f(y)|}{|x - y|}.$$

Remarque 6.3 \mathbb{L}^c est un sous-ensemble de \mathbb{B}^c et toute fonction de \mathbb{L}^c est Lipschitz sur \overline{E} avec $[w] \leq [w]_1$.

Remarque 6.4 Pour alléger l'écriture, nous posons pour tout $x \in \overline{E}$, $\lambda Qw(x) = \lambda(x)Qw(x)$.

Nous posons maintenant les hypothèses nécessaires pour la suite, donnant la régularité Lipschitz des caractéristiques locales du PDMP.

Hypothèse 6.1 L'intensité de saut λ est bornée et il existe $[\lambda]_1 \in \mathbb{R}^+$ tel que pour tout $(x, y) \in E^2$, $u \in [0, t^*(x) \wedge t^*(y)]$,

$$|\lambda(\phi(x, u)) - \lambda(\phi(y, u))| \leq [\lambda]_1 |x - y|.$$

Hypothèse 6.2 Le temps de sortie t^* est borné et Lipschitz sur E .

Hypothèse 6.3 Le noyau markovien Q est Lipschitz dans le sens qu'il existe $[Q] \in \mathbb{R}^+$ tel que pour toute fonction $w \in \mathbb{L}^c$ les deux conditions suivantes sont satisfaites :

1. pour tout $(x, y) \in E^2$, $u \in [0, t^*(x) \wedge t^*(y)]$, on a

$$|Qw(\phi(x, u)) - Qw(\phi(y, u))| \leq [Q] [w]_1 |x - y|,$$

2. pour tout $(x, y) \in E^2$, on a

$$|Qw(\phi(x, t^*(x))) - Qw(\phi(y, t^*(y)))| \leq [Q] [w]_* |x - y|.$$

Hypothèse 6.4 La fonction gain g associée au problème d'arrêt optimal appartient à \mathbb{L}^c .

On suppose de plus que la distribution de $X(0)$ est donnée par δ_{x_0} pour un point fixé $x_0 \in E$.

On considère alors le PAO_N pour un entier fixé N :

$$\sup_{\tau \in \mathcal{M}_N} \mathbb{E}_{x_0} [g(X(\tau))].$$

6.4 Interprétation probabiliste

La première étape de la méthode que nous proposons consiste à interpréter d'un point de vue probabiliste l'opérateur L , pour en particulier faire apparaître des espérances conditionnelles.

Avec l'Hypothèse 6.1, on peut simplifier l'écriture des opérateurs.

Lemme 6.1 *Sous l'Hypothèse 6.1, pour tout x dans E , l'horizon de temps de saut $s^*(x)$ défini par $t^*(x) \wedge \inf\{t \geq 0, e^{-\Lambda(x,t)} = 0\}$ est égal au temps de sortie $t^*(x)$.*

Preuve : D'après l'hypothèse 6.1, l'intensité de saut λ est bornée.

On a alors $\Lambda(x, t) < \infty$ pour tout x, t , et donc $e^{-\Lambda(x,t)} > 0$ pour tout $t \leq t^*(x)$.

On a ainsi $s^*(x) = t^*(x)$. \square

Les opérateurs H , I , J , K et L se récrivent alors :

$$\begin{aligned} Hf(x, t) &= f(\phi(x, t \wedge t^*(x)))e^{-\Lambda(x, t \wedge t^*(x))}, \\ Iw(x, t) &= \int_0^{t \wedge t^*(x)} \lambda Qw(\phi(x, s))e^{-\Lambda(x, s)} ds, \\ J(w, f)(x, t) &= Iw(x, t) + Hf(x, t), \\ Kw(x) &= \int_0^{t^*(x)} \lambda Qw(\phi(x, s))e^{-\Lambda(x, s)} ds + Qw(\phi(x, t^*(x)))e^{-\Lambda(x, t^*(x))}, \\ L(w, h)(x) &= \sup_{t \geq 0} J(w, h)(x, t) \vee Kw(x). \end{aligned}$$

L'interprétation probabiliste de ces opérateurs en fonction de la chaîne sous-jacente $(\Theta_n)_{n \in \mathbb{N}^*}$ où $\Theta_n = (Z_n, S_n)$, est le premier pas de notre démarche d'approximation de la fonction valeur.

Proposition 6.1 *Pour tout $x \in E$, $w \in \mathbb{B}(E)$, $f \in \mathbb{B}(\overline{E})$, $h \in \mathbb{B}^c$ et $t \geq 0$, on a*

$$\begin{aligned} Hf(x, t) &= \mathbb{E}\left[f(\phi(x, t \wedge t^*(x)))\mathbf{1}_{\{S_{n+1} \geq t \wedge t^*(x)\}} \middle| Z_n = x\right], \\ Iw(x, t) &= \mathbb{E}\left[w(Z_{n+1})\mathbf{1}_{\{S_{n+1} < t \wedge t^*(x)\}} \middle| Z_n = x\right], \\ Kw(x) &= \mathbb{E}\left[w(Z_{n+1}) \middle| Z_n = x\right], \\ L(w, h)(x) &= \sup_{u \leq t^*(x)} \left\{ \mathbb{E}\left[w(Z_{n+1})\mathbf{1}_{\{S_{n+1} < u\}} + h(\phi(x, u))\mathbf{1}_{\{S_{n+1} \geq u\}} \middle| Z_n = x\right] \right\} \\ &\quad \vee \mathbb{E}\left[w(Z_{n+1}) \middle| Z_n = x\right]. \end{aligned}$$

Preuve : Par définition, pour tout $f \in \mathbb{B}(\overline{E})$, $x \in E$ et $t \geq 0$,

$$Hf(x, t) = f(\phi(x, t \wedge t^*(x)))e^{-\Lambda(x, t \wedge t^*(x))}.$$

En remarquant que

$$t \wedge t^*(x) = \begin{cases} t & \text{sur } [0, t^*(x)[\\ t^*(x) & \text{sur } [t^*(x), \infty[, \end{cases} \quad (6.4.1)$$

on a

$$\begin{aligned} Hf(x, t) &= \mathbb{1}_{[0, t^*(x)[}(t) f(\phi(x, t)) e^{-\Lambda(x, t)} + \mathbb{1}_{[t^*(x), \infty[}(t) f(\phi(x, t^*(x))) e^{-\Lambda(x, t^*(x))} \\ &= \mathbb{1}_{\{t < t^*(x)\}} \mathbb{E}_x[f(\phi(x, t)) \mathbb{1}_{\{S_1 > t\}}] + \mathbb{1}_{\{t \geq t^*(x)\}} \mathbb{E}_x[f(\phi(x, t^*(x))) \mathbb{1}_{\{S_1 = t^*(x)\}}]. \end{aligned}$$

Avec (6.4.1), on a

$$\begin{aligned} Hf(x, t) &= \mathbb{E}_x[f(\phi(x, t \wedge t^*(x))) \mathbb{1}_{\{S_1 > t \wedge t^*(x)\}}] + \mathbb{E}_x[f(\phi(x, t \wedge t^*(x))) \mathbb{1}_{\{S_1 = t \wedge t^*(x)\}}] \\ &= \mathbb{E}_x[f(\phi(x, t \wedge t^*(x))) \mathbb{1}_{\{S_1 \geq t \wedge t^*(x)\}}] \\ &= f(\phi(x, t \wedge t^*(x))) \mathbb{P}_x(S_1 \geq t \wedge t^*(x)) \\ &= \mathbb{E}[f(\phi(x, t \wedge t^*(x))) \mathbb{1}_{\{S_{n+1} \geq t \wedge t^*(x)\}} | Z_n = x]. \end{aligned}$$

Par définition, pour tout $w \in \mathbb{B}(E)$, $x \in E$ et $t \geq 0$,

$$\begin{aligned} Iw(x, t) &= \int_0^{t \wedge t^*(x)} \lambda Qw(\phi(x, s)) e^{-\Lambda(x, s)} ds \\ &= \int_0^{t^*(x)} \lambda Qw(\phi(x, s)) e^{-\Lambda(x, s)} \mathbb{1}_{\{s < t \wedge t^*(x)\}} ds \\ &\quad + \lambda Qw(\phi(x, t^*(x))) e^{-\Lambda(x, t^*(x))} \mathbb{1}_{\{t^*(x) < t \wedge t^*(x)\}}. \end{aligned}$$

Comme le second terme à droite de l'égalité est nul, on a

$$\begin{aligned} Iw(x, t) &= \mathbb{E}_x[Qw(\phi(x, S_1)) \mathbb{1}_{\{S_1 < t \wedge t^*(x)\}}] = \mathbb{E}_x[w(Z_1) \mathbb{1}_{\{S_1 < t \wedge t^*(x)\}}] \\ &= \mathbb{E}[w(Z_{n+1}) \mathbb{1}_{\{S_{n+1} < t \wedge t^*(x)\}} | Z_n = x]. \end{aligned}$$

L'interprétation probabiliste des opérateurs K et L se déduit directement de leur définition et des preuves précédentes. \square

Ce résultat est un point clé de notre démarche. En effet, on constate que L ne dépend du PDMP qu'à travers la chaîne sous-jacente (Θ_n) . La discrétisation en temps n'est donc pas nécessaire puisque l'on a déjà une chaîne de Markov en temps discret.

6.5 Approximation de la fonction valeur

Pour approcher la suite des fonctions valeurs (V_n) , nous procédons en deux étapes. Premièrement, la maximisation en temps continu de l'opérateur L est convertie en une maximisation en temps discret en utilisant un schéma de discrétisation dépendant de Z_n , c'est-à-dire de la trajectoire du PDMP, ce qui nous donne un nouvel opérateur L^d . Dans un deuxième temps, les espérances conditionnelles de la chaîne de Markov (Θ_k) dans la définition de L^d sont remplacées par les espérances conditionnelles de l'approximation quantifiée $(\hat{\Theta}_k)$ définissant ainsi un opérateur \hat{L}^d .

Nous allons tout d'abord définir les grilles de discrétisation en temps comme suit.

Définition 6.4 Soit $z \in E$ et $\Delta(z) \in]0, t^*(z)[$. On définit $n(z) = \left\lceil \frac{t^*(z)}{\Delta(z)} \right\rceil - 1$, où $\lceil x \rceil$ est la partie entière de x . L'ensemble des points $(t_i)_{i \in \{0, \dots, n(z)\}}$ tel que $t_i = i\Delta(z)$ est noté $G(z)$.

$G(z)$ est la grille associée à l'intervalle de temps $[0, t^*(z)]$.

Remarque 6.5 Il est important de noter que, pour tout $z \in E$, non seulement $t^*(z) \notin G(z)$, mais encore $\max G(z) = t_{n(z)} \leq t^*(z) - \Delta(z)$. Cette propriété est cruciale pour la suite.

Définition 6.5 Soit $w \in \mathbb{B}(E)$ et $z \in E$,

$$\begin{aligned} L^d(w, g)(z) &= \max_{s \in G(z)} \left\{ \mathbb{E} \left[w(Z_1) \mathbf{1}_{\{S_1 < s\}} + g(\phi(y, u)) \mathbf{1}_{\{S_1 \geq u\}} \mid Z_0 = z \right] \right\} \\ &\quad \vee \mathbb{E} \left[w(Z_1) \mid Z_0 = z \right] \end{aligned}$$

Venons-en maintenant à la quantification de (Θ_n) . L'algorithme de quantification nous fournit une grille finie $\Gamma_n^\Theta \subset E \times \mathbb{R}_+$ à chaque temps $0 \leq n \leq N$ ainsi que des poids pour chaque point de la grille (on pourra se reporter par exemple à [3, 60, 63]). Soit $p \geq 1$ tel que Θ_n ait des moments finis au moins jusqu'à l'ordre p , et p_n la projection du plus proche voisin de $E \times \mathbb{R}_+$ sur Γ_n^Θ (pour la distance en norme p , s'il existe plusieurs proches voisins à même distance, on retient celui de plus petit indice). Alors la quantification de Θ_n est définie par

$$\hat{\Theta}_n = (\hat{Z}_n, \hat{S}_n) = p_n(Z_n, S_n).$$

Nous noterons Γ_n^Z la projection de Γ_n^Θ sur E et Γ_n^S la projection de Γ_n^Θ sur \mathbb{R}^+ .

En pratique, on calcule d'abord les grilles de quantification et les poids, et ensuite la grille adaptée discrétisée en temps $G(z)$ pour chaque $z \in \Gamma_n^Z$, pour tout $0 \leq n \leq N - 1$. Alors, il y a seulement un nombre fini de grilles à calculer, et tout comme les grilles de quantification, elles peuvent être calculées à l'avance et stockées dans une bibliothèque.

En utilisant ce qui précède ainsi que la caractérisation des opérateurs donnée par la Proposition 6.1, on en déduit la définition des opérateurs discrétisés.

Définition 6.6 Pour $k \in \{1, \dots, N\}$, $w \in \mathbb{B}(\Gamma_k^Z)$, $z \in \Gamma_{k-1}^Z$, et $s \in \mathbb{R}^+$

$$\begin{aligned} \hat{J}_k(w, g)(z, s) &= \mathbb{E} \left[w(\hat{Z}_k) \mathbf{1}_{\{\hat{S}_k < s\}} + g(\phi(z, s)) \mathbf{1}_{\{\hat{S}_k \geq s\}} \mid \hat{Z}_{k-1} = z \right], \\ \hat{K}_k(w)(z) &= \mathbb{E} \left[w(\hat{Z}_k) \mid \hat{Z}_{k-1} = z \right], \\ \hat{L}_k^d(w, g)(z) &= \max_{s \in G(z)} \left\{ \hat{J}_k(w, g)(z, s) \right\} \vee \hat{K}_k(w)(z). \end{aligned}$$

Notons que $\hat{\Theta}_n$ est une variable aléatoire prenant un nombre fini de valeurs, et donc que les espérances précédentes sont des sommes finies, la probabilité de chaque atome étant donnée par son poids sur la grille de quantification.

Nous pouvons maintenant donner la construction complète de la suite approchant (V_n) .

Définition 6.7 On considère $\widehat{v}_N(z) = g(z)$ où $z \in \Gamma_N^Z$ et pour $k \in \{1, \dots, N\}$, on pose

$$\widehat{v}_{k-1}(z) = \widehat{L}_k^d(\widehat{v}_k, g)(z), \quad (6.5.1)$$

où $z \in \Gamma_{k-1}^Z$

Définition 6.8 L'approximation de V_k est donnée par

$$\widehat{V}_k = \widehat{v}_k(\widehat{Z}_k), \quad (6.5.2)$$

pour $k \in \{0, \dots, N\}$.

6.6 Résultats intermédiaires

Afin d'obtenir une borne de l'erreur commise en approchant la fonction valeur, nous étudions ici les propriétés lipschitziennes des opérateurs J et K , ainsi que de la suite des fonctions valeurs.

6.6.1 Propriétés lipschitziennes de J et K

Dans cette partie, nous allons établir des propriétés lipschitz de J et K en temps et en espace.

Lemme 6.2 Soit $h \in \mathbb{L}^c$. Alors, pour tout $(x, y) \in E^2$ et $(t, u) \in \mathbb{R}_+^2$, on a

$$\begin{aligned} & \left| h(\phi(x, t \wedge t^*(x)))e^{-\Lambda(x, t \wedge t^*(x))} - h(\phi(y, u \wedge t^*(y)))e^{-\Lambda(x, u \wedge t^*(y))} \right| \\ & \leq D_1(h)|x - y| + D_2(h)|t - u|, \end{aligned}$$

où

– si $t < t^*(x)$ et $u < t^*(y)$,

$$D_1(h) = [h]_1 + C_h C_{t^*} [\lambda]_1, \quad D_2(h) = [h]_2 + C_h C_\lambda,$$

– si $t = t^*(x)$ et $u = t^*(y)$,

$$D_1(h) = [h]_* + C_h C_{t^*} [\lambda]_1 + C_h C_\lambda [t^*], \quad D_2(h) = 0,$$

– sinon,

$$D_1(h) = [h]_1 + C_h C_{t^*} [\lambda]_1 + [h]_2 [t^*] + C_h C_\lambda [t^*], \quad D_2(h) = [h]_2 + C_h C_\lambda.$$

Preuve : On note A la différence :

$$A = \left| h(\phi(x, t \wedge t^*(x)))e^{-\Lambda(x, t \wedge t^*(x))} - h(\phi(y, u \wedge t^*(y)))e^{-\Lambda(x, u \wedge t^*(y))} \right|.$$

Supposons ici que $t < t^*(x)$ et $u < t^*(x)$. Alors

$$\begin{aligned} A &= \left| h(\phi(x, t))e^{-\Lambda(x, t)} - h(\phi(y, u))e^{-\Lambda(y, u)} \right| \\ &\leq \underbrace{\left| h(\phi(x, t))e^{-\Lambda(x, t)} - h(\phi(x, u))e^{-\Lambda(x, u)} \right|}_{(1)} + \underbrace{\left| h(\phi(x, u))e^{-\Lambda(x, u)} - h(\phi(y, u))e^{-\Lambda(y, u)} \right|}_{(2)}. \end{aligned}$$

Alors, pour le premier terme à droite de l'inégalité, nous avons :

$$\begin{aligned} (1) &\leq \left| h(\phi(x, t))e^{-\Lambda(x, u)} - h(\phi(x, t))e^{-\Lambda(x, u)} \right| + \left| h(\phi(x, u))e^{-\Lambda(x, t)} - h(\phi(x, u))e^{-\Lambda(x, u)} \right| \\ &\leq \left| e^{-\Lambda(x, t)} \right| \left| h(\phi(x, t)) - h(\phi(x, u)) \right| + \left| h(\phi(x, u)) \right| \left| e^{-\Lambda(x, t)} - e^{-\Lambda(x, u)} \right| \end{aligned}$$

Comme $\Lambda \geq 0$, $\left| e^{-\Lambda(x, t)} \right| \leq 1$. De plus, $h \circ \phi$ est Lipschitz par rapport à sa deuxième variable, de constante de Lipschitz $[h]_2$ et bornée de constante C_h , et $t \mapsto e^{-\Lambda(\cdot, t)}$ est Lipschitz, de constante de Lipschitz C_λ . Donc

$$(1) \leq ([h]_2 + C_h C_\lambda) |t - u|.$$

De la même façon, pour le deuxième terme, nous avons

$$(2) \leq \left| e^{-\Lambda(x, u)} \right| \left| h(\phi(x, u)) - h(\phi(y, u)) \right| + \left| h(\phi(y, u)) \right| \left| e^{-\Lambda(x, u)} - e^{-\Lambda(y, u)} \right|.$$

On a $h \circ \phi$ Lipschitz par rapport à sa première variable, de constante de Lipschitz $[h]_1$. De plus, $x \mapsto e^x$ est Lipschitz de constante 1 sur \mathbb{R}^- . Donc

$$\begin{aligned} \left| e^{-\Lambda(x, u)} - e^{-\Lambda(y, u)} \right| &\leq \left| \Lambda(x, u) - \Lambda(y, u) \right| \\ &\leq [\lambda]_1 C_{t^*} |x - y|. \end{aligned}$$

On a donc

$$(2) \leq ([h]_1 + C_h [\lambda]_1 C_{t^*}) |x - y|,$$

ce qui prouve le résultat pour le premier cas.

Pour le deuxième et le troisième cas, en appliquant le même principe de décomposition que précédemment, et en utilisant les Hypothèses 6.1, 6.2 de la Section 6.3 et le fait que $h \in \mathbb{L}^c$, nous avons le résultat voulu. \square

Lemme 6.3 Soit $w \in \mathbb{B}(E)$. Alors, pour tout $x \in E$, $(t, u) \in \mathbb{R}_+^2$, on a

$$\left| J(w, g)(x, t) - J(w, g)(x, u) \right| \leq (C_w C_\lambda + [g]_2 + C_g C_\lambda) |t - u|.$$

Preuve : Par définition de J , nous avons

$$\begin{aligned} \left| J(w, g)(x, t) - J(w, g)(x, u) \right| &\leq \left| \int_{t \wedge t^*(x)}^{u \wedge t^*(x)} \lambda Qw(\phi(x, s)) e^{-\Lambda(x, s)} ds \right| \\ &\quad + \left| g(\phi(x, t \wedge t^*(x))) e^{-\Lambda(x, t \wedge t^*(x))} - g(\phi(x, u \wedge t^*(x))) e^{-\Lambda(x, u \wedge t^*(x))} \right|. \end{aligned}$$

Avec le Lemme 6.2, en prenant $h = g$, on a pour le deuxième terme à droite de l'inégalité :

$$\begin{aligned} &\left| g(\phi(x, t \wedge t^*(x))) e^{-\Lambda(x, t \wedge t^*(x))} - g(\phi(x, u \wedge t^*(x))) e^{-\Lambda(x, u \wedge t^*(x))} \right| \\ &\leq ([g]_2 + C_g C_\lambda) |t - u|. \end{aligned}$$

Pour le premier terme, en utilisant les Hypothèses 6.1-6.3 de la Section 6.3, nous avons assez facilement que

$$\left| \int_{t \wedge t^*(x)}^{u \wedge t^*(x)} \lambda Qw(\phi(x, s)) e^{-\Lambda(x, s)} ds \right| \leq C_\lambda C_w |t - u|,$$

d'où le résultat. \square

Lemme 6.4 Soit $w \in \mathbb{L}^c$. Alors, pour tout $(x, y) \in E^2$, $t \in \mathbb{R}^+$,

$$\left| J(w, g)(x, t) - J(w, g)(y, t) \right| \leq (C_w E_1 + [w]_1 E_2 + E_3) |x - y|,$$

où

$$\begin{aligned} E_1 &= C_\lambda [t^*] + C_{t^*} [\lambda]_1 (1 + C_{t^*} C_\lambda), \\ E_2 &= C_{t^*} C_\lambda [Q], \\ E_3 &= [g]_1 + [g]_2 [t^*] + C_g \{ C_{t^*} [\lambda]_1 + C_\lambda [t^*] \}. \end{aligned}$$

Preuve : Par définition, nous avons

$$\begin{aligned} &\left| J(w, g)(x, t) - J(w, g)(y, t) \right| \\ &\leq \left| \int_0^{t \wedge t^*(x)} \lambda Qw(\phi(x, s)) e^{-\Lambda(x, s)} ds - \int_0^{t \wedge t^*(y)} \lambda Qw(\phi(y, s)) e^{-\Lambda(y, s)} ds \right| \\ &\quad + \left| g(\phi(x, t \wedge t^*(x))) e^{-\Lambda(x, t \wedge t^*(x))} - g(\phi(y, t \wedge t^*(y))) e^{-\Lambda(y, t \wedge t^*(y))} \right|. \end{aligned}$$

Sans perte de généralité, on peut supposer que $t^*(x) \leq t^*(y)$. On a donc

$$\begin{aligned} &\left| J(w, g)(x, t) - J(w, g)(y, t) \right| \\ &\leq \int_0^{t \wedge t^*(x)} \left| \lambda Qw(\phi(x, s)) e^{-\Lambda(x, s)} - \lambda Qw(\phi(y, s)) e^{-\Lambda(y, s)} \right| ds \end{aligned} \quad (6.6.1)$$

$$+ \int_{t \wedge t^*(y)}^{t \wedge t^*(x)} \left| \lambda Qw(\phi(y, s)) e^{-\Lambda(y, s)} \right| ds \quad (6.6.2)$$

$$+ \left| g(\phi(x, t \wedge t^*(x))) e^{-\Lambda(x, t \wedge t^*(x))} - g(\phi(y, t \wedge t^*(y))) e^{-\Lambda(y, t \wedge t^*(y))} \right|. \quad (6.6.3)$$

Du Lemme 6.2 avec $h = g$, on a pour le terme (6.6.3),

$$(6.6.3) \leq \left([g]_1 + C_g C_{t^*} [\lambda]_1 + [g]_2 [t^*] + C_g C_\lambda [t^*] \right) |x - y| \\ \leq E_3 |x - y|.$$

Pour le terme (6.6.2), comme $|t \wedge t^*(x) - t \wedge t^*(y)| \leq |t^*(x) - t^*(y)|$, nous obtenons

$$(6.6.2) \leq C_\lambda C_w |t \wedge t^*(x) - t \wedge t^*(y)| \\ \leq C_\lambda C_w |t^*(x) - t^*(y)| \\ \leq C_\lambda C_w [t^*] |x - y|.$$

Enfin, pour le terme (6.6.1), on utilise le Lemme 6.2 avec $h = \lambda Qw$:

$$(6.6.1) \leq C_{t^*} \left([\lambda Qw]_1 + C_{\lambda Qw} C_{t^*} [\lambda]_1 \right) |x - y|.$$

Assez facilement, on a

$$[\lambda Qw]_1 = C_\lambda [Q] [w]_1 + C_w [\lambda]_1,$$

et

$$C_{\lambda Qw} = C_\lambda C_w.$$

Donc

$$(6.6.1) \leq C_{t^*} \left(C_\lambda [Q] [w]_1 + C_w [\lambda]_1 + C_\lambda C_w C_{t^*} [\lambda]_1 \right) |x - y|.$$

En réarrangeant les termes, on a le résultat attendu. \square

Lemme 6.5 Soit $w \in \mathbb{L}^c$. Alors, pour tout $(x, y) \in E^2$,

$$|Kw(x) - Kw(y)| \leq (C_w E_4 + [w]_1 E_2 + [w]_* [Q]) |x - y|,$$

où $E_4 = 2C_\lambda [t^*] + C_{t^*} [\lambda]_1 (2 + C_{t^*} C_\lambda)$.

Preuve : Par définition, on a

$$\left| Kw(x) - Kw(y) \right| \\ \leq \left| \int_0^{t^*(x)} \lambda Qw(\phi(x, s)) e^{-\Lambda(x, s)} ds - \int_0^{t^*(y)} \lambda Qw(\phi(y, s)) e^{-\Lambda(y, s)} ds \right| \\ + \left| Qw(\phi(x, t^*(x))) e^{-\Lambda(x, t^*(x))} - Qw(\phi(y, t^*(y))) e^{-\Lambda(y, t^*(y))} \right|.$$

Sans perte de généralité, on suppose comme dans la preuve du Lemme 6.4 que $t^*(x) \leq t^*(y)$. On a donc de la même façon

$$\left| Kw(x) - Kw(y) \right| \\ \leq \int_0^{t^*(x)} |\lambda Qw(\phi(x, s)) e^{-\Lambda(x, s)} - \lambda Qw(\phi(y, s)) e^{-\Lambda(y, s)}| ds \\ + \int_{t^*(x)}^{t^*(y)} |\lambda Qw(\phi(y, s)) e^{-\Lambda(y, s)}| ds \\ + \left| Qw(\phi(x, t^*(x))) e^{-\Lambda(x, t^*(x))} - Qw(\phi(y, t^*(y))) e^{-\Lambda(y, t^*(y))} \right|. \quad (6.6.4)$$

On a déjà calculé les deux premiers termes à droite de l'inégalité dans la preuve du Lemme 6.4. Pour (6.6.4), on utilise le Lemme 6.2 en posant $h = Qw$. Comme $[Qw]_* = [Q][w]_*$ et $C_{Qw} = C_w$, on a le résultat souhaité. \square

6.6.2 Propriétés lipschitziennes des fonctions valeurs

Dans cette partie, nous allons étudier la continuité lipschitzienne de la suite des fonctions valeurs (v_n) . Nous prouvons que sous certaines conditions, v_n appartient à \mathbb{L}^c pour tout $0 \leq n \leq N$. Nous allons de plus calculer la constante de Lipschitz de v_n sur \overline{E} , plus précise dans certains cas que $[v_n]_1$, comme on le verra dans la Remarque 6.6.

Nous commençons par donner des résultats plus détaillés sur l'opérateur J .

Lemme 6.6 *Soit $w \in \mathbb{L}^c$. Alors pour tout $x \in E$ et $(s, t) \in \mathbb{R}_+^2$,*

$$\left| \sup_{u \geq t} J(w, g)(x, u) - \sup_{u \geq s} J(w, g)(x, u) \right| \leq (C_w C_\lambda + [g]_2 + C_g C_\lambda) |t - s|.$$

Preuve : Sans perte de généralité, on peut supposer que $t \leq s$.

Alors on a

$$\begin{aligned} \left| \sup_{u \geq t} J(w, g)(x, u) - \sup_{u \geq s} J(w, g)(x, u) \right| \\ = \sup_{u \geq t} J(w, g)(x, u) - \sup_{u \geq s} J(w, g)(x, u). \end{aligned} \quad (6.6.5)$$

Or il existe $\bar{t} \in [t \wedge t^*(x), t^*(x)]$ tel que $\sup_{u \geq t} J(w, g)(x, u) = J(w, g)(x, \bar{t})$. Ainsi, si

$$\bar{t} \geq s, \text{ on a } \left| \sup_{u \geq t} J(w, g)(x, u) - \sup_{u \geq s} J(w, g)(x, u) \right| = 0.$$

Et si $\bar{t} \in [t \wedge t^*(x), s[$, alors

$$\sup_{u \geq t} J(w, g)(x, u) - \sup_{u \geq s} J(w, g)(x, u) \leq J(w, g)(x, \bar{t}) - J(w, g)(x, s).$$

Du Lemme 6.3, on obtient l'inégalité suivante :

$$\sup_{u \geq t} J(w, g)(x, u) - \sup_{u \geq s} J(w, g)(x, u) \leq (C_w C_\lambda + [g]_2 + C_g C_\lambda) |\bar{t} - s|. \quad (6.6.6)$$

Des équations (6.6.5) et (6.6.6), et comme $|\bar{t} - s| \leq |t - s|$ on a le résultat souhaité. \square

D'une façon similaire, nous avons le résultat suivant :

Lemme 6.7 *Soit $w \in \mathbb{L}^c$. Alors pour tout $(x, y) \in E^2$,*

$$\left| \sup_{t \leq t^*(x)} J(w, g)(x, t) - \sup_{t \leq t^*(y)} J(w, g)(y, t) \right| \leq (C_w E_5 + [w]_1 E_2 + E_6) |x - y|,$$

où $E_5 = E_1 + C_\lambda [t^*]$ et $E_6 = E_3 + ([g]_2 + C_g C_\lambda) [t^*]$.

