



HAL
open science

Méthodes ensemblistes garanties pour l'estimation de grandeurs physiques

Isabelle Braems

► **To cite this version:**

Isabelle Braems. Méthodes ensemblistes garanties pour l'estimation de grandeurs physiques. Sciences de l'ingénieur [physics]. Université Paris Sud, 2002. Français. NNT : . tel-03180799

HAL Id: tel-03180799

<https://hal.science/tel-03180799>

Submitted on 25 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

No D'ORDRE : 6972

UNIVERSITE PARIS XI
UFR SCIENTIFIQUE D'ORSAY

THÈSE

Présentée

Pour obtenir

Le GRADE de DOCTEUR EN SCIENCES

DE L'UNIVERSITÉ PARIS XI ORSAY

PAR

Isabelle BRAEMS

Sujet:

MÉTHODES ENSEMBLISTES GARANTIES POUR L'ESTIMATION DE
GRANDEURS PHYSIQUES

Soutenue le 24 Septembre 2002 devant la commission d'examen

M. MARC CONDAT	Président
MME SYLVIANE GENTIL	Rapporteur
M. WOLFRAM LUTHER	Rapporteur
M. MICHEL KIEFFER	Examineur
M. NACIM RAMDANI	Examineur
M. ERIC WALTER	Examineur

“Ainsi, au lieu de ce grand nombre de préceptes dont la logique est composée, je crus que j’aurois assez des quatre suivants, pourvu que je prisse une ferme et constante résolution de ne manquer pas une seule fois à les observer. [...] Le second, de diviser chacune des difficultés que j’examinerois, en autant de parcelles qu’il se pourroit, et qu’il seroit requis pour les mieux résoudre. Le troisième, de conduire par ordre mes pensées, en commençant par les objets les plus simples et les plus aisés à connoître, pour monter peu à peu comme par degrés jusques à la connoissance des plus composés, et supposant même de l’ordre entre ceux qui ne se précèdent point naturellement les uns les autres. Et le dernier, de faire partout des dénombrements si entiers et des revues si générales, que je fusse assuré de ne rien omettre.”

René Descartes
Discours de la méthode, 1637

Remerciements

Marc CONDAT, en acceptant d'être le président de mon jury de thèse, m'a fait un très grand honneur. Nos quelques discussions m'ont permis d'apprécier sa vaste culture et sa grande ouverture d'esprit. Qu'il en soit vivement remercié.

Je souhaite également exprimer toute ma gratitude à Sylviane GENTIL et à Wolfram LUTHER. En acceptant la lourde tâche de rapporteur, ils ont consacré beaucoup de temps à étudier avec minutie mon document, et apporté de nombreux commentaires très utiles, ainsi que de nouvelles perspectives.

Je remercie également Nacim RAMDANI, pour sa collaboration très riche et nos nombreuses discussions, ainsi que d'avoir accepté de faire partie de ce jury.

Je tiens à exprimer mes plus sincères remerciements aux trois personnes qui ont accepté de m'encadrer, membres du Groupe Incertitudes et Modèles, autant pour leur apport scientifique qu'humain.

Mon directeur de thèse, Eric WALTER, allié à une grande rigueur scientifique, une critique pertinente et une grande gentillesse. Nos discussions tant scientifiques que plus littéraires ont été pour moi de grandes sources d'enrichissement. Je n'en saurais trop le remercier de ses conseils avisés.

Ma plus profonde reconnaissance s'adresse également à Michel KIEFFER, qui a su, par sa présence et ses conseils, guider au mieux ma démarche. Je lui dois également, chose importante, de ne plus frémir face à un clavier d'ordinateur.

La participation de Luc JAULIN a été essentielle tout au long de cette thèse : il a su prodiguer sans cesse, avec dynamisme et enthousiasme, nombre de conseils essentiels et orienter utilement mes recherches.

Merci aussi à Fabienne BERTHIER, dont la collaboration m'a aussi beaucoup apporté, aussi pour sa bonne humeur et son énergie.

Mon travail de thèse a été enrichi de nombreuses rencontres...

Plusieurs professeurs m'ont prouvé que l'enthousiasme pour l'enseignement est une chose durable : je suis tout particulièrement reconnaissante envers Jean-Louis BALLOT, Robert MEGY et Guy DEMOMENT. Je tiens en particulier à remercier ce dernier de m'avoir accueillie au sein de son laboratoire et d'avoir toujours proposé une écoute attentive et de nombreux conseils inestimables. Merci également à Pierre BERTRAND.

Certains ont su, par leur gentillesse et leur constante bonne humeur, m'empêcher de me replier exclusivement sur mon travail de thèse. A Michel et Hélène, un très sincère merci pour leur amitié, et leur soutien. "Um beijo e um abraço" spéciaux à Gilney et Mariana, dont la constante gentillesse est un véritable cadeau. Un même merci à Sébastien et Veit, qui m'ont généreusement prêté une oreille attentive dans les moments d'angoisse!

Je remercie également Fabienne, Martha, Mohammed, Emmanuel, Florence et Claude, de leur

présence chaleureuse; l'équipe informatique et l'administration de leur bonne humeur et de leur diligence : Jacques, Jean, Sylvain, Yvette, Daniel, Simone et Hélène.

A Jean-Do que je n'ai jamais pu remercier du soutien qu'il m'a apporté.

Ma reconnaissance va à mes parents et Manue, qui par leur compréhension, leur attention et leur soutien, m'ont permis de finir ce travail dans de bonnes conditions.

Enfin, à azizam.

Table des matières

1	Introduction	7
1.1	Construction de modèles paramétriques	7
1.1.1	Etapes <i>a priori</i>	7
1.1.2	Etapes <i>a posteriori</i>	8
1.1.3	Garantie	9
1.2	Approche ensembliste	9
1.2.1	Ensemble solution	10
1.2.2	Représentation d'une grandeur incertaine	10
1.3	Mise en œuvre numérique	11
1.3.1	Représentations d'ensembles	11
1.3.2	Erreurs numériques du calculateur	12
1.4	Plan du document	14
2	Ensembles	17
2.1	Opérations sur les ensembles	17
2.1.1	Opérations ensemblistes pures	17
2.1.2	Opérations étendues	18
2.1.3	Fonction ensembliste	19
2.1.4	Récipients	20
2.2	Analyse par intervalles	22
2.2.1	Intervalles	22
2.2.2	Pavés	23
2.2.3	Fonction d'inclusion	24
2.3	Implémentation	31
2.3.1	Sous-pavages	31
2.3.2	Approximations intérieure et extérieure	33
2.3.3	Images d'un ensemble par une fonction	34
2.4	Conclusions	36
3	Contracteurs	39
3.1	Contractions de pavés	39
3.1.1	Définitions	40
3.1.2	Contracteurs de CSP linéaires	47
3.1.3	Contracteurs de CSP non-linéaires	48
3.2	Contractions par projection	51
3.2.1	Relation réelle, graphe et projection	51
3.2.2	Projection d'une contrainte primitive	52
3.2.3	Inversion explicite	54

3.2.4	Contraction par décomposition	56
3.2.5	Systèmes de contraintes	66
3.3	Nouveaux algorithmes ensemblistes	70
3.3.1	Caractérisation d'un ensemble	71
3.3.2	Projection d'un ensemble	76
3.3.3	Contrôle du pessimisme lié au phénomène d'enveloppement	83
3.4	Conclusion	88
4	Identification	89
4.1	Introduction	89
4.1.1	Erreur de mesure	91
4.1.2	Erreur de modèle	93
4.2	Modélisation d'une grandeur incertaine	94
4.2.1	Approche stochastique	94
4.2.2	Estimation à erreurs bornées	95
4.3	Facteurs expérimentaux connus	97
4.3.1	Identification paramétrique en électrochimie	97
4.3.2	Mécanisme rédox : une exponentielle	99
4.3.3	Mécanisme de Volmer-Heyrovsky : somme de deux exponentielles	103
4.3.4	Conclusions	107
4.4	Facteurs expérimentaux inconnus	107
4.4.1	Dispositif expérimental	108
4.4.2	Modèle à base de connaissance	109
4.4.3	Approche ensembliste	116
4.4.4	Connaissance exacte des facteurs expérimentaux	121
4.4.5	Prise en compte des incertitudes intrinsèques	126
4.4.6	Réduction du modèle	131
4.4.7	Conclusions et perspectives	132
4.5	Analyse critique de résultats	133
4.5.1	Introduction	134
4.5.2	Classification des données	134
4.5.3	Conclusions	135
4.6	Conclusions	136
5	Caractérisation de structures de modèle	139
5.1	Identifiabilité <i>a priori</i>	140
5.1.1	Identifiabilité structurelle et approches formelles	141
5.1.2	Approches numériques garanties	144
5.2	Discernabilité <i>a priori</i>	150
5.3	Conclusions	153
6	Incertitude de mesure d'un capteur logiciel	155
6.1	Capteurs	156
6.1.1	Caractéristiques métrologiques d'un capteur	158
6.1.2	Evaluation : approche stochastique	159
6.1.3	Evaluation : approche intervalles	160
6.2	Modélisation d'un capteur logiciel	162
6.3	Détermination <i>a priori</i> des caractéristiques d'un capteur	163
6.3.1	Problème direct	163

6.3.2	Problème inverse	165
6.4	Application: étude d'un thermomètre	169
6.4.1	Contexte	170
6.4.2	Problème direct	172
6.4.3	Problème inverse	174
6.5	Conclusions	177
7	Conclusions et perspectives	179
7.1	Les problèmes ensemblistes dans la construction de modèles	179
7.1.1	Avant le recueil des données	180
7.1.2	Après le recueil des données	180
7.2	Nouveaux algorithmes ensemblistes	180
7.3	Perspectives	181
7.3.1	Planification d'expériences	181
7.3.2	Commande	181
8	Annexe	183
8.1	Table des symboles	183
8.1.1	Notations mathématiques ponctuelles	183
8.1.2	Notations ensemblistes	183
8.1.3	Grandeurs relatives à l'estimation	183
8.1.4	Paramètres relatifs aux algorithmes	184
8.2	Mise en œuvre informatique	184
8.2.1	Arrondi sur les réels	184
8.2.2	Arrondi sur les intervalles	185
8.2.3	Garantie des calculs	186
8.3	Evaluation de la chaîne de mesure: cas de données multiples	188
8.3.1	n capteurs physiques	188
8.3.2	Cas d'un capteur unique	191

Chapitre 1

Introduction

1.1 Construction de modèles paramétriques

La construction de modèles paramétriques à partir de données expérimentales est un problème constamment rencontré dans les sciences physiques et les sciences de l'ingénieur.

Souvent, l'objectif d'une telle démarche est de chercher à décrire, à analyser, voire à prédire le comportement du système S siège d'un phénomène à étudier. Selon l'information disponible, deux types de structures de modèles peuvent être envisagés. Le *modèle comportemental* est un modèle purement mathématique dont le vecteur de paramètres \mathbf{x} n'a pas de signification physique particulière. Il est très souvent employé en ingénierie, lorsque les phénomènes étudiés semblent trop complexes pour qu'une description de ceux-ci aboutisse à un modèle utilisable. Sinon, on établit un modèle dit à *base de connaissance* à partir des lois physiques qui décrivent le phénomène. Les composantes de \mathbf{x} sont alors des grandeurs ayant un sens physique de valeur numérique inconnue.

Dans d'autres cas, l'objectif visé est légèrement différent. L'expérimentateur souhaite évaluer un vecteur de grandeurs physiques \mathbf{x} pour lesquelles il ne dispose pas d'appareil de mesure. Il cherche alors à mettre en évidence un effet de ces grandeurs physiques dans le comportement de S . Des données recueillies, il fournit une estimée $\hat{\mathbf{x}}$ de \mathbf{x} à partir du modèle à base de connaissance $\mathcal{M}(\hat{\mathbf{x}})$. Il réalise dans ce cas une mesure indirecte de \mathbf{x} .

Dans tous les cas, la démarche de construction de modèles est un processus itératif dont les étapes peuvent être regroupées en deux blocs, comme l'illustre la figure 1-1. Nous qualifierons d'*a priori* toute réflexion menée avant que les données ne soient recueillies, et d'*a posteriori* les études qui s'appuient sur ces données.

1.1.1 Etapes a priori

L'élaboration de la structure de modèle ou *caractérisation*, est une étape fondamentale. Elle consiste à sélectionner une structure pertinente vis-à-vis de l'information disponible et du comportement escompté du système S . Pour cela, il est nécessaire de pouvoir étudier les propriétés de cette structure de façon aussi indépendante que possible de la valeur que pourra prendre \mathbf{x} par la suite, afin de détecter au plus tôt les éventuels problèmes qui pourront être rencontrés. Ainsi, l'expérimentateur qui hésite entre deux structures de modèle peut s'aider d'une analyse de *discernabilité* entre ces structures. Il est de même fortement recommandé d'étudier l'*identifiabilité* de la structure de modèle $\mathcal{M}(\cdot)$ sélectionnée.

Cette étape franchie, l'expérimentateur doit choisir un ou plusieurs capteurs qui lui permettront de recueillir un vecteur de données expérimentales que nous noterons \mathbf{y} . Le choix de ces capteurs est généralement soumis à un cahier des charges, qui peut imposer par exemple la

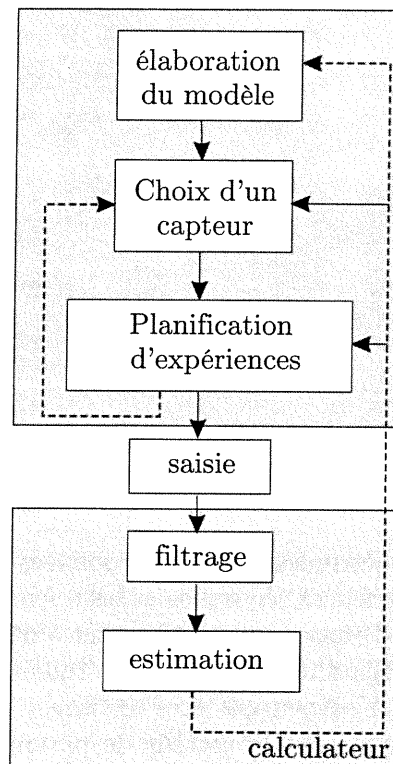


Figure 1-1: Etapes de construction d'un modèle paramétrique

précision avec laquelle \mathbf{x} doit être mesuré. Pour cela, l'expérimentateur s'en remet aux notices techniques des constructeurs, à ses propres tests, et à son expérience.

La *planification d'expériences* peut permettre à l'expérimentateur de définir *a priori* le protocole expérimental le plus adapté à l'objectif visé. Elle est particulièrement importante lorsque le protocole expérimental se révèle long ou coûteux, mais elle n'est pas encore systématiquement prévue. Nous ne l'aborderons pas dans ce document.

Comme aucune donnée n'a encore été recueillie, aucune valeur de \mathbf{x} n'est disponible. Les étapes de caractérisation et de choix d'un capteur consistent donc à tester respectivement la véracité, pour toute valeur de \mathbf{x} (dans un espace de recherche *a priori* si celui-ci est disponible) des propositions

- la structure de modèle $\mathcal{M}(\cdot)$ est pertinente,
- le capteur sélectionné satisfait le cahier des charges.

L'expérimentateur ne pourra alors pleinement être satisfait du résultat obtenu lors de ces tests que si celui-ci est vrai pour toutes les valeurs éventuelles de \mathbf{x} .

1.1.2 Etapes *a posteriori*

Une fois les données recueillies, elles sont généralement soumises à un pré-traitement, encore appelé filtrage, qui regroupe les opérations de fusions de données, de rééchantillonnage et de débruitage. L'estimation, ou identification, consiste ensuite à évaluer le vecteur de paramètres \mathbf{x} à partir des données recueillies et de la structure de modèle $\mathcal{M}(\cdot)$. Pour cela, l'estimateur

ajuste la valeur de \mathbf{x} de manière à ce que la différence de comportement entre le système S et son modèle $M(\mathbf{x})$ soit jugée acceptable, au sens d'un critère à définir.

Dans l'étape d'estimation, le calculateur ne cherche donc plus à vérifier une proposition, mais à caractériser une grandeur incertaine. Pour cela, il doit prendre en compte *l'environnement logique* de l'estimation, c'est-à-dire les hypothèses censées caractériser le contexte expérimental défini *a priori* (structure du modèle, perturbations, incertitudes). Si le résultat de l'estimation contredit les hypothèses formulées, les résultats obtenus lors de l'une des étapes pourront être remis en cause. On pourra critiquer la trop grande simplicité du modèle choisi (étape de caractérisation), le manque de pertinence des données recueillies (modification du protocole expérimental), la présence de données aberrantes (recueil des données) ou les erreurs numériques. Il est donc là aussi nécessaire que le résultat de l'estimation puisse être assuré, ou *garanti*.

1.1.3 Garantie

La construction d'un modèle paramétrique est donc également un processus bouclé, car les conclusions de chaque étape peuvent être mises à profit afin de modifier le processus, puis de le relancer. Afin d'obtenir des conclusions finales fiables, il est donc essentiel que les conclusions obtenues à chaque étape intermédiaire soient dignes de confiance.

Pour cela, l'approche envisagée doit respecter deux conditions. D'une part, elle doit assurer de fournir *toutes* les solutions au problème posé dans l'environnement logique défini. Et, lorsqu'aucune solution n'est acceptable, il est impératif que cette approche puisse le spécifier. D'autre part, il est nécessaire de pouvoir tenir compte des erreurs numériques introduites dans le processus par la présence du calculateur, tout spécialement lorsque des décisions doivent être prises à partir des valeurs numériques obtenues.