Preuve : Sans perte de généralité, on suppose $t^*(x) \leq t^*(y)$. Deux cas se présentent.

1. Soit il existe $t_x \in [t \wedge t^*(x), t^*(x)]$ tel que $\sup_{t \leq t^*(x)} J(w, g)(x, t) = J(w, g)(x, t_x)$. Comme $\sup_{t \leq t^*(y)} J(w, g)(y, t) \geq J(w, g)(y, t)$ pour tout t , on a :

$$\left| \sup_{t \leq t^*(x)} J(w, g)(x, t) - \sup_{t \leq t^*(y)} J(w, g)(y, t) \right| \leq \left| J(w, g)(x, t_x) - J(w, g)(y, t) \right|.$$

En particulier,

$$\left| \sup_{t \leq t^*(x)} J(w, g)(x, t) - \sup_{t \leq t^*(y)} J(w, g)(y, t) \right| \leq \left| J(w, g)(x, t_x) - J(w, g)(y, t_x) \right|.$$

Par le Lemme 6.4, on a alors le résultat.

2. Soit on suppose que la borne sup est atteinte sur $[t^*(x), t^*(y)]$. Il existe alors $t_y \in [t^*(x), t^*(y)]$ tel que $\sup_{t^*(x) \leq t \leq t^*(y)} J(w, g)(y, t) = J(w, g)(y, t_y)$. Et on a

$$\sup_{t \leq t^*(x)} J(w, g)(x, t) - \sup_{t^*(x) \leq t < t^*(y)} J(w, g)(y, t) \geq J(w, g)(x, t^*(x)) - J(w, g)(y, t_y).$$

Donc

$$\begin{aligned} & \sup_{t \leq t^*(x)} J(w, g)(x, t) - \sup_{t^*(x) \leq t < t^*(y)} J(w, g)(y, t) \\ & \geq - \left| J(w, g)(x, t^*(x)) - J(w, g)(y, t^*(x)) \right| - \left| J(w, g)(y, t^*(x)) - J(w, g)(y, t_y) \right|. \end{aligned}$$

Avec les Lemmes 6.3 et 6.4, on a

$$\begin{aligned} \sup_{t \leq t^*(x)} J(w, g)(x, t) - \sup_{t^*(x) \leq t < t^*(y)} J(w, g)(y, t) & \geq - (C_w E_1 + [w]_1 E_2 + E_3) |x - y| \\ & \quad - (C_w C_\lambda + [g]_2 + C_g C_\lambda) |t^*(x) - t_y|. \end{aligned}$$

De plus, $|t^*(x) - t_y| \leq |t^*(x) - t^*(y)| \leq [t_*] |x - y|$. Donc, en réarrangeant et en regroupant les constantes,

$$\sup_{t \leq t^*(x)} J(w, g)(x, t) - \sup_{t^*(x) \leq t < t^*(y)} J(w, g)(y, t) \geq - (C_w E_5 + [w]_1 E_2 + E_6) |x - y|,$$

et comme avec le Lemme 6.4 on a aussi

$$\begin{aligned} \sup_{t \leq t^*(x)} J(w, g)(x, t) - \sup_{t^*(x) \leq t < t^*(y)} J(w, g)(y, t) & \leq J(w, g)(x, t_x) - J(w, g)(y, t_x) \\ & \leq (C_w E_5 + [w]_1 E_2 + E_6) |x - y|, \end{aligned}$$

On a le résultat voulu. \square

Nous pouvons maintenant établir les résultats pour (v_n) . Par définition, on a pour tout $0 \leq n \leq N$ que (v_n) est bornée, avec $C_{v_n} = C_g$.

Proposition 6.2 Pour tout $0 \leq n \leq N$, $v_n \in \mathbb{L}^c$, $C_{v_n} = C_g$ et $v_n = \sup \mathbb{E}[g(X)]$,

$$\begin{aligned} [v_n]_1 &\leq e^{C_\lambda C_{t^*}} (2[v_{n+1}]_1 E_2 + C_g E_1 + C_g E_4 + C_g C_{t^*} [\lambda]_1 (1 + C_\lambda C_{t^*})) \\ &\quad + e^{C_\lambda C_{t^*}} \left\{ ([g]_1 + [g]_2 [t^*]) \vee ([v_{n+1}]_* [Q]) \right\}, \end{aligned} \quad (6.6.7)$$

$$[v_n]_2 \leq e^{C_\lambda C_{t^*}} \left\{ C_g C_\lambda (4 + C_\lambda C_{t^*}) + [g]_2 \right\}, \quad (6.6.8)$$

$$[v_n]_* \leq [v_n]_1 + [v_n]_2 [t^*],$$

$$[v_n] \leq [v_{n+1}]_1 E_2 + C_g E_4 + \left\{ E_7 \vee ([v_{n+1}]_* [Q]) \right\},$$

où $E_7 = [g]_1 + 2[t^*]([g]_2 + C_g C_\lambda)$.

Preuve : D'une façon évidente, on a $v_N = g \in \mathbb{L}^c$.

Supposons que v_{n+1} appartienne à \mathbb{L}^c . Alors, en utilisant la propriété de semi-groupe du flot ϕ , on peut montrer que pour tout $x \in E$, $t \in [0, t^*(x)]$, on a ([42, Eq. (8)])

$$v_n(\phi(x, t)) = e^{\Lambda(x, t)} \left\{ \left(\sup_{u \geq t} J(v_{n+1}, g)(x, u) \vee K v_{n+1}(x) \right) - I v_{n+1}(x, t) \right\}. \quad (6.6.9)$$

On remarque que pour $x \in E$, $t \in \mathbb{R}^+$, on a

$$\sup_{u \geq t} J(v_{n+1}, g)(x, u) \vee K v_{n+1}(x) \leq \sup_u J(v_{n+1}, g)(x, u) \vee K v_{n+1}(x) = v_n(x). \quad (6.6.10)$$

Soit $(x, y) \in E^2$ et $t \in [0, t^*(x) \wedge t^*(y)]$. On montre facilement que

$$\left| e^{\Lambda(x, t)} - e^{\Lambda(y, t)} \right| \leq e^{C_\lambda C_{t^*}} [\lambda]_1 C_{t^*} |x - y|, \quad (6.6.11)$$

$$\left| I v_{n+1}(x, t) - I v_{n+1}(y, t) \right| \leq \left(C_{v_{n+1}} E_1 + [v_{n+1}]_1 E_2 \right) |x - y|. \quad (6.6.12)$$

En effet, pour la première inégalité, on a pour $x > 0$

$$\left| e^{\Lambda(x, t)} - e^{\Lambda(y, t)} \right| \leq \left\{ \max_{[\Lambda(x, t), \Lambda(y, t)]} e^x \right\} \left| \Lambda(x, t) - \Lambda(y, t) \right|.$$

Or $\Lambda(x, t) \leq C_\lambda C_{t^*}$ donc $\max_{[\Lambda(x, t), \Lambda(y, t)]} e^x \leq e^{C_\lambda C_{t^*}}$, d'où le résultat.

La deuxième inégalité se trouve en utilisant le Lemme 6.4.

Les équations (6.6.9)-(6.6.12) nous donnent

$$\begin{aligned} \left| v_n(\phi(x, t)) - v_n(\phi(y, t)) \right| &\leq \left| e^{\Lambda(x, t)} \left\{ \sup_{u \geq t} J(v_{n+1}, g)(x, u) \vee K v_{n+1}(x) - I v_{n+1}(x, t) \right\} \right. \\ &\quad \left. - e^{\Lambda(y, t)} \left\{ \sup_{u \geq t} J(v_{n+1}, g)(y, u) \vee K v_{n+1}(y) - I v_{n+1}(y, t) \right\} \right| \end{aligned} \quad (6.6.13)$$

En intercalant des termes, on a

$$\begin{aligned} \left| v_n(\phi(x, t)) - v_n(\phi(y, t)) \right| &\leq \left| e^{\Lambda(x, t)} \left\{ \sup_{u \geq t} J(v_{n+1}, g)(x, u) \vee K v_{n+1}(x) - I v_{n+1}(x, t) \right\} \right| \\ &\quad - e^{\Lambda(y, t)} \left\{ \sup_{u \geq t} J(v_{n+1}, g)(x, u) \vee K v_{n+1}(x) - I v_{n+1}(x, t) \right\} \end{aligned} \quad (6.6.14)$$

$$\begin{aligned} &+ \left| e^{\Lambda(y, t)} \left\{ \sup_{u \geq t} J(v_{n+1}, g)(x, u) \vee K v_{n+1}(x) - I v_{n+1}(x, t) \right\} \right| \\ &\quad - e^{\Lambda(y, t)} \left\{ \sup_{u \geq t} J(v_{n+1}, g)(y, u) \vee K v_{n+1}(y) - I v_{n+1}(xy, t) \right\}. \end{aligned} \quad (6.6.15)$$

Pour la première différence à droite de l'inégalité, on a

$$\begin{aligned} (6.6.14) &\leq e^{C_\lambda C_{t^*}} [\lambda]_1 C_{t^*} \left| \sup_{u \geq t} J(v_{n+1}, g)(x, u) \vee K v_{n+1}(x) - I v_{n+1}(x, t) \right| \\ &\leq e^{C_\lambda C_{t^*}} [\lambda]_1 C_{t^*} \left\{ |v_n(x)| + |I v_{n+1}(x, t)| \right\} |x - y|. \end{aligned}$$

Pour la deuxième différence, on a

$$\begin{aligned} (6.6.15) &\leq e^{\Lambda(y, t)} \left\{ \sup_{u \geq t} |J(v_{n+1}, g)(x, u) - J(v_{n+1}, g)(y, u)| \vee |K v_{n+1}(x) - K v_{n+1}(y)| \right\} \\ &\quad + e^{\Lambda(y, t)} (C_{v_{n+1}} E_1 + [v_{n+1}]_1 E_2) |x - y|. \end{aligned}$$

On a ainsi

$$\begin{aligned} \left| v_n(\phi(x, t)) - v_n(\phi(y, t)) \right| &\leq \left\{ |v_n(x)| + |I v_{n+1}(x, t)| \right\} e^{C_\lambda C_{t^*}} [\lambda]_1 C_{t^*} |x - y| \\ &\quad + e^{\Lambda(y, t)} \left\{ \sup_{u \geq t} |J(v_{n+1}, g)(x, u) - J(v_{n+1}, g)(y, u)| \vee |K v_{n+1}(x) - K v_{n+1}(y)| \right\} \\ &\quad + e^{\Lambda(y, t)} (C_{v_{n+1}} E_1 + [v_{n+1}]_1 E_2) |x - y|. \end{aligned} \quad (6.6.16)$$

Pour $x \in E$, $t \in [0, t^*(x)]$ et $n \in \mathbb{N}$, on note que

$$e^{\Lambda(x, t)} \leq e^{C_\lambda C_{t^*}}, \quad |I v_{n+1}(x, t)| \leq C_\lambda C_{v_{n+1}} C_{t^*}, \text{ et } |v_{n+1}(x)| \leq C_g. \quad (6.6.17)$$

Alors, nous obtenons l'inégalité (6.6.7) en utilisant les équations (6.6.16), (6.6.17), les Lemmes 6.4, 6.6, et le fait que $C_g E_1 + E_3 = C_g E_4 + [g]_1 + [g]_2 [t^*]$.

Maintenant, posons $x \in E$ et $t, s \in [0, t^*(x)]$. D'une façon similaire, on a

$$\left| e^{\Lambda(x, t)} - e^{\Lambda(x, s)} \right| \leq e^{C_\lambda C_{t^*}} C_\lambda |t - s|, \quad (6.6.18)$$

$$\left| I v_{n+1}(x, t) - I v_{n+1}(x, s) \right| \leq C_\lambda C_{v_{n+1}} |t - s|. \quad (6.6.19)$$

En combinant les équations (6.6.9), (6.6.10), (6.6.18) et (6.6.19), on a

$$\begin{aligned} \left| v_n(\phi(x, t)) - v_n(\phi(x, s)) \right| &\leq \left\{ |v_n(x)| + |I v_{n+1}(x, t)| \right\} e^{C_\lambda C_{t^*}} C_\lambda |t - s| \\ &\quad + e^{\Lambda(x, t)} \left\{ \left| \sup_{u \geq t} J(v_{n+1}, g)(x, u) - \sup_{u \geq s} J(v_{n+1}, g)(x, u) \right| + C_\lambda C_{v_{n+1}} |t - s| \right\}. \end{aligned} \quad (6.6.20)$$

Pour finir, l'inégalité (6.6.8) vient des équations (6.6.17), (6.6.20) et du Lemme 6.5.

On a par définition pour tout $w \in \mathbb{L}^c$,

$$|w(\phi(x, t^*(x))) - w(\phi(y, t^*(y)))| \leq [w]_2[t^*]|x - y| + [w]_1|x - y|.$$

Donc on a $[w]_* \leq [w]_1 + [w]_2[t^*]$. En l'appliquant à v_n , on a $[v_n]_* \leq [v_n]_1 + [v_n]_2[t^*]$.

Pour la dernière inégalité, posons $(x, y) \in \overline{E}^2$. Par définition,

$$\begin{aligned} & |v_n(x) - v_n(y)| \\ & \leq \left| \sup_{u \leq t^*(x)} J(v_{n+1}, g)(x, u) - \sup_{u \leq t^*(y)} J(v_{n+1}, g)(y, u) \right| \vee |Kv_{n+1}(x) - Kv_{n+1}(y)|. \end{aligned}$$

Par les Lemmes 6.7 et 6.5, on a

$$\begin{aligned} & |v_n(x) - v_n(y)| \\ & \leq (C_g E_5 + [v_{n+1}]_1 E_2 + E_6)|x - y| \vee (C_g E_4 + [v_{n+1}]_1 E_2 + [v_{n+1}]_*[Q])|x - y|, \end{aligned}$$

et nous pouvons conclure en utilisant le fait que $E_4 = E_5 + C_{t^*}[\lambda]_1$ et en posant $E_7 = [g]_1 + 2[t^*]([g]_2 + C_g C_\lambda)$. \square

Remarque 6.6 On peut remarquer que $[v_n]$ est plus précise que $[v_n]_1$. Si, en plus de nos hypothèses, le flot ϕ est Lipschitz pour toutes ses variables, alors on a $[v_n]_i \leq [v_n][\phi]_i$ pour $i \in \{1, 2, *\}$, ce qui devrait nous donner une meilleure constante, comme nous le verrons dans l'exemple détaillé dans la Section 7.2.

6.7 Estimation de l'erreur pour la fonction valeur

Nous pouvons maintenant établir notre résultat principal, c'est-à-dire la convergence de notre schéma d'approximation et la détermination d'une borne supérieure pour la vitesse de convergence.

Théorème 6.2 Soit $n \in \{0, \dots, N-1\}$, et supposons que les $\Delta(z)$, pour $z \in \Gamma_n^z$, sont choisis tels que

$$\min_{z \in \Gamma_n^z} \{\Delta(z)\} > (C_\lambda)^{-1/2} (3[t^*]\|\widehat{Z}_n - Z_n\|_p + 2\|S_{n+1} - \widehat{S}_{n+1}\|_p)^{1/2}.$$

Alors, l'erreur de discrétisation pour V_n est bornée par :

$$\begin{aligned} \|V_n - \widehat{V}_n\|_p & \leq \|V_{n+1} - \widehat{V}_{n+1}\|_p + \alpha \|\Delta(\widehat{Z}_n)\|_p + \beta_n \|\widehat{Z}_n - Z_n\|_p + 2[v_{n+1}]\|\widehat{Z}_{n+1} - Z_{n+1}\|_p \\ & \quad + \left\{ \gamma \|\widehat{Z}_n - Z_n\|_p + \delta (3[t^*]\|\widehat{Z}_n - Z_n\|_p + 2\|S_{n+1} - \widehat{S}_{n+1}\|_p)^{1/2} \right\} \\ & \quad \vee \left\{ [v_{n+1}]_*[Q]\|\widehat{Z}_n - Z_n\|_p \right\}, \end{aligned}$$

où $\alpha = [g]_2 + 2C_g C_\lambda$, $\beta_n = [v_n] + [v_{n+1}]_1 E_2 + C_g E_4$, $\gamma = 2[g]_1 + [g]_2[t^*]$, $\delta = 4C_g(C_\lambda)^{1/2}$, et E_2 et E_4 définis dans la Section 6.6.

On rappelle que $V_N = g(Z_N)$ et $\widehat{V}_N = g(\widehat{Z}_N)$. Alors $\|V_N - \widehat{V}_N\|_p \leq [g] \|\widehat{Z}_N - Z_N\|_p$. Alors $|V_0 - \widehat{V}_0|$ peut être rendu arbitrairement petit par un choix adéquat des paramètres de discrétisation. Nous pouvons également remarquer que la racine carrée dans la dernière somme de l'erreur est le prix à payer pour l'intégration de fonctions non-continues, c'est-à-dire les fonctions indicatrices intervenant dans la définition de l'opérateur J .

Pour prouver le Théorème 6.2, on décompose la différence $\|V_n - \widehat{V}_n\|_p$ en quatre termes :

$$\|V_n - \widehat{V}_n\|_p \leq \sum_{i=1}^4 \Xi_i,$$

où

$$\begin{aligned} \Xi_1 &= \|v_n(Z_n) - v_n(\widehat{Z}_n)\|_p \\ \Xi_2 &= \|L(v_{n+1}, g)(\widehat{Z}_n) - L^d(v_{n+1}, g)(\widehat{Z}_n)\|_p \\ \Xi_3 &= \|L^d(v_{n+1}, g)(\widehat{Z}_n) - \widehat{L}_{n+1}^d(v_{n+1}, g)(\widehat{Z}_n)\|_p \\ \Xi_4 &= \|\widehat{L}_{n+1}^d(v_{n+1}, g)(\widehat{Z}_n) - \widehat{L}_{n+1}^d(\widehat{v}_{n+1}, g)(\widehat{Z}_n)\|_p. \end{aligned}$$

6.7.1 Premier terme

Le premier terme est assez simple à majorer grâce à la Proposition 6.2.

Lemme 6.8 *Une borne supérieure pour Ξ_1 est*

$$\|v_n(Z_n) - v_n(\widehat{Z}_n)\|_p \leq [v_n] \|Z_n - \widehat{Z}_n\|_p.$$

6.7.2 Deuxième terme

Dans cette partie, nous allons étudier l'erreur commise quand on remplace la borne supérieure de L sur tous les temps plus petits ou égaux à $t^*(z)$ par le maximum de L sur la grille finie $G(z)$.

Lemme 6.9 *Soit $w \in \mathbb{L}^c$. Alors pour tout $z \in E$,*

$$\left| \sup_{t \leq t^*(z)} J(w, g)(z, t) - \max_{s \in G(z)} J(w, g)(z, s) \right| \leq (C_w C_\lambda + [g]_2 + C_g C_\lambda) \Delta(z).$$

Preuve : Clairement, il existe $\bar{t} \in [0, t^*(z)]$ tel que

$$\sup_{t \leq t^*(z)} J(w, g)(z, t) = J(w, g)(z, \bar{t}),$$

et il existe $0 \leq i \leq n(z)$ tel que $\bar{t} \in [t_i, t_{i+1}]$ (avec $t_{n(z)+1} = t^*(z)$). Par conséquent, le Lemme 6.3 nous donne

$$\begin{aligned} 0 &\leq \sup_{t \leq t^*(z)} J(w, g)(z, t) - \max_{s \in G(z)} J(w, g)(z, s) \leq J(w, g)(z, \bar{t}) - J(w, g)(z, t_i) \\ &\leq (C_w C_\lambda + [g]_2 + C_g C_\lambda) |\bar{t} - t_i| \\ &\leq (C_w C_\lambda + [g]_2 + C_g C_\lambda) |t_{i+1} - t_i|, \end{aligned}$$

ce qui prouve le résultat. \square

Ainsi, en revenant sur le second terme, nous obtenons la borne suivante :

Lemme 6.10 *Une borne supérieure pour Ξ_2 est*

$$\|L(v_{n+1}, g)(\widehat{Z}_n) - L^d(v_{n+1}, g)(\widehat{Z}_n)\|_p \leq ([g]_2 + 2C_g C_\lambda) \|\Delta(\widehat{Z}_n)\|_p.$$

Preuve : Par définition, on a

$$L(v_{n+1}, g)(\widehat{Z}_n) = \sup_{t \leq t^*(\widehat{Z}_n)} J(v_{n+1}, g)(\widehat{Z}_n, t) \vee K_{v_{n+1}}(\widehat{Z}_n)$$

et

$$L^d(v_{n+1}, g)(\widehat{Z}_n) = \max_{s \in G(\widehat{Z}_n)} J(v_{n+1}, g)(\widehat{Z}_n, s) \vee K_{v_{n+1}}(\widehat{Z}_n).$$

D'où

$$\begin{aligned} & \|L(v_{n+1}, g)(\widehat{Z}_n) - L^d(v_{n+1}, g)(\widehat{Z}_n)\|_p \\ & \leq \left\| \sup_{t \leq t^*(\widehat{Z}_n)} J(v_{n+1}, g)(\widehat{Z}_n, t) - \max_{s \in G(\widehat{Z}_n)} J(v_{n+1}, g)(\widehat{Z}_n, s) \right\|_p. \end{aligned}$$

Et donc, d'après le Lemme 6.9, on a

$$\begin{aligned} & \|L(v_{n+1}, g)(\widehat{Z}_n) - L^d(v_{n+1}, g)(\widehat{Z}_n)\|_p \\ & \leq (C_{v_{n+1}} C_\lambda + [g]_2 + C_g C_\lambda) \|\Delta(\widehat{Z}_n)\|_p. \end{aligned}$$

Et comme $C_{v_{n+1}} = C_g$ (Proposition 6.2), on a bien le résultat. \square

6.7.3 Troisième terme

Nous arrivons ici à la partie cruciale de notre démonstration : nous avons besoin de comparer des espérances conditionnelles relatives à la vraie chaîne de Markov (Z_n, S_n) et celles relatives à l'approximation quantifiée $(\widehat{Z}_n, \widehat{S}_n)$ de la chaîne de Markov. La principale difficulté vient du fait que certaines fonctions utilisées dans le calcul des espérances sont des fonctions indicatrices et donc en particulier ne sont pas Lipschitz. Nous parvenons cependant à surmonter cette difficulté en prouvant que l'événement pour lequel il y a discontinuité a une probabilité suffisamment faible : c'est le but des deux lemmes suivants.

Lemme 6.11 *Pour tout $n \in \{0, \dots, N\}$ et $0 < \eta < \min_{z \in \Gamma_n^Z} \{\Delta(z)\}$,*

$$\left\| \mathbf{1}_{t^*(Z_n) < t^*(\widehat{Z}_n) - \eta} \right\|_p \leq \frac{[t^*] \|Z_n - \widehat{Z}_n\|_p}{\eta}.$$

Preuve : En appliquant l'inégalité de Chebychev, on a

$$\begin{aligned}\mathbb{E}\left[\mathbb{1}_{\{t^*(Z_n) < t^*(\widehat{Z}_n) - \eta\}}^p\right] &= \mathbb{P}\left(t^*(Z_n) < t^*(\widehat{Z}_n) - \eta\right) \\ &\leq \mathbb{P}\left(|t^*(Z_n) - t^*(\widehat{Z}_n)| > \eta\right) \leq \frac{[t^*]^p \|Z_n - \widehat{Z}_n\|_p^p}{\eta^p},\end{aligned}$$

ce qui montre le résultat. \square

Lemme 6.12 *Pour tout $n \in \{0, \dots, N-1\}$ et $0 < \eta < \min_{z \in \Gamma_n^Z} \{\Delta(z)\}$,*

$$\begin{aligned}\left\| \max_{s \in G(\widehat{Z}_n)} \mathbb{E}[\mathbb{1}_{\{S_{n+1} < s\}} - \mathbb{1}_{\{\widehat{S}_{n+1} < s\}} | \widehat{Z}_n] \right\|_p &\leq \frac{2}{\eta} \|S_{n+1} - \widehat{S}_{n+1}\|_p + C_\lambda \eta \\ &\quad + \frac{2[t^*] \|Z_n - \widehat{Z}_n\|_p}{\eta}.\end{aligned}$$

Preuve : Posons $0 < \eta < \min_{z \in \Gamma_n^Z} \{\Delta(z)\}$. On remarque que la différence des fonctions indicatrices est non nulle si et seulement si S_{n+1} et \widehat{S}_{n+1} sont de part et d'autre de s . Alors, si on note $D_s = |\mathbb{1}_{\{S_{n+1} < s\}} - \mathbb{1}_{\{\widehat{S}_{n+1} < s\}}|$, on a

$$|D_s| = |D_s \mathbb{1}_{\{|S_{n+1} - \widehat{S}_{n+1}| > \frac{\eta}{2}\}} + D_s \mathbb{1}_{\{|S_{n+1} - \widehat{S}_{n+1}| \leq \frac{\eta}{2}\}} [\mathbb{1}_{\{S_{n+1} - s \leq \frac{\eta}{2}\}} + \mathbb{1}_{\{S_{n+1} - s > \frac{\eta}{2}\}}]|.$$

Or, $|D_s| \leq 1$, donc

$$|D_s| = |\mathbb{1}_{\{S_{n+1} < s\}} - \mathbb{1}_{\{\widehat{S}_{n+1} < s\}}| \leq \mathbb{1}_{\{|S_{n+1} - \widehat{S}_{n+1}| > \frac{\eta}{2}\}} + \mathbb{1}_{\{|S_{n+1} - s| \leq \frac{\eta}{2}\}}.$$

Nous avons ainsi

$$\begin{aligned}\left\| \max_{s \in G(\widehat{Z}_n)} \mathbb{E}[\mathbb{1}_{\{S_{n+1} < s\}} - \mathbb{1}_{\{\widehat{S}_{n+1} < s\}} | \widehat{Z}_n] \right\|_p &\leq \left\| \mathbb{1}_{\{|S_{n+1} - \widehat{S}_{n+1}| > \frac{\eta}{2}\}} \right\|_p + \left\| \max_{s \in G(\widehat{Z}_n)} \mathbb{E}[\mathbb{1}_{\{s - \frac{\eta}{2} \leq S_{n+1} \leq s + \frac{\eta}{2}\}} | \widehat{Z}_n] \right\|_p.\end{aligned}\quad (6.7.1)$$

D'une part, l'inégalité de Chebychev nous donne

$$\left\| \mathbb{1}_{\{|S_{n+1} - \widehat{S}_{n+1}| > \frac{\eta}{2}\}} \right\|_p^p = \mathbb{P}(|S_{n+1} - \widehat{S}_{n+1}| > \frac{\eta}{2}) \leq \frac{2^p \|S_{n+1} - \widehat{S}_{n+1}\|_p^p}{\eta^p}.\quad (6.7.2)$$

D'autre part, comme $s \in G(\widehat{Z}_n)$ et par définition de η , on a $s + \eta < t^*(\widehat{Z}_n)$, (Remarque 6.5). Alors on a

$$\begin{aligned}\mathbb{E}[\mathbb{1}_{\{s - \frac{\eta}{2} \leq S_{n+1} \leq s + \frac{\eta}{2}\}} | \widehat{Z}_n] &= \mathbb{E}\left[\mathbb{E}[\mathbb{1}_{\{s - \frac{\eta}{2} \leq S_{n+1} \leq s + \frac{\eta}{2}\}} | Z_n] | \widehat{Z}_n\right] \\ &\leq \mathbb{E}\left[\int_{s - \frac{\eta}{2}}^{s + \frac{\eta}{2}} \lambda(\phi(\widehat{Z}_n, u)) du | \widehat{Z}_n\right] + \mathbb{E}[\mathbb{1}_{\{t^*(Z_n) \leq s + \frac{\eta}{2}\}} | \widehat{Z}_n] \\ &\leq \eta C_\lambda + \mathbb{E}[\mathbb{1}_{\{t^*(Z_n) \leq t^*(\widehat{Z}_n) - \frac{\eta}{2}\}} | \widehat{Z}_n].\end{aligned}\quad (6.7.3)$$

En combinant les Equations (6.7.1)-(6.7.3) et en utilisant le Lemme 6.11, on a le résultat. \square

Nous revenons maintenant aux conséquences qu'entraîne le changement de la chaîne de Markov (Z_n, S_n) par son approximation quantifiée $(\widehat{Z}_n, \widehat{S}_n)$ dans les espérances conditionnelles.

Lemme 6.13 *Soit $w \in \mathbb{L}^c$, alors on a*

$$\begin{aligned} & \left| \mathbb{E}[w(Z_{n+1}) | Z_n = \widehat{Z}_n] - \mathbb{E}[w(\widehat{Z}_{n+1}) | \widehat{Z}_n] \right| \\ & \leq (C_w E_4 + [w]_1 E_2 + [w]_*[Q]) \mathbb{E}[|Z_n - \widehat{Z}_n| | \widehat{Z}_n] + [w] \mathbb{E}[|Z_{n+1} - \widehat{Z}_{n+1}| | \widehat{Z}_n]. \end{aligned}$$

Preuve : Tout d'abord, notons que

$$\begin{aligned} \mathbb{E}[w(Z_{n+1}) | Z_n = \widehat{Z}_n] - \mathbb{E}[w(\widehat{Z}_{n+1}) | \widehat{Z}_n] &= \mathbb{E}[w(Z_{n+1}) | Z_n = \widehat{Z}_n] - \mathbb{E}[w(Z_{n+1}) | \widehat{Z}_n] \\ &\quad + \mathbb{E}[w(Z_{n+1}) | \widehat{Z}_n] - \mathbb{E}[w(\widehat{Z}_{n+1}) | \widehat{Z}_n]. \end{aligned}$$

D'une part, d'après la Remarque 6.3, on a

$$\left| \mathbb{E}[w(Z_{n+1}) | \widehat{Z}_n] - \mathbb{E}[w(\widehat{Z}_{n+1}) | \widehat{Z}_n] \right| \leq [w] \mathbb{E}[|Z_{n+1} - \widehat{Z}_{n+1}| | \widehat{Z}_n].$$

D'autre part, on se rappelle que par construction du processus quantifié, on a $(\widehat{Z}_n, \widehat{S}_n) = p_n(Z_n, S_n)$. Alors nous avons la propriété suivante : $\sigma\{\widehat{Z}_n\} \subset \sigma\{Z_n, S_n\}$. En utilisant la structure particulière du PDMP $\{X(t)\}$, on a $\sigma\{Z_n, S_n\} \subset \mathcal{F}_{T_n}$. Maintenant, en utilisant la propriété de Markov du processus $\{X(t)\}$, il suit

$$\mathbb{E}[w(Z_{n+1}) | \widehat{Z}_n] = \mathbb{E}[\mathbb{E}[w(Z_{n+1}) | \mathcal{F}_{T_n}] | \widehat{Z}_n] = \mathbb{E}[\mathbb{E}[w(Z_{n+1}) | Z_n] | \widehat{Z}_n].$$

L'Equation (6.1) nous donne

$$\begin{aligned} & \mathbb{E}[w(Z_{n+1}) | Z_n = \widehat{Z}_n] - \mathbb{E}[w(Z_{n+1}) | \widehat{Z}_n] \\ &= \mathbb{E}[\mathbb{E}[w(Z_{n+1}) | Z_n = \widehat{Z}_n] - \mathbb{E}[w(Z_{n+1}) | Z_n] | \widehat{Z}_n] \\ &= \mathbb{E}[Kw(\widehat{Z}_n) - Kw(Z_n) | \widehat{Z}_n]. \end{aligned}$$

En utilisant le Lemme 6.5, on a le résultat. \square

Maintenant, en combinant les lemmes précédents, on en déduit le troisième terme de l'erreur.

Lemme 6.14 *Pour tout $0 < \eta < \min_{z \in \Gamma_n^Z} \{\Delta(z)\}$, une borne supérieure pour Ξ_3 est*

$$\begin{aligned} & \|L^d(v_{n+1}, g)(\widehat{Z}_n) - \widehat{L}_{n+1}^d(v_{n+1}, g)(\widehat{Z}_n)\|_p \\ & \leq [v_{n+1}] \|\widehat{Z}_{n+1} - Z_{n+1}\|_p + (C_g E_4 + [v_{n+1}]_1 E_2) \|\widehat{Z}_n - Z_n\|_p \\ & \quad + \left\{ (2[g]_1 + [g]_2[t^*]) \|\widehat{Z}_n - Z_n\|_p + 2C_g \left(\frac{3[t^*]}{\eta} \|\widehat{Z}_n - Z_n\|_p \right. \right. \\ & \quad \left. \left. + C_\lambda \eta + \frac{2\|S_{n+1} - \widehat{S}_{n+1}\|_p}{\eta} \right) \right\} \vee \left\{ [v_{n+1}]_*[Q] \|\widehat{Z}_n - Z_n\|_p \right\}. \end{aligned}$$

Preuve : Pour simplifier les notations, posons

$$\Psi(x, y, t) = v_{n+1}(y)\mathbf{1}_{\{t < s\}} + g(\phi(x, t))\mathbf{1}_{\{t \geq s\}}.$$

De la définition de L^d et \widehat{L}_{n+1}^d , on obtient

$$\begin{aligned} & |L^d(v_{n+1}, g)(\widehat{Z}_n) - \widehat{L}_{n+1}^d(v_{n+1}, g)(\widehat{Z}_n)| \\ & \leq \max_{s \in G(\widehat{Z}_n)} \left| \mathbb{E}[\Psi(Z_n, Z_{n+1}, S_{n+1}) | Z_n = \widehat{Z}_n] - \mathbb{E}[\Psi(\widehat{Z}_n, \widehat{Z}_{n+1}, \widehat{S}_{n+1}) | \widehat{Z}_n] \right| \\ & \vee \left| \mathbb{E}[v_{n+1}(Z_{n+1}) | Z_n = \widehat{Z}_n] - \mathbb{E}[v_{n+1}(\widehat{Z}_{n+1}) | \widehat{Z}_n] \right|. \end{aligned} \quad (6.7.4)$$

D'une part, en utilisant le Lemme 6.13 et le fait que v_{n+1} est dans \mathbb{L}^c (Proposition 6.2), on obtient

$$\begin{aligned} & \left| \mathbb{E}[v_{n+1}(Z_{n+1}) | Z_n = \widehat{Z}_n] - \mathbb{E}[v_{n+1}(\widehat{Z}_{n+1}) | \widehat{Z}_n] \right| \\ & \leq [v_{n+1}] \mathbb{E}[|Z_{n+1} - \widehat{Z}_{n+1}| | \widehat{Z}_n] \\ & \quad + (C_g E_4 + [v_{n+1}]_1 E_2 + [v_{n+1}]_* [Q]) \mathbb{E}[|Z_n - \widehat{Z}_n| | \widehat{Z}_n]. \end{aligned} \quad (6.7.5)$$

D'autre part, des arguments similaires à ceux de la preuve du Lemme 6.13 nous donnent

$$\begin{aligned} & \mathbb{E}[\Psi(Z_n, Z_{n+1}, S_{n+1}) | Z_n = \widehat{Z}_n] - \mathbb{E}[\Psi(\widehat{Z}_n, \widehat{Z}_{n+1}, \widehat{S}_{n+1}) | \widehat{Z}_n] \\ & = \mathbb{E} \left[\mathbb{E}[\Psi(Z_n, Z_{n+1}, S_{n+1}) | Z_n = \widehat{Z}_n] - \mathbb{E}[\Psi(Z_n, Z_{n+1}, S_{n+1}) | Z_n = Z_n] \middle| \widehat{Z}_n \right] \\ & \quad + \mathbb{E}[\Psi(Z_n, Z_{n+1}, S_{n+1}) | \widehat{Z}_n] - \mathbb{E}[\Psi(\widehat{Z}_n, \widehat{Z}_{n+1}, \widehat{S}_{n+1}) | \widehat{Z}_n] \\ & = \Upsilon_1 + \Upsilon_2. \end{aligned} \quad (6.7.6)$$

Pour la seconde différence à droite de l'équation (6.7.6), notée Υ_2 , on a

$$\begin{aligned} |\Upsilon_2| & \leq \mathbb{E}[|\Psi(Z_n, Z_{n+1}, S_{n+1}) - \Psi(\widehat{Z}_n, \widehat{Z}_{n+1}, \widehat{S}_{n+1})| | \widehat{Z}_n] \\ & \leq \mathbb{E}[|v_{n+1}(Z_{n+1})\mathbf{1}_{\{S_{n+1} < s\}} - v_{n+1}(\widehat{Z}_{n+1})\mathbf{1}_{\{\widehat{S}_{n+1} < s\}}| \\ & \quad + |g(\phi(Z_n, S_{n+1}))\mathbf{1}_{\{S_{n+1} \geq s\}} - g(\phi(\widehat{Z}_n, \widehat{S}_{n+1}))\mathbf{1}_{\{\widehat{S}_{n+1} \geq s\}}| | \widehat{Z}_n] \\ & \leq \mathbb{E}[|v_{n+1}(Z_{n+1}) - v_{n+1}(\widehat{Z}_{n+1})| | \widehat{Z}_n] \\ & \quad + \mathbb{E}[|v_{n+1}(\widehat{Z}_{n+1})| |\mathbf{1}_{\{S_{n+1} < s\}} - \mathbf{1}_{\{\widehat{S}_{n+1} < s\}}| | \widehat{Z}_n] \\ & \quad + \mathbb{E}[|g(\phi(Z_n, S_{n+1})) - g(\phi(\widehat{Z}_n, \widehat{S}_{n+1}))| | \widehat{Z}_n] \\ & \quad + \mathbb{E}[|g(\phi(\widehat{Z}_n, \widehat{S}_{n+1}))| |\mathbf{1}_{\{S_{n+1} \geq s\}} - \mathbf{1}_{\{\widehat{S}_{n+1} \geq s\}}| | \widehat{Z}_n]. \end{aligned} \quad (6.7.7)$$

On a ainsi

$$\begin{aligned} |\Upsilon_2| & \leq [v_{n+1}] \mathbb{E}[|\widehat{Z}_{n+1} - Z_{n+1}| | \widehat{Z}_n] + [g]_1 \mathbb{E}[|\widehat{Z}_n - Z_n| | \widehat{Z}_n] \\ & \quad + 2C_g \mathbb{E}[|\mathbf{1}_{\{S_{n+1} < s\}} - \mathbf{1}_{\{\widehat{S}_{n+1} < s\}}| | \widehat{Z}_n]. \end{aligned} \quad (6.7.8)$$

Regardons maintenant la première différence de l'équation (6.7.6), notée Υ_1 . Nous rencontrons une autre difficulté ici. En effet, nous savons par construction que $s < t^*(\widehat{Z}_n)$, mais nous ne savons rien sur les positions relatives de s et $t^*(Z_n)$. Pour l'événement $s \leq t^*(Z_n)$, nous reconnaissons l'opérateur J dans les espérances. Pour l'événement opposé, $s > t^*(Z_n)$, nous pouvons grossièrement borner Ψ par $C_{v_{n+1}} + C_g = 2C_g$. Alors, nous obtenons

$$|\Upsilon_1| \leq \mathbb{E} \left[|J(v_{n+1}, g)(\widehat{Z}_n, s) - J(v_{n+1}, g)(Z_n, s)| \mathbf{1}_{\{s \leq t^*(Z_n)\}} \middle| \widehat{Z}_n \right] \\ + 2C_g \mathbb{E} [\mathbf{1}_{\{t^*(Z_n) < s\}} | \widehat{Z}_n].$$

Le Lemme 6.4 nous donne une borne supérieure pour le premier terme. Comme pour le calcul de la fonction indicatrice, par définition de $G(\widehat{Z}_n)$ et notre choix de η , nous avons $s < t^*(\widehat{Z}_n) - \eta$. Alors on a

$$|\Upsilon_1| \leq (C_g E_1 + [v_{n+1}]_1 E_2 + E_3) \mathbb{E} [|\widehat{Z}_n - Z_n| | \widehat{Z}_n] + 2C_g \mathbb{E} [\mathbf{1}_{\{t^*(Z_n) < t^*(\widehat{Z}_n) - \eta\}} | \widehat{Z}_n]. \quad (6.7.9)$$

Maintenant, en combinant (6.7.4), (6.7.5), (6.7.8) et (6.7.9), et le fait que $C_g E_1 + E_3 = C_g E_4 + [g]_1 + [g]_2 [t^*]$, on obtient

$$|L^d(v_{n+1}, g)(\widehat{Z}_n) - \widehat{L}_{n+1}^d(v_{n+1}, g)(\widehat{Z}_n)| \\ \leq [v_{n+1}] \mathbb{E} [|\widehat{Z}_{n+1} - Z_{n+1}| | \widehat{Z}_n] + \mathbb{E} [|\widehat{Z}_n - Z_n| | \widehat{Z}_n] (C_g E_4 + [v_{n+1}]_1 E_2) \\ + \left\{ 2C_g (\mathbb{E} [\mathbf{1}_{\{t^*(Z_n) < t^*(\widehat{Z}_n) - \eta\}} | \widehat{Z}_n] + \max_{s \in G(\widehat{Z}_n)} \mathbb{E} [|\mathbf{1}_{\{S_{n+1} < s\}} - \mathbf{1}_{\{\widehat{S}_{n+1} < s\}}| | \widehat{Z}_n]) \right. \\ \left. (2[g]_1 + [g]_2 [t^*]) \mathbb{E} [|\widehat{Z}_n - Z_n| | \widehat{Z}_n] \right\} \vee \left\{ [v_{n+1}]_* [Q] \mathbb{E} [|\widehat{Z}_n - Z_n| | \widehat{Z}_n] \right\}.$$

On conclut en prenant la norme L^p de chaque côté et en utilisant les Lemmes 6.12 et 6.11. \square

6.7.4 Quatrième terme

Le dernier terme de l'erreur est une simple comparaison de deux sommes finies.

Lemme 6.15 *Une borne supérieure de Ξ_4 est*

$$\|\widehat{L}_{n+1}^d(v_{n+1}, g)(\widehat{Z}_n) - \widehat{L}_{n+1}^d(\widehat{v}_{n+1}, g)(\widehat{Z}_n)\|_p \leq [v_{n+1}] \|\widehat{Z}_{n+1} - Z_{n+1}\|_p + \|V_{n+1} - \widehat{V}_{n+1}\|_p.$$

Preuve : Par définition de l'opérateur \widehat{L}_n^d , on a

$$\|\widehat{L}_{n+1}^d(v_{n+1}, g)(\widehat{Z}_n) - \widehat{L}_{n+1}^d(\widehat{v}_{n+1}, g)(\widehat{Z}_n)\|_p \\ = \left\| \max_{s \in G(\widehat{Z}_n)} \left\{ \mathbb{E} \left[v_{n+1}(\widehat{Z}_{n+1}) \mathbf{1}_{\{\widehat{S}_{n+1} < s\}} + g(\phi(\widehat{Z}_n, s)) \mathbf{1}_{\{\widehat{S}_{n+1} \geq s\}} \middle| \widehat{Z}_n \right] \right\} \vee \mathbb{E} \left[v_{n+1}(\widehat{Z}_{n+1}) \middle| \widehat{Z}_n \right] \right. \\ \left. - \max_{s \in G(\widehat{Z}_n)} \left\{ \mathbb{E} \left[\widehat{v}_{n+1}(\widehat{Z}_{n+1}) \mathbf{1}_{\{\widehat{S}_{n+1} < s\}} + g(\phi(\widehat{Z}_n, s)) \mathbf{1}_{\{\widehat{S}_{n+1} \geq s\}} \middle| \widehat{Z}_n \right] \right\} \vee \mathbb{E} \left[\widehat{v}_{n+1}(\widehat{Z}_{n+1}) \middle| \widehat{Z}_n \right] \right\|_p \\ \leq \|\mathbb{E} [v_{n+1}(\widehat{Z}_{n+1}) - \widehat{v}_{n+1}(\widehat{Z}_{n+1}) | \widehat{Z}_n]\|_p \\ \leq \|v_{n+1}(\widehat{Z}_{n+1}) - v_{n+1}(Z_{n+1})\|_p + \|v_{n+1}(Z_{n+1}) - \widehat{v}_{n+1}(\widehat{Z}_{n+1})\|_p.$$

On conclut en utilisant que $v_{n+1} \in \mathbb{L}^c$ (Proposition 6.2) et les définitions de V_n et \widehat{V}_n . \square

6.7.5 Preuve du Théorème

On peut maintenant prouver le Théorème 6.2. Les Lemmes 6.8, 6.10, 6.14 et 6.15 des sections précédentes nous donnent, pour tout $0 < \eta < \min_{z \in \Gamma_n^z} \{\Delta(z)\}$,

$$\begin{aligned} \|V_n - \widehat{V}_n\|_p &\leq [v_n] \|\widehat{Z}_n - Z_n\|_p + ([g]_2 + 2C_g C_\lambda) \|\Delta(\widehat{Z}_n)\|_p \\ &\quad + [v_{n+1}] \|\widehat{Z}_{n+1} - Z_{n+1}\|_p + (C_g E_4 + [v_{n+1}]_1 E_2) \|\widehat{Z}_n - Z_n\|_p \\ &\quad + \left\{ (2[g]_1 + [g]_2 [t^*]) \|\widehat{Z}_n - Z_n\|_p + 2C_g \left(\frac{3[t^*]}{\eta} \|\widehat{Z}_n - Z_n\|_p \right. \right. \\ &\quad \left. \left. + C_\lambda \eta + \frac{2\|S_{n+1} - \widehat{S}_{n+1}\|_p}{\eta} \right) \right\} \vee \left\{ [v_{n+1}]_* [Q] \|\widehat{Z}_n - Z_n\|_p \right\} \\ &\quad + [v_{n+1}] \|\widehat{Z}_{n+1} - Z_{n+1}\|_p + \|V_{n+1} - \widehat{V}_{n+1}\|_p. \end{aligned}$$

Le choix optimal pour η satisfait clairement

$$C_\lambda \eta = \frac{1}{\eta} (3[t^*] \|\widehat{Z}_n - Z_n\|_p + 2\|S_{n+1} - \widehat{S}_{n+1}\|_p),$$

à condition qu'il satisfasse aussi la condition $0 < \eta < \min_{z \in \Gamma_n^z} \{\Delta(z)\}$. On a donc

$$\eta = C_\lambda^{1/2} (3[t^*] \|\widehat{Z}_n - Z_n\|_p + 2\|S_{n+1} - \widehat{S}_{n+1}\|_p)^{1/2}.$$

Alors, en réarrangeant les termes précédents, on obtient le résultat :

$$\begin{aligned} \|V_n - \widehat{V}_n\|_p &\leq \|V_{n+1} - \widehat{V}_{n+1}\|_p + ([g]_2 + 2C_g C_\lambda) \|\Delta(\widehat{Z}_n)\|_p \\ &\quad + ([v_n] + [v_{n+1}]_1 E_2 + C_g E_4) \|\widehat{Z}_n - Z_n\|_p + 2[v_{n+1}] \|\widehat{Z}_{n+1} - Z_{n+1}\|_p \\ &\quad + \left\{ (2[g]_1 + [g]_2 [t^*]) \|\widehat{Z}_n - Z_n\|_p + 4C_g (C_\lambda)^{1/2} (3[t^*] \|\widehat{Z}_n - Z_n\|_p \right. \\ &\quad \left. + 2\|S_{n+1} - \widehat{S}_{n+1}\|_p)^{1/2} \right\} \vee ([v_{n+1}]_* [Q] \|\widehat{Z}_n - Z_n\|_p). \end{aligned}$$

\square

Nous avons ainsi un schéma d'approximation de V_n . L'algorithme nous permet de calculer en avance et de stocker des grilles de quantification avec le poids des cellules. De plus, nous obtenons des grilles en temps, que l'on peut également stocker, ou même ne pas conserver en mémoire, puisqu'elles sont de la forme $i\Delta(z)$ (il suffit de connaître $\Delta(z)$ pour avoir tous les pas de la grille). Ainsi, nous pouvons calculer instantanément la fonction valeur, et il n'est pas nécessaire de modifier le schéma de quantification si par exemple, on change la fonction gain du PAO. Dans le chapitre suivant, après avoir expliqué la construction d'un temps d'arrêt ϵ -optimal, nous donnerons un exemple numérique simple pour illustrer nos résultats. Dans [24], on pourra trouver une application industrielle de la démarche exposée ici.

Chapitre 7

Calcul d'un temps d'arrêt ϵ -optimal et Exemple

Dans la pratique, il est plus intéressant de connaître le temps d'arrêt optimal d'un PAO que sa fonction valeur. Dans ce chapitre, nous donnons la construction d'un temps d'arrêt ϵ -optimal pour le PAO_N , puis nous illustrerons ce résultat par un exemple numérique.

7.1 Calcul d'un temps d'arrêt ϵ -optimal

Après avoir expliqué rapidement le principe de notre démarche, nous donnerons une règle d'arrêt pour le PAO_N , et la preuve des théorèmes de convergence.

7.1.1 Temps d'arrêt ϵ -optimal du PAO_N

Dans [42, Théoreme 1], U.S. Gugerli définit un temps d'arrêt ϵ -optimal pour le problème initial. Ce temps d'arrêt ne dépend que de la chaîne de Markov sous-jacente (Θ_n) , et de la fonction valeur optimale. Un candidat naturel pour un temps d'arrêt ϵ -optimal pourrait donc être obtenu en remplaçant la chaîne de Markov (Θ_n) et la fonction valeur optimale par leur *approximation quantifiée*. Cependant, des difficultés apparaissent lorsque l'on veut comparer des termes dépendants de (Θ_n) et ceux de son approximation quantifiée. Il est donc loin d'être évident que cette méthode puisse fournir une règle d'arrêt ϵ -optimale. Néanmoins, en modifiant l'approche de U.S. Gugerli, en utilisant à la fois la vraie chaîne (Z_n, S_n) et sa quantifiée (\hat{Z}_n, \hat{S}_n) , nous sommes capables de proposer une construction numérique d'un temps d'arrêt ϵ -optimal du problème d'arrêt optimal initial.

7.1.2 Règle d'arrêt du PAO approché

Voici comment nous allons procéder. Tout d'abord, rappelons nous que p_n est la projection du plus proche voisin de $E \times \mathbb{R}^+$ sur Γ_n^Θ , et pour tout $(z, s) \in E \times \mathbb{R}^+$, on définit $(\hat{z}_n, \hat{s}_n) = p_n(z, s)$. Notons que \hat{z}_n et \hat{s}_n dépendent tous deux de z et s .

Maintenant, pour $n \in \{1, \dots, N\}$, définissons

$$s_n^*(z, s) = \min \left\{ t \in G(\widehat{z}_{n-1}) \mid \widehat{J}_n(\widehat{v}_n, g)(\widehat{z}_{n-1}, t) = \max_{u \in G(\widehat{z}_{n-1})} \widehat{J}_n(\widehat{v}_n, g)(\widehat{z}_{n-1}, u) \right\}$$

et

$$r_{n,\beta}(z, s) = \begin{cases} t^*(z) & \text{si } \widehat{K}_n \widehat{v}_n(\widehat{z}_{n-1}) > \max_{u \in G(\widehat{z}_{n-1})} \widehat{J}_n(\widehat{v}_n, g)(\widehat{z}_{n-1}, u), \\ s_n^*(z, s) \mathbb{1}_{\{s_n^*(z, s) < t^*(z)\}} + (t^*(z) - \beta) \mathbb{1}_{\{s_n^*(z, s) \geq t^*(z)\}} & \text{sinon.} \end{cases}$$

On peut noter que l'on utilise ici l'horizon réel de temps de saut $t^*(z)$ ainsi que les approximations quantifiées de K , J et (z, s) . Posons

$$\tau_1 = r_{N,\beta}(Z_0, S_0) \wedge T_1,$$

et pour $n \in \{1, \dots, N-1\}$, posons

$$\tau_{n+1} = \begin{cases} r_{N-n,\beta}(Z_0, S_0) & \text{si } T_1 > r_{N-n,\beta}(Z_0, S_0), \\ T_1 + \tau_n \circ \theta_{T_1} & \text{sinon.} \end{cases}$$

Cette construction est « récursive » dans le sens suivant : en partant au temps 0, on pose $r_{N,\beta}(Z_0, S_0)$ un temps d'arrêt « virtuel ». Le processus suit sa trajectoire jusqu'au minimum entre $r_{N,\beta}(Z_0, S_0)$ et le premier temps de saut T_1 . Si le minimum est $r_{N,\beta}(Z_0, S_0)$, le processus s'arrête. Si le minimum est T_1 , on calcule $r_{N-1,\beta}(Z_0, S_0)$ le prochain temps d'arrêt « virtuel » et on relance le processus, jusqu'au prochain temps de saut T_2 ou d'arrêt $r_{N-1,\beta}(Z_0, S_0)$, et ainsi de suite.

Notre règle d'arrêt est alors donnée par τ_N .

Remarque 7.1 Cette procédure est particulièrement intéressante car elle ne requiert pas plus de calculs : nous avons déjà calculé les valeurs de \widehat{K}_n et \widehat{J}_n sur les grilles. Il suffit juste de stocker le point où le maximum de \widehat{J}_n est atteint, ou le résultat de la comparaison de \widehat{K}_n et \widehat{J}_n .

Proposition 7.1 τ_N est un $\{\mathcal{F}_T\}$ -temps d'arrêt.

Preuve : Posons $U_1 = r_{1,\beta}(Z_0, S_0)$ et pour $2 \leq k \leq N$,

$$U_k = r_{k,\beta}(Z_{k-1}, S_{k-1}) \mathbb{1}_{\{r_{k-1,\beta}(Z_{k-2}, S_{k-2}) \geq S_{k-1}\}}.$$

Alors on a clairement que $\tau_N = \sum_{k=1}^N U_k \wedge S_k$ qui est un $\{\mathcal{F}_T\}$ -temps d'arrêt par la Proposition 1.4. □

Remarque 7.2 Ce résultat est très important car on obtient le vrai temps d'arrêt du vrai processus, ce qui n'est pas obtenu dans le cas des processus de diffusion.

Nous allons maintenant montrer que ce temps d'arrêt nous donne une bonne approximation de la fonction valeur V_0 . Pour tout $z \in E$, posons

$$\bar{v}_n(z) = \mathbb{E}[g(X_{\tau_{N-n}}) | Z_n = z],$$

et en accord avec notre notation précédente, pour $n \in \{1, \dots, N-1\}$

$$\bar{V}_n = \bar{v}_n(Z_n).$$

La comparaison entre V_0 et \bar{V}_0 est donnée par les deux théorèmes suivants.

Théorème 7.1 *Posons $n \in \{0, \dots, N-2\}$ et supposons que les paramètres de la discrétisation sont choisis tels qu'il existe $0 < a < 1$ satisfaisant*

$$\beta \frac{2a-1}{3a-2} = (C_\lambda)^{-1/2} \left(\frac{[t^*]}{1-a} \|\hat{Z}_n - Z_n\|_p + 2\|S_{n+1} - \hat{S}_{n+1}\|_p \right)^{1/2} < \min_{z \in \Gamma_n^z} \{\Delta(z)\}.$$

Alors on a

$$\begin{aligned} \|\bar{V}_n - V_n\|_p &\leq \|\bar{V}_{n+1} - V_{n+1}\|_p + \|\hat{V}_{n+1} - V_{n+1}\|_p + \|\hat{V}_n - V_n\|_p \\ &\quad + 2[v_{n+1}] \|Z_{n+1} - \hat{Z}_{n+1}\|_p + a_n \|Z_n - \hat{Z}_n\|_p \\ &\quad + (b_n \|Z_n - \hat{Z}_n\|_p) \vee \left(3[g]_1 \|Z_n - \hat{Z}_n\|_p \right. \\ &\quad \left. + 4C_g(C_\lambda)^{1/2} \left(\frac{[t^*]}{1-a} \|\hat{Z}_n - Z_n\|_p + 2\|S_{n+1} - \hat{S}_{n+1}\|_p \right)^{1/2} \right) \end{aligned}$$

avec $a_n = 2[v_{n+1}]_1 E_2 + 2C_g C_{t^*} [\lambda]_1 (2 + C_{t^*} C_\lambda)$ et $b_n = 4C_g C_\lambda [t^*] + 2[v_{n+1}]_* [Q]$.

Le Théorème 7.1 donne une estimation récursive de l'erreur. Voici le pas d'initialisation :

Théorème 7.2 *On suppose que les paramètres de discrétisation sont choisis tels qu'il existe $0 < a < 1$ satisfaisant*

$$\beta \frac{2a-1}{3a-2} = (C_\lambda)^{-1/2} \left(\frac{[t^*]}{1-a} \|\hat{Z}_{N-1} - Z_{N-1}\|_p + 2\|S_N - \hat{S}_N\|_p \right)^{1/2} < \min_{z \in \Gamma_{N-1}^z} \{\Delta(z)\}.$$

Alors on a

$$\begin{aligned} \|\bar{V}_{N-1} - V_{N-1}\|_p &\leq \|\hat{V}_{N-1} - V_{N-1}\|_p + 2[g] \|Z_N - \hat{Z}_N\|_p + a_{N-1} \|Z_{N-1} - \hat{Z}_{N-1}\|_p \\ &\quad + (b_{N-1} \|Z_{N-1} - \hat{Z}_{N-1}\|_p) \vee \left(3[g]_1 \|Z_{N-1} - \hat{Z}_{N-1}\|_p \right. \\ &\quad \left. + 4C_g(C_\lambda)^{1/2} \left(\frac{[t^*]}{1-a} \|\hat{Z}_{N-1} - Z_{N-1}\|_p + 2\|S_N - \hat{S}_N\|_p \right)^{1/2} \right) \end{aligned}$$

avec $a_{N-1} = 2[g]_1 E_2 + 2C_g C_{t^*} [\lambda]_1 (2 + C_{t^*} C_\lambda)$ et $b_{N-1} = 4C_g C_\lambda [t^*] + 2[g]_* [Q]$.