Si ces deux conditions sont respectées, nous dirons que le résultat est *garanti*. L'objectif de cette thèse est de montrer qu'il est possible de fournir une réponse *numérique garantie* aux différentes étapes du processus de construction de modèles, en développant des algorithmes de nature ensembliste à l'aide de l'analyse par intervalles. L'approche ensembliste est présentée au paragraphe suivant.

1.2 Approche ensembliste

Cette approche consiste à décrire la solution d'un problème non pas sous la forme d'une unique solution ponctuelle, c'est-à-dire une unique valeur de l'inconnue \mathbf{x} , mais sous la forme d'un ensemble de valeurs \mathbf{x} jugées acceptables. Elle nécessite pour cela l'utilisation d'algorithmes complets.

Un algorithme de résolution d'un problème, ou *résolveur*, est dit *complet* [Gra98] s'il calcule toutes les solutions de ce problème. Dans notre démarche, la propriété de complétude est essentielle : à chaque question posée, le calculateur doit délivrer l'ensemble des réponses correctes. Un algorithme ponctuel peut se révéler suffisant lorsque cette réponse est unique. Dans d'autres cas, il existe un ensemble discret ou continu de réponses, ou encore il n'existe pas de réponse correcte. Il est alors nécessaire que le calculateur puisse rendre compte de ce résultat, ce qui suppose qu'il puisse délivrer en sortie l'ensemble vide, un ensemble continu ou encore un ensemble discret de valeurs. Par opposition aux algorithmes ponctuels, les algorithmes ensemblistes fournissent une évaluation (généralement approchée) d'un ensemble.

Comme les problèmes posés par les étapes *a priori* diffèrent de ceux des étapes *a posteriori*, les techniques classiquement utilisées sont différentes. Nous montrons dans les paragraphes suivants que l'approche ensembliste s'adapte parfaitement à ces deux contextes.

1.2.1 Ensemble solution

Dans les étapes *a priori*, aucune valeur de \mathbf{x} n'a encore été déterminée, seul un éventuel espace de recherche \mathbb{X} *a priori* est disponible. L'expérimentateur souhaite alors déterminer si la propriété \mathcal{P} est vraie pour toute valeur de \mathbf{x} dans \mathbb{X} . L'approche formelle est utilisée le plus fréquemment, mais elle peut se révéler inutilisable ou très coûteuse en charge mémoire, et les résultats obtenus ne sont pas toujours aisément manipulables. A contrario, il est possible de répondre à cette question de façon numérique et garantie en déterminant l'ensemble \mathbb{S} des valeurs de \mathbf{x} qui satisfont \mathcal{P} . S'il est prouvé que \mathbb{S} est vide, alors aucune valeur de \mathbf{x} ne satisfait la propriété \mathcal{P} . Dans le cas contraire, en plus de prouver qu'il existe des valeurs de \mathbf{x} qui satisfont \mathcal{P} , il sera possible de fournir l'ensemble de ces valeurs.

Remarque 1.2.1 Dans l'approche ensembliste, la proposition "la propriété \mathcal{P} est vraie sur \mathbb{S} " implique qu'elle soit vraie pour chacun des éléments de \mathbb{S} . Cependant, la proposition " \mathcal{P} n'est pas vraie sur \mathbb{S} " n'implique pas qu'elle soit fausse sur tout \mathbb{S} , mais seulement qu'il existe des éléments de \mathbb{S} qui ne satisfont pas \mathcal{P} . Les deux propositions " \mathcal{P} n'est pas vraie sur \mathbb{S} " et " \mathcal{P} est fausse sur \mathbb{S} " ne sont donc pas équivalentes.

1.2.2 Représentation d'une grandeur incertaine

Dans les étapes *a posteriori*, les données ont été recueillies. Le vecteur \mathbf{x} caractérise désormais une grandeur incertaine, de valeur numérique inconnue. L'objectif est alors de décrire cette grandeur incertaine.

De nombreuses techniques d'estimation de paramètres ou d'état s'inscrivent dans un cadre probabiliste. Elles fournissent alors l'estimée $\hat{\mathbf{x}}$ de \mathbf{x} sous la forme d'un point, associé éventuellement à une région de confiance. L'approche ensembliste est alors utilisée dans la description de la région de confiance, à partir d'un calcul sur les lois de probabilités de \mathbf{x} et \mathbf{y} (voir la figure 1-2), et de la sortie de modèle notée $\boldsymbol{\eta}(\mathbf{x})$.

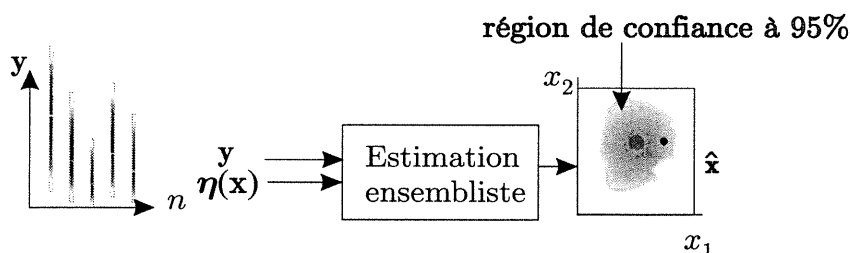


Figure 1-2: Estimation ensembliste dans un contexte statistique

Dans cette approche, la grandeur incertaine \mathbf{x} de \mathbb{R}^m est modélisée par une variable aléatoire X à laquelle on associe une densité de probabilité $\mathcal{P}_X(\cdot)$. Cette représentation nous permet de manipuler à la fois le *support* de X , défini par

$$\{\mathbf{x} \in \mathbb{R}^m \mid \mathcal{P}_X(x) \neq 0\}$$

et la distribution de cette variable aléatoire sur son support, comme le montre l'exemple suivant.

Exemple 1.2.1 Soient deux grandeurs incertaines x et y modélisées par deux variables aléatoires indépendantes X et Y de densité de probabilités uniformes respectives $\mathcal{P}_X(\cdot)$ et $\mathcal{P}_Y(\cdot)$

égales, définies sur $[0, 1]$. La grandeur $Z = X.Y$ est également une variable aléatoire de densité de probabilité $\mathcal{P}_Z(\cdot)$ définie comme suit :

$$\begin{aligned}\mathcal{P}_Z(z) &= -\ln(z) \text{ si } z \in]0, 1] \\ \mathcal{P}_Z(z) &= 0 \text{ si } z \geq 1 \text{ ou } z < 0.\end{aligned}$$

Ce calcul relativement simple peut se révéler très compliqué lorsque les densités $\mathcal{P}_X(\cdot)$ et $\mathcal{P}_Y(\cdot)$ ne sont ni uniformes, ni gaussiennes. Le calcul de la loi de probabilité associée à une estimée fait ainsi l'objet de nombreux problèmes inverses en statistique.

La représentation ensembliste suivante est un outil de représentation plus pauvre, mais qui permet d'effectuer très simplement de nombreux calculs.

Exemple 1.2.1 (suite) Modélisons les grandeurs x , y et z par leur trois supports $\mathbb{X} = [0, 1]$, $\mathbb{Y} = [0, 1]$ et \mathbb{Z} , sans information sur leur distribution de probabilité. Un calcul très simple, que nous détaillerons dans le chapitre 2, permet d'obtenir le support de z : $\mathbb{Z} = \mathbb{X}.\mathbb{Y} = [0, 1]$, et ceci quelles que soient les densités de probabilité éventuelles $\mathcal{P}_X(\cdot)$ et $\mathcal{P}_Y(\cdot)$. Modéliser une grandeur incertaine x par son support \mathbb{X} ne revient pas à modéliser x par une variable aléatoire de densité de probabilité $\mathcal{P}_X(\cdot)$ uniforme. Il s'agit d'un autre choix de modélisation, qui permet d'accéder plus simplement au support d'une grandeur incertaine.

Cette approche peut être utilisée pour modéliser les données \mathbf{y} ainsi que l'estimée \mathbf{x} . Il s'agit alors d'un contexte d'estimation à erreurs bornées (voir la figure 1-3). Seul un support $[y_i]$ est attribué à chaque donnée y_i . Ceci permet de calculer, avec les outils de l'analyse par intervalles, des approximations garanties du support de l'estimée $\hat{\mathbf{x}}$ noté \mathbb{S} .

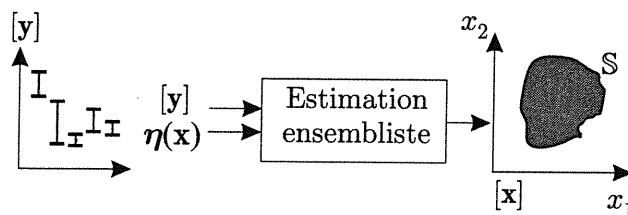


Figure 1-3: Mesure ensembliste dans un contexte déterministe

La description ensembliste représente ici également une approche originale. Elle s'inscrit donc comme un outil intéressant, capable de traiter à la fois des problèmes gérés par des outils de calcul formel ou par des approches probabilistes. Cependant, pour que cette approche soit applicable, il est nécessaire de disposer d'une représentation d'ensembles sous une forme facilement manipulable à l'aide d'un ordinateur.

1.3 Mise en œuvre numérique

1.3.1 Représentations d'ensembles

Parmi les nombreuses techniques de descriptions d'ensembles disponibles, celle que nous allons sélectionner doit faciliter :

- la représentation d'un ensemble en machine,

- la réalisation d'opérations sur ces ensembles,
- la vérification d'une propriété \mathcal{P} pour tout élément de l'ensemble.

Enumérons quelques représentations envisageables [Dor00], dont le tableau 1.1 résume les caractéristiques.

Dans la description par un *nuage de points*, le sous-ensemble \mathbb{X} de \mathbb{R}^m est représenté par un sous-ensemble fini $\mathcal{N}(\mathbb{X})$ de \mathbb{X} . Les éléments \mathbf{x} de $\mathcal{N}(\mathbb{X})$ sont obtenus généralement par des techniques d'exploration aléatoire (méthodes de Monte Carlo) ou déterministes (utilisation d'une grille). Cette sous-représentation est de manipulation aisée, mais il n'est pas possible d'établir de façon garantie les propriétés de \mathbb{X} à partir de celles de $\mathcal{N}(\mathbb{X})$ (si ce n'est que $\mathcal{N}(\mathbb{X}) \neq \emptyset \Rightarrow \mathbb{X} \neq \emptyset$)

La description de \mathbb{X} par un *système d'équations et d'inégalités* est exacte et complète, mais l'étude des propriétés de \mathbb{X} nécessite la manipulation souvent excessivement lourde de ce système, même si les approches formelles sont, dans certains cas, très utiles.

La description par *recouvrement* suppose de disposer d'une classe d'objets, que nous appellerons *réceptifs*, telle qu'il soit possible de recouvrir \mathbb{X} par un nombre fini de ces réceptifs. Cette représentation est intéressante puisqu'elle peut permettre la vérification d'une propriété sur tout élément de l'ensemble. Le principal obstacle de cette approche réside dans le choix du réceptif, dont la représentation en machine doit être aisée, et pour lequel des règles de calcul doivent être établies.

La description par *encadrement* de \mathbb{X} par deux ensembles $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$ tels que

$$\underline{\mathbb{X}} \subset \mathbb{X} \subset \overline{\mathbb{X}},$$

permet également d'assurer la vérification de certaines propriétés de \mathbb{X} à partir des propriétés de $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$. Par exemple, si $\underline{\mathbb{X}}$ n'est pas vide, alors \mathbb{X} n'est pas vide, et il est possible d'encadrer le volume de \mathbb{X} selon $\text{Vol}(\underline{\mathbb{X}}) \leq \text{Vol}(\mathbb{X}) \leq \text{Vol}(\overline{\mathbb{X}})$. La qualité de cette représentation repose sur un choix judicieux de la nature des deux ensembles $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$. Il s'agit en effet de réaliser un compromis entre une description fidèle et une représentation et une manipulation simples de \mathbb{X} .

L'approche que nous exploiterons dans ce document combine avantageusement l'*encadrement* et le *recouvrement*. L'encadrement de \mathbb{X} par $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$ nous permettra de vérifier une propriété \mathcal{P} de façon garantie sur \mathbb{X} . Le recouvrement nous permettra d'obtenir une représentation en machine simple et aisément manipulable des ensembles $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$.

	manipulation	résultat
nuage de points	aisée	non-garanti
système d'équations et d'inégalités	lourde	garanti
encadrement-recouvrement	aisée	garanti

Tableau 1.1: Propriétés des représentations ensemblistes

Cette étape franchie, il s'agit maintenant de pouvoir garantir les calculs numériques que nous allons effectuer sur ces ensembles, et plus précisément sur les éléments de $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$.

1.3.2 Erreurs numériques du calculateur

1.3.2.1 Arithmétique à virgule flottante

Selon la norme IEEE 754 [A.N85], la représentation en machine d'un nombre réel x est un nombre flottant \hat{x} codé à l'aide d'une mantisse $b = b_0b_1\dots b_p$ et d'un exposant e contraint à

appartenir à l'intervalle $[\underline{e}, \bar{e}]$, sous la forme $\hat{x} = \pm b \times \beta^e$. La base β est un entier (en général 2), et la longueur p caractérise le nombre de bits accordés à la description de x . Comme cette longueur est nécessairement finie (p vaut typiquement 53 en format double), l'ensemble des nombres flottants est un ensemble discret et fini, noté \mathcal{R} , qui ne recouvre pas \mathbb{R} . Les nombres x qui ne sont pas exactement représentables en machine sont donc arrondis dès leur entrée dans l'ordinateur.

Exemple 1.3.1 *Comme 0.1 n'est pas exactement représentable en machine, il est automatiquement arrondi au moment même où il est entré. Ainsi, sans un contrôle de l'arrondi, un solveur exact sur les nombres flottants ne trouve pas de solution à l'équation $x - \frac{1}{10} = 0$.*

De ce fait, l'arithmétique à virgule flottante des ordinateurs est une approximation de l'arithmétique exacte. Supposons que les nombres entrés x et y soient exactement représentables, et effectuons l'opération $x \circ y$. Le résultat $z = x \circ y$ n'est pas nécessairement représentable. Dans ce cas, il est automatiquement tronqué ou arrondi. S'il intervient dans d'autres opérations ultérieures, l'erreur commise se propage, ce qui peut entraîner une dérive. Ainsi, même si l'erreur commise à chaque opération est faible, la dérive due à l'accumulation d'erreurs numériques peut être catastrophique. Ce phénomène est particulièrement flagrant lorsqu'on soustrait deux nombres de valeur proche, comme le montre l'exemple suivant.

Exemple 1.3.2 *Soit l'équation quadratique*

$$x^2 - 2sx + r = 0$$

où $s > 0$. Les deux solutions de cette équation sont donc

$$x_1 = s + \sqrt{s^2 - r}, \quad x_2 = s - \sqrt{s^2 - r},$$

de sorte que si r est très petit, l'erreur commise sur x_2 peut être importante. Prenons par exemple $s = 10^{18}$ et calculons x_2 pour $r = 10^{-15}, 10^{-5}, 10^{-1}, 10^{10}$. Pour ces quatre valeurs, le calculateur fournit $x_2 = -21.1875$ pour une représentation en double (voir le tableau 1.2).

r	10^{-15}	10^{-5}	10^{-1}	10^{10}
$x_2 = s - \sqrt{s^2 - r}$	-21.1875	-21.1875	-21.1875	-21.1875
$x_2 = \frac{r}{s + \sqrt{s^2 - r}}$	5.10^{-34}	5.10^{-24}	5.10^{-20}	5.10^{-9}

Tableau 1.2: Evaluations de x_2 pour différentes valeurs de r dans l'exemple 1.3.2

Pour contrer cet effet, connu sous le terme d'*effet d'annulation*¹, une des approches préconisées est de manipuler symboliquement l'expression de manière à regrouper les termes de valeur numérique voisine.

Exemple 1.3.2 (suite) *L'erreur commise précédemment peut être notablement réduite en écrivant $x_2 = \frac{r}{s + \sqrt{s^2 - r}}$. Elle est alors de l'ordre de $\frac{r}{2s}$. Les valeurs obtenues pour cette expression de x_2 sont spécifiées au tableau 1.2.*

¹ *cancellation effect* dans la langue de Wilde

Afin d'insister sur l'importance de la prise en compte de ces erreurs, la société SUN a regroupé sur la page Web [Web1] un panel de catastrophes célèbres dont la cause est imputée aux erreurs numériques du calculateur. Une version détaillée peut également être obtenue à [Web2].

Il est donc essentiel de développer des techniques capables de maîtriser ces erreurs numériques. Pour cela, il est nécessaire de pouvoir analyser l'erreur commise lors de l'exécution d'un programme sur un calculateur.

1.3.2.2 Analyse des erreurs numériques

Deux types de méthode d'analyse des erreurs d'arrondi dans un programme ont été recensées [Che95].

Les méthodes *a priori* regroupent les analyses régressive ou directe. L'analyse régressive quantifie ces erreurs à partir des dérivées partielles du résultat en fonction des entrées et des résultats intermédiaires. L'analyse directe majore l'erreur commise pour chaque opération intermédiaire. Ces deux approches fournissent des résultats très intéressants mais nécessitent une démarche spécifique pour chaque programme testé.

Les méthodes *a posteriori* [BBC⁺97] regroupent les approches probabiliste et déterministe, qui évaluent toutes deux directement sur le calculateur une estimée ou une borne supérieure de l'erreur commise sur le résultat d'intérêt lors de l'exécution du programme. Dans l'approche probabiliste, chaque erreur d'arrondi est modélisée par une variable aléatoire, dont la propagation au sein du programme est évaluée à l'aide de l'arithmétique stochastique. La méthode CESTAC (Contrôle et Estimation STochastique des Arrondis de Calcul) consiste ainsi à exécuter plusieurs fois le même programme en propageant différemment les erreurs d'arrondi. La dispersion des résultats permet d'évaluer l'impact de ces erreurs, et d'en fournir une estimée.

L'approche *a posteriori* dite déterministe est celle que nous suivons dans ce document. Elle consiste à délaissier la représentation en virgule flottante au profit de la représentation par intervalles. Chaque réel x est donc directement codé par un intervalle $[x]$ qui le contient, et dont les bornes sont exactement représentables en machine. L'extension des opérations exactes effectuées sur les réels aux opérations sur les intervalles est effectuée dans le cadre de l'arithmétique des intervalles, de manière à assurer que l'intervalle résultat contient le résultat de l'opération exacte effectuée sur les réels [Moo59] [AM01]. Les erreurs d'arrondi sont donc majorées pour chaque opération effectuée par le programme. Nous disposons donc d'une classe de récipients naturelle pour laquelle les opérations sont garanties, celle des intervalles.

1.4 Plan du document

L'objectif de ce mémoire est de montrer comment l'approche ensembliste garantie peut assister l'expérimentateur dans plusieurs des nombreuses prises de décision qui jalonnent la démarche de construction de modèles. La caractérisation garantie d'ensembles a fait l'objet de nombreux travaux au sein du L2S, et de trois thèses [Jau94], [Did97] et [Kie99]. C'est pourquoi nous nous bornerons à rappeler très brièvement dans le chapitre 2 la mise en œuvre des algorithmes ensemblistes à l'aide de l'analyse par intervalles, qui est l'outil de base que nous allons exploiter.

Ces algorithmes s'appuient sur des techniques de partitionnement qui limitent leur application à des problèmes de faible dimension. Pour les problèmes de dimension plus élevée que nous allons traiter dans ce mémoire, il est nécessaire de faire appel à des outils, appelés contracteurs, encore confidentiels dans la communauté de l'analyse par intervalles. Le chapitre 3 introduit ces contracteurs et présente leur mise en œuvre dans les estimateurs ensemblistes classiques. Nous proposons également dans ce chapitre de nouveaux algorithmes ensemblistes dédiés à des tâches plus spécifiques, telles que l'évaluation du volume d'un ensemble et la caractérisation de

la projection d'un ensemble sur un sous-espace. Nous les rencontrerons tout au long des chapitres suivants, qui abordent chacune des étapes de la construction de modèles précédemment évoquée.

La deuxième partie de ce mémoire, qui illustre les résultats obtenus à l'aide de ces nouveaux algorithmes, regroupe les principales contributions de cette thèse. Nous l'avons décomposée en deux sous-parties. Bien que ce choix ne respecte pas l'ordre chronologique, nous commençons par l'étude des étapes *a posteriori*, et ceci pour deux raisons. D'une part, cette étude nous permet d'introduire très simplement les idées principales des techniques ensemblistes. D'autre part, elle illustre la possibilité d'obtenir des résultats en contournant certaines étapes *a priori*, et en particulier l'étape de caractérisation.

Dans les étapes *a posteriori*, les ensembles manipulés représentent les supports de grandeurs incertaines. Deux parties constituent le chapitre 4, qui aborde l'estimation ensembliste garantie dans un contexte à erreurs bornées. Lorsque le vecteur de paramètres à identifier regroupe toutes les grandeurs incertaines recensées, l'inversion ensembliste permet de caractériser l'ensemble des valeurs des paramètres compatibles avec les données recueillies et la structure de modèle sélectionnée. L'illustration de cette approche sur des exemples issus de l'électrochimie est proposée au lecteur désireux de se familiariser avec les techniques ensemblistes. Elle souligne la possibilité d'obtenir des conclusions fiables sur les hypothèses sous-jacentes d'un protocole de mesure expérimental, là où les estimateurs classiques sont généralement piégés et fournissent des conclusions fausses, même pour des modèles très simples. L'objectif de la seconde application consiste à évaluer un dispositif novateur de mesure indirecte de paramètres thermo-physiques. Dans ce cas, il est nécessaire de tenir compte des incertitudes liées aux grandeurs intrinsèques au dispositif, même si la valeur prise par ces grandeurs au cours de l'expérience n'a que peu d'intérêt pour l'expérimentateur. Pour répondre à ce problème, nous exploitons l'algorithme de projection d'ensembles exposé au chapitre précédent. Les résultats obtenus, qui tiennent compte des incertitudes liées au dispositif, sont à nouveau garantis, contrairement aux techniques utilisées jusqu'à présent.

Avec le chapitre 5 nous abordons les étapes *a priori* où, puisqu'aucune donnée n'est disponible, il s'agit de fournir une réponse la plus indépendante possible de la valeur que pourra prendre le vecteur de paramètres après la mesure. Cependant, les propriétés d'une structure de modèle sont classiquement étudiées à l'aide de tests qui ne fournissent pas systématiquement une réponse pour toutes les valeurs de l'espace de recherche. Pour répondre aux attentes légitimes d'un expérimentateur désireux de caractériser une structure sur tout cet espace, nous proposons de nouveaux critères numériques d'identifiabilité et de discernabilité de modèles. Il est alors possible de tester des structures de modèles de façon garantie à l'aide des algorithmes ensemblistes. Cette démarche, qui est à notre connaissance originale, est illustrée par plusieurs exemples.

Enfin, le chapitre 6 aborde le problème de la sélection d'un capteur sous un angle, selon nous, inédit. En particulier y est présentée une méthode d'évaluation garantie des performances de la chaîne de mesure indirecte en fonction des caractéristiques du capteur physique étudié. Nous y développons aussi l'approche complémentaire qui consiste à déterminer les caractéristiques du capteur physique qui permettront de satisfaire les contraintes de précision requises par un cahier des charges. Un exemple d'application y est détaillé.

Le lecteur trouvera aussi dans l'annexe une table des notations et symboles.

L'ensemble de ce travail a fait l'objet des articles [BBJ⁺01b] et [JKBW01], des chapitres d'ouvrages collectifs [BJKW01b], [BBJ⁺01a], [JBKW01] et [KJBW01], et des actes de conférence [BJKW01a], [KWBJ01] et [BJKW02]. Enfin, une application à la synthèse robuste de PID, qui ne sera pas abordée dans ce mémoire, a été acceptée récemment [JBW02].

Chapitre 2

Ensembles

Ce chapitre aborde l'élaboration d'algorithmes ensemblistes. Les règles générales des opérations sur des ensembles sont rappelées dans un premier temps. Le cas particulier des intervalles et des pavés, qui fait appel à l'analyse par intervalles, est ensuite détaillé. Nous présentons enfin la mise en œuvre des algorithmes ensemblistes à l'aide de sous-pavages, qu'illustre le calcul des images directe et inverse d'un ensemble par une fonction ensembliste. Dans ce chapitre ainsi que dans le suivant, les problèmes de mise en œuvre ne sont pas abordés pour plus de clarté. Ils sont traités dans l'annexe. Sauf mention explicite, toutes les définitions concernent donc des nombres réels. La majorité d'entre elles possèdent donc une version spécifique adaptée à la représentation en flottants (voir l'annexe).

2.1 Opérations sur les ensembles

Deux types d'opérations peuvent être envisagées sur les ensembles [Jau00b] :

- les opérations ensemblistes pures, telles que l'intersection, l'union, le produit cartésien et la projection,
- les opérations étendues, dont la définition est obtenue par extension des opérations sur les éléments de ces ensembles.

2.1.1 Opérations ensemblistes pures

Elles se formulent à l'aide d'opérateurs logiques. Soient \mathbb{X} et \mathbb{Y} deux sous-ensembles de \mathbb{R}^m et \mathbb{R}^n . Le produit cartésien \mathbb{Z} de ces deux ensembles est défini par

$$\mathbb{Z} = \mathbb{X} \times \mathbb{Y} = \left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \in \mathbb{R}^{m+n} \mid \mathbf{x} \in \mathbb{X} \wedge \mathbf{y} \in \mathbb{Y} \right\}.$$

L'ensemble \mathbb{Z}' est un sous-ensemble de \mathbb{Z} si $\mathbb{Z}' \subset \mathbb{Z}$, c'est-à-dire si $\forall z' \in \mathbb{Z}', z' \in \mathbb{Z}$. La *projection* $\pi_{\mathbb{X}}(\mathbb{Z}')$ de \mathbb{Z}' sur \mathbb{X} est définie par

$$\pi_{\mathbb{X}}(\mathbb{Z}') = \left\{ \mathbf{x} \in \mathbb{X} \mid \exists \mathbf{y} \in \mathbb{Y}, \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \in \mathbb{Z}' \right\}.$$

Enfin, si $m = n$, l'intersection et l'union de \mathbb{X} et \mathbb{Y} sont définies par

$$\begin{aligned}\mathbb{X} \cap \mathbb{Y} &= \{x \in \mathbb{R}^m \mid x \in \mathbb{X} \wedge y \in \mathbb{Y}\}, \\ \mathbb{X} \cup \mathbb{Y} &= \{x \in \mathbb{R}^m \mid x \in \mathbb{X} \vee y \in \mathbb{Y}\}.\end{aligned}$$

2.1.2 Opérations étendues

Soit $\mathcal{P}(\mathbb{X})$ l'ensemble des parties de \mathbb{X} . L'opération \circ définie sur $\mathbb{X} \times \mathbb{Y}$ est étendue sur $\mathcal{P}(\mathbb{X} \times \mathbb{Y})$ suivant le schéma générique suivant :

$$\mathbb{X}' \circ \mathbb{Y}' \triangleq \{x \circ y, x \in \mathbb{X}', y \in \mathbb{Y}'\}, \quad (2.1)$$

où \mathbb{X}' et \mathbb{Y}' sont deux sous-ensembles quelconques de \mathbb{X} et \mathbb{Y} . L'expression (2.1) qui assure de ne perdre aucun élément, garantit l'opération étendue \circ . Par exemple, la somme de deux sous-ensembles \mathbb{X} et \mathbb{Y} de \mathbb{R}^p est définie sur $\mathcal{P}(\mathbb{R}^{2p})$ par :

$$\mathbb{X} + \mathbb{Y} \triangleq \{x + y, x \in \mathbb{X}, y \in \mathbb{Y}\}. \quad (2.2)$$

Remarque 2.1.1 *De façon générale, les opérations utilisées seront toujours définies sur des espaces fermés. Dans le cas d'ensembles réels, l'espace considéré est donc $\mathcal{P}(\overline{\mathbb{R}})$ où $\overline{\mathbb{R}}$ est la fermeture de \mathbb{R} . Cet espace contient en particulier l'ensemble vide \emptyset et le résultat de la division d'un ensemble \mathbb{X} par un ensemble \mathbb{Y} contenant 0*

$$\frac{\mathbb{X}}{\mathbb{Y}} \triangleq \left\{ \frac{x}{y}, x \in \mathbb{X}, y \in \mathbb{Y} \right\}.$$

L'extension sur $\mathcal{P}(\mathbb{X})$ d'une opération \circ définie sur \mathbb{X} ne conserve pas toutes ses propriétés. En particulier, si (\mathbb{X}, \circ) forme un groupe, $(\mathcal{P}(\mathbb{X}), \circ)$ ne forme qu'un monoïde. Ainsi, si chaque élément x de (\mathbb{X}, \circ) possède un élément symétrique, ce n'est pas le cas de chaque élément de $\mathcal{P}(\mathbb{X})$. La manipulation des opérations étendues requiert donc beaucoup de précautions. Par exemple, les deux assertions

$$\begin{aligned}\mathbb{Z} &= \mathbb{X} + \mathbb{Y}, \\ \mathbb{X} &= \mathbb{Z} - \mathbb{Y}.\end{aligned}$$

ne sont pas équivalentes. L'exemple suivant permet de s'en convaincre.

Exemple 2.1.1 *Prenons $\mathbb{X} = [0, 1]$, $\mathbb{Y} = [-1, 0]$ et $\mathbb{Z} = [0]$. Bien que les règles de calcul sur les intervalles ne soient présentées qu'au paragraphe 2.2, il est facile de vérifier que*

$$\mathbb{Z} - \mathbb{Y} = [0] - [-1, 0] = [0, 1] = \mathbb{X},$$

tandis que

$$\mathbb{X} + \mathbb{Y} = [0, 1] + [-1, 0] = [-1, 1] \not\subseteq \mathbb{Z}.$$

Enfin, les relations d'ordre et d'équivalence sont définies par

$$\begin{aligned}\mathbb{X} \subset \mathbb{Y} &\Leftrightarrow \forall x \in \mathbb{X}, x \in \mathbb{Y}, \\ \mathbb{X} = \mathbb{Y} &\Leftrightarrow (\mathbb{X} \subset \mathbb{Y}) \wedge (\mathbb{Y} \subset \mathbb{X}).\end{aligned}$$

Il reste maintenant à étendre la notion de fonction sur un ensemble.

2.1.3 Fonction ensembliste

Soit \mathbb{X} un ensemble de \mathbb{R}^m , \mathbb{Y} un ensemble de \mathbb{R}^n , et f une fonction de \mathbb{R}^m dans \mathbb{R}^n . On définit l'image directe de \mathbb{X} par f par

$$f(\mathbb{X}) = \{y \in \mathbb{R}^n \mid \exists x \in \mathbb{X}, y = f(x)\} \quad (2.3)$$

et l'image inverse de \mathbb{Y} par f par

$$f^{-1}(\mathbb{Y}) = \{x \in \mathbb{R}^m \mid \exists y \in \mathbb{Y}, f(x) = y\}. \quad (2.4)$$

Les propriétés suivantes seront essentielles à l'élaboration des algorithmes présentés dans ce document. Elles découlent directement des définitions (2.3) et (2.4).

Soient \mathbb{X}_1 et \mathbb{X}_2 (respectivement \mathbb{Y}_1 et \mathbb{Y}_2) deux sous-ensembles de \mathbb{X} (respectivement \mathbb{Y}). Il vient, pour l'intersection :

$$\begin{aligned} f(\mathbb{X}_1 \cap \mathbb{X}_2) &\subset f(\mathbb{X}_1) \cap f(\mathbb{X}_2), \\ f^{-1}(\mathbb{Y}_1 \cap \mathbb{Y}_2) &= f^{-1}(\mathbb{Y}_1) \cap f^{-1}(\mathbb{Y}_2), \end{aligned}$$

et pour l'union

$$\begin{aligned} f(\mathbb{X}_1 \cup \mathbb{X}_2) &= f(\mathbb{X}_1) \cup f(\mathbb{X}_2), \\ f^{-1}(\mathbb{Y}_1 \cup \mathbb{Y}_2) &= f^{-1}(\mathbb{Y}_1) \cup f^{-1}(\mathbb{Y}_2). \end{aligned}$$

La composition des fonctions suit les propriétés

$$\begin{aligned} f \circ f^{-1}(\mathbb{Y}) &\subset \mathbb{Y}, \\ f^{-1} \circ f(\mathbb{X}) &\supset \mathbb{X}, \end{aligned}$$

et

$$\begin{aligned} \mathbb{X}_1 \subset \mathbb{X}_2 &\Rightarrow f(\mathbb{X}_1) \subset f(\mathbb{X}_2) \\ \mathbb{Y}_1 \subset \mathbb{Y}_2 &\Rightarrow f^{-1}(\mathbb{Y}_1) \subset f^{-1}(\mathbb{Y}_2). \end{aligned}$$

Notons enfin que si $\mathbb{X} \subset \mathbb{X}_1 \times \mathbb{X}_2$, alors $\mathbb{X} \subset \pi_{\mathbb{X}_1}(\mathbb{X}) \times \pi_{\mathbb{X}_2}(\mathbb{X})$.

Les réécritures de fonctions classiques dans le cadre des ensembles sont également à manipuler avec une très grande prudence. Pour cela, le théorème suivant nous sera très utile.

Théorème 2.1.1 *Soit la fonction f définie sur \mathbb{X}^m à valeurs dans $\mathbb{Y} \in \mathbb{R}$, pour laquelle nous disposons d'une expression explicite comportant uniquement des fonctions et des opérateurs. Soit \mathbb{F} la fonction de $\mathcal{P}(\mathbb{X}^m)$ dans $\mathcal{P}(\mathbb{Y})$ obtenue en remplaçant chaque occurrence de la variable x_i , $i = 1, \dots, m$, par l'ensemble \mathbb{X}_i , $i = 1, \dots, m$. Alors, en posant*

$$f(\mathbb{X}_1, \dots, \mathbb{X}_m) = \{f(x_1, \dots, x_m) \mid x_1 \in \mathbb{X}_1, \dots, x_m \in \mathbb{X}_m\},$$

on a

$$f(\mathbb{X}_1, \dots, \mathbb{X}_m) \subset \mathbb{F}(\mathbb{X}_1, \dots, \mathbb{X}_m). \quad (2.5)$$

Si l'expression explicite de f utilisée ne fait intervenir qu'une seule occurrence de chaque variable

x_i , $i = 1, \dots, m$, alors on a l'égalité

$$f(\mathbb{X}_1, \dots, \mathbb{X}_m) = \mathbb{F}(\mathbb{X}_1, \dots, \mathbb{X}_m). \quad (2.6)$$

R. Moore a fourni une preuve de ce théorème dans le cas où f est une combinaison de fonctions continues et les ensembles \mathbb{X}_i sont des intervalles fermés [Moo79]. Dans le cas général, la preuve peut être obtenue à l'aide du théorème de propagation que nous présenterons au chapitre 3.

L'inclusion (2.5) est une conséquence du *problème de dépendance*¹ inhérent au calcul ensembliste. Cet effet est illustré par l'exemple suivant.