Comme dans la Section 6.7, il est maintenant clair qu'un choix adéquat des paramètres de discrétisation apporte arbitrairement des erreurs petites si on utilise le temps d'arrêt τ_N .

Nous donnons dans la section suivante la définition des Théorèmes 7.1 et 7.2.

7.1.3 Convergence

Nous donnons ici la preuve des Théorèmes 7.1 et 7.2.

Preuve du Théorème 7.1 : La définition de τ_n et la propriété forte de Markov du processus $\{X(t)\}$ donne

$$\begin{aligned}\bar{v}_n(Z_n) &= \mathbb{E}[g(X_{r_{n+1,\beta}(Z_n, S_n)}) \mathbb{1}_{\{S_{n+1} > r_{n+1,\beta}(Z_n, S_n)\}} | Z_n] + \mathbb{E}[\bar{v}_{n+1}(Z_{n+1}) \mathbb{1}_{\{S_{n+1} \leq r_{n+1,\beta}(Z_n, S_n)\}} | Z_n] \\ &= \mathbb{1}_{\{r_{n+1,\beta}(Z_n, S_n) \geq t^*(Z_n)\}} K \bar{v}_{n+1}(Z_n) + \mathbb{1}_{\{r_{n+1,\beta}(Z_n, S_n) < t^*(Z_n)\}} J(\bar{v}_{n+1}, g)(Z_n, r_{n+1,\beta}(Z_n, S_n)).\end{aligned}$$

Cependant, notre définition de $r_{n,\beta}$ et l'utilisation particulière du paramètre β implique

$$\{r_{n+1,\beta}(Z_n, S_n) \geq t^*(Z_n)\} = \left\{ \hat{K}_{n+1} \hat{v}_{n+1}(\hat{Z}_n) > \max_{s \in G(\hat{Z}_n)} \hat{J}_{n+1}(\hat{v}_{n+1}, g)(\hat{Z}_n, s) \right\}.$$

Donc

$$\begin{aligned}\hat{K}_{n+1} \hat{v}_{n+1}(\hat{Z}_n) \vee \max_{s \in G(\hat{Z}_n)} \hat{J}_{n+1}(\hat{v}_{n+1}, g)(\hat{Z}_n, s) \\ = \mathbb{1}_{\{r_{n+1,\beta}(Z_n, S_n) \geq t^*(Z_n)\}} \hat{K}_{n+1} \hat{v}_{n+1}(\hat{Z}_n) + \mathbb{1}_{\{r_{n+1,\beta}(Z_n, S_n) < t^*(Z_n)\}} \max_{s \in G(\hat{Z}_n)} \hat{J}_{n+1}(\hat{v}_{n+1}, g)(\hat{Z}_n, s).\end{aligned}$$

Ainsi, nous obtenons

$$\begin{aligned}\bar{v}_n(Z_n) &= \hat{K}_{n+1} \hat{v}_{n+1}(\hat{Z}_n) \vee \max_{s \in G(\hat{Z}_n)} \hat{J}_{n+1}(\hat{v}_{n+1}, g)(\hat{Z}_n, s) \\ &\quad + \mathbb{1}_{\{r_{n+1,\beta}(Z_n, S_n) \geq t^*(Z_n)\}} \left[K \bar{v}_{n+1}(Z_n) - \hat{K}_{n+1} \hat{v}_{n+1}(\hat{Z}_n) \right] \\ &\quad + \mathbb{1}_{\{r_{n+1,\beta}(Z_n, S_n) < t^*(Z_n)\}} \left[J(\bar{v}_{n+1}, g)(Z_n, r_{n+1,\beta}(Z_n, S_n)) - \max_{s \in G(\hat{Z}_n)} \hat{J}_{n+1}(\hat{v}_{n+1}, g)(\hat{Z}_n, s) \right].\end{aligned} \tag{7.1.1}$$

Etudions le terme avec l'opérateur K . Tout d'abord, nous insérons le terme V_n pour pouvoir utiliser le travail effectué dans la section précédente (nous ne pouvons directement l'appliquer à \bar{v}_n car on ne sait pas s'il est Lipschitz). On a donc clairement

$$\begin{aligned}\left| K \bar{v}_{n+1}(Z_n) - \hat{K}_{n+1} \hat{v}_{n+1}(\hat{Z}_n) \right| &\leq \mathbb{E}[|\bar{V}_{n+1} - V_{n+1}| | Z_n] \\ &\quad + \left| K v_{n+1}(Z_n) - \hat{K}_{n+1} \hat{v}_{n+1}(\hat{Z}_n) \right|. \end{aligned} \tag{7.1.2}$$

Des calculs similaires à ceux des Lemmes 6.5, 6.13 et 6.15, et l'Equation (6.7.5) donnent

$$\begin{aligned}\left| K v_{n+1}(Z_n) - \hat{K}_{n+1} \hat{v}_{n+1}(\hat{Z}_n) \right| \\ \leq (C_g E_4 + [v_{n+1}]_1 E_2 + [v_{n+1}]_* [Q]) \left(|Z_n - \hat{Z}_n| + \mathbb{E}[|Z_n - \hat{Z}_n| | \hat{Z}_n] \right) \\ + 2[v_{n+1}] \mathbb{E}[|Z_{n+1} - \hat{Z}_{n+1}| | \hat{Z}_n] + \mathbb{E}[|V_{n+1} - \hat{V}_{n+1}| | \hat{Z}_n].\end{aligned} \tag{7.1.3}$$

Nous regardons maintenant le terme avec l'opérateur J . Posons $R_n = r_{n+1,\beta}(Z_n, S_n)$. Nous étudions tout d'abord le cas où $R_n = s_{n+1}^*(Z_n, S_n) < t^*(Z_n)$. Par définition, on a

$$\widehat{J}_{n+1}(\widehat{v}_{n+1}, g)(\widehat{Z}_n, R_n) = \max_{s \in G(\widehat{Z}_n)} \widehat{J}_{n+1}(\widehat{v}_{n+1}, g)(\widehat{Z}_n, s).$$

Comme précédemment, nous insérons V_n et nous obtenons

$$\begin{aligned} & \left| \left\{ J(\bar{v}_{n+1}, g)(Z_n, R_n) - \max_{s \in G(\widehat{Z}_n)} \widehat{J}_{n+1}(\widehat{v}_{n+1}, g)(\widehat{Z}_n, s) \right\} \mathbb{1}_{\{R_n = s_{n+1}^*(Z_n, S_n)\}} \right| \\ & \leq \mathbb{E}[|\bar{V}_{n+1} - V_{n+1}| | Z_n] \mathbb{1}_{\{R_n = s_{n+1}^*(Z_n, S_n)\}} \\ & \quad + \left| J(v_{n+1}, g)(Z_n, R_n) - \widehat{J}_{n+1}(\widehat{v}_{n+1}, g)(\widehat{Z}_n, R_n) \right| \mathbb{1}_{\{R_n = s_{n+1}^*(Z_n, S_n)\}}. \end{aligned} \quad (7.1.4)$$

Encore une fois, des arguments similaires à ceux des Lemmes 6.4, 6.11 et 6.15, et les Equations (6.7.6), (6.7.8) et (6.7.9) donnent, sur $\{R_n = s_{n+1}^*(Z_n, S_n)\}$

$$\begin{aligned} & \left| J(v_{n+1}, g)(Z_n, R_n) - \widehat{J}_{n+1}(\widehat{v}_{n+1}, g)(\widehat{Z}_n, R_n) \right| \\ & \leq ([v_{n+1}]_1 E_2 + [g]_1 + C_g C_{t^*} [\lambda]_1 (2 + C_{t^*} C_\lambda)) \left(|Z_n - \widehat{Z}_n| + \mathbb{E}[|Z_n - \widehat{Z}_n| | \widehat{Z}_n] \right) \\ & \quad + 2[v_{n+1}] \mathbb{E}[|Z_{n+1} - \widehat{Z}_{n+1}| | \widehat{Z}_n] + \mathbb{E}[|V_{n+1} - \widehat{V}_{n+1}| | \widehat{Z}_n] \\ & \quad + [g]_1 \mathbb{E}[|Z_n - \widehat{Z}_n| | \widehat{Z}_n] + 2C_g \mathbb{E}[\mathbb{1}_{\{S_{n+1} < R_n\}} - \mathbb{1}_{\{\widehat{S}_{n+1} < R_n\}} | \widehat{Z}_n]. \end{aligned} \quad (7.1.5)$$

On remarque que toutes les constantes avec un facteur $[t^*]$ sont supprimées car nous savons ici que $R_n < t^*(Z_n)$ et $R_n < t^*(\widehat{Z}_n)$ sur $\{R_n = s_{n+1}^*(Z_n, S_n)\}$.

Enfin, sur $\{s^*(Z_n) \geq t^*(Z_n) = R_n + \beta\}$, par construction de la grille $G(\widehat{Z}_n)$ (Remarque 6.5), on a pour tout $0 < \eta < \min_{z \in \Gamma_n^Z} \{\Delta(z)\}$,

$$R_n = t^*(Z_n) - \beta < s^*(Z_n) < t^*(\widehat{Z}_n) - \eta.$$

Par conséquent, en utilisant la borne la plus grossière,

$$\left| J(\bar{v}_{n+1}, g)(Z_n, R_n) \right| + \left| \max_{s \in G(\widehat{Z}_n)} \widehat{J}_{n+1}(\widehat{v}_{n+1}, g)(\widehat{Z}_n, s) \right| \leq 2C_g,$$

on obtient

$$\begin{aligned} & \left| J(\bar{v}_{n+1}, g)(Z_n, r_{n+1,\beta}(Z_n, S_n)) - \max_{s \in G(\widehat{Z}_n)} \widehat{J}_{n+1}(\widehat{v}_{n+1}, g)(\widehat{Z}_n, s) \right| \mathbb{1}_{\{r_{n+1,\beta}(Z_n, S_n) = t^*(Z_n) - \beta\}} \\ & \leq 2C_g \left| \mathbb{1}_{\{t^*(Z_n) - \beta < t^*(\widehat{Z}_n) - \eta\}} \right|. \end{aligned} \quad (7.1.6)$$

Maintenant, en combinant les Equations (7.1.1)-(7.1.6) et les Lemmes 6.12 et 6.11,

on obtient pour tout $\beta < \eta < \min_{z \in \Gamma_n^Z} \{\Delta(z)\}$

$$\begin{aligned} \|\bar{V}_n - \hat{V}_n\|_p &\leq \|\bar{V}_{n+1} - V_{n+1}\|_p + \|V_{n+1} - \hat{V}_{n+1}\|_p + 2[v_{n+1}] \|Z_{n+1} - \hat{Z}_{n+1}\|_p \\ &\quad + \|Z_n - \hat{Z}_n\|_p \left(2[v_{n+1}]_1 E_2 + 2C_g C_{t^*} [\lambda]_1 (2 + C_{t^*} C_\lambda) \right) \\ &\quad + \left\{ (4C_g C_\lambda [t^*] + 2[v_{n+1}]_* [Q]) \|Z_n - \hat{Z}_n\|_p \right\} \\ &\quad + \vee \left\{ 3[g]_1 \|Z_n - \hat{Z}_n\|_p + 2C_g (C_\lambda \eta + \frac{2}{\eta} \|S_{n+1} - \hat{S}_{n+1}\|_p \right. \\ &\quad \left. + \frac{3\eta - 2\beta}{\eta(\eta - \beta)} [t^*] \|Z_n - \hat{Z}_n\|_p \right\}. \end{aligned}$$

Supposons maintenant qu'il existe $0 < a < 1$ tel que $\eta = \beta \frac{2a - 1}{3a - 2}$. Alors, le choix optimal pour η satisfait

$$C_\lambda \eta = \frac{1}{\eta} \left(\frac{[t^*]}{1 - a} \|\hat{Z}_n - Z_n\|_p + 2\|S_{n+1} - \hat{S}_{n+1}\|_p \right),$$

et cela satisfait également la condition $0 < \eta < \min_{z \in \Gamma_n^Z} \{\Delta(z)\}$, d'où le résultat. \square

Preuve du Théorème 7.2 : Comme précédemment, la propriété forte de Markov du processus $\{X(t)\}$ donne

$$\begin{aligned} \bar{v}_{N-1}(Z_{N-1}) &= \mathbb{E}[g(X_{r_{N,\beta}(Z_{N-1}, S_{N-1})}) \mathbb{1}_{\{S_N > r_{N,\beta}(Z_{N-1}, S_{N-1})\}} | Z_{N-1}] \\ &\quad + \mathbb{E}[g(Z_N) \mathbb{1}_{\{S_N \leq r_{N,\beta}(Z_{N-1}, S_{N-1})\}} | Z_{N-1}] \\ &= \mathbb{1}_{\{r_{N,\beta}(Z_{N-1}, S_{N-1}) \geq t^*(Z_{N-1})\}} K g(Z_{N-1}) \\ &\quad + \mathbb{1}_{\{r_{N,\beta}(Z_{N-1}, S_{N-1}) < t^*(Z_{N-1})\}} J(g, g)(Z_{N-1}, r_{N,\beta}(Z_{N-1}, S_{N-1})). \end{aligned}$$

Le reste de la preuve est similaire à celle du Théorème précédent. \square

7.2 Exemple et résultats numériques

Nous rappelons ici l'exemple détaillé dans la Section 1.3.2 pour appliquer notre méthode numérique et nous présentons ensuite les résultats numériques.

- On pose $E = [0, 1[$ et $\partial E = \{1\}$.
- Le flot est défini sur $[0, 1]$ par

$$\phi(x, t) = x + ct$$

avec c une constante positive.

- L'intensité de saut λ est définie sur $[0, 1]$ par

$$\lambda(x) = \beta x^\alpha$$

avec $\beta > 0$ et $\alpha \geq 1$.

- On définit pour tout $x \in [0, 1]$ $Q(x, \cdot)$ comme la loi uniforme sur $[0, 1/2]$.

Alors le processus avance à vitesse constante c vers 1, mais plus il s'approche de la frontière 1, plus la probabilité de revenir sur $[0, 1/2]$ est grande. La Figure 7.1 montre deux trajectoires de ce processus pour $x_0 = 0$, $c = \alpha = 1$ et $\beta = 3$ jusqu'au 10-ème saut.

La fonction gain g est définie sur $[0, 1]$ par $g(x) = x$. Nos hypothèses sont donc clairement vérifiées, et nous sommes même dans le cas spécifique où le flot est Lipschitz (Remarque 6.6). Toutes les constantes des Théorèmes 6.2 et 7.1 peuvent se calculer explicitement.

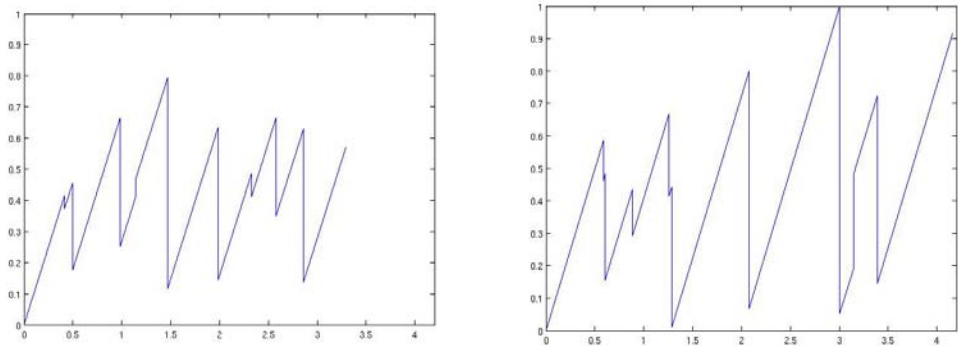


FIG. 7.1 – Deux trajectoires du PDMP, avec le temps en abscisse.

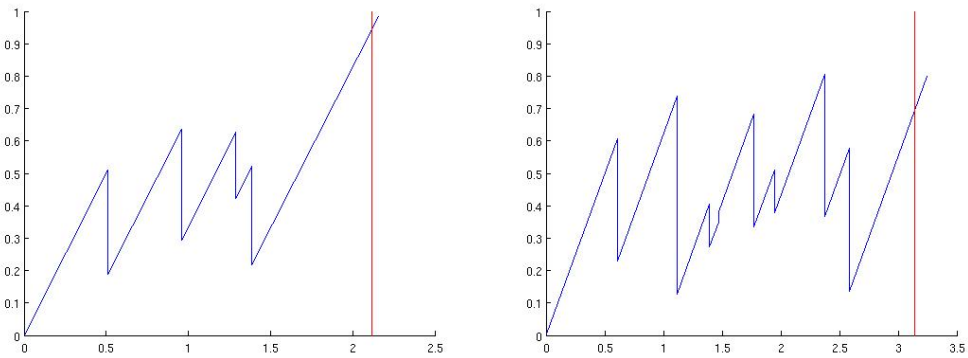


FIG. 7.2 – Deux trajectoires stoppées du PDMP, avec le temps en abscisse.

La fonction valeur $V_0 = v_0(x_0)$ n'est pas connue, mais, comme notre règle d'arrêt τ_N est un temps d'arrêt dominé par T_N , il est clair que l'on a

$$\bar{V}_0 = \mathbf{E}_{x_0}[g(X(\tau_N))] \leq V_0 = \sup_{\tau \in \mathcal{M}_N} \mathbf{E}_{x_0}[g(X(\tau))] \leq \mathbf{E}_{x_0}\left[\sup_{0 \leq t \leq T_N} g(X(t))\right]. \quad (7.2.1)$$

Le premier et le dernier terme peuvent être évalués par la méthode de Monte Carlo, ce qui nous donne un autre indicateur de la précision de notre méthode numérique. Pour 10^6 simulations, on obtient

$$\mathbb{E}_{x_0} \left[\sup_{0 \leq t \leq T_N} g(X(t)) \right] = 0.9878.$$

Les résultats de la simulation (pour $d = 2$, $x_0 = 0$, $c = \alpha = 1$, $\beta = 3$, jusqu'au 10-ème saut et pour 10^5 simulations) sont donnés dans le Tableau 7.1. La Figure 7.2 donne des exemples de trajectoires stoppées. On peut constater que l'ordre de grandeur des erreurs théoriques est très important. Ce résultat était prévisible, car ces bornes sont valables pour une très grande classe de PDMP. Notons cependant que, comme espéré, elles diminuent quand l'erreur de quantification décroît. De l'équation (7.2.1), il vient

$$V_0 - \bar{V}_0 \leq \mathbb{E}_{x_0} \left[\sup_{0 \leq t \leq T_N} g(X(t)) \right] - \bar{V}_0,$$

ce qui nous permet d'avoir une borne supérieure empirique de l'erreur.

Pt	QE	Δ	\hat{V}_0	\bar{V}_0	B_1	B_2	B_3
10	0.0943	0.151	0.7760	0.8173	0.1705	74.64	897.0
50	0.0418	0.100	0.8298	0.8785	0.1093	43.36	511.5
100	0.0289	0.083	0.8242	0.8850	0.1028	34.15	400.3
500	0.0133	0.056	0.8432	0.8899	0.0989	21.03	243.1
900	0.0102	0.049	0.8514	0.8968	0.0910	17.98	206.9

Pt	Nombre de points sur chaque grille de quantification
QE	Erreur de quantification : $QE = \max_{0 \leq k \leq N} \ \Theta_k - \hat{\Theta}_k\ _2$
Δ	Pour tout z , $\Delta(z) = \Delta$
B_1	Borne empirique $\mathbb{E}_{x_0} \left[\sup_{0 \leq t \leq T_N} g(X(t)) \right] - \bar{V}_0$
B_2	Borne théorique donnée par le Théorème 6.2
B_3	Borne théorique donnée par les Théorèmes 7.1 et 7.2

TAB. 7.1 – Résultats de la simulation.

Dans cette partie de la thèse, nous avons donc étudié le problème d'arrêt optimal à horizon aléatoire. En nous appuyant sur les résultats développés pour le problème d'arrêt optimal à horizon fini et la méthode numérique pour les processus de diffusion, avec notamment l'utilisation de la quantification, nous avons prouvé la convergence de notre schéma de discrétisation tout en étant capable de donner une estimation de l'erreur commise. Nous pouvons en plus donner la construction d'un temps d'arrêt ϵ -optimal pour le PAO_N .

Les résultats numériques obtenus sur l'exemple simple (Section 7.2) permettent d'envisager des applications sur des exemples plus complexes. Les travaux de B. de Saporta, F. Dufour, H. Zhang et C. Elegbede ([24]) appliquent notre démarche dans le cas de la corrosion de structure.

Conclusion générale

Nous finissons cette contribution à l'étude des PDMP par un rapide bilan des résultats développés dans ce mémoire et par quelques pistes d'améliorations ou d'applications possibles.

Bilan

Dans la Partie I de ce mémoire, l'étude exhaustive des trois variables relatives au réservoir (hauteur, température, mode) nous a permis de comprendre et de formaliser avec précision leurs interactions, et de simplifier le problème, en éliminant les modes inatteignables. De plus, l'étude qualitative et quantitative de l'évolution de la hauteur et de la température, avec notamment la résolution du système différentiel (\mathcal{S}), s'est révélée nécessaire dans les deux approches pour calculer les probabilités de défaillance du réservoir. Les deux méthodes numériques que nous avons proposées utilisent la modélisation par les PDMP. En effet, les PDMP, par leurs spécificités, ont l'avantage de décrire au mieux le comportement réel d'un système dynamique au cours du temps, car ils prennent en compte les phénomènes aléatoires liés aux défaillances d'unités et l'évolution déterministe du système soumis à des lois physiques.

Le calcul des probabilités de défaillance du réservoir dans le cas de la première méthode numérique repose sur la résolution du système différentiel et la méthode de Monte Carlo. En connaissant le flot (c'est-à-dire la hauteur et la température) à chaque instant, nous avons ainsi pu calculer les temps de saut déterministes du PDMP. Pour estimer les temps de saut aléatoires, nous avons dû utiliser un algorithme de simulation d'une variable aléatoire d'intensité non constante ([14]), car l'intensité du PDMP dépendant de la position du réservoir, les algorithmes génériques ne permettent pas de les calculer.

Dans la deuxième approche, nous avons calculé la probabilité de surchauffe du réservoir, en utilisant les résultats de M.H.A. Davis sur les espérances de fonctionnelles d'un PDMP. Nous avons vu que l'équation intégro-différentielle devient un système de 17 équations couplées auxquelles sont associées 17 conditions au bord. L'étude des domaines de définition, nécessaire pour déterminer les conditions au bord des équations, s'est révélée plus complexe qu'il n'y paraissait, à cause, en particulier, des domaines qui bouclent entre eux (E_{24} et E_{27}), du domaine sans bord (E_{25}), et de la forme particulière des domaines, non inclus les uns dans les autres.

La différence majeure entre ces deux méthodes est que dans le premier cas, nous avons calculé, par la méthode de Monte Carlo, les probabilités de défaillance à temps *fini* (horizon de 1000h), alors que la deuxième méthode calcule une approximation de la probabilité de surchauffe à horizon *infini*. Dans les deux cas, les résultats, comparés à d'autres méthodes, sont cohérents et satisfaisants, notamment au niveau du temps de simulation, ce qui est en partie dû à l'économie réalisée en ne discrétisant pas le système différentiel, puisque l'on a trouvé une solution analytique pour chaque cas, pour la première méthode. Dans les deux cas, il n'y a pas discrétisation du processus.

L'intérêt d'étudier par deux approches ce cas d'école nous permet de conclure de la pertinence de la modélisation par les PDMP pour des systèmes hybrides représentatifs des problématiques industrielles, en relevant cependant la limitation de l'adaptabilité de nos méthodes à des cas significativement plus complexes.

Alors que la Partie I est une étude exploratoire sur le développement de méthodes numériques utilisant les PDMP, la Partie II apporte une contribution théorique dans l'étude des PDMP. Dans cette partie, nous avons étudié le PAO à horizon aléatoire pour les PDMP, en nous appuyant sur les travaux réalisés par U.S. Gugerli concernant les PAO à horizon infini pour les PDMP : U.S. Gugerli donne des résultats théoriques sur l'existence d'un temps d'arrêt ϵ -optimal du PAO.

Nous avons donc cherché, d'une part, à transposer les résultats de U.S. Gugerli dans le cadre des PAO à horizon aléatoire et à construire une méthode numérique permettant de calculer un temps d'arrêt ϵ -optimal du PAO, d'autre part d'obtenir la convergence de la fonction valeur approchée vers la fonction valeur du PAO originel, ainsi qu'une vitesse de convergence. Pour cela, nous avons procédé en deux étapes :

- l'interprétation probabiliste des opérateurs introduits par U.S. Gugerli s'est révélée nécessaire pour approcher la fonction valeur. Nous avons transformé la maximisation en temps continu de l'opérateur L en une maximisation en temps discret, en utilisant un schéma de discrétisation en temps dépendant de la position après-saut du PDMP ;
- les espérances conditionnelles de la chaîne de Markov (Θ_n) sont remplacées par des espérances conditionnelles de la chaîne quantifiée $(\hat{\Theta}_n)$. Nous avons utilisé pour cela, les résultats sur la quantification développés dans le cadre des processus de diffusion.

Le travail réalisé dans cette partie apporte un résultat intéressant car, à ce jour, seuls O.L.V. Costa et M.H.A. Davis ont proposé une méthode numérique pour résoudre le PAO pour un PDMP, en prouvant la convergence de leur schéma sans, cependant, donner une estimation de la vitesse de convergence. D'une part, comme nous l'avons remarqué dans la section 5.3, nous construisons une méthode numérique pour les PDMP en tenant compte de leur structure particulière, qui ne permet pas d'utiliser directement les résultats développés pour les diffusions. D'autre part, nous sommes en mesure de prouver la convergence du schéma construit et de calculer en plus la vitesse de convergence du schéma numérique. La méthode développée permet alors de proposer une règle d'arrêt qui est aussi un temps d'arrêt ϵ -optimal du PAO originel.

Perspectives

Le travail effectué au cours de cette thèse est une étude préliminaire qui a initié le projet ANR FauToCoES (Fault Tolerant Control of Embedded Systems, ANR-09-SEGI-004, 2009-2013) du programme ANR ARPEGE. Dans ce cadre, plusieurs pistes d'améliorations ou d'approfondissements vont être exploitées.

Il serait par exemple intéressant, dans l'approche par les EDP pour le calcul de surchauffe du réservoir, de choisir des grilles adaptées au flot et non pas des grilles cartésiennes fixes, ce qui permettrait d'éviter l'étape d'interpolation des fonctions. De plus, la base de nos deux méthodes repose sur la résolution analytique du système différentiel. La connaissance de l'évolution de la hauteur et de la température à chaque instant nous permet, entre autres, de déterminer le temps d'atteinte des frontières. Dans notre première approche, connaître la solution analytique de (\mathcal{S}) se révèle nécessaire, alors que l'on peut éventuellement s'en affranchir dans la deuxième méthode, en discrétisant le système différentiel.

La méthode utilisée pour calculer la probabilité de surchauffe dans le réservoir peut s'appliquer pour le calcul des deux autres probabilités d'événements redoutés (assèchement et surchauffe). Ces calculs n'ont pas été menés, car notre but était uniquement d'explorer cette méthode. Mais un travail similaire, c'est-à-dire en étudiant les modes concernés, les domaines de définition des fonctions associées aux modes et les bords de ces domaines, permettrait de le faire.

Aussi bien dans la première partie concernant le réservoir, que dans la deuxième sur les problèmes d'arrêt optimal, il serait intéressant d'appliquer les méthodes numériques développées à des systèmes dynamiques plus complexes. Dans ce sens, une application du problème d'arrêt optimal au cas du réservoir, et à la propagation de fissures sont à l'étude. Un exemple d'application industrielle proposée par EADS Astrium pour les problèmes d'arrêt optimal sur la corrosion de structure dans le cadre de la maintenance préventive est déjà présenté dans [24]. Un autre axe naturel de recherche intéressant mais difficile serait d'étendre nos résultats au problème de contrôle impulsionnel où non seulement le contrôleur doit choisir ses instants d'intervention, mais il doit aussi choisir le nouveau point de départ du processus. Du point de vue des applications, ceci correspond à des maintenances successives avec des réparations éventuellement partielles.