Exemple 2.1.2 Soit la fonction f définie sur \mathbb{R} par: $f : \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = x - x$. L'extension de cette fonction nulle impose que $f(\mathbb{X}) = \{0\}$ pour tout sous-ensemble \mathbb{X} de \mathbb{R} . Or l'image de tout ensemble \mathbb{X} par la fonction \mathbb{F} vaut $\mathbb{F}(\mathbb{X}) = \mathbb{X} - \mathbb{X}$, qui n'est pas réduit au singleton zéro si \mathbb{X} lui-même n'est pas un singleton. Par exemple, $\mathbb{F}([-1, 1]) = [-2, 2]$. Ce pessimisme est dû au fait que l'ensemble $\mathbb{X} - \mathbb{X} = \{x - y, x \in \mathbb{X}, y \in \mathbb{X}\}$ est différent de l'ensemble $\{x - x, x \in \mathbb{X}\} = \{0\}$.

Lorsque l'égalité (2.6) n'est pas vérifiée, on dit que la fonction \mathbb{F} est *pessimiste*. Dans le cas contraire, elle est dite *optimale*.

2.1.4 Récipients

Les opérations définies précédemment permettent théoriquement de manipuler les ensembles comme des objets. Mais en pratique, il est difficile de coder des ensembles \mathbb{X} de forme quelconque. Au lieu de développer des algorithmes purement ensemblistes, on sélectionne le plus souvent un type d'objet, appelé *récipient*, qui nous permettra de décrire \mathbb{X} .

Soit $\mathbb{I}\mathcal{X}$ l'ensemble des récipients possibles pour \mathbb{X} . Le choix du récipient \mathcal{X} résulte d'un compromis :

- le calcul des opérations ensemblistes sur \mathcal{X} doit être le plus simple possible,
- afin d'assurer la garantie du calcul effectué, le récipient \mathcal{X} doit être une *approximation extérieure* de \mathbb{X} , c'est-à-dire $\mathbb{X} \subset \mathcal{X}$.

Des récipients très variés ont été utilisés dans la littérature. Parmi les plus connus, on trouve les intervalles, les pavés, les zonotopes, les parallélotopes, les ellipsoïdes [MNPLW96]. Nous imposons que $\mathbb{I}\mathcal{X}$ vérifie deux propriétés fondamentales :

1. $\mathbb{I}\mathcal{X}$ ne contient que des éléments convexes,
2. $\mathbb{I}\mathcal{X}$ est fermé pour l'intersection :

$$\mathcal{X}_1 \in \mathbb{I}\mathcal{X}, \mathcal{X}_2 \in \mathbb{I}\mathcal{X} \Rightarrow \mathcal{X}_1 \cap \mathcal{X}_2 \in \mathbb{I}\mathcal{X}.$$

C'est pourquoi la classe de récipients que nous allons considérer par la suite est celle des intervalles et des pavés. Considérons à nouveau une opération \circ de $\mathbb{X} \times \mathbb{Y}$ dans \mathbb{Z} . L'extension de cette opération a été définie, pour deux sous-ensembles \mathbb{X}' et \mathbb{Y}' de \mathbb{X} et \mathbb{Y} selon les règles du paragraphe 2.1.2

$$\mathbb{X}' \circ \mathbb{Y}' = \{x' \circ y' \mid x' \in \mathbb{X}', y' \in \mathbb{Y}'\}.$$

¹appelé *dependency effect* dans la langue de Shakespeare.

Nous étendons alors cette opération au cas des récipients par

$$\mathbb{X}' [\circ] \mathbb{Y}' = [\{x' \circ y' \mid x' \in \mathbb{X}', y' \in \mathbb{Y}'\}] = [\mathbb{X}' \circ \mathbb{Y}'],$$

où $[\mathbb{X}]$ est le plus petit récipient contenant \mathbb{X} , appelé *enveloppe* de \mathbb{X} . On vérifie alors bien $\mathbb{X}' \circ \mathbb{Y}' \subset \mathbb{X}' [\circ] \mathbb{Y}'$.

Parmi l'ensemble de ces opérations étendues, on distinguera en particulier le cas de l'union. On définit pour cela l'opérateur \sqcup par

$$[x] \sqcup [y] = [[x] \cup [y]].$$

De même, on étend les fonctions de la façon suivante

$$[f] (\mathbb{X}') = [\{f(x) \mid x \in \mathbb{X}'\}],$$

de sorte que

$$f (\mathbb{X}') \subset [f] (\mathbb{X}'). \quad (2.7)$$

L'inclusion (2.7) traduit le *phénomène d'enveloppement*² qui caractérise le pessimisme introduit en se contentant à chaque fois d'une approximation extérieure de l'ensemble. L'exemple 2.1.3 illustre le pessimisme dû à l'enveloppement d'un ensemble non connexe. Nous verrons au paragraphe 2.2.2 un deuxième effet de ce phénomène dans le cas multi-dimensionnel.

Exemple 2.1.3 Soit l'ensemble $\mathbb{X} = [1, 4]$. On souhaite déterminer l'ensemble \mathbb{Y} tel que

$$\mathbb{Y}^2 = \mathbb{X} = \{y \in \mathbb{R} \mid y^2 = x, x \in \mathbb{X}\}.$$

Comme l'illustre la figure 2-1, \mathbb{Y} est l'union de deux intervalles

$$\mathbb{Y} = \mathbb{Y}_1 \cup \mathbb{Y}_2 = [-2, -1] \cup [1, 2].$$

Si le récipient \mathcal{Y} choisi pour décrire \mathbb{Y} est un intervalle, alors au mieux $\mathcal{Y} = [-2, 2] \supseteq \mathbb{Y}$.

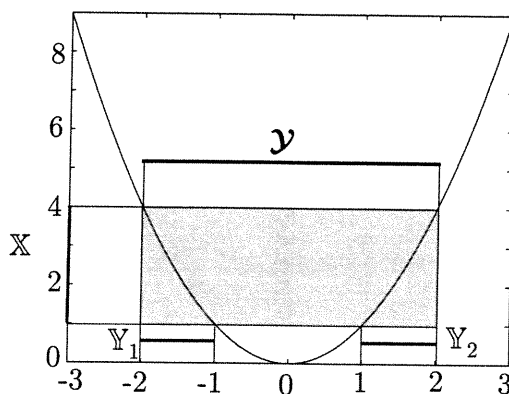


Figure 2-1: Ensemble image \mathbb{Y} et récipient \mathcal{Y} de l'ensemble \mathbb{X} décrits dans l'exemple 2.1.3

²appelé *wrapping effect* dans la langue de Shelley.

Soit $h = f \circ g$. L'extension aux récipients de h , notée $[h]$ ne vérifie alors que l'inclusion

$$[h](\mathbb{X}) \subset [f]([g](\mathbb{X})).$$

Dans le cas où les récipients choisis sont des intervalles et des pavés, ces règles de calcul peuvent être explicitées. Elles forment le cœur de l'analyse par intervalles.

2.2 Analyse par intervalles

L'arithmétique des intervalles, apparue dès les années 1960 ([Moo66], [Han69]), a été initialement développée afin de pouvoir contrôler les erreurs numériques des calculateurs utilisant une représentation à virgule flottante (voir le chapitre 1). Pour cela, elle spécialise les règles du calcul ensembliste au cas où les récipients considérés sont des intervalles dont les bornes sont exactement représentables en machine.

Dans cette partie, nous rappelons très brièvement ces règles de calcul sur des intervalles quelconques. Pour une étude plus complète de cet aspect mathématique, nous renvoyons le lecteur aux ouvrages [Moo79], [Han92] et [Neu90]. La mise en œuvre de ces règles sur ordinateur à l'aide de contrôle d'arrondi, requis pour assurer le caractère garanti du résultat est abordée dans l'annexe. Pour de plus amples informations, consulter entre autres [Wal98], [KM81] [Kul96].

2.2.1 Intervalles

2.2.1.1 Représentation

Définition 2.2.1 *Un intervalle réel³, noté $[x] = [\underline{x}, \bar{x}]$ est l'ensemble satisfaisant*

$$[x] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}. \quad (2.8)$$

Il existe d'autres représentations moins répandues d'un intervalle. Par exemple, il est nécessaire pour l'arithmétique affine [ACS94] de représenter $[x]$ sous la forme

$$[x] = [x_0, r],$$

où x_0 est le centre de l'intervalle $[x]$

$$x_0 \triangleq \text{mid}([x]) = \frac{\underline{x} + \bar{x}}{2},$$

et r est son rayon

$$r \triangleq \frac{1}{2}w([x]) = \frac{\bar{x} - \underline{x}}{2}.$$

La quantité $w([x])$ sera également appelée la taille de $[x]$. Dans la suite, nous utiliserons la notation (2.8). L'ensemble des intervalles sera noté \mathbb{IR} .

Le lecteur remarquera que nous appelons intervalle, comme dans tout le reste du manuscrit si rien n'est spécifié, un intervalle fermé. En particulier, un intervalle $[x]$ est un sous-ensemble fermé borné de \mathbb{R} .

³ Pour alléger le texte, l'adjectif réel ne sera plus spécifié par la suite. Tout intervalle sera donc un intervalle réel.

2.2.1.2 Arithmétique des intervalles

Afin de satisfaire les règles imposées par le calcul ensembliste, on peut formuler une extension intervalles de tous les opérateurs de l'arithmétique des réels. L'avantage principal de cette description est que la quasi-totalité des opérations sur des intervalles peut être obtenue à partir de calculs sur les bornes de ces intervalles. Dans certains cas, le nombre des opérations requises dans un calcul ensembliste manipulant des intervalles est simplement le double de celui correspondant au même calcul effectué sur des réels.

Prenons par exemple le calcul de la somme de deux intervalles. Soit $[x]$ et $[y]$ deux intervalles, on définit alors

$$[z] = [x] + [y] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}].$$

Par exemple⁴, $[1, 2] + [-5, 9] = [-4, 11]$.

L'opérateur de division pose, lui, de plus amples problèmes à cause du fait que la division n'est pas définie partout si l'intervalle diviseur contient zéro. En particulier, il est possible de remplacer l'évaluation pessimiste

$$\frac{2}{[-1, 1]} =]-\infty, \infty[\quad (2.9)$$

par le résultat exact

$$\frac{2}{[-1, 1]} =]-\infty, -2] \cup [2, \infty[.$$

Les deux approches ont été envisagées et comparées dans [Hyv92]. Dans ce document, nous choisirons la première définition (2.9), au détriment de la qualité du résultat, afin d'éviter l'explosion combinatoire due au maniement d'intervalles disjoints.

2.2.2 Pavés

Un pavé (ou un vecteur d'intervalles) $[\mathbf{x}]$ est un sous-ensemble de \mathbb{R}^m résultat du produit cartésien de m intervalles :

$$[\mathbf{x}] = [x_1] \times \dots \times [x_m].$$

Le *pavé enveloppe* $[\mathbb{X}]$ d'un sous-ensemble \mathbb{X} de \mathbb{R}^m est le plus petit pavé contenant \mathbb{X} .

La *taille* $w([\mathbf{x}])$ du pavé $[\mathbf{x}]$ est la plus grande taille de ses composantes :

$$w([\mathbf{x}]) = \max_{i=1, \dots, m} w([x_i]).$$

L'extension au cas multi-dimensionnel des opérations précédentes est immédiat. Il est cependant essentiel de remarquer qu'un deuxième effet du phénomène d'enveloppement survient dans ce cas. L'exemple 2.2.1 illustre le problème dû à la déformation du récipient par une transformation non-linéaire.

Exemple 2.2.1 Soit le pavé $\mathbb{X} = [0, 1]^{x_2}$. On cherche à déterminer l'ensemble \mathbb{Y} défini par

$$\mathbb{Y} = \{ \mathbf{y} \mid y_1 = x_1^2 \text{ et } y_2 = x_1^2 + x_1, \mathbf{x} \in \mathbb{X} \}.$$

⁴Une calculatrice par intervalles, développée en Java, est accessible à [Webh]

Ces deux ensembles sont représentés à la figure 2-2. L'ensemble \mathbb{Y} diffère du pavé $[\mathbb{Y}] = [\hat{y}_1] \times [\hat{y}_2]$ calculé comme

$$\begin{aligned} [\hat{y}_1] &= [0, 1]^2 = [0, 1] \\ [\hat{y}_2] &= [0, 1] + [0, 1] = [0, 2] \end{aligned}$$

(qui est le pavé enveloppe de \mathbb{Y}). Ceci illustre le phénomène d'enveloppement inhérent à notre choix de récipient.

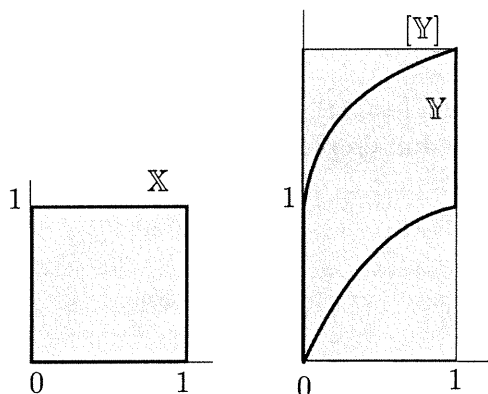


Figure 2-2: Déformation d'un pavé et phénomène d'enveloppement

2.2.3 Fonction d'inclusion

2.2.3.1 Définitions

Définition 2.2.2 Soit une fonction \mathbf{f} de \mathbb{R}^m à valeurs dans \mathbb{R}^n . La fonction intervalles $[\mathbf{f}] : \mathbb{IR}^m \rightarrow \mathbb{IR}^n$ est une fonction d'inclusion pour \mathbf{f} si

$$\forall [x] \in \mathbb{IR}^m, \mathbf{f}([x]) \subset [\mathbf{f}]([x]).$$

La fonction

$$\begin{aligned} [\mathbf{f}] : \mathbb{IR}^m &\rightarrow \mathbb{IR}^n \\ [x] &\mapsto \mathbb{R}^n \end{aligned}$$

est bien une fonction d'inclusion pour n'importe quelle fonction \mathbf{f} de \mathbb{R}^m à valeurs dans \mathbb{R}^n , mais elle est si pessimiste qu'elle n'a aucun intérêt. L'objectif principal de l'analyse par intervalles est de construire des fonctions d'inclusion peu pessimistes, pour lesquelles la taille de $[\mathbf{f}]([x])$ n'est pas trop élevée, tout en assurant la garantie. Sur la figure 2-3 sont représentées les images du même pavé $[x]$ par différentes fonctions d'inclusion.

En raison des phénomènes de dépendance (paragraphe 2.1.3) et d'enveloppement (paragraphe 2.1.4), il est souvent difficile d'élaborer des fonctions d'inclusion optimales. Afin d'assurer la convergence des algorithmes ensemblistes que nous allons utiliser, nous imposerons que toutes les fonctions d'inclusion élaborées vérifient les propriétés du paragraphe suivant.

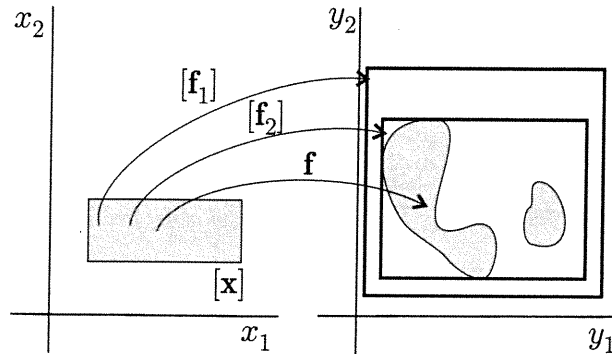


Figure 2-3: Images du pavé $[x]$ par la fonction f et par deux fonctions d'inclusion associée $[f_1]$ et $[f_2]$. $[f_2]$ est moins pessimiste que $[f_1]$

2.2.3.2 Monotonie et convergence

Définition 2.2.3 Une fonction d'inclusion $[f]$ est dite monotone au sens de l'inclusion si

$$[x] \subset [y] \Rightarrow [f]([x]) \subset [f]([y]).$$

Définition 2.2.4 Une fonction d'inclusion $[f]$ est dite convergente si, pour toute suite de pavés $[x](k)$,

$$\lim_{k \rightarrow \infty} w([x](k)) = 0 \Rightarrow \lim_{k \rightarrow \infty} w([f]([x](k))) = 0.$$

Nous présentons succinctement dans le paragraphe suivant les fonctions d'inclusion classiques que nous utiliserons dans ce document, pour une fonction quelconque f définie comme suit

$$\begin{aligned} f : \mathbb{R}^m &\rightarrow \mathbb{R} \\ (x_1, \dots, x_m) &\mapsto f(x_1, \dots, x_m) \end{aligned}$$

Remarque 2.2.1 Un moyen simple de construire une fonction d'inclusion pour f à valeurs dans \mathbb{R}^n est d'élaborer une fonction d'inclusion pour chacune de ses composantes $f_i, i = 1, \dots, n$. On choisit alors

$$[f] = [f_1] \times \dots \times [f_n].$$

Pour tenter de caractériser les performances relatives de ces différentes fonctions d'inclusion, R. Moore a établi un critère de convergence [Moo79]. Ainsi, l'ordre de convergence d'une fonction d'inclusion est défini comme l'entier α le plus grand tel que

$$\exists \beta \mid w([f]([x])) - w(f([x])) \leq \beta w([x])^\alpha,$$

lorsque $w([x])$ tend vers zéro. Ce critère est partiel au sens où il ne caractérise les performances d'une fonction que sur des pavés de taille infinitésimale.

2.2.3.3 Fonctions d'inclusion de fonctions monotones

De nombreuses fonctions d'inclusion de fonctions f sont obtenues à partir des fonctions d'inclusion pour les fonctions mathématiques de base telles que les fonctions logarithme (\ln), exponentielle (\exp) et puissance, ou les fonctions trigonométriques (\cos , \sin , \tan) ou hyperboliques

(cosh, sinh, tanh). Il est donc essentiel de disposer de fonctions d'inclusion peu pessimistes pour ces fonctions. Dans le cas de fonctions monotones ou monotones par morceaux, la mise en œuvre de ces fonctions d'inclusion s'appuie sur le théorème suivant.

Théorème 2.2.1 *Soit $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction continue croissante (respectivement décroissante) sur un intervalle $[x] \subset \mathbb{R}$. La fonction ensembliste $[f]$ qui à tout sous-intervalle $[x']$ de $[x]$ associe $[f(\underline{x}'), f(\overline{x}')] (respectivement $[f(\overline{x}'), f(\underline{x}'))$ est une fonction d'inclusion optimale de f sur $[x]$.$*

La mise en œuvre des fonctions mathématiques de base monotones telles que les fonctions \ln , \log et \exp découle directement de ce théorème, ce qui explique que le nombre d'opérations effectuées pour calculer l'image d'un intervalle n'est que le double du même calcul effectué sur les réels. En décomposant $[x]$ en une union d'intervalles $[x_i]_{i=1, \dots, n}$ sur lesquels f est monotone, nous obtenons des fonctions d'inclusion pour les fonctions mathématiques de base (puissance, fonctions trigonométriques et trigonométriques hyperboliques).

Remarque 2.2.2 *L'élaboration des fonctions d'inclusion des fonctions transcendantes requiert en pratique, des traitements supplémentaires. En effet, l'erreur numérique de l'évaluation de ces fonctions en un point est mal contrôlé. Sans un traitement approprié, même la garantie peut être perdue.*

Les études de fonction requises par cette approche sont relativement fastidieuses et malaisées pour des utilisateurs éventuels. D'autres types de fonctions d'inclusion ont alors été proposés. Nous les étudions dans le paragraphe suivant.

2.2.3.4 Fonction d'inclusion naturelle

Elle s'appuie sur le théorème suivant qui est une simple extension au cas des intervalles du théorème 2.1.1.

Théorème 2.2.2 *La fonction ensembliste \mathbb{F} obtenue en remplaçant chaque occurrence des variables $x_i, i = 1, \dots, m$ dans l'expression formelle de f par l'intervalle correspondant est une fonction d'inclusion monotone au sens de l'inclusion, appelée fonction d'inclusion naturelle. Si de plus, l'ensemble des opérateurs et fonctions élémentaires qui composent cette fonction sont continus, alors \mathbb{F} est convergente. Si chaque x_i n'apparaît qu'une seule fois dans l'expression de \mathbb{F} , alors \mathbb{F} est optimale.*

Preuve.

Découle directement du théorème 2.1.1. ■

En raison des phénomènes d'enveloppement et de dépendance, la fonction d'inclusion naturelle \mathbb{F} est très rarement optimale. L'importance d'un pré-traitement symbolique, visant à diminuer le nombre d'occurrences de chaque variable est donc essentielle, comme le montre l'exemple suivant.

Exemple 2.2.2 *Considérons les trois expressions équivalentes sur \mathbb{R}*

$$\begin{aligned} f_1(x) &= (x-1)^2, \\ f_2(x) &= (x-1)(x-1), \\ f_3(x) &= x^2 - 2x + 1, \end{aligned}$$

et déterminons l'image de l'intervalle $[x] = [-2, 2]$ par $\mathbb{F}_1, \mathbb{F}_2$ et \mathbb{F}_3 fonctions d'inclusion naturelle de f_1, f_2 et f_3 :

$$\begin{aligned}\mathbb{F}_1([-2, 2]) &= ([-2, 2] - 1)^2 = [-3, 1]^2 = [0, 9], \\ \mathbb{F}_2([-2, 2]) &= ([-2, 2] - 1)([-2, 2] - 1) = [-3, 1] [-3, 1] = [-3, 9], \\ \mathbb{F}_3([-2, 2]) &= [-2, 2]^2 - 2[-2, 2] + 1 = [0, 4] - [-4, 4] + 1 = [-3, 9].\end{aligned}$$

La fonction d'inclusion \mathbb{F}_1 est optimale, alors que \mathbb{F}_2 et \mathbb{F}_3 sont pessimistes.

La fonction d'inclusion naturelle \mathbb{F} peut également être très pessimiste pour des fonctions f prolongeables par continuité.

Exemple 2.2.3 Soit la fonction

$$f : [-0.5, 0.5] \rightarrow \mathbb{R} \\ x \mapsto \frac{\ln(1+x)}{x},$$

qui est continue par prolongement par continuité puisque $\lim_{x \rightarrow 0^-} f(x) = \lim_{x \rightarrow 0^+} f(x) = 1$. L'image par cette fonction f de tout compact inclus dans $[-0.5, 0.5]$ est donc un intervalle borné. En raison de la division par un intervalle contenant 0, $\mathbb{F}([-0.5, 0.5]) = \frac{\ln(1+[-0.5, 0.5])}{[-0.5, 0.5]} = [-\infty, \infty]$. L'algorithme LNC présenté au tableau 2.1 fournit une évaluation optimale de f ($[-1, 1]$) en utilisant les propriétés du paragraphe 2.2.3.3.

Algorithme LNC($e : [x], s : [y]$)	
1	si $(0 \notin [x])$ $[y] := \mathbb{F}([x])$;
2	si $(\underline{x} = 0)$ $\bar{y} := 1$;
3	sinon $\bar{y} := \frac{\ln(1+\underline{x})}{\underline{x}}$;
4	si $(\bar{x} = 0)$ $\underline{y} := 1$;
5	sinon $\underline{y} := \frac{\ln(1+\bar{x})}{\bar{x}}$;

Tableau 2.1: Evaluation optimale de la fonction de l'exemple 2.2.3

Il sera donc essentiel d'évaluer, chaque fois que nous le pourrons, l'image d'une fonction prolongeable par continuité à l'aide d'un algorithme similaire à celui du tableau 2.1.

La convergence des fonctions d'inclusion naturelle est au moins linéaire ($\alpha \geq 1$). La fonction d'inclusion suivante, plus complexe, permet d'atteindre un ordre de convergence plus élevé.

2.2.3.5 Fonction d'inclusion centrée et centrée mixte

La fonction d'inclusion centrée s'appuie sur le théorème suivant, obtenu par extension aux intervalles du théorème de la valeur moyenne.

Théorème 2.2.3 (valeur moyenne) Soit f une fonction de \mathbb{R}^m dans \mathbb{R} différentiable sur le pavé $[x]$. Alors

$$\forall x \in [x], \exists \xi \in [x] \text{ tel que } f(x) = f(x_0) + \mathbf{g}^T(\xi)(x - x_0),$$

où \mathbf{g} est le vecteur gradient de f , et x_0 le milieu de $[x]$.

Sous l'hypothèse que f est différentiable, nous pouvons donc écrire

$$\forall \mathbf{x} \in [\mathbf{x}], f(\mathbf{x}) \in f(\mathbf{x}_0) + [\mathbf{g}^T]([\mathbf{x}]) (\mathbf{x} - \mathbf{x}_0).$$

La fonction d'inclusion centrée $[f_c]$ pour f est alors définie comme

$$f_c([\mathbf{x}]) = f(\mathbf{x}_0) + [\mathbf{g}^T]([\mathbf{x}]) ([\mathbf{x}] - \mathbf{x}_0),$$

où $[\mathbf{g}^T]$ est une fonction d'inclusion de \mathbf{g}^T (par exemple, la fonction d'inclusion naturelle).

Le principal avantage de la fonction d'inclusion centrée est sa convergence, car

$$\frac{w([f_c]([\mathbf{x}]))}{w(f([\mathbf{x}]))} \rightarrow 1 \text{ lorsque } w([\mathbf{x}]) \rightarrow 0.$$

L'ordre de convergence de la fonction d'inclusion centrée est au moins quadratique ($\alpha \geq 2$).

La fonction d'inclusion centrée mixte est une amélioration de la fonction centrée dans le cas multi-dimensionnel [Han68]. Pour plus de simplicité, nous présentons son élaboration dans le cas où $m = 3$.

Appliquons le théorème de la valeur moyenne pour chaque variable x_i successivement, en commençant par exemple, par x_3 . Il vient :

$$\forall \mathbf{x} \in [\mathbf{x}], f(x_1, x_2, x_3) \in f(x_1, x_2, x_{03}) + g_3(x_1, x_2, [x_3]) ([x_3] - x_{03})$$

puis

$$\forall \mathbf{x} \in [\mathbf{x}], f(x_1, x_2, x_{03}) \in f(x_1, x_{02}, x_{03}) + g_2(x_1, [x_2], x_{03}) ([x_2] - x_{02})$$

et

$$\forall \mathbf{x} \in [\mathbf{x}], f(x_1, x_{02}, x_{03}) \in f(x_{01}, x_{02}, x_{03}) + g_1([x_1], x_{02}, x_{03}) ([x_1] - x_{01}),$$

d'où en regroupant,

$$\begin{aligned} \forall \mathbf{x} \in [\mathbf{x}], f(x_1, x_2, x_3) \subset & f(x_{01}, x_{02}, x_{03}) \\ & + g_1([x_1], x_{02}, x_{03}) ([x_1] - x_{01}) \\ & + g_2([x_1], [x_2], x_{03}) ([x_2] - x_{02}) \\ & + g_3([x_1], [x_2], [x_3]) ([x_3] - x_{03}). \end{aligned}$$

Sur ce principe, de nombreuses autres fonctions d'inclusions peuvent être élaborées. En particulier, les fonctions d'inclusion de Taylor (voir en particulier [Neu]) utilisent un ordre de dérivation plus élevé. Cependant, il est indispensable de garder à l'esprit qu'il existe un compromis entre la qualité de la fonction d'inclusion utilisée et la complexité algorithmique qui en découle. Aucune fonction d'inclusion ne se révèle universellement la meilleure. De plus, le critère que nous avons utilisé pour caractériser les fonctions d'inclusion s'appuie sur la convergence de la fonction d'inclusion lorsque le pavé $[\mathbf{x}]$ est de taille infinitésimale. Lorsque cette hypothèse n'est pas vérifiée, il est néanmoins possible de contrôler le pessimisme engendré à l'aide de l'outil de bisection, présenté dans le paragraphe suivant.

2.2.3.6 Bissection

Soit un pavé $[\mathbf{x}] \subset \mathbb{R}^m$:

$$[\mathbf{x}] = [x_1] \times \dots \times [x_m]$$

et soit i l'indice correspondant à la composante de $[\mathbf{x}]$ de taille la plus grande (éventuellement, si $w([x_j]) = w([x_k]) = w([\mathbf{x}])$, alors $i = \min(j, k)$).

Soient les pavés

$$L[\mathbf{x}] = [x_1] \times \dots \times [\underline{x}_i, x_{0i}] \times \dots \times [x_m],$$

$$R[\mathbf{x}] = [x_1] \times \dots \times [x_{0i}, \bar{x}_i] \times \dots \times [x_m].$$

L'opération qui consiste à obtenir les pavés $L[\mathbf{x}]$ et $R[\mathbf{x}]$ à partir de $[\mathbf{x}]$ est appelée *bissection*. Les pavés $L[\mathbf{x}]$ et $R[\mathbf{x}]$ sont frères, et fils du même pavé père $[\mathbf{x}]$. L'opération inverse de la bissection, qui consiste à former le pavé $[\mathbf{x}]$ à partir des pavés frères $L[\mathbf{x}]$ et $R[\mathbf{x}]$ est appelée *réunification*. Un exemple de bissection est présenté à la figure 2-4.

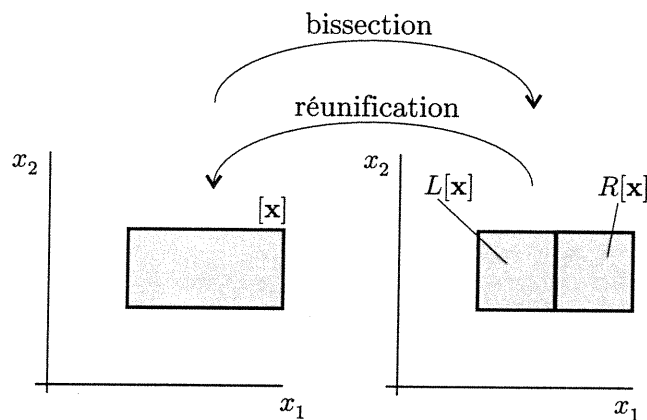


Figure 2-4: Bissection du pavé $[\mathbf{x}]$ et réunification

Puisque les fonctions d'inclusion sont particulièrement efficaces sur des pavés de faible taille, il est possible d'évaluer plus efficacement l'image du pavé $[\mathbf{x}]$ par une fonction \mathbf{f} en évaluant l'image des pavés fils de $[\mathbf{x}]$ par \mathbf{f} , puisque, lorsque $[\mathbf{f}]$ est monotone au sens de l'inclusion,

$$\mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}](L[\mathbf{x}]) \cup [\mathbf{f}](R[\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}]).$$

Preuve.

Comme $L[\mathbf{x}] \subset [\mathbf{x}]$ et $R[\mathbf{x}] \subset [\mathbf{x}]$, si $[\mathbf{f}]$ est monotone au sens de l'inclusion, alors $[\mathbf{f}](L[\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}])$ et $[\mathbf{f}](R[\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}])$, donc

$$[\mathbf{f}](L[\mathbf{x}]) \cup [\mathbf{f}](R[\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}]).$$

Soit $\mathbf{x} \in [\mathbf{x}]$. Alors $\mathbf{x} \in L[\mathbf{x}]$ ou $\mathbf{x} \in R[\mathbf{x}]$. Par définition de la fonction d'inclusion, on a donc $\mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}](L[\mathbf{x}])$ ou $\mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}](R[\mathbf{x}])$, donc

$$\forall \mathbf{x} \in [\mathbf{x}], \mathbf{f}(\mathbf{x}) \subset [\mathbf{f}](L[\mathbf{x}]) \cup [\mathbf{f}](R[\mathbf{x}]), \quad (2.10)$$

et $f([x]) \subset [f](L[x]) \cup [f](R[x])$.

La figure 2-5 présente un schéma générique de cette méthode.

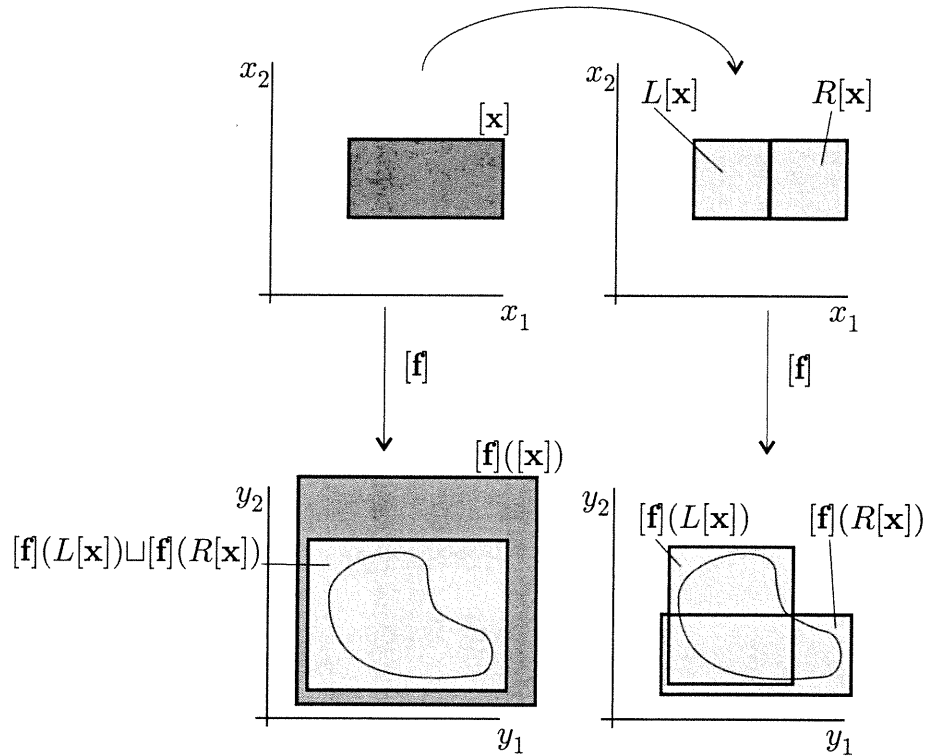


Figure 2-5: Principe de l'évaluation de l'image d'une fonction par bisection

L'amélioration fournie par cette approche peut être conséquente, comme le montre l'exemple suivant.

Exemple 2.2.4 Reprenons le problème de l'exemple 2.2.2 et bissectons l'intervalle $[x]$. Il vient

$$L[x] = [-2, 0], R[x] = [0, 2].$$

Calculons l'image de ces deux intervalles par la fonction f_2 . Il vient

$$\begin{aligned} f_2(L[x]) &= ([-2, 0] - 1) ([-2, 0] - 1) = [-3, -1] [-3, -1] = [1, 9], \\ f_2(R[x]) &= ([0, 2] - 1) ([0, 2] - 1) = [-1, 1] [-1, 1] = [-1, 1], \end{aligned}$$

d'où

$$f([x]) = f_2(L[x]) \sqcup f_2(R[x]) = [-1, 9],$$

qui est moins pessimiste que $\mathbb{F}_2([x]) = [-3, 9]$: la bisection permet de réduire le pessimisme de l'évaluation.

Remarquons enfin sur la figure 2-5 que se limiter à un pavé solution introduit un pessimisme important : ainsi, la description de $f([x])$ par $[f](L[x]) \cup [f](R[x])$ est bien moins pessimiste que $[f](L[x]) \sqcup [f](R[x])$.