Table des figures

1.1	Trajectoire du PDMP au premier temps de saut T_1 aléatoire.	14
1.2	Trajectoire du PDMP au deuxième temps de saut T_2 déterministe. . .	15
1.3	Deux trajectoires du PDMP (temps en abscisse).	17
2.1	Schéma du réservoir thermique.	28
2.2	Diagramme des transitions.	30
2.3	Fonction $a(\theta)$, avec la température θ en abscisse.	30
2.4	Graphe des 37 modes.	35
3.1	Algorithme de rejet (C. Coccozza-Thivent) sur la courbe d'une intensité variable λ	50
3.2	Graphe des transitions de l'exemple.	51
3.3	Probabilités de défaillance pour la méthode PDMP pour $N = 10^3$ et $N = 10^4$ comparés à la courbe de référence ($N = 10^7$), avec le temps (heure) en abscisse.	52
3.4	Exemple d'une trajectoire du PDMP jusqu'au 14-ème temps de saut, avec la hauteur en mètre sur le graphique du haut, et la température en degré sur le graphique du bas.	53
3.5	Probabilités de défaillance pour la méthode PDMP pour $N = 10^4$ comparé aux probabilités de défaillance calculées avec l'approche Monte Carlo (solution référence) et la méthode CCCMT (avec $n_h = 30$ et $n_\theta = 60$), avec le temps (en heure) en abscisse.	56
3.6	Probabilités de défaillance pour la méthode PDMP pour $N = 10^4$ comparé aux probabilités de défaillance calculées avec la méthode ASH, avec le temps (en heure) en abscisse.	58
3.7	Probabilités de défaillance pour la méthode PDMP pour $N = 10^4$ comparé aux probabilités de défaillance calculées avec la méthode des volumes finis avec $\delta h = 0.5$, $\delta \theta = 2$, $\delta t = 0.03$	59
4.1	Graphe des transitions en partant de m_1	63
4.2	Graphe des transitions en partant de m_{27}	64
4.3	Graphe des 17 modes.	65
4.4	Domaine E_{29} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.	67
4.5	Domaine E_{36} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.	68
4.6	Domaines E_{24} et E_{27} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.	72

4.7	Domaine E_{24} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.	72
4.8	Domaine E_{27} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.	73
4.9	Domaines E_{36} et E_{25} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.	78
4.10	Domaines E_{24} et E_{27} , avec la hauteur (en mètre) en abscisse et la température (en degré) en ordonnée.	82
4.11	Grilles cartésiennes superposées D_{27} et E_{36}	83
4.12	Interpolation d'un point de E_{36} sur D_{27}	84
4.13	Graphes des 17 modes dans l'ordre de calcul.	86
4.14	Fonction f_{36} , avec la température en X, la hauteur en Y et la probabilité en Z.	90
4.15	Fonction f_{35} , avec la température en X, la hauteur en Y et la probabilité en Z.	90
4.16	Fonction g_{34} , avec la température en X, la hauteur en Y et la probabilité en Z.	91
4.17	Fonction f_{29} , avec la température en X, la hauteur en Y et la probabilité en Z.	91
4.18	Fonction f_{25} , avec la température en X, la hauteur en Y et la probabilité en Z.	92
4.19	Fonction g_{28} , avec la température en X, la hauteur en Y et la probabilité en Z.	92
4.20	Fonction g_{26} , avec la température en X, la hauteur en Y et la probabilité en Z.	93
4.21	Fonction f_{14} , avec la température en abscisse et la probabilité en ordonnée.	93
4.22	Fonction f_{10} , avec la température en abscisse et la probabilité en ordonnée.	94
4.23	Fonction f_9 , avec la hauteur en abscisse et la probabilité en ordonnée.	94
4.24	Fonction f_3 , avec la hauteur en abscisse et la probabilité en ordonnée.	95
5.1	Diagramme de Voronoï de la loi gaussienne centrée réduite, pour 200 points et 100 000 itérations.	106
7.1	Deux trajectoires du PDMP, avec le temps en abscisse.	139
7.2	Deux trajectoires stoppées du PDMP, avec le temps en abscisse.	139

Liste des tableaux

2.1	Tableau des correspondances mode-coefficient.	31
2.2	Constantes liées au système différentiel (\mathcal{S}) (avec $\gamma_1(\alpha)$, $\gamma_2(\alpha)$ et $\gamma_3(\alpha)$ en $m.h^{-1}$; q , en $m.h^{-1}$, une mesure du débit des unités; θ_{in} , en $^{\circ}C$, la température du liquide délivré par les unités 1 et 2; et K , en $m.^{\circ}C.h^{-1}$, un paramètre fixe lié aux variables physiques du réservoir.)	32
2.3	Tableau des correspondances coefficient-mode.	36
2.4	Configurations en fonction du coefficient α	36
2.5	Evolution de la hauteur et de la température en fonction du cas. . . .	41
3.1	Intensité de saut λ_m en fonction du mode m	45
3.2	Noyau de saut Q en fonction du mode m	47
3.3	Tableau des temps de franchissement de frontière $t_m^*(x)$	47
3.4	Temps de saut d'une trajectoire calculés par l'algorithme de C. Cocozza-Thivent, avec m le mode, h_0 et θ_0 la hauteur et la température initiale correspondant au mode m , τ_1 , τ_2 , et τ_3 les temps de panne des unités 1, 2 et 3, t^* le temps de saut déterministe et T le temps de saut du processus.	50
3.5	Probabilités d'événements redoutés pour $N = 10^4$ histoires.	54
3.6	Temps de simulation pour la méthode PDMP et erreur relative de l'approche Monte Carlo et de la méthode PDMP.	55
3.7	Erreur relative pour la méthode CCCMT.	56
3.8	Temps de calcul et erreur relative d'estimation de $p_1(1000)$ associés à différents pas de discrétisation pour le schéma de volumes finis.	59
4.1	Equations pour les domaines simples.	69
4.2	Equations associées aux modes m_{24} , m_{26} , m_{27} , m_{28} et m_{34}	73
4.3	Equations des fonctions redressées associées aux modes m_{24} , m_{26} , m_{27} , m_{28} et m_{34}	80
4.4	Tableau des différents pas de discrétisation.	81
4.5	Les fonctions f_{36} , f_{35} , g_{34} , f_{29} , f_{25} , g_{28} et g_{26} en $(7, \theta_1)$ pour différents pas de discrétisation.	87
4.6	Les fonctions f_{14} et f_{10} en θ_1 pour différents pas de discrétisation. . .	87
4.7	Les fonctions f_9 et f_3 en $h = 7$ pour différents pas de discrétisation. .	88
4.8	Les fonctions f_8 , f_7 et f_2 pour différents pas de discrétisation.	88
4.10	La fonction f_1 en $(7, \theta_1)$ pour différents pas de discrétisation.	88
4.11	Temps de calcul pour différents pas de discrétisation.	88
4.9	La fonction g_{27} en $(7, \theta_1)$ pour différents pas de discrétisation.	88

7.1	Résultats de la simulation.	140
-----	-------------------------------------	-----

Annexe A

Tableaux

Cette annexe est composée de trois tableaux. Le premier (Annexe A.1) résume tous les modes étudiés dans la Section 2.2.1 et transitions du réservoir. On l'utilisera notamment pour la détermination de l'intensité λ_m du PDMP (Section 3.1.3) et du noyau markovien Q_m (Section 3.1.4). Le deuxième (Annexe A.2) donne les équations du système différentiel (\mathcal{S}) décrites dans la Section 4.2. Enfin, le troisième tableau (Annexe A.3) résume les équations discrétisées selon le schéma de discrétisation donné dans la Section 4.3.2.

A.1 Tableau récapitulatif des modes du réservoir et de leurs transitions

Mode m	Domaines et Frontières	Cas	Etats entrants	Etats sortants
$m_1 = (O, F, O)$	$E_1 = \{7\} \times \{\theta_1\}$ $\partial E_1 = \emptyset$	3	\times	$\xrightarrow{\lambda^1} m_{15} = (Ob, F, O), m_{27} = (Fb, F, O)$ $\xrightarrow{\lambda^2} m_4 = (O, Ob, O), m_7 = (O, Fb, O)$ $\xrightarrow{\lambda^3} m_2 = (O, F, Ob), m_3 = (O, F, Fb)$
$m_2 = (O, F, Ob)$	$E_2 = \{7\} \times \{\theta_1\}$ $\partial E_2 = \emptyset$	3	$\xleftarrow{\lambda^3} m_1 = (O, F, O)$	$\xrightarrow{\lambda^1} m_{16} = (Ob, F, Ob), m_{28} = (Fb, F, Ob)$ $\xrightarrow{\lambda^2} m_5 = (O, Ob, Ob), m_8 = (O, Fb, Ob)$
$m_3 = (O, F, Fb)$	$E_3 = [7, 8[\times\{\theta_1\}$ $\partial E_3 = \{8\} \times \{\theta_1\}$	1.1	$\xleftarrow{\lambda^3} m_1 = (O, F, O)$	$\xrightarrow{\lambda^1} m_{17} = (Ob, F, Fb), m_{29} = (Fb, F, Fb)$ $\xrightarrow{\lambda^2} m_6 = (O, Ob, Fb), m_9 = (O, Fb, Fb)$ $\xrightarrow{L^2} m_{10} = (F, F, Fb)$
$m_4 = (O, Ob, O)$	pas de contribution	1.3	$\xleftarrow{\lambda^2} m_1 = (O, F, O)$	$\xrightarrow{\lambda^1} m_{18} = (Ob, Ob, O), m_{30} = (Fb, Ob, O)$ $\xrightarrow{\lambda^3} m_5 = (O, Ob, Ob), m_6 = (O, Ob, Fb)$ $\xrightarrow{L^2} m_{11} = (F, Ob, O)$
$m_5 = (O, Ob, Ob)$	pas de contribution	1.3	$\xleftarrow{\lambda^2} m_2 = (O, F, Ob)$ $\xleftarrow{\lambda^3} m_4 = (O, Ob, O)$	$\xrightarrow{\lambda^1} m_{19} = (Ob, Ob, Ob), m_{32} = (Fb, Ob, Ob)$ $\xrightarrow{L^2} m_{12} = (F, Ob, Ob)$
$m_6 = (O, Ob, Fb)$	pas de contribution	1.2	$\xleftarrow{\lambda^2} m_3 = (O, F, Fb)$ $\xleftarrow{\lambda^3} m_4 = (O, Ob, O)$	$\xrightarrow{\lambda^1} m_{20} = (Ob, Ob, Fb), m_{33} = (Fb, Ob, Fb)$ $\xrightarrow{L^2} m_{13} = (F, Ob, Fb)$
$m_7 = (O, Fb, O)$	$E_7 = \{7\} \times \{\theta_1\}$ $\partial E_7 = \emptyset$	3	$\xleftarrow{\lambda^2} m_1 = (O, F, 0)$	$\xrightarrow{\lambda^1} m_{21} = (Ob, Fb, O), m_{34} = (Fb, Fb, O)$ $\xrightarrow{\lambda^3} m_8 = (O, Fb, Ob), m_9 = (O, Fb, Fb)$
$m_8 = (O, Fb, Ob)$	$E_8 = \{7\} \times \{\theta_1\}$ $\partial E_8 = \emptyset$	3	$\xleftarrow{\lambda^2} m_2 = (O, F, Ob)$ $\xleftarrow{\lambda^3} m_7 = (O, Fb, O)$	$\xrightarrow{\lambda^1} m_{22} = (Ob, Fb, Ob), m_{36} = (Fb, Fb, Ob)$
$m_9 = (O, Fb, Fb)$	$E_9 = [7, 8[\times\{\theta_1\}$ $\partial E_9 = \{8\} \times \{\theta_1\}$	1.1	$\xleftarrow{\lambda^2} m_3 = (O, F, Fb)$ $\xleftarrow{\lambda^3} m_7 = (O, Fb, O)$	$\xrightarrow{\lambda^1} m_{23} = (Ob, Fb, Fb), m_{37} = (Fb, Fb, Fb)$ $\xrightarrow{L^2} m_{14} = (F, Fb, Fb)$

Mode m	Domaines et Frontières	Cas	Etats entrants	Etats sortants
$m_{10} = (F, F, Fb)$	$E_{10} = \{8\} \times [\theta_1, 100[$ $\partial E_{10} = \{8\} \times \{100\}$	2	$\xleftarrow{L^2} m_3 = (O, F, Fb)$	$\xrightarrow{\lambda^1} m_{17} = (Ob, F, Fb), m_{29} = (Fb, F, Fb)$ $\xrightarrow{\lambda^2} m_{13} = (F, Ob, Fb), m_{14} = (F, Fb, Fb)$
$m_{11} = (F, Ob, O)$	pas de contribution	3	$\xleftarrow{L^2} m_4 = (O, Ob, O)$	$\xrightarrow{\lambda^1} m_{18} = (Ob, Ob, O), m_{30} = (Fb, Ob, O)$ $\xrightarrow{\lambda^3} m_{12} = (F, Ob, Ob), m_{13} = (F, Ob, Fb)$
$m_{12} = (F, Ob, Ob)$	pas de contribution	3	$\xleftarrow{L^2} m_5 = (O, Ob, Ob)$ $\xleftarrow{\lambda^3} m_{11} = (F, Ob, O)$	$\xrightarrow{\lambda^1} m_{19} = (Ob, Ob, Ob), m_{32} = (Fb, Ob, Ob)$
$m_{13} = (F, Ob, Fb)$	pas de contribution	1.1	$\xleftarrow{L^2} m_6 = (O, Ob, Fb)$ $\xleftarrow{\lambda^2} m_{10} = (F, F, Fb)$ $\xleftarrow{\lambda^3} m_{11} = (F, Ob, O)$	$\xrightarrow{\lambda^1} m_{20} = (Ob, Ob, Fb), m_{33} = (Fb, Ob, Fb)$
$m_{14} = (F, Fb, Fb)$	$E_{14} = \{8\} \times [\theta_1, 100[$ $\partial E_{14} = \{8\} \times \{100\}$	2	$\xleftarrow{L^2} m_9 = (O, Fb, Fb)$ $\xleftarrow{\lambda^2} m_{10} = (F, F, Fb)$	$\xrightarrow{\lambda^1} m_{23} = (Ob, Fb, Fb), m_{37} = (Fb, Fb, Fb)$
$m_{15} = (Ob, F, O)$	pas de contribution	3	$\xleftarrow{\lambda^1} m_1 = (O, F, O)$	$\xrightarrow{\lambda^2} m_{18} = (Ob, Ob, O), m_{21} = (Ob, Fb, O)$ $\xrightarrow{\lambda^3} m_{16} = (Ob, F, Ob), m_{17} = (Ob, F, Fb)$
$m_{16} = (Ob, F, Ob)$	pas de contribution	3	$\xleftarrow{\lambda^1} m_2 = (O, F, Ob)$ $\xleftarrow{\lambda^3} m_{15} = (Ob, F, O)$	$\xrightarrow{\lambda^2} m_{19} = (Ob, Ob, Ob), m_{22} = (Ob, Fb, Ob)$
$m_{17} = (Ob, F, Fb)$	pas de contribution	1.1	$\xleftarrow{\lambda^1} m_3 = (O, F, Fb)$ $\xleftarrow{\lambda^1} m_{10} = (F, F, Fb)$ $\xleftarrow{\lambda^3} m_{15} = (Ob, F, O)$	$\xrightarrow{\lambda^2} m_{20} = (Ob, Ob, Fb), m_{23} = (Ob, Fb, Fb)$
$m_{18} = (Ob, Ob, O)$	pas de contribution	1.3	$\xleftarrow{\lambda^1} m_4 = (O, Ob, O)$ $\xleftarrow{\lambda^1} m_{11} = (F, Ob, O)$ $\xleftarrow{\lambda^3} m_{15} = (Ob, F, O)$	$\xrightarrow{\lambda^3} m_{19} = (Ob, Ob, Ob), m_{20} = (Ob, Ob, Fb)$

Mode m	Domaines et Frontières	Cas	Etats entrants	Etats sortants
$m_{19} = (Ob, Ob, Ob)$	pas de contribution	1.3	$\xleftarrow{\lambda^1} m_5 = (O, Ob, Ob)$ $\xleftarrow{\lambda^1} m_{12} = (F, Ob, Ob)$ $\xleftarrow{\lambda^2} m_{16} = (Ob, F, Ob)$ $\xleftarrow{\lambda^3} m_{18} = (Ob, Ob, O)$	$\longrightarrow \times$
$m_{20} = (Ob, Ob, Fb)$	pas de contribution	1.2	$\xleftarrow{\lambda^1} m_6 = (O, Ob, Fb)$ $\xleftarrow{\lambda^1} m_{13} = (F, Ob, Fb)$ $\xleftarrow{\lambda^2} m_{17} = (Ob, F, Fb)$ $\xleftarrow{\lambda^3} m_{18} = (Ob, Ob, O)$	$\longrightarrow \times$
$m_{21} = (Ob, Fb, O)$	pas de contribution	3	$\xleftarrow{\lambda^1} m_7 = (O, Fb, 0)$ $\xleftarrow{\lambda^2} m_{15} = (Ob, F, 0)$	$\xrightarrow{\lambda^3} m_{22} = (Ob, Fb, Ob), m_{23} = (Ob, Fb, Fb)$
$m_{22} = (Ob, Fb, Ob)$	pas de contribution	3	$\xleftarrow{\lambda^1} m_8 = (O, Fb, Ob)$ $\xleftarrow{\lambda^2} m_{16} = (Ob, F, Ob)$ $\xleftarrow{\lambda^3} m_{21} = (Ob, Fb, 0)$	$\longrightarrow \times$
$m_{23} = (Ob, Fb, Fb)$	pas de contribution	1.1	$\xleftarrow{\lambda^1} m_9 = (O, Fb, Fb)$ $\xleftarrow{\lambda^1} m_{14} = (F, Fb, Fb)$ $\xleftarrow{\lambda^2} m_{17} = (Ob, F, Fb)$ $\xleftarrow{\lambda^3} m_{21} = (Ob, Fb, 0)$	$\longrightarrow \times$
$m_{24} = (Fb, O, F)$	E_{24} ∂E_{24}	1.1	$\xleftarrow{L^1} m_{27} = (Fb, F, O)$	$\xrightarrow{\lambda^2} m_{31} = (Fb, Ob, F), m_{35} = (Fb, Fb, F)$ $\xrightarrow{\lambda^3} m_{25} = (Fb, O, Ob), m_{26} = (Fb, O, Fb)$ $\xrightarrow{L^2} m_{27} = (Fb, F, O)$

Mode m	Domaines et Frontières	Cas	Etats entrants	Etats sortants
$m_{25} = (Fb, O, Ob)$	E_{25} pas de frontière	3	$\xleftarrow{\lambda^3} m_{24} = (Fb, O, F)$ $\xleftarrow{L^1} m_{28} = (Fb, F, Ob)$	$\xrightarrow{\lambda^2} m_{32} = (Fb, Ob, Ob), m_{36} = (Fb, Fb, Ob)$
$m_{26} = (Fb, O, Fb)$	E_{26} ∂E_{26}	1.1	$\xleftarrow{\lambda^3} m_{24} = (Fb, O, F)$	$\xrightarrow{\lambda^2} m_{33} = (Fb, Ob, Fb), m_{37} = (Fb, Fb, Fb)$ $\xrightarrow{L^2} m_{29} = (Fb, F, Fb)$
$m_{27} = (Fb, F, O)$	E_{27} ∂E_{27}	4	$\xleftarrow{\lambda^1} m_1 = (O, F, O)$ $\xleftarrow{L^2} m_{24} = (Fb, O, F)$	$\xrightarrow{\lambda^2} m_{30} = (Fb, Ob, O), m_{34} = (Fb, Fb, O)$ $\xrightarrow{\lambda^3} m_{28} = (Fb, F, Ob), m_{29} = (Fb, F, Fb)$ $\xrightarrow{L^1} m_{24} = (Fb, O, F)$
$m_{28} = (Fb, F, Ob)$	E_{28} ∂E_{28}	4	$\xleftarrow{\lambda^1} m_2 = (O, F, Ob)$ $\xleftarrow{\lambda^3} m_{27} = (Fb, F, O)$	$\xrightarrow{\lambda^2} m_{32} = (Fb, Ob, Ob), m_{36} = (Fb, Fb, Ob)$ $\xrightarrow{L^1} m_{25} = (Fb, O, Ob)$
$m_{29} = (Fb, F, Fb)$	E_{29} ∂E_{29}	2	$\xleftarrow{\lambda^1} m_3 = (O, F, Fb)$ $\xleftarrow{\lambda^1} m_{10} = (F, F, Fb)$ $\xleftarrow{L^2} m_{26} = (Fb, O, Fb)$ $\xleftarrow{\lambda^3} m_{27} = (Fb, F, O)$	$\xrightarrow{\lambda^2} m_{33} = (Fb, Ob, Fb), m_{37} = (Fb, Fb, Fb)$
$m_{30} = (Fb, Ob, O)$	$E_{30} \subset [6, 8] \times [\theta_2, 100[$ Pas de bord	3	$\xleftarrow{\lambda^1} m_4 = (O, Ob, O)$ $\xleftarrow{\lambda^1} m_{11} = (F, Ob, O)$ $\xleftarrow{\lambda^2} m_{27} = (Fb, F, O)$ $\xleftarrow{L^2} m_{31} = (Fb, Ob, F)$	$\xrightarrow{\lambda^3} m_{32} = (Fb, Ob, Ob), m_{33} = (Fb, Ob, Fb)$
$m_{31} = (Fb, Ob, F)$	pas de contribution	1.1	$\xleftarrow{\lambda^2} m_{24} = (Fb, O, F)$	$\xrightarrow{\lambda^3} m_{32} = (Fb, Ob, Ob), m_{33} = (Fb, Ob, Fb)$ $\xrightarrow{L^2} m_{30} = (Fb, Ob, O)$

Mode m	Domaines et Frontières	Cas	Etats entrants	Etats sortants
$m_{32} = (Fb, Ob, Ob)$	pas de contribution	3	$\xleftarrow{\lambda^1} m_5 = (O, Ob, Ob)$ $\xleftarrow{\lambda^1} m_{12} = (F, Ob, Ob)$ $\xleftarrow{\lambda^2} m_{25} = (Fb, O, Ob)$ $\xleftarrow{\lambda^2} m_{28} = (Fb, F, Ob)$ $\xleftarrow{\lambda^3} m_{30} = (Fb, Ob, O)$ $\xleftarrow{\lambda^3} m_{31} = (Fb, Ob, F)$	$\longrightarrow \times$
$m_{33} = (Fb, Ob, Fb)$	pas de contribution	1.1	$\xleftarrow{\lambda^1} m_6 = (O, Ob, Fb)$ $\xleftarrow{\lambda^1} m_{13} = (F, Ob, Fb)$ $\xleftarrow{\lambda^2} m_{26} = (Fb, O, Fb)$ $\xleftarrow{\lambda^2} m_{29} = (Fb, F, Fb)$ $\xleftarrow{\lambda^3} m_{30} = (Fb, Ob, O)$ $\xleftarrow{\lambda^3} m_{31} = (Fb, Ob, F)$	$\longrightarrow \times$
$m_{34} = (Fb, Fb, O)$	E_{34} ∂E_{34}	4	$\xleftarrow{\lambda^1} m_7 = (O, Fb, O)$ $\xleftarrow{\lambda^3} m_{27} = (Fb, F, O)$	$\xrightarrow{\lambda^3} m_{36} = (Fb, Fb, Ob), m_{37} = (Fb, Fb, Fb)$ $\xrightarrow{L^1} m_{35} = (Fb, Fb, F)$
$m_{35} = (Fb, Fb, F)$	E_{35} ∂E_{35}	2	$\xleftarrow{\lambda^2} m_{24} = (Fb, O, F)$ $\xleftarrow{L^1} m_{34} = (Fb, Fb, O)$	$\xrightarrow{\lambda^3} m_{36} = (Fb, Fb, Ob), m_{37} = (Fb, Fb, Fb)$
$m_{36} = (Fb, Fb, Ob)$	E_{36} ∂E_{36}	4	$\xleftarrow{\lambda^1} m_8 = (O, Fb, Ob)$ $\xleftarrow{\lambda^2} m_{25} = (Fb, O, Ob)$ $\xleftarrow{\lambda^2} m_{28} = (Fb, F, Ob)$ $\xleftarrow{\lambda^3} m_{34} = (Fb, Fb, O)$ $\xleftarrow{\lambda^3} m_{35} = (Fb, Fb, F)$	$\longrightarrow \times$

Mode m	Domaines et Frontières	Cas	Etats entrants	Etats sortants
$m_{37} = (Fb, Fb, Fb)$	$E_{37} \subset [6, 8] \times [\theta_1, 100[$ $\partial E_{37} = \{(h_0, 100)\}$	2	$\xleftarrow{\lambda^1} m_9 = (O, Fb, Fb)$ $\xleftarrow{\lambda^1} m_{14} = (F, Fb, Fb)$ $\xleftarrow{\lambda^2} m_{26} = (Fb, O, Fb)$ $\xleftarrow{\lambda^2} m_{29} = (Fb, F, Fb)$ $\xleftarrow{\lambda^3} m_{34} = (Fb, Fb, O)$ $\xleftarrow{\lambda^3} m_{35} = (Fb, Fb, F)$	

A.2 Tableau des équations

MODE	CAS	DOMAINE	FRONTIERE
$m_1 = (O, F, O)$	3	E_1	∂E_1
$f_1(7, \theta_1) = \frac{1}{\ell_{1,2,3}(\theta_1)} [\lambda^1(\theta_1)f_{27}(7, \theta_1) + \lambda^2(\theta_1)f_7(7, \theta_1) + \lambda^3(\theta_1)(f_2(7, \theta_1) + f_3(7, \theta_1))]$			
$m_2 = (O, F, Ob)$	3	E_1	∂E_1
$f_2(7, \theta_1) = \frac{1}{\ell_{1,2}(\theta_1)} [\lambda^1(\theta_1)f_{28}(7, \theta_1) + \lambda^2(\theta_1)f_8(7, \theta_1)]$			
$m_3 = (O, F, Fb)$	1.1	E_3	∂E_3
$q \frac{\partial}{\partial h} f_3(h, \theta_1) + \lambda^1(\theta_1)f_{29}(h, \theta_1) + \lambda^2(\theta_1)f_9(h, \theta_1) - \ell_{1,2}(\theta_1)f_3(h, \theta_1) = 0$ $f_3(8, \theta_1) = f_{10}(8, \theta_1)$			
$m_7 = (O, Fb, O)$	3	E_1	∂E_1
$f_7(7, \theta_1) = \frac{1}{\ell_{1,3}(\theta_1)} [\lambda^1(\theta_1)f_{34}(7, \theta_1) + \lambda^3(\theta_1)(f_8(7, \theta_1) + f_9(7, \theta_1))]$			
$m_8 = (O, Fb, Ob)$	3	E_1	∂E_1
$f_8(7, \theta_1) = \frac{1}{2}f_{36}(7, \theta_1)$			
$m_9 = (O, Fb, Fb)$	1.1	E_3	∂E_3
$q \frac{\partial}{\partial h} f_9(h, \theta_1) + \lambda^1(\theta_1) - \ell_1(\theta_1)f_9(h, \theta_1) = 0$ $f_9(8, \theta_1) = f_{14}(8, \theta_1)$			
$m_{10} = (F, F, Fb)$	2	E_{10}	∂E_{10}
$\frac{K}{8} \frac{\partial}{\partial \theta} f_{10}(8, \theta) + \lambda^1(\theta)f_{29}(8, \theta) + \lambda^2(\theta)f_{14}(8, \theta) - \ell_{1,2}(\theta)f_{10}(8, \theta) = 0$ $f_{10}(8, 100) = 1$			
$m_{14} = (F, Fb, Fb)$	2	E_{10}	∂E_{10}
$\frac{K}{8} \frac{\partial}{\partial \theta} f_{14}(8, \theta) + \lambda^1(\theta) - \ell_1(\theta)f_{14}(8, \theta) = 0$ $f_{14}(8, 100) = 1$			
$m_{24} = (Fb, O, F)$	1.1	E_{24}	∂E_{24}
$q \frac{\partial}{\partial h} f_{24}(h, \theta) + \frac{q(\theta_1 - \theta)}{h} \frac{\partial}{\partial \theta} f_{24}(h, \theta) + \lambda^2(\theta)f_{35}(h, \theta) + \lambda^3(\theta)(f_{25}(h, \theta) + f_{26}(h, \theta)) - \ell_{2,3}(\theta)f_{24}(h, \theta) = 0$ $f_{24}(8, \theta) = f_{27}(8, \theta)$			

MODE	CAS	DOMAINE	FRONTIERE
$m_{25} = (Fb, O, Ob)$	3	E_{25}	∂E_{25}
$\frac{q(\theta_1 - \theta)}{h} \frac{\partial}{\partial \theta} f_{25}(h, \theta) + \lambda^2(\theta) f_{36}(h, \theta) - \ell_2(\theta) f_{25}(xh, \theta) = 0$ $f_{25}(h, \theta_1) = \frac{1}{2} f_{36}(h, \theta_1)$			
$m_{26} = (Fb, O, Fb)$	1.1	E_{24}	∂E_{24}
$q \frac{\partial}{\partial h} f_{26}(h, \theta) + \frac{q(\theta_1 - \theta)}{h} \frac{\partial}{\partial \theta} f_{26}(h, \theta) + \lambda^2(\theta) - \ell_2(\theta) f_{26}(h, \theta) = 0$ $f_{26}(8, \theta) = f_{29}(8, \theta)$			
$m_{27} = (Fb, F, F)$	4	E_{27}	∂E_{27}
$-q \frac{\partial}{\partial h} f_{27}(h, \theta) + \frac{K}{h} \frac{\partial}{\partial \theta} f_{27}(h, \theta) + \lambda^2(\theta) f_{34}(h, \theta) + \lambda^3(\theta) (f_{28}(h, \theta) + f_{29}(h, \theta))$ $- \ell_{2,3}(\theta) f_{27}(h, \theta) = 0$ $f_{27}(6, \theta) = f_{24}(6, \theta)$			
$m_{28} = (Fb, F, Ob)$	4	E_{27}	∂E_{27}
$-q \frac{\partial}{\partial h} f_{28}(h, \theta) + \frac{K}{h} \frac{\partial}{\partial \theta} f_{28}(h, \theta) + \lambda^2(\theta) f_{36}(h, \theta) - \ell_2(\theta) f_{28}(h, \theta) = 0$ $f_{28}(6, \theta) = f_{25}(6, \theta)$			
$m_{29} = (Fb, F, Fb)$	2	E_{29}	∂E_{29}
$\frac{K}{h} \frac{\partial}{\partial \theta} f_{29}(h, \theta) + \lambda^2(\theta) - \ell_2(\theta) f_{29}(h, \theta) = 0$ $f_{29}(h, 100) = 1$			
$m_{34} = (Fb, Fb, O)$	4	E_{27}	∂E_{27}
$-q \frac{\partial}{\partial h} f_{34}(h, \theta) + \frac{K}{h} \frac{\partial}{\partial \theta} f_{34}(h, \theta) + \lambda^3(\theta) (f_{36}(h, \theta) + 1) - \ell_3(\theta) f_{34}(h, \theta) = 0$ $f_{34}(6, \theta) = f_{35}(6, \theta)$			
$m_{35} = (Fb, Fb, F)$	2	E_{29}	∂E_{29}
$\frac{K}{h} \frac{\partial}{\partial \theta} f_{35}(h, \theta) + \lambda^3(\theta) (f_{36}(h, \theta) + 1) - \ell_3(\theta) f_{35}(h, \theta) = 0$ $f_{35}(h, 100) = 1$			
$m_{36} = (Fb, Fb, Ob)$	4	E_{36}	∂E_{36}
$-q \frac{\partial}{\partial h} f_{36}(h, \theta) + \frac{K}{h} \frac{\partial}{\partial \theta} f_{36}(h, \theta) = 0$ $f_{36}(4, \theta) = 0 \text{ avec } \theta \in [\theta_1, 100[$ $f_{36}(h, 100) = 1 \text{ avec } h \in [4, 8]$			

A.3 Tableau des équations discrétisées

MODE	CAS	DOMAINE	FRONTIERE
$m_{36} = (Fb, Fb, Ob)$	4	E_{36}	∂E_{36}
pour $i = 1$ à N_h et $j = 1$ à N_θ			
$-q \frac{f_{36}(i+1, j) - f_{36}(i, j)}{\delta h} + \frac{K}{h_i} \frac{f_{36}(i, j+1) - f_{36}(i, j)}{\delta \theta} = 0$			
$f_{36}(1, j) = 0$ pour $j = 1$ à N_θ $f_{36}(i, N_\theta + 1) = 1$ pour $i = 1$ à $N_h + 1$			
$m_{35} = (Fb, Fb, F)$	2	E_{29}	∂E_{29}
pour $i = 1$ à $N_h/2 + 1$ et $j = N_\theta$ à 1			
$\frac{K}{h_i} \frac{f_{35}(i, j+1) - f_{35}(i, j)}{\delta \theta} + \lambda^3(\theta_j)(f_{36}(i, j) + 1) - \ell_3(\theta_j)f_{35}(i, j) = 0$			
$f_{35}(i, N_\theta + 1) = 1$			
$m_{34} = (Fb, Fb, O)$	4	D_{27}	∂D_{27}
pour $i = N_h/2 + 1$ à N_h et $k \in \mathbb{N}^*$			
$-q \frac{g_{34}(i+1, k+1) - g_{34}(i, k+1)}{\delta h} + \lambda^3(w_{i+1, k+1})(f_{36}(i+1, w_{i+1, k+1}) + 1) - \ell_3(w_{i+1, k+1})g_{34}(i+1, k+1) = 0$			
pour $i = N_h/2 + 1$ à $3N_h/4 + 1$			
$-q \frac{g_{34}(i+1, 1) - g_{34}(i, 1)}{\delta h} + \lambda^3(w_{(i+1, 1)})(f_{36}(w_{(i+1, 1)}) + 1) - \ell_3(w_{(i+1, 1)})g_{34}(i+1, 1) = 0$			
$g_{34}(N_h/2 + 1, k+1) = f_{35}(N_h/2 + 1, w_{N_h/2+1, k+1})$			
$m_{29} = (Fb, F, Fb)$	2	E_{29}	∂E_{29}
pour $i = N_h/2 + 1$ à $N_h + 1$ et $j = N_\theta$ à 1			
$\frac{K}{h_i} \frac{f_{29}(i, j+1) - f_{29}(i, j)}{\delta \theta} + \lambda^2(\theta_j) - \ell_2(\theta_j)f_{29}(i, j) = 0$			
$f_{29}(i, N_\theta + 1) = 1$			

MODE	CAS	DOMAINE	FRONTIERE
$m_{25} = (Fb, O, Ob)$	3	E_{25}	∂E_{25}
pour $i = N_h/2 + 1$ à $N_h + 1$ et $j = 1$ à N_θ			
$\frac{q(\theta_1 - \theta_{j+1})}{h_i} \frac{f_{25}(i, j+1) - f_{25}(i, j)}{\delta\theta} + \lambda^2(\theta_{i+1})f_{36}(i, j+1) - \ell_2(\theta_{j+1})f_{25}(i, j+1) = 0$ $f_{25}(i, 1) = \frac{1}{2}f_{36}(i, 1)$			
$m_{28} = (Fb, F, Ob)$	4	D_{27}	∂D_{27}
pour $i = N_h/2 + 1$ à N_h et $k \in \mathbb{N}^*$			
$-q \frac{g_{28}(i+1, k+1) - g_{28}(i, k+1)}{\delta h} + \lambda^2(w_{i+1, k+1})f_{36}(i+1, w_{i+1, k+1})$ $-\ell_2(w_{i+1, k+1})g_{28}(i, k+1) = 0$			
pour $i = N_h/2 + 1$ à $3N_h/4 + 1$			
$-q \frac{g_{34}(i+1, 1) - g_{34}(i, 1)}{\delta h} + \lambda^2(w_{i+1, 1})(f_{36}(w_{i+1, 1})) - \ell_2(w_{i+1, 1})g_{34}(i+1, 1) = 0$ $g_{28}(N_h/2 + 1, k+1) = f_{25}(N_h/2 + 1, w_{N_h/2+1, k+1})$			
$m_{26} = (Fb, O, Fb)$	1.1	D_{24}	∂D_{24}
pour $i = N_h$ à $N_h/2 + 1$ et $k \in \mathbb{N}^*$			
$q \frac{g_{26}(i+1, k) - g_{26}(i, k)}{\delta h} + \lambda^2(z_{i+1, k+1}) - \ell_2(z_{i+1, k+1})g_{26}(i, k) = 0$ $g_{26}(N_h + 1, k) = f_{29}(N_h + 1, z_{N_h+1, k})$			
$m_{14} = (F, Fb, Fb)$	2	E_{10}	∂E_{10}
pour $j = N_\theta$ à 1			
$\frac{K}{8} \frac{f_{14}(j+1) - f_{14}(j)}{\delta\theta} + \lambda^1(\theta_j) - \ell_1(\theta_j)f_{14}(j) = 0$ $f_{14}(N_\theta + 1) = 1$			

MODE	CAS	DOMAINE	FRONTIERE
$m_{10} = (F, F, Fb)$	2	E_{10}	∂E_{10}
pour $j = N_\theta$ à 1			
$\frac{K}{8} \frac{f_{10}(j+1) - f_{10}(j)}{\delta\theta} + \lambda^1(\theta_j) f_{29}(N_h + 1, j) + \lambda^2(\theta_j) f_{14}(j) - \ell_{1,2}(\theta_j) f_{10}(j) = 0$ $f_{10}(N_\theta + 1) = 1$			
$m_9 = (O, Fb, Fb)$	1.1	E_3	∂E_3
pour $i = N_h$ à $3N_h/4 + 1$			
$q \frac{f_9(i+1) - f_9(i)}{\delta h} + \lambda^1(\theta_1) - \ell_1(\theta_1) f_9(i) = 0$ $f_9(N_h + 1) = f_{14}(1)$			
$m_8 = (O, Fb, Ob)$	3	E_1	∂E_1
$f_8 = \frac{1}{2} f_{36}(3N_h/4 + 1, 1)$			
$m_7 = (O, Fb, O)$	3	E_1	∂E_1
$f_7 = \frac{1}{\ell_{1,3}(\theta_1)} [\lambda^1(\theta_1) g_{34}(3N_h/4 + 1, 1) + \lambda^3(\theta_1) (f_8 + f_9(3N_h/4 + 1))]$			
$m_2 = (O, F, Ob)$	3	E_1	∂E_1
$f_2 = \frac{1}{\ell_{1,2}(\theta_1)} (\lambda^1(\theta_1) g_{28}(3N_h/4 + 1, 1) + \lambda^2(\theta_1) f_8)$			
$m_3 = (O, F, Fb)$	1.1	E_3	∂E_3
pour $i = N_h$ à $3N_h/4 + 1$			
$q \frac{f_3(i+1) - f_3(i)}{\delta h} + \lambda^1(\theta_1) f_{29}(i, 1) + \lambda^2(\theta_1) f_9(i) - \ell_{1,2}(\theta_1) f_3(i) = 0$ $f_3(N_h + 1) = f_{10}(1)$			
$m_{24} = (Fb, O, F)$	1.1	D_{24}	∂D_{24}
pour $i = N_h$ à $N_h/2 + 1$ et $k \in \mathbb{N}^*$			
$q \frac{g_{24}(i+1, k) - g_{24}(i, k)}{\delta h} + \lambda^2(z_{i,k}) f_{35}(i, z_{i,k}) + \lambda^3(z_{i,k}) (f_{25}(i, z_{i,k}) + g_{26}(i, k) - \ell_{2,3}(z_{i,k}) g_{24}(i, k)) = 0$ $g_{24}(N_h + 1, j) = g_{27}(N_h + 1, j)$			

MODE	CAS	DOMAINE	FRONTIERE
$m_{27} = (Fb, F, F)$	4	D_{27}	∂D_{27}
pour $i = N_h/2 + 1$ à N_h et $k \in \mathbb{N}^*$			
$-q \frac{g_{27}(i+1, k+1) - g_{27}(i, k+1)}{\delta h} + \lambda^2(w_{i+1, k+1})g_{34}(i+1, k+1)$ $+ \lambda^3(w_{i+1, k+1})(g_{28}(i+1, k+1) + f_{29}(i+1, w_{i+1, k+1})) - \ell_{2,3}(w_{i+1, k+1})g_{27}(i+1, k+1) = 0$			
pour $i = N_h/2 + 1$ à $3N_h/4 + 1$			
$-q \frac{g_{27}(i+1, 1) - g_{27}(i, 1)}{\delta h} + \lambda^2(w_{i+1, 1})g_{34}(i+1, 1)$ $+ \lambda^3(w_{i+1, 1})(g_{28}(i+1, 1) + f_{29}(i+1, w_{i+1, 1})) - \ell_{2,3}(w_{i+1, 1})g_{27}(i+1, 1) = 0$ $g_{27}(N_h/2 + 1, k+1) = g_{24}(N_h/2 + 1, k)$			
$m_1 = (O, F, O)$	3	E_1	∂E_1
$f_1 = \frac{1}{\ell_{1,2,3}(\theta_1)}(\lambda^1(\theta_1)g_{27}(3N_h/4 + 1, 1) + \lambda^2(\theta_1)f_7 + \lambda^3(\theta_1)(f_2 + f_3(3N_h/4 + 1)))$			

Annexe B

Programmes

Dans cette annexe sont présentés les programmes informatiques pour les deux méthodes numériques permettant de calculer les probabilités de défaillance du réservoir. La première méthode est codée en C++, la deuxième en Scilab.

B.1 Méthode de Monte Carlo

Nous détaillons ici le programme utilisé pour la première méthode numérique de calcul des probabilités de défaillance du réservoir (l'algorithme est décrit dans le Chapitre 3). Le Fichier principal définit les paramètres et appelle les fonctions définies dans le fichier *Calcul du temps de saut et du nouveau mode* pour calculer les probabilités de défaillance du réservoir. Le fichier *Calcul du temps de saut et du nouveau mode* est découpé en plusieurs parties.

- La première fonction (*alpha_recode*) convertit les modes en configurations.
- La fonction *calcul_a_theta* permet de calculer la fonction $a(\theta)$, la hauteur et la température en fonction de la configuration α .
- La fonction *saut_poisson* calcule le temps de saut aléatoire, suivant l'algorithme de C. Cocozza-Thivent.
- La fonction *calcul_nouvel_etat* donne le nouveau mode en fonction du type de saut (aléatoire ou déterministe). Il s'agit en fait de l'implémentation de la mesure de transition Q_m du PDMP.
- La fonction *tstar_nouveau_flot_initial* permet de calculer le temps de saut, le type de saut et remet à jour le flot en fonction de la configuration.

Le fichier principal fait donc appel aux cinq fonctions présentées ci-dessus. Après la déclaration de toutes les constantes nécessaires ($\lambda_1, \lambda_2, \lambda_3, Q, K, \dots$), on introduit une boucle générant un certain nombre d'histoires N (on prendra $N = 10^3, 10^4, 10^5$ et 10^7) afin de comptabiliser le nombre d'occurrences d'événements redoutés grâce à des compteurs : p_1 pour $P(h(t) < 4)$, p_2 pour $P(h(t) > 10)$, et p_3 pour $P(\theta(t) > 100)$. Pour chaque itération de la boucle, on calcule tout d'abord le temps de saut déterministe et le type de frontière correspondant, grâce à la fonction *tstar_nouveau_flot_initial* et le vecteur τ des temps de saut aléatoire de chaque composant avec l'appel de la fonction *saut_poisson*. On choisit le minimum des quatre temps ($t_*, \tau_1, \tau_2, \tau_3$) qu'on appelle *tau_total* et le composant incriminé associé (on déclare une variable $io = 1, 2, 3$ pour τ_1, τ_2, τ_3 et $io = 4$ pour t_*).

On calcule ensuite la hauteur et la température à l'instant τ_{total} de nouveau grâce à la fonction *tstar_nouvel_flot_initial* et l'on définit seulement à ce moment-là le nouveau mode du système en fonction de la hauteur et de la température calculées. Ainsi, si le nouveau mode conduit à un événement redouté, on incrémente le compteur p_i associé. A la fin de la boucle, on a ainsi les probabilités cumulées p_1 , p_2 et p_3 en divisant le nombre d'occurrences par le nombre d'histoires observées.

B.1.1 Fichier principal

```
#ifndef MAIN_PROGRAMM_COMPILED
#define MAIN_PROGRAMM_COMPILED

#include<string>
#include<iostream>
#include<vector>
#include"constant.hh"

using namespace std;

////////////////////////////////////
//Fonction Mise à jour du flot////////
////////////////////////////////////

extern void tstar_nouvel_flot_initial(int,
double &, int &,
double &, double &,
double,
string, string, string,
double, double);

////////////////////////////////////
//Fonction Calcul du temps de saut////////
////////////////////////////////////

extern double saut_poisson(string,string,string,double,
double,int,double);

////////////////////////////////////
//Fonction Détermination du nouveau mode////////
////////////////////////////////////

extern void calcul_nouvel_etat(string&, string&, string&,
double&, double&,string,string,string,double,double,int,
double,int);

////////////////////////////////////
//Paramètres et initialisation////////
////////////////////////////////////

int main() {

    lambda[1] = 2.0*2.2831e-3;
    lambda[2] = 2.0*2.8571e-3;
    lambda[3] = 2.0*1.5625e-3;

    Q=1.5;
    K=23.88915;
    theta_in=15;

    b1=3.0295; b2=0.7578;
    bc=0.05756;
    bd=0.2301;
    srand48(9);

    FILE *stat = fopen("stat","w");
    double T_TOTAL=1000.0;
    int I = 1000;
    double *TG = new double [I];
    int *probl = new int [I];
    int *prob2 = new int [I];
    int *prob3 = new int [I];
    for(int it=0;it<I;++it) {
        probl[it] = 0;
        prob2[it] = 0;
        prob3[it] = 0;
        TG[it] = (T_TOTAL/I)*it;
    }

    vector<double> Tn;

    int N=100000;
    int it = 0;

    int p1 = 0, p2=0, p3=0;

////////////////////////////////////
//Monte Carlo////////
////////////////////////////////////

for(int hist=0;hist<N;++hist) {
    // cout<<"hist = "<<hist<<": ";
    double T=0.0, h0=7.0, theta0=30.9261;
    string alpha1="0", alpha2="F", alpha3="0";
    int n=0;
    int type_front=0;
    double tau_total, tstar;
    string alphain, alpha2n, alpha3n;
    double h0n, theta0n;

    Tn.clear();
    Tn.push_back(0.0);
    while (T<T_TOTAL) {
        n=n+1;
#ifdef DETAIL_AFFICHAGE
        printf("n= %d T= %f\n",n,T);
#endif

////////////////////////////////////
//Mise à jour du flot////////
////////////////////////////////////

        tstar_nouvel_flot_initial(1,
tstar,type_front,
h0n,theta0n,
tau_total,
alpha1,alpha2,alpha3,theta0,h0);

////////////////////////////////////
//Calcul du temps de saut////////
////////////////////////////////////

        for(int composant=1;composant<=3;++composant) {
            tau[composant]=saut_poisson(alpha1,alpha2,alpha3,
theta0,h0,composant,tstar);
        }

        int j=1;
        double temp=tau[1];
        double tau_min;
        int io;
        for(int i=2;i<=3;i++) {
            if(tau[i]<temp)
                j=i,temp=tau[i];
        }
        io=j, tau_min=temp;

        int io_comp;
        if(tau_min<tstar) {
            tau_total=tau_min;
            io_comp=io;
        }
        else {
            tau_total=tstar;
            io_comp=4;
        }
#ifdef DETAIL_AFFICHAGE
        printf("h0= %f, theta0=%f => ",h0,theta0);
#endif

////////////////////////////////////
//Mise à jour du flot////////
////////////////////////////////////

        tstar_nouvel_flot_initial(0,
tstar,type_front,
h0n,theta0n,
tau_total,
alpha1,alpha2,alpha3,theta0,h0);
        h0=h0n,theta0=theta0n;
#ifdef DETAIL_AFFICHAGE
```

```

        printf("h0= %f, theta0=%f\n",h0,theta0);
    #endif

    //////////////////////////////////////
    ///Détermination du nouveau mode////////
    //////////////////////////////////////

    calcul_nouvel_etat(alpha1n, alpha2n,alpha3n, h0n, theta0n,
        alpha1,alpha2,alpha3,theta0,h0,io_comp,
        tau_total,type_front);

    #ifdef DETAIL_AFFICHAGE
        cout<<"alpha=("<<alpha1<<","<<alpha2<<","<<alpha3<<")
        => ";
    #endif
    alpha1=alpha1n, alpha2=alpha2n, alpha3=alpha3n;
    #ifdef DETAIL_AFFICHAGE
        cout<<"alpha=("<<alpha1<<","<<alpha2<<","<<alpha3<<")\n ";
    #endif

    //////////////////////////////////////
    ///Calcul des probabilités de défaillance////////
    //////////////////////////////////////

    if(alpha1=="arret") {
        it = (int) (T/TG[1]);
        if(type_front==4) {
            ++p1;
            for(int k=it;k<I;++k)
                prob1[k] = prob1[k]+1;
        }

        if(type_front==10) {
            ++p2;
            for(int k=it;k<I;++k)
                prob2[k] = prob2[k]+1;
        }
        if(type_front==100) {
            ++p3;
            for(int k=it;k<I;++k)
                prob3[k] = prob3[k]+1;
        }

        T=T_TOTAL;
    }
    else
        T=T+tau_total;
    }

    if(hist%1000==0)
        printf("p1 = %f p2 = %f p3=%f\n",
            (double) p1/(hist+1),
            (double) p2/(hist+1),(double) p3/(hist+1));
    }
    for(int it=0;it<I;++it)
        fprintf(stat, "%f %f %f %f\n",TG[it],
            (double) prob1[it]/N,
            (double) prob2[it]/N, (double) prob3[it]/N);
    fclose(stat);
}

#endif

```

B.1.2 Calcul du temps de saut et du nouveau mode

```

#include<iostream>
#include<string>
#include<math.h>
#include"constant.hh"

using namespace std;

int poisson(double pmean) {
    double bound= exp(-pmean);
    int count=0;
    double lprod=1.0;
    while(lprod>bound) {
        lprod=lprod*drand48();
        count=count+1;
    }
    return count-1;
}

double pow(double a, double b) {
    return exp(b*log(a));
}

////////////////////////////////////
////Conversion mode-configuration/////
////////////////////////////////////

void alpha_recode(int a_bis[], string alpha1, string alpha2,
string alpha3) {
    if ((alpha1=="0")||(alpha1=="0b"))
        a_bis[1]=1;
    else
        a_bis[1]=0;

    if ((alpha2=="0")||(alpha2=="0b"))
        a_bis[2]=1;
    else
        a_bis[2]=0;

    if ((alpha3=="0")||(alpha3=="0b"))
        a_bis[3]=1;
    else
        a_bis[3]=0;
}

////////////////////////////////////
////Calcul de la fonction a , de la hauteur et de /////
////la température en fonction de la configuration /////
////////////////////////////////////

void calcul_a_theta(double &a_theta, double &h, double &theta,
    string alpha1, string alpha2, string alpha3,
    double theta0, double h0, double t, int composant){
    alpha_recode(a_bis, alpha1,alpha2,alpha3);
    int a1=a_bis[1];
    int a2=a_bis[2];
    int a3=a_bis[3];

    double gamma1=(a1+a2-a3)*Q;
    double gamma2=(a1+a2)*Q*theta_in+K;
    double gamma3=(a1+a2)*Q;

    double lamb = lambda[composant];

    //////////////////////////////////
    ///Cas 1/////
    //////////////////////////////////

    if ((a1==0 && a2==1 &&a3==0)||
        (a1==1 && a2==0 && a3==0)||
        (a1==1 && a2==1 && a3==0)||
        (a1==1 && a2==1 && a3==1)) {
        double beta_user=gamma3/gamma1;
        h=gamma1*t+h0;
        theta=pow(h0,beta_user)*theta0/pow(h,beta_user)
+gamma2/gamma3-gamma2*pow(h0,beta_user)
/(gamma3*pow(h,beta_user));
        a_theta=lamb*(b1*exp(bc*(theta-20))
+b2*exp(-bd*(theta-20)))/(b1+b2);
        type_flot=1;
    }

    //////////////////////////////////
    ///Cas 2/////
    //////////////////////////////////

    else if (a1==0 && a2==0 &&a3==0) {
        h=h0;
        theta=theta0+gamma2*t/h0;

        a_theta=lamb*(b1*exp(bc*(theta-20))
+b2*exp(-bd*(theta-20)))/(b1+b2);
        type_flot=2;
    }

    //////////////////////////////////
    ///Cas 3/////
    //////////////////////////////////

    if ((a1==0 && a2==1 &&a3==1)||
        (a1==1 && a2==0 && a3==1)) {
        h=h0;
        theta=theta0*exp(-gamma3*t/h0)+gamma2/gamma3-
gamma2*exp(-gamma3*t/h0)/gamma3;
        a_theta=lamb*(b1*exp(bc*(theta-20))
+b2*exp(-bd*(theta-20)))/(b1+b2);
        type_flot=3;
    }

    //////////////////////////////////
    ///Cas 4/////
    //////////////////////////////////

    if ((a1==0 && a2==0 && a3==1)) {
        h=gamma1*t+h0;
        if(h<0) {
            #ifdef DETAIL_AFFICHAGE
                cout<<"gamma1= "<<gamma1<<" alpha1 "<<alpha1
                <<" alpha2 "<<alpha2 <<" alpha3 "<<alpha3<<"\n";
                printf(" theta0 = %f,h0 = %f,t = %f,
                composant = %d\n",theta0,h0,t, composant);
                printf("attention h negativen\n");
            #endif
            theta=theta0+gamma2*log(gamma1*t/h0+1)/gamma1;
            a_theta=lamb*(b1*exp(bc*(theta-20))
+b2*exp(-bd*(theta-20)))/(b1+b2);
            type_flot=4;
        }
    }

    //////////////////////////////////
    ///Calcul du temps de saut aléatoire /////
    ///Algorithme de C. Cocozza-Thivent /////
    //////////////////////////////////

    double saut_poisson(string alpha1,string alpha2,
    string alpha3,double theta0, double h0, int composant,
    double tstar) {

        alpha_recode(a_bis,alpha1,alpha2,alpha3);

        if(composant==1)
            if(alpha1=="Fb"||alpha1=="0b")
                return 1000000.0;

        if(composant==2)
            if(alpha2=="Fb"||alpha2=="0b")
                return 1000000.0;

        if(composant==3)
            if(alpha3=="Fb"||alpha3=="0b")
                return 1000000.0;

        double a=0;
        double A=1.0;
        double tau=1000000;
        double M, w;
        w = 100000;
        double a_theta1, a_theta2;
        while (a<tstar) {
            double h,theta;
            calcul_a_theta(a_theta1, h, theta, alpha1,alpha2,alpha3,
            theta0,h0,a,composant);
            calcul_a_theta(a_theta2, h, theta, alpha1,alpha2,alpha3,
            theta0,h0,a+A,composant);
            M = max(a_theta1,a_theta2);
            double n= poisson(M*A);
            int trouve=0;
            if(n!=0) {
                for(int i=0;i<n; ++i) {
                    double u = a + drand48()*A;
                    double v = drand48()*M;
                    double a_theta;
                    calcul_a_theta(a_theta, h, theta,
                    alpha1,alpha2,alpha3,theta0,h0,u,composant);
                    if (v<=a_theta) {
                        trouve=1;
                        if(u<w)
                            w = u;
                    }
                }
            }
        }
    }
}

```

```

if(trouve) {
    return w;
}
    }
    }
    a=a+A;
}
return tau;
}

////////////////////////////////////
///Calcul du nouveau mode en fonction du type de saut ///
////////////////////////////////////

void calcul_nouvel_etat(string &m1, string &m2, string &m3,
double &theta_n, double &h_n,
string alpha1, string alpha2, string alpha3,
double theta0, double h0, int i_o, double tot,
int type_front) {

////////////////////////////////////
/// Sélection du nouveau mode si l'unité 1 est en panne///
////////////////////////////////////

    if(i_o==1) {
#ifdef DETAIL_AFFICHAGE
        printf("Le composant 1 tombe en panne \n");
#endif
        if((alpha1!="Ob")&&(alpha1!="Fb")) {
            int bernoulli=drand48()<0.5?1:0;
            if (bernoulli==1) {
#ifdef DETAIL_AFFICHAGE
                printf("Bernoulli = T, Composant 1 est Ouvert bloque\n");
#endif
                m1="Ob";
            }
            else {
#ifdef DETAIL_AFFICHAGE
                printf("Bernoulli = F, Composant 1 est Ferme bloque\n");
#endif
                m1="Fb";
            }
            else {
                m1=alpha1;
#ifdef DETAIL_AFFICHAGE
                cout <<"hello je suis ici m1\n";
#endif
            }
            m2=alpha2;
            m3=alpha3;
        }

////////////////////////////////////
/// Sélection du nouveau mode si l'unité 2 est en panne///
////////////////////////////////////

        if (i_o==2) {
#ifdef DETAIL_AFFICHAGE
            printf("Le composant 2 tombe en panne\n");
#endif
            if((alpha2!="Ob")&&(alpha2!="Fb")) {
                int bernoulli=drand48()<0.5?1:0;
                if (bernoulli==1) {
#ifdef DETAIL_AFFICHAGE
                    printf("Bernoulli = T, Composant 2 est Ouvert bloque\n");
#endif
                    m2="Ob";
                }
                else {
#ifdef DETAIL_AFFICHAGE
                    printf("Bernoulli = F, Composant 2 est Ferme bloque\n");
#endif
                    m2="Fb";
                }
                else {
                    m2=alpha2;
#ifdef DETAIL_AFFICHAGE
                    cout <<"hello je suis ici m2\n";
#endif
                }
                m1=alpha1;
                m3=alpha3;
            }

////////////////////////////////////
/// Sélection du nouveau mode si l'unité 3 est en panne///
////////////////////////////////////

            if (i_o==3) {
#ifdef DETAIL_AFFICHAGE
                printf("Le composant 3 est en panne\n");
#endif
                if ((alpha3!="Ob")&&(alpha3!="Fb")) {
                    int bernoulli=drand48()<0.5 ;
                    if (bernoulli==1) {
#ifdef DETAIL_AFFICHAGE
                        printf("Bernoulli = T, Composant 3 est Ouvert bloque\n");
#endif
                        m3="Ob";
                    }
                    else {
#ifdef DETAIL_AFFICHAGE
                        printf("Bernoulli = F, Composant 3 est Ferme bloque\n");
#endif
                        m3="Fb";
                    }
                    else {
                        m3=alpha3;
#ifdef DETAIL_AFFICHAGE
                        cout <<"hello je suis ici m3\n";
#endif
                    }
                }
                m1=alpha1;
                m2=alpha2;
            }

////////////////////////////////////
/// Sélection du nouveau mode si l'unité 1 est en panne///
////////////////////////////////////

            if (i_o==4) { // on passe les barrieres
                if (type_front==6) {
#ifdef DETAIL_AFFICHAGE
                    printf("On passe en dessous de 6m appel de la loi de commande \n");
#endif
                }
                if ((alpha1!="Ob")&&(alpha1!="Fb")) {
                    m1="O";
                }
                else {
                    m1=alpha1;
                }

                if ((alpha2!="Ob")&&(alpha2!="Fb")) {
                    m2="O";
                }
                else {
                    m2=alpha2;
                }

                if ((alpha3!="Ob")&&(alpha3!="Fb")) {
                    m3="F";
                }
                else {
                    m3=alpha3;
                }
            }

////////////////////////////////////
/// atteinte de 6 m --> pas de loi de commande ///

            if (type_front==5) {
#ifdef DETAIL_AFFICHAGE
                printf("On passe au dessus de 6m\n");
#endif
            }
            m1=alpha1;
            m2=alpha2;
            m3=alpha3;
        }

////////////////////////////////////
/// atteinte de 8 m --> loi de commande ///

            if (type_front==8) {
#ifdef DETAIL_AFFICHAGE
                printf("On passe au dessus de 8m, appel de la loi de commande \n");
#endif
            }
            if ((alpha1!="Ob")&&(alpha1!="Fb")) {
                m1="F";
            }
            else {
                m1=alpha1;
            }

            if ((alpha2!="Ob")&&(alpha2!="Fb")) {
                m2="F";
            }
            else {
                m2=alpha2;
            }
        }
    }
}