Ces remarques vont nous amener tout naturellement à décrire un ensemble non plus par un seul récipient mais par une union de pavés obtenus par bisections successives.

2.3 Implémentation

Les pavés sont des récipients très utiles à la description d'ensembles. Mais en raison du phénomène d'enveloppement, la description d'un ensemble \mathbb{X} par un pavé extérieur peut se révéler trop pessimiste. Pour remédier à ce problème, il a été proposé de caractériser tout ensemble \mathbb{X} à l'aide d'une représentation plus adéquate, qui conserve certains attributs de \mathbb{X} dont sa forme, au moins approximative. Cette représentation consiste à décrire \mathbb{X} par recouvrement de pavés.

L'approche par recouvrement est classique dans la représentation des ensembles. Les techniques varient selon le type de récipient utilisé. A nouveau, nous nous contenterons de manipuler des pavés. De plus, les unions de pavés que nous manipulerons seront telles que les pavés ne se chevauchent pas, nous les appellerons des *sous-pavages* [Kie99].

Il existe différentes techniques pour structurer ces unions de pavés. Dans tout ce mémoire, nous considérons uniquement deux structures :

- la liste –éventuellement le stockage temporaire dans la pile–, qui correspond à une structure très sommaire; cette structure a pour principal avantage de permettre une lecture aisée des algorithmes présentés dans les chapitres suivants;
- la structure en arbre binaire, dont le développement et l'implémentation sont au cœur de la thèse de M. Kieffer [Kie99] [KJBW01].

2.3.1 Sous-pavages

Un sous-pavage $\mathbb{X} \subset \mathbb{R}^m$ est une union de pavés $([\mathbf{x}])_{i=1,\dots,n}$ dont les intérieurs sont disjoints. Pour calculer les éléments d'un sous-pavage, nous allons les soumettre à un test.

2.3.1.1 Test d'inclusion

L'ensemble des intervalles booléens \mathbb{IB} est un sous-ensemble de \mathbb{IR} défini par extension de l'ensemble des booléens $\mathbb{B} = \{0, 1\}$:

$$\mathbb{IB} = \{\emptyset, [0], [1], [0, 1]\},$$

c'est-à-dire que \mathbb{IB} est l'ensemble des extensions intervalles des éléments de \mathbb{B} . De même que 0 représente *faux* et 1 représente *vrai*, \emptyset représente *impossible*, $[0]$ représente *faux*, $[1]$ représente *vrai* et $[0, 1]$ représente *indéterminé*. Puisque $\mathbb{IB} \subset \mathbb{IR}$, les opérations sur les intervalles booléens sont obtenues par fermeture sur \mathbb{IB} des opérations définies sur les intervalles réels. Il est également ajouté l'opérateur de complémentation logique \neg :

$$\neg[x] = \{\neg x \mid x \in [x]\}.$$

La notion de fonction d'inclusion est étendue aux booléens comme pour les réels. Ainsi $[\beta] : \mathbb{IB}^m \rightarrow \mathbb{IB}$ est une fonction d'inclusion pour $\beta : \mathbb{B}^m \rightarrow \mathbb{B}$ si

$$\forall [\mathbf{x}] \in \mathbb{IB}^m, \beta([\mathbf{x}]) \subset [\beta]([\mathbf{x}]).$$

Un *test* est une fonction $t : \mathbb{R}^m \rightarrow \mathbb{B}$. Par exemple, la fonction caractéristique $\chi_{\mathbb{X}}(\cdot)$ d'un ensemble $\mathbb{X} \subset \mathbb{R}^m$ est un test, puisqu'elle vérifie pour tout élément \mathbf{x} de \mathbb{R}^m : $\chi_{\mathbb{X}}(\mathbf{x}) = 1$ si

$\mathbf{x} \in \mathbb{X}$, sinon $\chi_{\mathbb{X}}(\mathbf{x}) = 0$. Nous appellerons *test associé* à l'ensemble \mathbb{X} le test $t : \mathbb{R}^m \rightarrow \mathbb{B}$ tel que

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^m, t(\mathbf{x}) = 1\}. \quad (2.11)$$

La fonction $[t]$ est un *test d'inclusion* pour t si $\forall [\mathbf{x}] \in \mathbb{IR}^m$

$$[t]([\mathbf{x}]) = 1 \implies \forall \mathbf{x} \in [\mathbf{x}], t(\mathbf{x}) = 1, \quad (2.12)$$

$$[t]([\mathbf{x}]) = 0 \implies \forall \mathbf{x} \in [\mathbf{x}], t(\mathbf{x}) = 0. \quad (2.13)$$

Nous dirons qu'un test d'inclusion est *optimal* si

$$\forall [\mathbf{x}] \in \mathbb{IR}^m, [t]([\mathbf{x}]) = \{t(\mathbf{x}), \mathbf{x} \in [\mathbf{x}]\}.$$

De plus, nous considérerons dans toute la suite des tests d'inclusion convergents. Pour obtenir un sous-pavage $\widehat{\mathbb{X}}$ recouvrant l'ensemble \mathbb{X} , nous utiliserons un test d'inclusion pour le test associé à \mathbb{X} .

2.3.1.2 Accumulation

Soit $\mathbb{X} \subset \mathbb{R}^m$ défini par $\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^m \mid t(\mathbf{x}) = 1\}$. La fonction $t(\cdot)$ constitue une représentation de \mathbb{X} . Pour obtenir un sous-pavage $\widehat{\mathbb{X}}$ recouvrant \mathbb{X} , nous utiliserons le test d'inclusion $[t]$. Soit $[\mathbf{x}] \in \mathbb{IR}^m$. L'algorithme de caractérisation garantie de \mathbb{X} s'appuie sur les propriétés suivantes :

- si $[t]([\mathbf{x}]) = 1$, d'après (2.11) et (2.12), $[\mathbf{x}] \subset \mathbb{X}$.
- si $[t]([\mathbf{x}]) = 0$, d'après (2.11) et (2.13), $[\mathbf{x}] \cap \mathbb{X} = \emptyset$.
- si $[t]([\mathbf{x}]) = [0, 1]$, il n'est pas possible de conclure, nous appliquons alors (2.10), et bissectons $[\mathbf{x}]$ pour une analyse plus fine.

L'algorithme SIVIA_{∞} (*Set Inverter via Interval Analysis*), présenté au tableau 2.2, permet d'obtenir un sous-pavage $\widehat{\mathbb{X}}$ recouvrant l'ensemble \mathbb{X} défini par (2.11).

Algorithme SIVIA_{∞} (e: $[t], [\mathbf{x}]$, s: $\widehat{\mathbb{X}}$)	
1	$test := [t]([\mathbf{x}]);$
2	si ($test = 1$) renvoyer $[\mathbf{x}]$;
3	sinon si ($test = 0$) renvoyer \emptyset ;
4	$\widehat{\mathbb{X}} := \text{SIVIA}_{\infty}([t], L[\mathbf{x}]) \cup \text{SIVIA}_{\infty}([t], R[\mathbf{x}]);$

Tableau 2.2: Algorithme SIVIA_{∞}

Cet algorithme récursif partitionne un sous-espace de \mathbb{R}^m , en effectuant deux opérations : un test d'appartenance, et une bisection éventuelle. Pour cette raison, la terminologie anglaise le qualifie de *branch-and-bound algorithm*. Le principe de cet algorithme est de bissecter pour conclure. Ceci constitue aussi son défaut majeur, puisque dans de nombreux cas, il sera nécessaire de bissecter jusqu'à obtenir des pavés $[\mathbf{x}]$ infinitésimaux. Ainsi, en raison du phénomène d'enveloppement, tout pavé $[\mathbf{x}]$ intersectant la frontière de \mathbb{X} sera soumis à un nombre de bisections important, éventuellement infini (nous l'avons dénommé SIVIA_{∞} pour cette raison). Nous dirons que l'algorithme s'accumule sur la frontière de \mathbb{X} , qui sera appelée *zone d'accumulation*. Il n'est pas possible de recouvrir exactement un ensemble de forme quelconque par une union de récipients exactement représentables en machine. SIVIA_{∞} n'est donc pas utilisable.

Une idée naturelle est alors d'imposer un critère d'arrêt, en introduisant le paramètre ε , qui quantifie la taille maximale des pavés à bissecter. On obtient alors l'algorithme SIVIA, développé par L. Jaulin [JW93b] dans le cadre de sa thèse, et similaire à [Moo92], développé indépendamment. Cet algorithme fournit alors deux sous-pavages $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$ qui encadrent l'ensemble \mathbb{X} au sens de l'inclusion.

2.3.2 Approximations intérieure et extérieure

La description par recouvrement exact est particulièrement coûteuse en temps de calcul et en mémoire. Le sous-pavage $\widehat{\mathbb{X}}$ obtenu en introduisant le critère d'arrêt, sera donc seulement une approximation de l'ensemble \mathbb{X} . L'ensemble des pavés $[\mathbf{x}]$ de taille inférieure à ε constitue un ensemble $\Delta\mathbb{X}$ que nous appellerons *couche d'incertitude*, dont le volume dépend directement de ε . Le sous-pavage $\underline{\mathbb{X}}$ constitué des pavés satisfaisant le test d'inclusion

$$\underline{\mathbb{X}} = \{[\mathbf{x}] \in \mathbb{R}^m \mid ([t]([\mathbf{x}]) = 1)\} \subset \mathbb{X},$$

est une *approximation intérieure* de \mathbb{X} . Le sous-pavage $\overline{\mathbb{X}} = \underline{\mathbb{X}} \cup \Delta\mathbb{X}$, vérifie $\mathbb{X} \subset \overline{\mathbb{X}}$. Il s'agit donc d'une *approximation extérieure* de \mathbb{X} . La caractérisation de ces deux approximations est proposée au tableau 2.3 où la sortie est le couple de sous-pavages $(\underline{\mathbb{X}}, \overline{\mathbb{X}})$. Au pas 4, nous définissons l'union de deux couples de sous-pavages $(\underline{\mathbb{Z}}, \overline{\mathbb{Z}}) = (\underline{\mathbb{X}}, \overline{\mathbb{X}}) \cup (\underline{\mathbb{Y}}, \overline{\mathbb{Y}})$ par

$$\begin{aligned} \underline{\mathbb{Z}} &= \underline{\mathbb{X}} \cup \underline{\mathbb{Y}}, \\ \overline{\mathbb{Z}} &= \overline{\mathbb{X}} \cup \overline{\mathbb{Y}}. \end{aligned}$$

Algorithme SIVIA (e: $[t], [\mathbf{x}], \varepsilon$, s: $(\underline{\mathbb{X}}, \overline{\mathbb{X}})$)	
1	$test := [t]([\mathbf{x}]);$
2	si ($test = 1$) renvoyer $([\mathbf{x}], [\mathbf{x}]);$
3	sinon si ($test = 0$) renvoyer $(\emptyset, \emptyset);$
4	sinon si ($w([\mathbf{x}]) < \varepsilon$) renvoyer $(\emptyset, [\mathbf{x}]);$
5	$(\underline{\mathbb{X}}, \overline{\mathbb{X}}) := \text{SIVIA}([t], L[\mathbf{x}], \varepsilon) \cup \text{SIVIA}([t], R[\mathbf{x}], \varepsilon).$

Tableau 2.3: Algorithme SIVIA

En conclusion, l'algorithme SIVIA encadre l'ensemble \mathbb{X} par deux sous-pavages $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$ tels que

$$\underline{\mathbb{X}} \subset \mathbb{X} \subset \overline{\mathbb{X}}, \quad (2.14)$$

où

$$\overline{\mathbb{X}} = \underline{\mathbb{X}} \cup \Delta\mathbb{X}.$$

L'introduction du paramètre ε transforme donc SIVIA $_{\infty}$ en un algorithme fini qui termine en moins de $\left(\frac{w(\mathbb{X})}{\varepsilon} + 1\right)^m$ itérations (voir [JW93a] pour une étude détaillée des propriétés de SIVIA). Les grandeurs m et ε sont deux facteurs essentiels qui influent sur les performances de SIVIA :

- le temps de calcul t_c croît exponentiellement avec m : SIVIA est performant pour des problèmes à faible nombre de paramètres,

- comme le paramètre ε , à choisir par l'utilisateur, influe sur le volume de la couche d'incertitude $\Delta\mathbb{X}$, il est nécessaire de réaliser un compromis entre la précision de l'encadrement souhaité et le temps de calcul (ainsi que la charge mémoire). Dans la suite, nous serons amenés à quantifier la précision de cet encadrement par le rapport

$$\delta = \frac{1}{\text{Vol}(\Delta\mathbb{X})}, \quad (2.15)$$

qui dépend de ε .

L'avantage de l'encadrement (2.14) est qu'il permet de conclure, pour un ensemble \mathbb{X} défini par une propriété \mathcal{P}

$$\mathbb{X} = \{\mathbf{x} \mid \mathcal{P} \text{ est vraie}\},$$

que

Propriété 2.3.1 *Si $\overline{\mathbb{X}}$ est vide, alors \mathbb{X} est vide; aucun point ne satisfait la propriété \mathcal{P} testée.*

Propriété 2.3.2 *$\text{Vol}(\underline{\mathbb{X}}) \leq \text{Vol}(\mathbb{X}) \leq \text{Vol}(\overline{\mathbb{X}})$.*

Propriété 2.3.3 *Si $\underline{\mathbb{X}}$ n'est pas vide, alors \mathbb{X} n'est pas vide; il existe au moins un point satisfaisant la propriété \mathcal{P} testée.*

La majorité des algorithmes que nous présenterons seront des algorithmes *branch and bound* qui s'appuient sur la structure de SIVIA. Il est donc crucial de souligner dès à présent que ces algorithmes sont de complexité exponentielle, ce qui limite leur application à la description de sous-ensembles de \mathbb{R}^m , où m reste faible.

Dans le paragraphe suivant, nous présentons très brièvement le calcul des images directe et inverse d'un sous-pavage par une fonction ensembliste.

2.3.3 Images d'un ensemble par une fonction

Comme nous utilisons les sous-pavages pour représenter des ensembles, il nous faut définir des algorithmes de mise en œuvre des fonctions ensemblistes. Nous avons vu comment la structure de SIVIA permet de calculer un encadrement de \mathbb{X} . Nous allons nous inspirer de cette structure pour déterminer les images directe et inverse d'un sous-pavage par une fonction ensembliste [Kie99] [KJBW01]. Il sera en particulier nécessaire d'établir un test d'inclusion adapté à chacun de ces deux problèmes.

2.3.3.1 Image inverse

Soit une fonction \mathbf{f} de \mathbb{R}^m dans \mathbb{R}^n pour laquelle nous disposons d'une fonction d'inclusion $[\mathbf{f}]$ convergente. Soit \mathbb{Y} un sous-pavage de \mathbb{R}^n et \mathbb{X} l'ensemble défini par

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}). \quad (2.16)$$

Soit le test t défini par

$$\begin{aligned} t & : \mathbb{R}^m \rightarrow \mathbb{B} \\ \mathbf{x} & \longmapsto \begin{cases} 1 & \text{si } \mathbf{f}(\mathbf{x}) \in \mathbb{Y}, \\ 0 & \text{sinon.} \end{cases} \end{aligned}$$

Supposons qu'on dispose d'une fonction d'inclusion $[f]$ pour f . Alors le test

$$[t] \quad : \quad \mathbb{R}^m \rightarrow \mathbb{B}$$

$$\mathbf{x} \quad \mapsto \quad \begin{cases} [1] & \text{si } [f]([\mathbf{x}]) \in \mathbb{Y}, \\ [0] & \text{si } [f]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset, \\ [0, 1] & \text{sinon.} \end{cases}$$

est un test d'inclusion pour t . Il suffit donc d'implanter ce test d'inclusion $[t]$ dans l'algorithme SIVIA vu au paragraphe précédent pour obtenir des approximations intérieure et extérieure de \mathbb{X} défini par (2.16).

Exemple 2.3.1 Soit

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$\mathbf{x} \quad \mapsto \quad \begin{pmatrix} (x_1 - 1)^2 + x_2 \\ x_1^2 + x_2^2 \end{pmatrix},$$

pour laquelle la fonction d'inclusion naturelle \mathbb{F} est optimale, et \mathbb{S} l'ensemble défini par

$$\mathbb{S} = \{\mathbf{x} \in \mathbb{X} \mid f(\mathbf{x}) \in [\mathbf{y}]\} = f^{-1}([\mathbf{y}]) \cap \mathbb{X},$$

avec $\mathbb{X} = [-5, 5]^{\times 2}$ et $[\mathbf{y}] = [0, 2] \times [1, +\infty]$. Les trois approximations intérieure et extérieure obtenues par SIVIA pour trois valeurs différentes de ε sont représentées sur la figure 2-6. On observe bien l'amélioration de la qualité de la représentation (la couche d'incertitude $\Delta\mathbb{S}$ représentée en blanc diminue) lorsque ε diminue. En contrepartie, le nombre de bisections n_b et le temps de calcul t_c , représentés en fonction de ε sur la figure 2-7, varient suivant une loi en ε^{-1} .