```

```

        if ((alpha3!="0b")&&(alpha3!="Fb")) {
m3="0";
        }
        else {
m3=alpha3;
        }
    }

    ////// atteinte de 10 m --> événement redouté //////

    if (type_front==10) {
#ifdef DETAIL_AFFICHAGE
        printf("Evenement rare la frontiere
depassee est 10m\n");
#endif
        m1="arret";
        m2="arret";
        m3="arret";
    }

    ////// atteinte de 4 m --> événement redouté //////

    if (type_front==4) {
#ifdef DETAIL_AFFICHAGE
        printf("Evenement rare la frontiere
depassee est 4m\n");
#endif
        m1="arret";
        m2="arret";
        m3="arret";
    }

    ////// atteinte de 100 degrés --> événement redouté //////

    if (type_front==100) {
#ifdef DETAIL_AFFICHAGE
        printf("Evenement rare la frontiere
depassee est 100degres\n");
#endif
        m1="arret";
        m2="arret";
        m3="arret";
    }

    ////// pas de frontière atteinte //////

    if (type_front==0) {
#ifdef DETAIL_AFFICHAGE
        cout<<"alpha1 = "<<alpha1<<" alpha2 = "<<alpha2
<<" alpha3 = "<<alpha3<<"\n";
        cout<<"calcul_nouvel_etat : calcul_temps infini\n";
#endif
    }
}

///////////////////////////////////////////////////////////////////
// Calcul du temps de saut et mise à //
// jour du flot en fonction du mode //
///////////////////////////////////////////////////////////////////

void tstar_nouvel_flot_initial(int cle,
double &tstar, int &type_front,
double &hn, double &theta0n,
double tau_total,
string alpha1, string alpha2, string alpha3,
double theta0, double h0) {
    alpha_recode(a_bis, alpha1,alpha2,alpha3);

    int a1=a_bis[1];
    int a2=a_bis[2];
    //int a3=a_bis[3];

    double gamma2=(a1+a2)*Q*theta_in+K;
    double gamma3=(a1+a2)*Q;

    ////// hauteur initiale supérieure à 8m //////

    if(h0>8.0) {

    //////////CAS 1.1 :(0,1,0) et (1,0,0)

        if((alpha1=="F"&&alpha2=="0b"&&alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="Fb")) {
            if(cle) {
                tstar=(10-h0)/Q;
                type_front=10;
            }
        }
    }
}

```

```

    }
    else {
        hn=Q*tau_total+h0;
        theta0n=(h0*theta0)/hn+(Q*theta_in+K)/Q-(h0*(Q*theta_in+K))
/(hn*Q);
    }
}

//////////CAS 1.2 :(1,1,0)

    else if ((alpha1=="0" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="0b" && alpha3=="F")||
(alpha1=="0" && alpha2=="0b" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="0" && alpha3=="F")||
(alpha1=="0b" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="F")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="Fb")) {
        if(cle) {
            tstar=(10.0-h0)/(2.0*Q);
            type_front=10;
        }
        else {
            hn=2.0*Q*tau_total+h0;
            theta0n=(h0*theta0)/hn+(2.0*Q*theta_in+K)/(2.0*Q)
-(h0*(2.0*Q*theta_in+K))/(2.0*hn*Q);
        }
    }

    //////////CAS 1.3 :(1,1,1)

    else if((alpha1=="0" && alpha2=="0b" && alpha3=="0")||
(alpha1=="0" && alpha2=="0b" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="0" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="0")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="0b")) {
        if(cle) {
            tstar=(10.0-h0)/Q ;
            type_front=10;
        }
        else {
            hn=Q*tau_total+h0;
            theta0n=(h0*h0*theta0)/(hn*hn)+(2.0*Q*theta_in+K)/(2.0*Q)-
(h0*h0*(Q*theta_in+K))/(hn*hn*Q);
        }
    }

    //////////CAS 2 :(0,0,0)

    else if ((alpha1=="F" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="Fb")) {
        if(cle) {
            tstar=(100.0-theta0)*h0/K ;
            type_front=100;
        }
        else {
            hn=h0;
            theta0n=theta0+K*tau_total/h0;
        }
    }

    //////////CAS 3 :(0,1,1)

    else if ((alpha1=="F" && alpha2=="0b" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0b")){
        if(cle) {
            tstar=1000000;
            type_front=0;
        }
        else {
            hn=h0;
            theta0n=theta0*exp(-gamma3*tau_total/h0)+gamma2/gamma3-
gamma2*exp(-gamma3*tau_total/h0)/gamma3;
        }
    }

    //////////CAS 3 :(1,0,1)

    else if((alpha1=="0b" && alpha2=="F" && alpha3=="0")||
(alpha1=="0b" && alpha2=="F" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="Fb")) {
        if(cle) {
            tstar=1000000 ;
            type_front=0;
        }
        else {
            hn=h0;
            theta0n=theta0*exp(-gamma3*tau_total/h0)+gamma2/gamma3-
gamma2*exp(-gamma3*tau_total/h0)/gamma3;
        }
    }
}

```



```

//////////modes impossibles

else if ((alpha1=="Fb" && alpha2=="0" && alpha3=="F")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="0" && alpha2=="0" && alpha3=="F")||
(alpha1=="0" && alpha2=="0" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="0b")||
(alpha1=="0" && alpha2=="F" && alpha3=="0")||
(alpha1=="0" && alpha2=="F" && alpha3=="0b")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="0b")||
(alpha1=="F" && alpha2=="F" && alpha3=="0")||
(alpha1=="F" && alpha2=="F" && alpha3=="0b")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="0b")) {
    if(cle) {
tstar=100000000.0;
type_front=1000000;
cout<<"alpha1 = "<<alpha1<<" alpha2 = "<<alpha2
<<" alpha3 = "<<alpha3<<"\n";
printf("saut_deterministe() cle = %d provoquer
erreur fatale\n mode ",cle);
cout <<"h0 = "<<h0<<" theta0 = "<<theta0<<"\n";
exit(1);
    }
    else {
cout<<"alpha1 = "<<alpha1<<" alpha2 = "<<alpha2
<<" alpha3 = "<<alpha3<<"\n";
printf("tstar_nouvel_flot_initial() cle = %d
provoquer erreur fatale\n mode ",
cle);
exit(1);
    }
}
////////mode 8 : (0,0,1)
////////impossible
}

//////// hauteur initiale égale à 8m //////////

else if(h0==8) {

//////////CAS 1.1 : (0,1,0) et (1,0,0)
if ((alpha1=="F" && alpha2=="0b" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="Fb")) {
    if(cle) {
tstar=(10-h0)/Q;
type_front=10;
    }
    else {
hn=Q*tau_total+h0;
theta0n=(h0*theta0)/hn+(Q*theta_in+K)/(h0*(Q*theta_in+K))
/(hn*Q);
    }
}

//////////CAS 1.2 : (1,1,0)

else if ((alpha1=="0" && alpha2=="0" && alpha3=="F")||
(alpha1=="0" && alpha2=="0b" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="0b" && alpha3=="F")||
(alpha1=="0b" && alpha2=="0" && alpha3=="F")||
(alpha1=="0b" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="F")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="Fb")) {
    if(cle) {
tstar=(10-h0)/(2.0*Q);
type_front=10;
    }
    else {
hn=2.0*Q*tau_total+h0;
theta0n=(h0*theta0)/hn+(2.0*Q*theta_in+K)/(2.0*Q)-
(h0*(2.0*Q*theta_in+K))/(2.0*hn*Q);
    }
}

//////////CAS 1.3 : (1,1,1)

else if ((alpha1=="0" && alpha2=="0" && alpha3=="0b")||
(alpha1=="0" && alpha2=="0b" && alpha3=="0")||

```

```

(alpha1=="0" && alpha2=="0b" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="0" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="0")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="0b")) {
    if(cle) {
tstar=(10.0-h0)/Q ;
type_front=10;
    }
    else {
hn=Q*tau_total+h0;
theta0n=(h0*theta0)/(hn*hn)+(2.0*Q*theta_in+K)/(2.0*Q)
-(h0*theta0*(Q*theta_in+K))/(hn*hn*Q);
    }
}

//////////CAS 2 : (0,0,0)

else if ((alpha1=="F" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="Fb")){
    if(cle) {
tstar=(100.0-theta0)*h0/K ;
type_front=100;
    }
    else {
hn=h0;
theta0n=theta0+K*tau_total/h0;
    }
}

//////////CAS 3 : (0,1,1)

else if ((alpha1=="F" && alpha2=="0b" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0b")) {
    if(cle) {
tstar=1000000;
type_front=0;
    }
    else {
hn=h0;
theta0n=theta0*exp(-gamma3*tau_total/h0)+gamma2/gamma3-
gamma2*exp(-gamma3*tau_total/h0)/gamma3;
    }
}

//////////CAS 3 : (1,0,1)

else if ((alpha1=="0" && alpha2=="F" && alpha3=="0")||
(alpha1=="0" && alpha2=="F" && alpha3=="0b")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="F" && alpha3=="0")||
(alpha1=="0b" && alpha2=="F" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="0b")) {
    if(cle) {
tstar=1000000.0 ;
type_front=0;
    }
    else {
hn=h0;
theta0n=theta0*exp(-gamma3*tau_total/h0)+gamma2/gamma3-
gamma2*exp(-gamma3*tau_total/h0)/gamma3;
    }
}

//////////CAS 4 : (0,0,1)

else if ((alpha1=="F" && alpha2=="F" && alpha3=="0")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="0b")){
    if(cle){
tstar=min((h0-6.0)/Q,(1-exp((theta0-100.0)*Q/K))*h0/Q);
if(((h0-6)/Q)<((1-exp((theta0-100.0)*Q/K))*h0/Q)) {
    type_front=6;
}
}
else {
type_front=100;
}
}
else {
hn=Q*tau_total+h0;
theta0n=theta0-(K/Q)*log(-Q*tau_total/h0+1);
}
}

```

```

    }
}

//////// hauteur initiale comprise entre 6m et 8m //////////

else if (h0>6 && h0<8) {

/////////CAS 1.1 : (0,1,0) et (1,0,0)

    if ((alpha1=="F" && alpha2=="Ob" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="Ob" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="Ob" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="Fb")||
(alpha1=="Ob" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="Ob" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="Ob" && alpha2=="Fb" && alpha3=="Fb")) {
    if(cle) {
tstar=(8-h0)/Q;
type_front=8;
    }
    else {
hn=Q*tau_total+h0;
theta0n=(h0*theta0)/hn+(Q*theta_in+K)/Q-(h0*(
Q*theta_in+K))/(hn*Q);
    }
}

/////////CAS 1.2 : (1,1,0)

    else if ((alpha1=="0" && alpha2=="0" && alpha3=="F")||
(alpha1=="0" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="Ob" && alpha3=="F")||
(alpha1=="0" && alpha2=="Ob" && alpha3=="Fb")||
(alpha1=="Ob" && alpha2=="0" && alpha3=="F")||
(alpha1=="Ob" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="Ob" && alpha2=="Ob" && alpha3=="F")||
(alpha1=="Ob" && alpha2=="Ob" && alpha3=="Fb")) {
    if(cle) {
tstar=(8-h0)/(2.0*Q);
type_front=8;
    }
    else {
hn=2.0*Q*tau_total+h0;
theta0n=(h0*theta0)/hn+(2.0*Q*theta_in+K)/(2.0*Q-
(h0*(2.0*Q*theta_in+K))/(2.0*hn*Q);
    }
}

/////////CAS 1.3 : (1,1,1)

    else if ((alpha1=="0" && alpha2=="0" && alpha3=="Ob")||
(alpha1=="0" && alpha2=="Ob" && alpha3=="0")||
(alpha1=="0" && alpha2=="Ob" && alpha3=="Ob")||
(alpha1=="Ob" && alpha2=="0" && alpha3=="Ob")||
(alpha1=="Ob" && alpha2=="Ob" && alpha3=="0")||
(alpha1=="Ob" && alpha2=="Ob" && alpha3=="Ob")) {
    if(cle) {
tstar=(8.0-h0)/Q ;
type_front=8;
    }
    else {
hn=Q*tau_total+h0;
theta0n=(h0+h0*theta0)/(hn*hn)+(2.0*Q*theta_in+K)/(2.0*Q)
-(h0*h0*(Q*theta_in+K))/(hn*hn*Q);
    }
}

/////////CAS 2 : (0,0,0)

    else if ((alpha1=="F" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="Fb")){
    if(cle) {
tstar=(100.0-theta0)*h0/K ;
type_front=100;
    }
    else {
hn=h0;
theta0n=theta0+K*tau_total/h0;
    }
}

/////////CAS 3 : (0,1,1)

    else if ((alpha1=="F" && alpha2=="Ob" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="Ob" && alpha3=="Ob")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="Ob")||
(alpha1=="Fb" && alpha2=="Ob" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="Ob" && alpha3=="Ob")||
(alpha1=="0" && alpha2=="Ob" && alpha3=="0")||
(alpha1=="0" && alpha2=="Ob" && alpha3=="Ob")||
(alpha1=="Ob" && alpha2=="Ob" && alpha3=="0")||
(alpha1=="Ob" && alpha2=="Ob" && alpha3=="Ob")){
    if(cle) {
tstar=(6.0-h0)/Q ;
type_front=5;
    }
}

```

```

    (alpha1=="Fb" && alpha2=="Ob" && alpha3=="Ob")) {
    if(cle) {
tstar=1000000;
type_front=0;
    }
    else {
hn=h0;
theta0n=theta0*exp(-gamma3*tau_total/h0)+gamma2/gamma3-
gamma2*exp(-gamma3*tau_total/h0)/gamma3;
    }
}

/////////CAS 3 : (1,0,1)

    else if ((alpha1=="0" && alpha2=="F" && alpha3=="0")||
(alpha1=="0" && alpha2=="F" && alpha3=="Ob")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="Ob")||
(alpha1=="Ob" && alpha2=="F" && alpha3=="0")||
(alpha1=="Ob" && alpha2=="F" && alpha3=="Ob")||
(alpha1=="Ob" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="Ob" && alpha2=="Fb" && alpha3=="Ob")) {
    if(cle) {
tstar=1000000.0 ;
type_front=0;
    }
    else {
hn=h0;
theta0n=theta0*exp(-gamma3*tau_total/h0)+gamma2/gamma3-
gamma2*exp(-gamma3*tau_total/h0)/gamma3;
    }
}

/////////CAS 4 : (0,0,1)

    else if ((alpha1=="F" && alpha2=="F" && alpha3=="0")||
(alpha1=="F" && alpha2=="F" && alpha3=="Ob")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="Ob")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="Ob")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="Ob")) {
    if(cle){
tstar=min((h0-6.0)/Q,(1-exp((theta0-100.0)*Q/K))*h0/Q);
if(((h0-6)/Q)<((1-exp((theta0-100.0)*Q/K))*h0/Q)) {
type_front=6;
}
    }
    else {
type_front=100;
    }
}
    else {
hn=-Q*tau_total+h0;
theta0n=theta0-(K/Q)*log(-Q*tau_total/h0+1);
    }
}

//////// hauteur initiale inférieure à 6m //////////

else if (h0<6) {

/////////CAS 1.1 : (0,1,0) et (1,0,0)

    if((alpha1=="Fb" && alpha2=="0" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="Ob" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="Ob" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="Fb")||
(alpha1=="Ob" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="Ob" && alpha2=="Fb" && alpha3=="Fb")) {
    if(cle) {
tstar=(6-h0)/Q;
type_front=5;
    }
    else {
hn=Q*tau_total+h0;
theta0n=(h0*theta0)/hn+(Q*theta_in+K)/Q-(h0*(Q*theta_in+K)
)/(hn*Q);
    }
}

/////////CAS 1.2 : (1,1,0)
/////////impossible

/////////CAS 1.3 : (1,1,1)

    else if ((alpha1=="0" && alpha2=="0" && alpha3=="Ob")||
(alpha1=="0" && alpha2=="Ob" && alpha3=="Ob")||
(alpha1=="Ob" && alpha2=="0" && alpha3=="Ob")||
(alpha1=="Ob" && alpha2=="Ob" && alpha3=="Ob")) {
    if(cle) {
tstar=(6.0-h0)/Q ;
type_front=5;
    }
}

```

```

    }
    else {
hn=Q*tau_total+h0;
theta0n=(h0*h0*theta0)/(hn*hn)+(2.0*Q*theta_in+K)/(2.0*Q)
-(h0*h0*(Q*theta_in+K))/(hn*hn*Q);
    }
}

//////////CAS 2 : (0,0,0)

    else if ((alpha1=="Fb" && alpha2=="Fb" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="Fb")) {
    if(cle) {
tstar=(100.0-theta0)*h0/K ;
type_front=100;
    }
    else {
hn=h0;
theta0n=theta0+K*tau_total/h0;
    }
}

//////////CAS 3 : (0,1,1)

    else if ((alpha1=="Fb" && alpha2=="0b" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0b")) {
    if(cle) {
tstar=1000000;
type_front=0;
    }
    else {
hn=h0;
theta0n=theta0*exp(-gamma3*tau_total/h0)+gamma2/gamma3-
gamma2*exp(-gamma3*tau_total/h0)/gamma3;
    }
}

//////////CAS 3 : (1,0,1)

    else if ((alpha1=="0" && alpha2=="F" && alpha3=="0b")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="F" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="0b")) {
    if(cle) {
tstar=1000000;
type_front=0;
    }
    else {
hn=h0;
theta0n=theta0*exp(-gamma3*tau_total/h0)+gamma2/gamma3-
gamma2*exp(-gamma3*tau_total/h0)/gamma3;
    }
}

//////////CAS 4 : (0,0,1)

    else if ((alpha1=="F" && alpha2=="F" && alpha3=="0b")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="0b")) {
    if(cle) {
tstar=min((h0-4)/Q,(1-exp((theta0-100.0)*Q/K))*h0/Q);
if(((h0-4)/Q)<((1-exp((theta0-100.0)*Q/K))*h0/Q))
type_front=4;
    else
type_front=100;
    }
    else {
hn=-Q*tau_total+h0;
theta0n=theta0-(K/Q)*log(-Q*tau_total/h0+1);
    }
}

//////////modes impossibles

    else if ((alpha1=="F" && alpha2=="0b" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="0b" && alpha3=="0")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="0")||
(alpha1=="F" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="F" && alpha2=="0b" && alpha3=="0")||
(alpha1=="0" && alpha2=="F" && alpha3=="0")||
(alpha1=="0b" && alpha2=="F" && alpha3=="0")||
(alpha1=="F" && alpha2=="F" && alpha3=="0")||
(alpha1=="F" && alpha2=="F" && alpha3=="0")||

```

```

(alpha1=="0" && alpha2=="0" && alpha3=="F")||
(alpha1=="0" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="0b" && alpha3=="F")||
(alpha1=="0" && alpha2=="0b" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="0" && alpha3=="F")||
(alpha1=="0b" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="F")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="Fb")) {
    if(cle) {
tstar=1000000000.0;
type_front=1000000;
cout<<"alpha1 = "<<alpha1<<" alpha2 = "<<alpha2
<<" alpha3 = "<<alpha3<<"\n";
printf("tstar_nouvel_flot_initial () cle = %d
provoque erreur fatale\n, mode ",
cle);
exit(1);
    }
    else {
cout<<"alpha1 = "<<alpha1<<" alpha2 = "<<alpha2
<<" alpha3 = "<<alpha3<<"\n";
printf("tstar_nouvel_flot_initial () cle = %d
provoque erreur fatale\n mode ",
cle);
exit(1);
    }
}

///// hauteur initiale égale à 6m //////////

    else if (h0==6) {

//////////CAS 1.1 : (0,1,0) et (1,0,0)

        if ((alpha1=="F" && alpha2=="0b" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="F")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="F")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="Fb")) {
            if(cle) {
tstar=(8-h0)/Q;
type_front=8;
            }
            else {
hn=Q*tau_total+h0;
theta0n=(h0*theta0)/hn+(Q*theta_in+K)/Q-(h0*(Q*theta_in+K))
/(hn*Q);
            }
        }

//////////CAS 1.2 : (1,1,0)

        else if ((alpha1=="0" && alpha2=="0" && alpha3=="F")||
(alpha1=="0" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="0" && alpha2=="0b" && alpha3=="F")||
(alpha1=="0b" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="0" && alpha3=="Fb")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="F")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="Fb")) {
            if(cle) {
tstar=(8-h0)/(2.0*Q);
type_front=8;
            }
            else {
hn=2.0*Q*tau_total+h0;
theta0n=(h0*theta0)/hn+(2.0*Q*theta_in+K)/(2.0*Q)-
(h0*(2.0*Q*theta_in+K))/(2.0*hn*Q);
            }
        }

//////////CAS 1.3 : (1,1,1)

        else if ((alpha1=="0" && alpha2=="0" && alpha3=="0b")||
(alpha1=="0" && alpha2=="0b" && alpha3=="0")||
(alpha1=="0" && alpha2=="0b" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="0" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="0")||
(alpha1=="0b" && alpha2=="0b" && alpha3=="0b")) {
            if(cle) {
tstar=(8.0-h0)/Q ;
type_front=8;
            }
            else {
hn=Q*tau_total+h0;
theta0n=(h0*h0*theta0)/(hn*hn)+(2.0*Q*theta_in+K)/(2.0*Q)
-(h0*h0*(Q*theta_in+K))/(hn*hn*Q);
            }
        }
    }
}

```

```

////////CAS 2 : (0,0,0)

else if ((alpha1=="F" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="Fb")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="Fb")){
    if(cle) {
tstar=(100.0-theta0)*h0/K ;
type_front=100;
    }
    else {
hn=h0;
theta0n=theta0+K*tau_total/h0;
    }
}

////////CAS 3 : (0,1,1)

else if ((alpha1=="F" && alpha2=="0b" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="0" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="0b" && alpha3=="0b")) {
    if(cle) {
tstar=1000000;
type_front=0;
    }
    else {
hn=h0;
theta0n=theta0*exp(-gamma3*tau_total/h0)+gamma2/gamma3-
gamma2*exp(-gamma3*tau_total/h0)/gamma3;
    }
}

////////CAS 3 : (1,0,1)

else if ((alpha1=="0" && alpha2=="F" && alpha3=="0")||
(alpha1=="0" && alpha2=="F" && alpha3=="0b")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="0" && alpha2=="Fb" && alpha3=="0b"))||

```

```

(alpha1=="0b" && alpha2=="F" && alpha3=="0")||
(alpha1=="0b" && alpha2=="F" && alpha3=="0b")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="0b" && alpha2=="Fb" && alpha3=="Fb")) {
    if(cle) {
tstar=1000000.0 ;
type_front=0;
    }
    else {
hn=h0;
theta0n=theta0*exp(-gamma3*tau_total/h0)+gamma2/gamma3-
gamma2*exp(-gamma3*tau_total/h0)/gamma3;
    }
}

////////CAS 4 : (0,0,1)

else if ((alpha1=="F" && alpha2=="F" && alpha3=="0")||
(alpha1=="F" && alpha2=="F" && alpha3=="0b")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="F" && alpha2=="Fb" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="F" && alpha3=="0b")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="0")||
(alpha1=="Fb" && alpha2=="Fb" && alpha3=="0b")) {
    if(cle){
tstar=min((h0-4.0)/Q, (1-exp((theta0-100.0)*Q/K))*h0/Q);
if(((h0-4)/Q)<((1-exp((theta0-100.0)*Q/K))*h0/Q)) {
    type_front=4;
}
    }
    else {
type_front=100;
    }
    }
    else {
hn=-Q*tau_total+h0;
theta0n=theta0-(K/Q)*log(-Q*tau_total/h0+1);
    }
}
}
}

```

B.2 Approche par les EDP

Le programme permettant de calculer la probabilité de surchauffe du réservoir (l'algorithme est décrit dans le Chapitre 4) est composé de quatre fichiers.

- Le fichier *fichier principal* appelle les fonctions calculées dans le fichier *Equations* dans l'ordre d'implémentation permettant d'obtenir la probabilité de surchauffe en partant du mode initial.
- Le fichier *Paramètres* définit tous les paramètres du réservoir, le schéma de discrétisation utilisé, ainsi que les suites permettant de définir les domaines de définition E_{24} et E_{27} .
- Le fichier *Equations* calcule dans l'ordre d'implémentation les fonctions selon le schéma de discrétisation défini dans la Section 4.3.2.
- Le fichier *Interpolations* calcule les fonctions interpolées sur les grilles redressées.