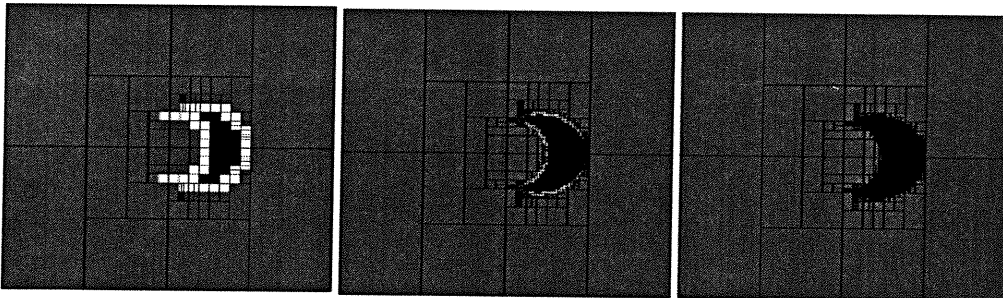


Figure 2-6: Approximation intérieure (en noir) et couche d'incertitude (en blanc) fournies par SIVIA pour l'ensemble \mathbb{S} de l'exemple 2.3.1; de gauche à droite : $\varepsilon = 0.5$, $\varepsilon = 0.1$, $\varepsilon = 0.01$

2.3.3.2 Image directe

Le calcul de l'image directe d'un sous-pavage $\mathbb{Y} = f(\mathbb{X})$ est plus ardu, puisqu'aucun test d'inclusion $t(\mathbf{y}) = (\mathbf{y} \in f(\mathbb{X}))$ n'est disponible sous forme intervalles. De plus, l'évaluation directe de l'image de chaque pavé $[\mathbf{x}]$ de \mathbb{X} est très souvent pessimiste, en raison des phénomènes d'enveloppement et de dépendance. Pour les contrer, l'algorithme IMAGESP, développé par M. Kieffer [Kie99], s'appuie sur une structure particulière qui fait apparaître trois étapes (voir sa mise en œuvre au tableau 2.4) :

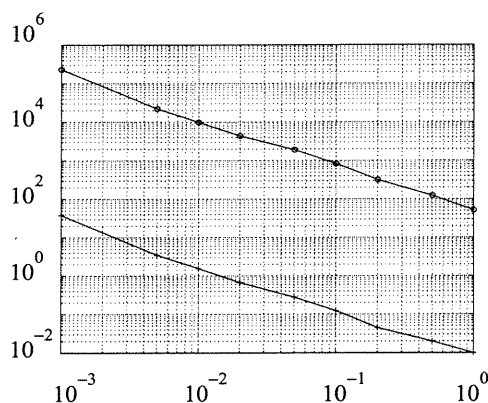


Figure 2-7: Temps de calcul t_c (-) et nombre de bisections n_b (+) effectuées par SIVIA en fonction de ε pour l'ensemble \mathbb{S} de l'exemple 2.3.1

1. la *hachage* (pas 2) permet de construire un sous-pavage régulier contenant \mathbb{X} noté \mathbb{X}_ε , dont tous les pavés sont de taille inférieure à ε ,
2. l'*évaluation* (pas 3) calcule l'image de chaque pavé $[\mathbf{x}]$ de \mathbb{X}_ε par $[\mathbf{f}]$ et les range dans une liste \mathcal{L} ,
3. la *régularisation* (pas 4) consiste à construire un sous-pavage régulier \mathbb{Y} à partir des pavés de la liste \mathcal{L} ; il est obtenu comme l'image inverse de l'union des pavés $[\mathbf{y}]$ de \mathcal{L} à l'aide de SIVIA.

Algorithme IMAGESP(e: $[\mathbf{f}]$, \mathbb{X} , ε ; s: \mathbb{Y})	
1	$\mathcal{L} := \emptyset$;
2	$\mathbb{X}_\varepsilon := \text{Hachage}(\mathbb{X}, \varepsilon)$;
3	Pour chaque $[\mathbf{x}]$ de \mathbb{X}_ε , $\mathcal{L} := \mathcal{L} \cup \{[\mathbf{f}]([\mathbf{x}])\}$;
4	$\mathbb{Y} := \text{SIVIA}([\text{Id}], [\mathbf{y}] \in \mathcal{L}, \varepsilon)$.

Tableau 2.4: Algorithme IMAGESP

Les propriétés d'IMAGESP ont été étudiées dans [KJW02]. Une illustration est proposée à l'exemple 2.3.2.

Exemple 2.3.2 On considère la transformation du pavé $[\mathbf{x}] = [-1, 1]^{\times 2}$ par la rotation ρ d'angle $\frac{\pi}{4}$. En raison du phénomène d'enveloppement, l'évaluation directe de $\rho([\mathbf{x}])$ est très pessimiste. Elle est représentée sur la figure 2-8a). Grâce au hachage, il est possible de diminuer notablement le pessimisme, comme le montrent les figures 2-8b), 2-8c) et 2-8d), obtenues pour des valeurs de ε décroissantes.

2.4 Conclusions

Deux phénomènes dégradent la mise en œuvre des outils ensemblistes.

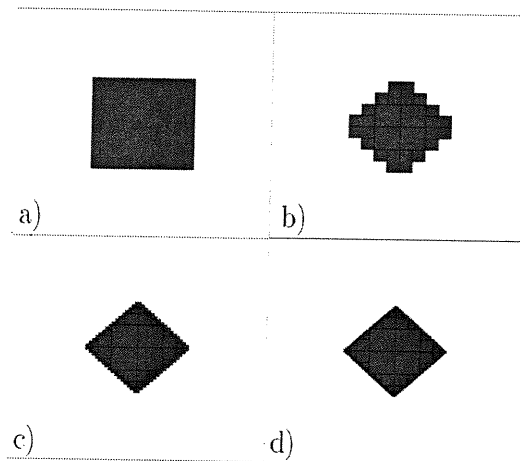


Figure 2-8: Sous-pavages obtenus par IMAGESP dans le cadre de l'exemple 2.3.2 pour des ha-chages de plus en plus fins; pour chaque cas, le cadre représente le pavé $[-5, 5]^2$

D'une part, le phénomène de dépendance rend pessimiste la simple extension \mathbb{F} sur $\mathcal{P}(\mathbb{R}^m)$ d'une fonction f multi-occurente. Lorsque l'ensemble à considérer est un intervalle ou un pavé, l'analyse par intervalles fournit des fonctions d'inclusion plus élaborées, capables de contrer ce phénomène. Cependant, les performances de ces fonctions ne sont généralement comparées que sur des intervalles ou des pavés infinitésimaux.

D'autre part, le phénomène d'enveloppement est lié au pessimisme introduit en représentant un ensemble de forme quelconque par un récipient de nature prédéfinie, – dans notre cas des intervalles et des pavés. Pour limiter ce pessimisme, il a alors été envisagé de représenter ces ensembles par des unions de récipients, ici des sous-pavages.

Les algorithmes de partitionnement SIVIA et IMAGESP, en utilisant de nombreuses bisections, permettent de traiter des pavés de taille faible, ce qui permet d'exploiter les caractéristiques de fonctions d'inclusion, et représentent les ensembles à l'aide de sous-pavages, afin de restreindre également le phénomène d'enveloppement. En contrepartie, la complexité de ces algorithmes est exponentielle selon la dimension m des pavés traités, ce qui restreint leur application à des problèmes de faible dimension.

Nous présentons au chapitre suivant des opérateurs, appelés contracteurs, qui, parce qu'ils n'utilisent aucune bisection, permettent d'alléger les temps de calcul requis, pour obtenir la même qualité de description que les algorithmes de partitionnement présentés jusqu'ici. Ces outils nous permettront également de proposer de nouveaux algorithmes de caractérisation ensembliste, et en particulier un algorithme rapide de projection d'ensembles.

Chapitre 3

Contracteurs

Caractériser l'ensemble solution d'un système d'inégalités par deux sous-pavages est un problème dit NP-difficile, pour lequel il n'existe pas d'algorithme de complexité polynômiale. Ainsi la complexité des algorithmes présentés au chapitre 2 qui s'appuient sur l'utilisation conjointe d'un test d'inclusion $t(\cdot)$ et d'une loi de partitionnement, est exponentielle. C'est le prix à payer pour ces bisections dans chaque direction de l'espace des paramètres, qui limite l'utilisation de cette approche lorsque la dimension m de cet espace est élevée.

Ceci donne tout son intérêt à un autre type d'outils, les *contracteurs*, capables de réduire la taille du pavé $[x]$ à traiter sans bisection. Les contracteurs les plus connus sont obtenus à partir d'une extension sur \mathbb{IR}^m de solveurs de systèmes définis sur \mathbb{R}^m . Ils ne sont efficaces que lorsque la taille de $[x]$ est déjà faible. Nous présentons un autre type de contracteur, qui permet de réduire les pavés $[x]$ de taille importante, en s'appuyant sur la notion de *propagation de contraintes*.

Nous abordons dans un premier temps le principe de la contraction d'un pavé (paragraphe 3.1). Pour cela, nous introduisons en particulier la notion de problème de satisfaction de contraintes. Nous décrivons brièvement quelques contracteurs issus des techniques de l'analyse par intervalles en soulignant leurs consignes d'utilisation. Au paragraphe 3.2, nous introduisons un contracteur efficace lorsque la taille du pavé à contracter est relativement grande. Ce contracteur repose sur le principe de la projection de contraintes. Il sera alors temps d'étudier la mise en œuvre de ces contracteurs pour la contraction d'ensembles (paragraphe 3.3). A cette occasion nous présentons au paragraphe 3.3.1 le solveur SIVIA_P, dont nous montrons qu'il permet de caractériser de façon plus rapide que SIVIA l'ensemble S recherché. L'ensemble des techniques exposées permet également de proposer d'autres solveurs : nous exposons le principe de calcul de la projection de S sur un hyperplan au paragraphe 3.3.2. Enfin, nous présentons un solveur dédié à l'estimation précise de la forme et du volume d'un ensemble au paragraphe 3.3.3.

Le contracteur de propagation-rétropropagation fait l'objet des publications [JKBW01], [JBKW01], tandis que [BJKW01b] et [BBJ⁺01a] illustrent le paragraphe 3.3.3.

3.1 Contractions de pavés

La contraction de domaines est une notion partagée par les communautés de l'analyse par intervalles, de l'intelligence artificielle, et de l'informatique. Aussi avons-nous choisi de replacer le problème de la description d'un ensemble défini par un système d'égalités ou d'inégalités dans le cadre plus général des problèmes de satisfaction de contraintes, en nous inspirant de l'approche suivie dans [JKDW01] et [Gra98].

Dans cette partie, nous considérons pour des raisons de simplicité que l'espace de recherche

initial est un pavé

$$\mathbb{X} = [\mathbf{x}_0] = [x_{01}] \times \dots \times [x_{0m}]. \quad (3.1)$$

Remarque 3.1.1 Dans cette thèse, le problème de la détermination de cet espace de recherche n'est pas évoqué, puisque nous supposerons toujours qu'il est obtenue à partir d'une connaissance a priori (fournie par l'expérimentateur). Dans le cas contraire, on pourra se reporter à [BL01] pour la détermination de cet espace lorsque \mathbf{f} est une fonction polynômiale.

3.1.1 Définitions

3.1.1.1 Problème de satisfaction de contraintes

Considérons le système d'équations à résoudre

$$\mathbf{f}(\mathbf{x}) = 0 \Leftrightarrow \begin{cases} f_1(\mathbf{x}) = 0 \\ \dots \\ f_j(\mathbf{x}) = 0 \\ \dots \\ f_n(\mathbf{x}) = 0 \end{cases}, \quad (3.2)$$

où $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^m$, et f_i est une fonction de \mathbf{x} . On appelle *problème de satisfaction de contraintes* (CSP) \mathcal{H} , la recherche de l'ensemble des solutions de (3.2) contenues dans \mathbb{X} , et on notera

$$\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in \mathbb{X}). \quad (3.3)$$

Remarque 3.1.2 Lorsque $n = 1$, nous parlerons plus simplement de *contrainte*. Dans le cas général, un CSP \mathcal{H} est donc composé d'un ensemble de contraintes \mathcal{H}_j

$$\mathcal{H} : \left(\begin{array}{l} \mathcal{H}_1 : f_1(\mathbf{x}) = 0, \mathbf{x} \in \mathbb{X} \\ \dots \\ \mathcal{H}_j : f_j(\mathbf{x}) = 0, \mathbf{x} \in \mathbb{X} \\ \dots \\ \mathcal{H}_n : f_n(\mathbf{x}) = 0, \mathbf{x} \in \mathbb{X} \end{array} \right).$$

L'ensemble solution \mathbb{S}_j d'une contrainte \mathcal{H}_j est défini comme

$$\mathbb{S}_j = \mathbb{S}_j(\mathcal{H}_j) = \{\mathbf{x} \in \mathbb{X}, f_j(\mathbf{x}) = 0\},$$

et l'ensemble \mathbb{S} solution du CSP \mathcal{H} est alors

$$\mathbb{S} = \mathbb{S}(\mathcal{H}) = \{\mathbf{x} \in \mathbb{X}, \mathbf{f}(\mathbf{x}) = \mathbf{0}\} = \bigcap_{j \in \{1, \dots, n\}} \mathbb{S}_j. \quad (3.4)$$

Enfin, deux CSP \mathcal{H}^1 et \mathcal{H}^2 sont dits *équivalents*, ce que nous noterons $\mathcal{H}^1 \equiv \mathcal{H}^2$, si leurs ensembles solutions sont égaux :

$$\mathbb{S}(\mathcal{H}^1) = \mathbb{S}(\mathcal{H}^2).$$

3.1.1.2 Consistances

Soit un pavé $[\mathbf{x}] \subset \mathbb{X}$. Considérons dans un premier temps une unique contrainte \mathcal{H}_j .

La valeur $x_i \in [x_i]$ est dite *consistante*¹ avec \mathcal{H}_j s'il est possible d'instancier les autres composantes x_j de \mathbf{x} de sorte que la relation $f_j(\mathbf{x}) = 0$ soit satisfaite, ce qui se traduit par

$$\begin{aligned} \exists x_1 &\in [x_1], \dots, x_{i-1} \in [x_{i-1}], x_{i+1} \in [x_{i+1}], \dots, x_m \in [x_m], \\ \mathbf{x} &= (x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_m) \in \mathbb{S}_j. \end{aligned}$$

Ceci revient à dire que x_i appartient à la projection sur $[x_i]$ de $\mathbb{S}_j \cap [\mathbf{x}]$, notée $\pi_i(\mathbb{S}_j \cap [\mathbf{x}])$. Le domaine $[x_i]$ est dit *consistant* avec la contrainte \mathcal{H}_j si toute valeur x_i de $[x_i]$ est consistante avec \mathcal{H}_j :

$$[x_i] = \pi_i(\mathbb{S}_j \cap [\mathbf{x}]).$$

Dans le cas d'un ensemble de plusieurs contraintes, on distingue deux types de consistances sur les domaines.

Le domaine $[x_i]$ est dit *globalement consistant* ou *consistant* avec le CSP \mathcal{H} si toute valeur x_i de $[x_i]$ est consistante avec toutes les contraintes prises simultanément,

$$[x_i] = \pi_i(\mathbb{S} \cap [\mathbf{x}]) = \pi_i\left(\bigcap_{j \in \{1, \dots, n\}} \mathbb{S}_j \cap [\mathbf{x}]\right). \quad (3.5)$$

Il est très difficile de vérifier en pratique la consistance globale d'un domaine. D'autres propriétés peuvent alors être considérées.

Le domaine $[x_i]$ est dit *localement consistant* (plus précisément arc consistant) avec le CSP \mathcal{H} si

$$[x_i] = \bigcap_{j \in \{1, \dots, n\}} \pi_i(\mathbb{S}_j \cap [\mathbf{x}]). \quad (3.6)$$

Remarque 3.1.3 *Supposons $[x_i]$ globalement consistant avec le CSP \mathcal{H} . Comme deux ensembles \mathbb{A} et \mathbb{B} quelconques satisfont la propriété*

$$\pi_i(\mathbb{A} \cap \mathbb{B}) \subset \pi_i(\mathbb{A}) \cap \pi_i(\mathbb{B}), \quad (3.7)$$

alors

$$[x_i] = \pi_i\left(\bigcap_{j \in \{1, \dots, n\}} \mathbb{S}_j \cap [\mathbf{x}]\right) \subset \bigcap_{j \in \{1, \dots, n\}} \pi_i(\mathbb{S}_j \cap [\mathbf{x}]) \subset [x_i],$$

et donc $[x_i] = \bigcap_{j \in \{1, \dots, n\}} \pi_i(\mathbb{S}_j \cap [\mathbf{x}])$. La consistance globale implique donc la consistance locale. La figure 3-1 illustre ces notions sur un exemple très simple où seules les deux premières composantes $[x_1]$ et $[x_2]$ sont représentées.

Le CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in \mathbb{X})$ est *globalement consistant* si tous ses domaines \mathbb{X}_i sont consistants. Dans ce cas, le pavé $[\mathbf{x}] \subset \mathbb{X}$ est le pavé enveloppe de $\mathbb{S} \cap [\mathbf{x}]$, que nous noterons

$$[\mathbf{x}] = [\mathbb{S} \cap [\mathbf{x}]]. \quad (3.8)$$

¹Conscients que consistance est une traduction rapide et contestable du terme anglais consistency, et que cohérence semblerait préférable, nous l'avons conservé par souci de cohérence avec les sources bibliographiques dont nous nous sommes inspirés.

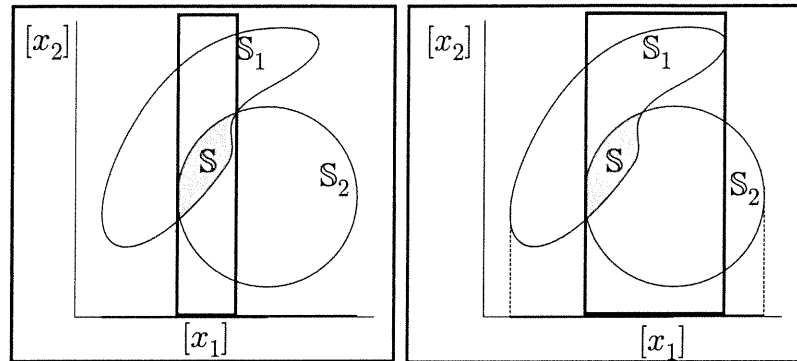


Figure 3-1: A gauche, l'intervalle $[x_1]$ est globalement consistant avec le CSP \mathcal{H} d'ensemble solution $\mathbb{S} = \mathbb{S}_1 \cap \mathbb{S}_2$. A droite, l'intervalle $[x_1]$ est seulement localement consistant avec le CSP \mathcal{H} .