B.2.1 Fichier principal

```
stacksize(100000000);

exec('parametres.sce');

getf('equations2.sci');

getf('interpolations.sci');

f36=equation(36);
//plot3d(theta,h((Nh/2+1):(Nh+1)),f36);

f35=equation(35);
//plot3d(theta,h((Nh/2+1):(Nh+1)),f35);

[psi36 psi036]=interpolation1(f36,36);

psi35=interpolationCI(f35,27);

[g34 g034]=equation6_7(34,psi35);
//plot3d(w(:,Nh/2+1),h((Nh/2+1):(Nh+1)),g34);
//plot2d(w0,g034);

f29=equation(29);
//plot3d(theta,h(VNh_2Nh1),f29);

f25=equation(25);
//plot3d(theta,h((Nh/2+1):(Nh+1)),f25);

psi25=interpolationCI(f25,27);

[g28 g028]=equation6_7(28,psi25);
//plot3d(w(:,Nh/2+1),h((Nh/2+1):(Nh+1)),g28);
//plot2d(w(:,Nh/2+1),g28);
//plot2d(w0,g028);

psi29=interpolationCI(f29,24);

g26=equation(26);

//plot3d(z(:,Nh/2+1),h(VNh_2Nh1),g26);

f14=equation(14);
//plot2d(theta,f14);

f10=equation(10);
//plot2d(theta,f10);

f9=equation(9);
//plot2d(h(VNh_34Nh1),f9);

f8=equation(8);

f7=equation(7);

f2=equation(2);

f3=equation(3);
//plot2d(h(VNh_34Nh1),f3);

[psi29b psi029b]=interpolation1(f29,36);
//plot3d(w(:,Nh/2+1),h((Nh/2+1):(Nh+1)),psi29b);

[psi35b psi035b]=interpolation1(f35,24);
//plot3d(w(:,Nh/2+1),h((Nh/2+1):(Nh+1)),psi35b);

[psi25b psi025b]=interpolation1(f25,24);
//plot3d(w(:,Nh/2+1),h((Nh/2+1):(Nh+1)),psi25b);

psi24=ones(compt+1,Nh/2+1);

[g0]=equation24_27(psi24);
//plot3d(w(:,Nh/2+1),h((Nh/2+1):(Nh+1)),g27(:,VNh_21));
//plot3d(z(:,Nh/2+1),h(VNh_2Nh1),g26);
//g27=equation24_27(2);
//g27=equation(27);

f1=equation(1);
```

B.2.2 Fichier Paramètres

```

clear
//////////
//parametres//
//////////

K=23.88915;
Q=1.5;
theta_in=15;
theta_1=30.9261;
b1=3.0295;
b2=0.7578;
bc=0.05756;
bd=0.2301; //3.0295
l=[2.2831E-3,2.8751E-3,1.5625E-3]; //vecteur des lambda ^

dtheta=input("choisir un pas") //pas de la grille
dh=(4*Q*dtheta*0.95)/K;
Ntheta=floor((100-theta_1)/dtheta)
Nh=floor(4/dh)

VNh1=1:(Nh+1); /// 4:8
VNH1:Nh;
VNth1=1:(Ntheta+1);
VNth1:Ntheta;
VNh_21=1:(Nh/2+1); ///4:6 (ou 6:8)
VNh_2=1:Nh/2;
VNh_41=1:(Nh/4+1); /// 4:5 (ou 6:7)
VNh_4=1:Nh/4;
VNh_2Nh1=(Nh/2+1):(Nh+1); /// 6:8
VNh_34Nh1=((3/4)*Nh+1):(Nh+1); /// 7:8
VNh_21Nh34=(Nh/2+1):(3*Nh/4+1); ///6:7

////////Fonction lambda//////////

function[Lambda]=Lambda(t,c)
lambda=l(c)*(b1*exp(bc*(t-20))+b2*exp(-bd*(t-20)))/(b1+b2);
endfunction

//////////Vecteurs theta et h//////////

theta=theta_1+(VNth1-1)*dtheta;
h=4+(VNH1-1)*dh;

//////////definition de la suite vn//////////
//////////
v0=theta_1+(K*log(7/8))/Q;

vinf=(3*K/Q)*log(4/3)+theta_1;

V=vinf-v0;

compt=0;

while V>=(dtheta/100),
    V=(1/4)*(vinf-theta_1)+(3/4)*(V+(K/Q)*log(3/4));
    compt=compt+1;
end;

////////determination du n0//////////

v26=((3/4)*26)*(K/Q)*log((7/6)*((3/4)^4))+vinf;

v=((3/4)^(1:compt))*(K/Q)*log((7/6)*((3/4)^4))+vinf;
v=[v vinf]; //////////on met v0 a part

Vcompt=1:(compt+1);
Vcompt2=1:(compt+2);

//////////determination de la matrice w//////////

w0=zeros(1,Nh/4+1);

w0=v0-(K/Q)*log(h(VNh_21Nh34)/8); //creation de w0 a part
//sur 6:7

w=ones(compt+1,Nh/2+1);

for n=Vcompt;
    w(n,:)=v(n)-(K/Q)*log(h(VNh_2Nh1)/8);
end;

entw0=floor((w0-theta_1)/dtheta)+1;

entw=floor((w-theta_1)/dtheta)+1;

L0=(w0-theta_1)/dtheta-entw0+1; /// on normalise par dtheta

L=(w-theta_1)/dtheta-entw+1; /// on normalise par dtheta

//////////definition de la suite un//////////
//////////

u1=theta_1+(K/Q)*log(7/6);

uinf=(4*K/Q)*log(4/3)+theta_1;

U=uinf-u1;

comptu=0;

while U>=(dtheta/100),
    U=(1/4)*(uinf-theta_1)+(3/4)*U-(K/Q)*log(4/3);
    comptu=comptu+1;
end;

u26=((3/4)^25)*(K/Q)*log((7/6)*((3/4)^4))+uinf;

u=((3/4)^(2:(comptu)))*(K/Q)*log((7/6)*((3/4)^4))+uinf;

u=[u1 u uinf];

//////////determination de la matrice z//////////

z=ones(comptu+1,Nh/2+1);

for n=Vcompt;
    z(n,:)=6*((h(VNh_2Nh1)).^(-1)).*(u(n)-theta_1)+theta_1;
end;

entz=floor((z-theta_1)/dtheta)+1;

Lu=(z-theta_1)/dtheta-entz+1;

//////////vecteurs lambda//////////

L1=Lambda(theta,1);
L2=Lambda(theta,2);
L3=Lambda(theta,3);
L2w=Lambda(w,2);
L2w0=Lambda(w0,2);
L3w=Lambda(w,3);
L3w0=Lambda(w0,3);
L2z=Lambda(z,2);
L3z=Lambda(z,3);
L1th=Lambda(theta_1,1);
L2th=Lambda(theta_1,2);
L3th=Lambda(theta_1,3);

```

B.2.3 Fichier Equations

```

function[f]=equation(n)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 36 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==36 then,
    f=zeros(Ntheta+1,Nh/2+1);

f(Ntheta+1,:)=1;
gprime=ones(1,Nh+1);
///initialiser la première ligne
fprime=zeros(1,Nh+1);    ///ligne courante

for i=Ntheta:-1:1;

    for j=VNh+1;

        fprime(j)=(dh*K)/(dtheta*Q)*(h(j-1).^(-1))
        .*(gprime(j-1)-fprime(j-1))+fprime(j-1);

        end;
        gprime=fprime;
        f(i,:)=fprime(VNh_2Nh1);
        end;
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 35 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==35 then,
    f=ones(Ntheta+1,Nh/2+1);
    f(Ntheta+1,:)=1;

    for i=Ntheta:-1:1;

        f(i,:)=h(VNh_2Nh1)*dtheta.*(K+2*L3(i)*dtheta
        *h(VNh_2Nh1).^(-1)).*(K.*(h(VNh_2Nh1)*dtheta
        .^(-1)).*f(i+1,:)+L3(i)*(f36(i,:)+1));

        end;
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 29 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==29 then,
    f=zeros(Ntheta+1,Nh/2+1);
    f(Ntheta+1,:)=1;

    for i=Ntheta:-1:1;
        f(i,:)=(K+2*L2(i)*dtheta*h(VNh_2Nh1).^(-1)).*(K*f(i+1,:)
        +dtheta*L2(i).*h(VNh_2Nh1));
        end;
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==25 then,
    f=zeros(Ntheta+1,Nh/2+1);
    f(1,VNh_21)=0.5*f36(1,VNh_21);
    for i=VNth;
        A=(Q*(theta(i+1)-theta_1)/dtheta)*(h(VNh_2Nh1).^(-1));
        f(i+1,:)=f(1,:)+((A+2*L2(i+1)).^(-1)).*(A.*(f(i,:)-f(1,:))
        +L2(i+1).*(f36(i+1,VNh_21)-2*f(1,:)));
        end;
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 26 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==26 then,
    f=zeros(comptu+1,Nh/2+1);
    f(:,Nh/2+1)=psi29(:);
    for j=Nh/2:-1:1;
        f(Vcompt,j)=(2*L2z(Vcompt,j)+Q/dh).^(-1)).*((Q/dh)
        *f(Vcompt,j+1)
        +L2z(Vcompt,j));
        end;
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==14 then,
    f=zeros(Ntheta+1,1);
    f($)=1;
    for i=Ntheta:-1:1;
        f(i)=(K/(8*dtheta)+2*L1(i)).^(-1)).*(L1(i)
        +K/(8*dtheta))*f(i+1));
        end;
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==10 then,
    f=zeros(Ntheta+1,1);
    f($)=1;
    for i=Ntheta:-1:1;
        f(i)=(K/(8*dtheta)+2*(L1(i)+L2(i))).^(-1))
        *(K/(8*dtheta)
        *f(i+1)+L1(i)*f29(i,Nh/2+1)+L2(i)*f14(i));
        end;
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==9 then,
    f=zeros(1,Nh/4+1);
    f($)=f14(1);
    for j= Nh/4:-1:1;
        f(j)=(dh/(2*L1th*dh+Q))*((Q/dh)*f(j+1)+L1th);
        end;
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==8 then,
    f=zeros(1,1);
    f=(1/2)*f36(1,Nh/4+1);
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==7 then,
    f=zeros(1,1);
    f=(1/(2*(L1th+L3th)))*(L1th*g034(Nh/4+1)+L3th*(f8+f9(1)));
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==2 then,
    f=zeros(1,1);
    f=(1/(2*(L1th+L2th)))*(L1th*g028(Nh/4+1)+L2th*f8);
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==3 then,
    f=zeros(1,Nh/4+1);
    f($)=f10(1);
    for j=Nh/4:-1:1;
        f(j)=((Q/dh+2*(L1th+L2th)).^(-1))*((Q/dh)*f(j+1)
        +L1th*f29(1,j+Nh/4)
        +L2th*f9(j));
        end;
    end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if n==1 then,
    f=zeros(1,1);
    f=(1/(2*(L1th+L2th+L3th)))*(L1th*g0(Nh/4+1)+L2th
    *f7+L3th*(f2+f3(1)));
    end;

```

```

endfunction;

//-----//

function[g0]=equation24_27(psi);

f=zeros(compt+1,Nh/2+1);
g=zeros(compt+1,Nh/2+1);
g0=zeros(1,Nh/4+1);
g(compt+1,1)=psi(compt+1,1);

norme=1;
x=0;
while norme>(dtheta/10000000);
    for j=VNh_2;
        g(compt+1,j+1)=((Q/dh+2*(L2w(compt+1,j+1)+L3w(compt+1,j+1)))
            .^(-1)).*((Q/dh)*g(compt+1,j)+L2w(compt+1,j+1)
            .*g34(compt+1,j+1)+L3w(compt+1,j+1).*(g28(compt+1,j+1)
            +psi29b(compt+1,j+1)));
    end;

    f(compt+1,Nh/2+1)=g(compt+1,Nh/2+1);

    for j=Nh/2:-1:1;
        f(compt+1,j)=((Q/dh+2*(L2z(compt+1,j)+L3z(compt+1,j)))
            .^(-1)).*((Q/dh)*f(compt+1,j+1)+L2z(compt+1,j+1).*psi35b(compt+1,j)
            +L3z(compt+1,j).*(psi25b(compt+1,j)+g26(compt+1,j)));
    end;

    norme=max(abs(g(compt+1,1)-f(compt+1,1))); //test d'arrêt

    g(compt+1,1)=f(compt+1,1);
    x=x+1;
end;
disp(x)

for i=compt:-1:1;
    g(i,1)=f(i+1,1);
    for j=VNh_2;
        g(i,j+1)=((Q/dh+2*(L2w(i,j+1)+L3w(i,j+1)))
            .^(-1)).*((Q/dh)
            *g(i,j)+L2w(i,j+1).*g34(i,j+1)+L3w(i,j+1).*(g28(i,j+1)
            +psi29b(i,j+1)));
    end;

    f(i,Nh/2+1)=g(i,Nh/2+1);

    for j=Nh/2:-1:1;
        f(i,j)=((Q/dh+2*(L2z(i,j)+L3z(i,j)))
            .^(-1)).*((Q/dh)
            *f(i,j+1)+L2z(i,j).*psi35b(i,j)+L3z(i,j)

```

```

            .*(psi25b(i,j)+g26(i,j)));
    end;
end;

g0(1)=f(1,1);

for j=VNh_4;
    g0(j+1)=((Q/dh+2*(L2w0(j+1)+L3w0(j+1)))
        .^(-1)).*((Q/dh)
        *g0(j)+L2w0(j+1).*g034(j+1)+L3w0(j+1).*(g028(j+1)
        +psi029b(j+1)));
end;

endfunction;

//-----//

function[f,f0]=equation6_7(n,psi);

f=zeros(compt+1,Nh/2+1);
f0=zeros(1,Nh/4+1);
f(:,1)=1;
f0(1)=psi;
if n==34 then,

    for j=VNh_2;
        f(Vcompt,j+1)=((Q/dh+2*L3w(Vcompt,j+1))
            .^(-1)).*((Q/dh)
            *f(Vcompt,j)+L3w(Vcompt,j+1).*(psi36(Vcompt,j+1)+1));
    end;

    for j=VNh_4;
        f0(j+1)=((Q/dh+2*L3w0(j+1))
            .^(-1)).*((Q/dh)*f0(j)
            +L3w0(j+1).*(psi036(j+1)+1));
    end;
end;

if n==28 then,

    for j=VNh_2;
        f(Vcompt,j+1)=((Q/dh+2*L2w(Vcompt,j+1))
            .^(-1)).*((Q/dh)
            *f(Vcompt,j)+L2w(Vcompt,j+1).*psi36(Vcompt,j+1));
    end;

    for j=VNh_4;
        f0(j+1)=((Q/dh+2*L2w0(j+1))
            .^(-1)).*((Q/dh)*f0(j)+L2w0(j+1)
            .*psi036(j+1));
    end;
end;
endfunction;

```


B.2.4 Fichier Interpolations

```
//-----
//-----

function[psi,psi0]=interpolation1(f,p)

    if p==36 then,

        psi=zeros(compt+1,Nh/2+1);
        psi0=zeros(1,Nh/4+1);

        for j=VNh_21;
            psi(Vcompt,j)=(1-L(Vcompt,j)).*f(entw(Vcompt,j)-1,j)
            +L(Vcompt,j).*f(entw(Vcompt,j),j);
        end;

        for j=VNh_41;
            psi0(j)=(1-L0(j)).*f(entw0(j),j)+L0(j).*f(entw0(j)+1,j);
        end;

    end;

    if p==24 then,

        psi=zeros(comptu+1,Nh/2+1);
        psi0=zeros(1,1);
        psi0=0;

        for j=VNh_21;
            psi(Vcompt,j)=(1-Lu(Vcompt,j)).*f(entz(Vcompt,j)-1,j)
            +Lu(Vcompt,j).*f(entz(Vcompt,j),j);
        end;

    end;

endfunction;

//-----
//-----

function[psi]=interpolationCI(f,p)

//////////Conditions initiales pour 34 et 28
    if p==27 then,
        psi=zeros(1,1);
        psi=(1-L0(1)).*f(entw0(1)-1,1)+L0(1).*f(entw0(1),1);
    end;

////////// conditions initiales pour 26
    if p==24 then,
        psi=zeros(comptu+1,1);
        psi(:)=(1-Lu(:,Nh/2+1)).*f(entz(:,Nh/2+1)-1,Nh/2+1)
        +Lu(:,Nh/2+1).*f(entz(:,Nh/2+1),Nh/2+1);
    end;

endfunction;
```


Bibliographie

- [1] T. Aldemir. Computer-assisted markov failure modeling of process control systems. *IEEE Trans. on Reliability*, 36(1) :133–144, 1987.
- [2] T. Aldemir. Utilization of the cell-to-cell mapping technique to construct markov failure models for process control systems. *PSAM Proceedings*, pages 1431–1436, 1991.
- [3] V. Bally and G. Pagès. A quantization algorithm for solving multi-dimensional discrete-time optimal stopping problems. *Bernoulli*, 9(6) :1003–1049, 2003.
- [4] V. Bally, G. Pagès, and J. Printems. A quantization tree method for pricing and hedging multidimensional American options. *Math. Finance*, 15(1) :119–168, 2005.
- [5] S.M. Berman. Optimal timing of antiviral therapy in HIV infection. *J. Appl. Probab.*, 31A :3–15, 1994. Studies in applied probability.
- [6] F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. Polit. Econ.*, 81(3) :637–659, 1973.
- [7] B. Bollobás. The optimal arrangement of producers. *J. London Math. Soc. (2)*, 6 :605–613, 1973.
- [8] E.-K. Boukas, Q. Zhu, and Q. Zhang. Piecewise deterministic Markov process model for flexible manufacturing systems with preventive maintenance. *J. Optim. Theory Appl.*, 81(2) :259–275, 1994.
- [9] N. Bouleau. *Probabilités de l'ingénieur*, volume 1418 of *Actualités Scientifiques et Industrielles*. Hermann, Paris, 1986. Variables aléatoires et simulation.
- [10] J.A. Bucklew and S. Cambanis. Estimating random integrals from noisy observations : sampling designs and their performance. *IEEE Trans. Inform. Theory*, 34(1) :111–127, 1988.
- [11] Y.-H. Chien, S.-H. Sheu, Z.G. Zhang, and E. Love. An extended optimal replacement model of systems subject to shocks. *European J. Oper. Res.*, 175(1) :399–412, 2006.
- [12] J. Chiquet and N. Limnios. A method to compute the transition function of a piecewise deterministic Markov process with application to reliability. *Statist. Probab. Lett.*, 78(12) :1397–1403, 2008.
- [13] J. Chiquet, N. Limnios, and M. Eid. Piecewise deterministic Markov processes applied to fatigue crack growth modelling. *J. Statist. Plann. Inference*, 139(5) :1657–1667, 2009.

- [14] C. Coccozza-Thivent. *Processus stochastiques et fiabilité des systèmes*, volume 28 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer-Verlag, Berlin, 1997.
- [15] O.L.V. Costa. Impulse control of piecewise-deterministic processes via linear programming. *IEEE Trans. Automat. Control*, 36(3) :371–375, 1991.
- [16] O.L.V. Costa and M.H.A. Davis. Approximations for optimal stopping of a piecewise-deterministic process. *Math. Control Signals Systems*, 1(2) :123–146, 1988.
- [17] O.L.V. Costa and M.H.A. Davis. Impulse control of piecewise-deterministic processes. *Math. Control Signals Systems*, 2(3) :187–206, 1989.
- [18] O.L.V. Costa and F. Dufour. Stability and ergodicity of piecewise deterministic Markov processes. *SIAM J. Control Optim.*, 47(2) :1053–1077, 2008.
- [19] O.L.V. Costa, C.A.B. Raymundo, and F. Dufour. Optimal stopping with continuous control of piecewise deterministic Markov processes. *Stochastics Stochastics Rep.*, 70(1-2) :41–73, 2000.
- [20] M.H.A. Davis. Piecewise-deterministic Markov processes : a general class of nondiffusion stochastic models. *J. Roy. Statist. Soc. Ser. B*, 46(3) :353–388, 1984. With discussion.
- [21] M.H.A. Davis. *Markov models and optimization*, volume 49 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 1993.
- [22] B. de Saporta, F. Dufour, and K. Gonzalez. Numerical method for optimal stopping of hybrid processes. In *Preprints of the 3rd IFAC International Conference on Analysis and Design of Hybrid Systems 3rd IFAC International Conference on Analysis and Design of Hybrid Systems*, pages 114–119, Zaragoza Espagne, 2009.
- [23] B. de Saporta, F. Dufour, and K. Gonzalez. Numerical method for optimal stopping of piecewise deterministic markov processes. *Ann. Appl. Probab.*, 20(5) :1607–1637, 2010.
- [24] B. de Saporta, F. Dufour, H. Zhang, and C. Elegbede. Optimal stopping for predictive maintenance. In *Proceedings of Lambda-Mu 17 Conference*, 2010.
- [25] M.A.H. Dempster. Optimal control of piecewise deterministic Markov processes. In *Applied stochastic analysis (London, 1989)*, volume 5 of *Stochastics Monogr.*, pages 303–325. Gordon and Breach, New York, 1991.
- [26] M.A.H. Dempster and J.J. Ye. Impulse control of piecewise deterministic Markov processes. *Ann. Appl. Probab.*, 5(2) :399–423, 1995.
- [27] D.L. Deoss and N. Siu. A simulation model for dynamic system availability analysis. *MITNE-287*, 1989.
- [28] J. Devooght. Dynamic reliability. In Jeffery Lewins and Martin Becker, editors, *Advances in Nuclear Science and Technology*, volume 25 of *Advances in Nuclear Science and Technology*, pages 215–278. Springer US, 1997.
- [29] J. Devooght and C. Smidts. Probabilistic reactor dynamics. i : The theory of continuous event trees. *Nuclear Science and Engineering*, 111(3) :229–240, 1992.

- [30] L. Devroye. *Nonuniform random variate generation*. Springer-Verlag, New York, 1986.
- [31] F. Dufour and O.L.V. Costa. Stability of piecewise-deterministic Markov processes. *SIAM J. Control Optim.*, 37(5) :1483–1502 (electronic), 1999.
- [32] Y. Dutuit, E. Châtelet, J.-P. Signoret, and P. Thomas. Dependability modelling and evaluation by using stochastic petri nets : application to two test cases. *Rel. Eng. and System Safety*, 55 :117–124, 1997.
- [33] R.J. Elliott. *Stochastic calculus and applications*, volume 18 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 1982.
- [34] M.H.C. Everdij and H.A.P. Blom. Piecewise deterministic Markov processes represented by dynamically coloured Petri nets. *Stochastics*, 77(1) :1–29, 2005.
- [35] M. Farid and M.H.A. Davis. Optimal consumption and exploration : a case study in piecewise-deterministic Markov modelling. *Ann. Oper. Res.*, 88 :121–137, 1999. Optimal control and differential games (Vienna, 1997).
- [36] T.S. Ferguson. Who solved the secretary problem ? *Statist. Sci.*, 4(3) :282–296, 1989. With comments and a rejoinder by the author.
- [37] K.R. Fister, S. Lenhart, and J.S. McNally. Optimizing chemotherapy in an HIV model. *Electron. J. Differential Equations*, pages No. 32, 12 pp. (electronic), 1998.
- [38] D. Gatarek. On first-order quasi-variational inequalities with integral terms. *Appl. Math. Optim.*, 24(1) :85–98, 1991.
- [39] A. Gersho and R.M. Gray. *Vector quantization and signal compression*. The Kluwer International Series in Engineering and Computer Science. Springer, 2000.
- [40] S. Graf and H. Luschgy. *Foundations of quantization for probability distributions*, volume 1730 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2000.
- [41] R.M. Gray and D.L. Neuhoff. Quantization. *IEEE Trans. Inform. Theory*, 44(6) :2325–2383, 1998. Information theory : 1948–1998.
- [42] U.S. Gugerli. Optimal stopping of a piecewise-deterministic Markov process. *Stochastics*, 19(4) :221–236, 1986.
- [43] T. Hill. Savoir quand s’arreter. *Pour la Science*, (381) :24–30, 2009.
- [44] H.R. Joshi. Optimal control of an HIV immunology model. *Optimal Control Appl. Methods*, 23(4) :199–213, 2002.
- [45] C. Kermisch and P.-E. Labeau. Approche dynamique de la fiabilité des systèmes. *R.E.E.*, 2002. Rapport du Projet ISdF 6/2000.
- [46] D. Kirschner, S. Lenhart, and S. Serbin. Optimal control of the chemotherapy of HIV. *J. Math. Biol.*, 35(7) :775–792, 1997.
- [47] B. Krzykacz-Hausmann. An approximate sensitivity analysis of results from complex computer models in the presence of epistemic and aleatory uncertainties. *Rel. Eng. and System Safety*, 91 :1210–1218, 2006.

- [48] H.J. Kushner. *Probability methods for approximations in stochastic control and for elliptic equations*. Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1977. Mathematics in Science and Engineering, Vol. 129.
- [49] P.-E. Labeau. Approche dynamique de la sûreté de fonctionnement. *R.E.E.*, (8) :52–58, 2008.
- [50] P.-E. Labeau and Y. Dutuit. Fiabilité dynamique et disponibilité de production : Un cas illustratif. In *Proceedings of Lambda-Mu 14 Conference*, Bourges, France, 2004.
- [51] W. Lair, R. Ziani, S. Mercier, and M. Roussignol. Processus markoviens déterministes par morceaux et quantification déterministe avec un schéma de volumes finis : un cas d’étude. In *Proceedings of Lambda-Mu 17 Conference*, 2010.
- [52] A. Lannoy. Maîtrise des risques et sûreté de fonctionnement - repères historiques et méthodologiques. In *Sciences du risque et du danger*, Notes de synthèse et de recherche. Lavoisier, 2008.
- [53] S. Lenhart and Y.C. Liao. Integro-differential equations associated with optimal stopping time of a piecewise-deterministic process. *Stochastics*, 15(3) :183–207, 1985.
- [54] H. Loubergé, S. Villeneuve, and M. Chesney. Long-term risk management of nuclear waste : a real options approach. *J. Econom. Dynam. Control*, 27(1) :157–180, 2002.
- [55] M. Marseguerra and E. Zio. The cell-to-boundary method in monte carlo-based dynamic psa. *Reliability Engineering and System Safety*, 48 :199–204, 1995.
- [56] M. Marseguerra and E. Zio. Monte carlo approach to psa for dynamic process systems. *Reliability Engineering and System Safety*, 52(3) :227–241, 1996.
- [57] S. Mercier. *Modèles stochastiques et méthodes numériques pour la fiabilité*, 2008. Habilitation à diriger des recherches, Université Paris-Est, France.
- [58] R. Myneni. The pricing of the American option. *Ann. Appl. Probab.*, 2(1) :1–23, 1992.
- [59] A. Okabe, B. Boots, K. Sugihara, and S.N. Chiu. *Spatial tessellations : concepts and applications of Voronoi diagrams*. Wiley Series in Probability and Statistics. John Wiley & Sons Ltd., Chichester, second edition, 2000. With a foreword by D. G. Kendall.
- [60] G. Pagès. A space quantization method for numerical integration. *J. Comput. Appl. Math.*, 89(1) :1–38, 1998.
- [61] G. Pagès and H. Pham. Optimal quantization methods for nonlinear filtering with discrete-time observations. *Bernoulli*, 11(5) :893–932, 2005.
- [62] G. Pagès, H. Pham, and J. Printems. An optimal Markovian quantization algorithm for multi-dimensional stochastic control problems. *Stoch. Dyn.*, 4(4) :501–545, 2004.
- [63] G. Pagès, H. Pham, and J. Printems. Optimal quantization methods and applications to numerical problems in finance. In *Handbook of computational and*

- numerical methods in finance*, pages 253–297. Birkhäuser Boston, Boston, MA, 2004.
- [64] G.A. Perez Castaneda. *Evaluation par simulation de la sûreté de fonctionnement de systèmes en contexte dynamique hybride*. PhD thesis, Institut National Polytechnique de Lorraine, 2009.
 - [65] H. Pham. Optimal stopping, free boundary, and American option in a jump-diffusion model. *Appl. Math. Optim.*, 35(2) :145–164, 1997.
 - [66] H. Pham, W. Runggaldier, and A. Sellami. Approximation by quantization of the filter process and applications to optimal stopping problems under partial observation. *Monte Carlo Methods Appl.*, 11(1) :57–81, 2005.
 - [67] S. Rangan and K. Poolla. Time-domain validation for sample-data uncertainty models. *IEEE Trans. Automat. Control*, 41(7) :980–991, 1996.
 - [68] M. Roussignol and D. Flipo. Files d’attente et fiabilité. <http://daniel.flipo.free.fr/cours/ffa.pdf>.
 - [69] A. Sellami. *Méthodes de quantification optimale pour le filtrage et applications à la finance*. PhD thesis, Université Paris Dauphine, 2005.
 - [70] S.M. Shechter, M.D. Bailey, A.J. Schaefer, and M.S. Roberts. The optimal time to initiate hiv therapy under ordered health states. *Operations Research*, 56(1) :20–33, 2008.
 - [71] W. Stadje. Optimal stopping in a cumulative damage model. *OR Spektrum*, 13(1) :31–35, 1991.
 - [72] W. Stadje and D. Zuckerman. Optimal strategies for some repair replacement models. *Adv. in Appl. Probab.*, 22(3) :641–656, 1990.
 - [73] M. Taylor, Howard. Optimal replacement under additive damage and other failure models. *Nav. Res. Log. Quart.*, 22(1) :1–18, 1975.
 - [74] B. Tombuyses and T. Aldemir. Continuous cell-to-cell mapping and dynamic psa. In *Proceedings of ICONE 4 Conference*, 1996.
 - [75] M. Tyran-Kamińska. Substochastic semigroups and densities of piecewise deterministic Markov processes. *J. Math. Anal. Appl.*, 357(2) :385–402, 2009.
 - [76] B. Ycart. *Modèles et algorithmes markoviens*, volume 39 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer-Verlag, Berlin, 2002.
 - [77] S. Zeiser, U. Franz, and V. Liebscher. Autocatalytic genetic networks modeled by piecewise-deterministic Markov processes. *J. Math. Biol.*, 60(2) :207–246, 2010.
 - [78] H. Zhang, F. Dufour, F. Innal, and Y. Dutuit. Dynamic reliability and piecewise-deterministic processes. In *Proceedings of Lambda-Mu 15 Conference*, Lille France, 2006.
 - [79] H. Zhang, K. Gonzalez, F. Dufour, and Y. Dutuit. Piecewise deterministic Markov processes and dynamic reliability. *Proceedings of the Institution of Mechanical Engineers Part O Journal of Risk and Reliability*, 222(04) :545–551, 12 2008.

- [80] D. Zuckerman. Optimal stopping in a semi-Markov shock model. *J. Appl. Probab.*, 15(3) :629–634, 1978.

Résumé

Les Processus Markoviens Déterministes par Morceaux (PDMP) ont été introduits dans la littérature par M.H.A Davis comme une classe générale de modèles stochastiques. Les PDMP forment une famille de processus markoviens qui décrivent une trajectoire déterministe ponctuée par des sauts aléatoires. Dans une première partie, les PDMP sont utilisés pour calculer des probabilités d'événements redoutés pour un cas-test de la fiabilité dynamique (le réservoir chauffé) par deux méthodes numériques différentes : la première est basée sur la résolution du système différentiel décrivant l'évolution physique du réservoir et la seconde utilise le calcul de l'espérance de la fonctionnelle d'un PDMP par un système d'équations intégral-différentielles. Dans la seconde partie, nous proposons une méthode numérique pour approcher la fonction valeur du problème d'arrêt optimal pour un PDMP. Notre approche est basée sur la quantification de la position après saut et le temps inter-sauts de la chaîne de Markov sous-jacente au PDMP, et la discrétisation en temps adaptée à la trajectoire du processus. Ceci nous permet d'obtenir une vitesse de convergence de notre schéma numérique et de calculer un temps d'arrêt ϵ -optimal.

Mots-clés Processus Markoviens Déterministes par Morceaux, méthodes numériques, fiabilité dynamique, fonctionnelle d'un PDMP, problème d'arrêt optimal, quantification, temps d'arrêt ϵ -optimal, vitesse de convergence.

Abstract

Piecewise Deterministic Markov Processes (PDMP's) have been introduced in the literature by M.H.A. Davis as a general class of stochastic models. PDMP's are a family of Markov processes involving deterministic motion punctuated by random jumps. In a first part, PDMP's are used to compute probabilities of top events for a case-study of dynamic reliability (the heated tank system) with two different methods : the first one is based on the resolution of the differential system giving the physical evolution of the tank and the second uses the computation of the functional of a PDMP by a system of integro-differential equations. In the second part, we propose a numerical method to approximate the value function for the optimal stopping problem of a PDMP. Our approach is based on quantization of the post-jump location and inter-arrival time of the Markov chain naturally embedded in the PDMP, and path-adapted time discretization grids. It allows us to derive bounds for the convergence rate of the algorithm and to provide a computable ϵ -optimal stopping time.

Key-words Piecewise Deterministic Markov Processes, numerical methods, dynamic reliability, functional of a PDMP, optimal stopping problem, quantization, ϵ -optimal stopping time, convergence rate.