Remarque 3.1.4 Si le pavé \mathbb{X} est consistant avec le CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in \mathbb{X})$, alors $\mathbb{X} = [\mathbb{S} \cap \mathbb{X}]$, d'après (3.8). Or d'après (3.4), $\mathbb{S} \subset \mathbb{X}$, donc $\mathbb{X} = [\mathbb{S}]$.

Sur la figure 3-2, le CSP $\mathcal{H}^x : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$ est globalement consistant, mais pas le CSP $\mathcal{H}^y : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{y}])$.

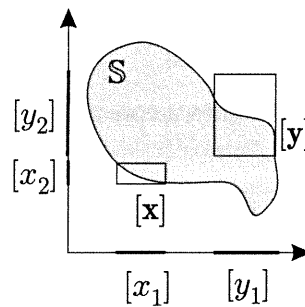


Figure 3-2: $[x_1]$ et $[x_2]$ sont consistants avec le CSP \mathcal{H} d'ensemble solution \mathbb{S} , donc \mathcal{H} est globalement consistant sur $[\mathbf{x}]$. Par contre \mathcal{H} n'est pas globalement consistant sur $[\mathbf{y}]$ car $[y_2]$ n'est pas consistant.

Remarque 3.1.5 De même qu'au chapitre précédent, les définitions que nous venons de présenter ne s'appliquent qu'à des intervalles théoriques. En effet, la mise en oeuvre de ces consistances appliquées à des domaines continus sur un ordinateur de précision finie ne peut permettre d'obtenir l'évaluation exacte de l'ensemble solution \mathbb{S} , car celui-ci peut contenir des éléments non-représentables en machine. De nouvelles consistances qui approximent l'arc-consistance sont alors définies. La consistance d'enveloppe est ainsi implémentée en lieu et place de la consistance d'arc. La mise en oeuvre de ces consistances est abordée dans l'annexe 8.2 afin de faciliter la lecture de ce chapitre, où nous continuerons à utiliser le terme de consistance d'arc.

3.1.1.3 Sous-vecteurs

Le vecteur $\mathbf{p} = (u_1, u_2, \dots, u_p)^\top$ est un *sous-vecteur* de $\mathbf{x} = (x_1, \dots, x_m)^\top$ s'il existe un sous-ensemble $\mathcal{I} = \{i_1, \dots, i_{n_{\mathcal{I}}}\}$ de $\{1, \dots, m\}$ tel que $\mathbf{p} = (x_{i_1}, \dots, x_{i_{n_{\mathcal{I}}}})^\top$. On notera alors $\mathbf{p} = \mathbf{x}_{\mathcal{I}}$.

L'ensemble \mathcal{J} tel que $\mathcal{I} \cap \mathcal{J} = \emptyset$ et $\mathcal{I} \cup \mathcal{J} = \{1, \dots, m\}$ est appelé *ensemble complémentaire* de \mathcal{I} . Nous appellerons alors *sous-vecteur complémentaire* le sous-vecteur $\mathbf{q} = \mathbf{x}_{\mathcal{J}}$ de dimension $q = m - p$, et nous noterons pour plus de simplicité, $\mathbf{x} = (\mathbf{p}, \mathbf{q})$.

Le sous-vecteur $\mathbf{p} = \mathbf{x}_{\mathcal{I}}$ est consistant avec le CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$ s'il existe un sous-vecteur \mathbf{q} complémentaire de sorte que $\mathbf{x} = (\mathbf{p}, \mathbf{q}) \in \mathbb{S}(\mathcal{H})$.

Soit $\mathcal{H}^1(\mathbf{p}, \mathbf{g}, [\mathbf{p}])$ et $\mathcal{H}^2(\mathbf{x}, \mathbf{f}, [\mathbf{x}])$ deux CSP. Si toute solution \mathbf{p} de \mathcal{H}^1 est un sous-vecteur de \mathbf{x} ($\mathbf{p} = \mathbf{x}_{\mathcal{I}}$) consistant avec \mathcal{H}^2 alors

$$\pi_{\mathcal{I}}(\mathbb{S}(\mathcal{H}^2)) = \mathbb{S}(\mathcal{H}^1),$$

où $\pi_{\mathcal{I}}(\cdot)$ est l'opérateur de projection d'un ensemble de dimension m sur le sous-espace $\mathbb{X}_{\mathcal{I}} = \mathbb{X}_{i_1} \times \dots \times \mathbb{X}_{i_{n_{\mathcal{I}}}}$.

L'ensemble solution $\mathbb{S}(\mathcal{H}^1)$ peut donc être déduit de l'ensemble solution $\mathbb{S}(\mathcal{H}^2)$. Nous dirons que \mathcal{H}^2 est une *approximation extérieure* de \mathcal{H}^1 , ce que nous noterons par $\mathcal{H}^2 \Rightarrow \mathcal{H}^1$.

Remarque 3.1.6 *Il devient maintenant très facile de traiter des CSP définis par des inégalités du type :*

$$\mathcal{H}^{ineg} : (\mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \mathbf{x} \in \mathbb{X}), \quad (3.9)$$

en définissant une approximation extérieure de \mathcal{H}^{ineg} , à l'aide de variables supplémentaires, comme le montre l'exemple 3.1.1.

Exemple 3.1.1 *Soit le CSP $\mathcal{H}^{ineg} : (p_1 + p_2 \leq 0, \mathbf{p} \in \mathbb{R}^{\times 2})$. Le CSP*

$$\mathcal{H} : (x_1 + x_2 + x_3 = 0, \mathbf{x} \in \mathbb{R}^{\times 2} \times [0, +\infty[)$$

est une approximation extérieure de \mathcal{H}^{ineg} . En particulier, la projection $\pi_{12}(\mathbb{S}(\mathcal{H}))$ sur $[x_1] \times [x_2]$ de l'ensemble solution \mathbb{S} du CSP \mathcal{H} est égale à l'ensemble solution $\mathbb{S}(\mathcal{H}^{ineg})$.

Nous montrerons qu'il est parfois possible d'améliorer les performances d'un contracteur associé à un problème \mathcal{H}^1 en appliquant ce contracteur à une approximation extérieure \mathcal{H}^2 du problème \mathcal{H}^1 .

La résolution des CSP est un problème qui a reçu une attention considérable. Les premiers CSP visaient à décrire des ensembles discrets ([Wal75], [Mac77]). Des approches utilisables quand les ensembles \mathbb{X}_i sont continus sont apparues plus récemment ([Cle87], [Dav87], [Lho93], et les références contenues dans [BSV⁺01]). Elles utilisent la notion de *contracteur*.

3.1.1.4 Contracteur

Un *contracteur* \mathcal{C} d'un CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$ est un opérateur capable de remplacer le pavé $[\mathbf{x}]$ par un pavé $\mathcal{C}([\mathbf{x}])$ de taille inférieure, tout en préservant l'intégralité de l'ensemble solution \mathbb{S} de \mathcal{H} . On peut donc écrire

$$\mathbb{S} \subset \mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}]. \quad (3.10)$$

Un contracteur vise donc à éliminer des valeurs inconsistantes des domaines $[x_i]$. Si toutes les valeurs inconsistantes ont été éliminées, le contracteur \mathcal{C} est dit *optimal*. Dans ce cas, le pavé $\mathcal{C}([\mathbf{x}])$ correspond à l'enveloppe $[\mathbb{S}]$ de \mathbb{S} , d'après la remarque 3.1.4.

Un contracteur \mathcal{C} d'un CSP \mathcal{H} d'ensemble solution \mathbb{S} satisfait donc les propriétés

$$\forall [\mathbf{x}] \subset \mathbb{X}, \mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}], \quad (3.11)$$

$$\forall [\mathbf{x}] \subset \mathbb{X}, [\mathbf{x}] \cap \mathbb{S} \subset \mathcal{C}([\mathbf{x}]). \quad (3.12)$$

qui découlent de (3.10).

Un contracteur est *monotone* si

$$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow \mathcal{C}([\mathbf{x}]) \subset \mathcal{C}([\mathbf{y}]),$$

il est *idempotent* si

$$\mathcal{C} \circ \mathcal{C}([\mathbf{x}]) = \mathcal{C}([\mathbf{x}]). \quad (3.13)$$

Remarque 3.1.7 *Tous les contracteurs présentés dans ce chapitre seront monotones [Gra98], mais pas nécessairement idempotents.*

Le pavé $[\mathbf{x}_\infty]$ est un *point fixe* du contracteur \mathcal{C} si

$$\mathcal{C}([\mathbf{x}_\infty]) = [\mathbf{x}_\infty].$$

Remarque 3.1.8 *Le pavé enveloppe $[\mathbb{S}]$ de $\mathbb{S}(\mathcal{H})$ est un point fixe de tout contracteur pour \mathcal{H} .*

On dira que \mathcal{C}_1 est plus *efficace* que \mathcal{C}_2 sur le domaine $[\mathbf{x}]$ (noté $\mathcal{C}_1 \prec \mathcal{C}_2$) si

$$\mathcal{C}_1([\mathbf{x}]) \subset \mathcal{C}_2([\mathbf{x}]).$$

Deux types de contracteur du CSP \mathcal{H} défini par $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$ peuvent être construits. Les contracteurs du premier type utilisent le calcul formel (en particulier les bases de Gröbner, voir [Gra98] et les projets LOCO, CONTRAINTES [Webg] et COCONUT [BSV⁺01]) pour minimiser le pessimisme de l'évaluation de \mathbf{f} sur le pavé $[\mathbf{x}]$ dû au problème de dépendance, en minimisant les occurrences des variables x_i dans l'expression de \mathbf{f} . Ces contracteurs sont optimaux si chaque variable x_i apparaît au plus une fois dans l'expression de \mathbf{f} obtenue.

Exemple 3.1.2 *Soit le CSP $\mathcal{H} : (f(x) = x + x = 0, x \in [-5, 5])$. L'ensemble solution $\mathbb{S} = [0, 0]$ est immédiatement obtenu en remplaçant l'expression de $f(x)$ par $2x$.*

Les contracteurs du second type utilisent le calcul numérique et correspondent à une extension aux intervalles d'un type particulier d'algorithme, appelé *sous-résolveur fini*. C'est à eux que nous allons consacrer le reste de ce chapitre.

3.1.1.5 Résolveur et sous-résolveur

Un *sous-résolveur* du CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$ est un algorithme qui calcule les valeurs de certaines variables x_i quand d'autres variables x_j sont connues.

Soient $\mathbf{u} = \mathbf{x}_{\mathcal{I}}$ et $\mathbf{v} = \mathbf{x}_{\mathcal{J}}$ deux sous-vecteurs de \mathbf{x} . Un *sous-résolveur fini* associé à \mathcal{H} est un algorithme fini de nature ensembliste $\phi : \mathbf{u} \mapsto \phi(\mathbf{u})$ tel que

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \Rightarrow \mathbf{v} \in \phi(\mathbf{u}). \quad (3.14)$$

On appellera \mathbf{u} et \mathbf{v} respectivement l'entrée et la sortie de ϕ . Dans la suite, nous serons amenés à utiliser abondamment les sous-résolveurs finis ϕ_i d'entrée ${}^i\mathbf{x} = (x_1 \dots x_{i-1} x_{i+1} \dots x_m)^T$ et de sortie x_i .

Remarque 3.1.9 Les sous-vecteurs \mathbf{u} et \mathbf{v} ne sont pas nécessairement complémentaires, comme le montre la figure 3-3, où $\mathbf{u} = (x_1 \ x_2 \ x_4 \ x_5)^T$ et $\mathbf{v} = (x_3)$.

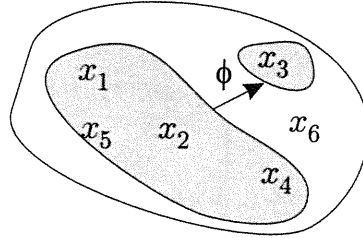


Figure 3-3: Exemple d'un sous-résolveur

Il est possible de définir très simplement des contracteurs en étendant aux intervalles des sous-résolveurs finis.

Théorème 3.1.1 Supposons qu'il existe un sous-résolveur ϕ_i d'entrée ${}^i\mathbf{x}$ et de sortie x_i associé au CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$. Soit $[\phi_i]$ une fonction d'inclusion pour ϕ_i . On peut écrire

$$\pi_i(\mathbb{S} \cap [\mathbf{x}]) \subset [\phi_i]([{}^i\mathbf{x}]) \cap [x_i].$$

Preuve.

Soit ϕ_i un sous-résolveur d'entrée ${}^i\mathbf{x}$ et de sortie x_i associé au CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$. D'après (3.14),

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \Rightarrow x_i \in \phi_i({}^i\mathbf{x}). \tag{3.15}$$

D'où

$$\begin{aligned} x_i \in \pi_i(\mathbb{S} \cap [\mathbf{x}]) &\iff \exists {}^i\mathbf{x} \mid (x_i, {}^i\mathbf{x}) \in \mathbb{S} \text{ et } (x_i, {}^i\mathbf{x}) \in [\mathbf{x}] \\ &\iff \exists {}^i\mathbf{x} \in [{}^i\mathbf{x}] \mid \mathbf{f}(\mathbf{x}) = \mathbf{0} \text{ où } \mathbf{x} = (x_i, {}^i\mathbf{x}) \text{ et } x_i \in [x_i] \\ &\stackrel{(3.15)}{\Rightarrow} \exists {}^i\mathbf{x} \in [{}^i\mathbf{x}] \mid x_i \in \phi_i({}^i\mathbf{x}) \text{ et } x_i \in [x_i] \\ &\Rightarrow x_i \in \phi_i([{}^i\mathbf{x}]) \cap [x_i]. \end{aligned}$$

Donc $\pi_i(\mathbb{S} \cap [\mathbf{x}]) \subset \phi_i([{}^i\mathbf{x}]) \cap [x_i]$. ■

Lemme 3.1.1 (Inversion explicite) Supposons qu'il soit possible d'isoler la variable x_i dans l'expression $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, c'est-à-dire qu'il existe un sous-résolveur ϕ_i d'entrée ${}^i\mathbf{x}$ et de sortie x_i satisfaisant

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \iff x_i \in \phi_i({}^i\mathbf{x}). \tag{3.16}$$

Si ϕ_i est une fonction continue et $[\phi_i]$ une fonction d'inclusion minimale de ϕ_i , alors

$$\pi_i(\mathbb{S} \cap [\mathbf{x}]) = [\phi_i]([{}^i\mathbf{x}]) \cap [x_i].$$

Preuve.

Découle directement du théorème 3.1.1, en remplaçant l'implication par une équivalence. Lorsque $[\phi_i]$ est une fonction d'inclusion minimale, $\phi_i([\mathbf{x}]) \cap [x_i] = [\phi_i]([\mathbf{x}]) \cap [x_i]$. ■

Théorème 3.1.2 (Extension aux sous-vecteurs) *Soit \mathcal{I} un sous-ensemble quelconque de $\{1, \dots, m\}$ et $\mathcal{J} = \{j_1, \dots, j_{n_{\mathcal{J}}}\}$ son ensemble complémentaire. Soit ϕ un sous-résolveur fini d'un CSP \mathcal{H} , d'entrée $\mathbf{x}_{\mathcal{I}}$ et de sortie $\mathbf{x}_{\mathcal{J}}$. L'opérateur \mathcal{C} défini par*

$$\mathcal{C} : [\mathbf{x}] = ([\mathbf{x}_{\mathcal{I}}], [\mathbf{x}_{\mathcal{J}}]) \longmapsto \left(\begin{array}{c} [x_{j_1}] \cap [\phi_{j_1}]([\mathbf{x}_{\mathcal{I}}]) \\ \dots \\ [\mathbf{x}_{\mathcal{I}}], [x_{j_i}] \cap [\phi_{j_i}]([\mathbf{x}_{\mathcal{I}}]) \\ \dots \\ [x_{n_{\mathcal{J}}}] \cap [\phi_{n_{\mathcal{J}}}]([\mathbf{x}_{\mathcal{I}}]) \end{array} \right), \quad (3.17)$$

où $[\phi_i]$ est une fonction d'inclusion de ϕ_i , est un contracteur pour \mathcal{H} .

Preuve.

Soit $[\mathbf{x}]$ un élément quelconque de \mathbb{X} . Il suffit de montrer que (3.11) et (3.12) sont satisfaites. L'inclusion $\mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}]$ est évidente d'après la construction de \mathcal{C} selon (3.17). Soit \mathbf{x} un élément de $[\mathbf{x}] \cap \mathcal{S}$. Alors

$$\begin{aligned} \mathbf{x} &\in [\mathbf{x}] \text{ et } \mathbf{f}(\mathbf{x}) = \mathbf{0}, \\ &\Rightarrow \mathbf{x} \in [\mathbf{x}] \text{ et } \exists \mathbf{x}_{\mathcal{I}} \in [\mathbf{x}_{\mathcal{I}}] \mid \mathbf{x}_{\mathcal{J}} \in \phi(\mathbf{x}_{\mathcal{I}}), \\ &\Rightarrow \mathbf{x} \in [\mathbf{x}] \text{ et } \mathbf{x}_{\mathcal{J}} \in \phi([\mathbf{x}_{\mathcal{I}}]). \end{aligned}$$

■

Un sous-résolveur d'entrée et de sortie \mathbf{x} est appelé *résolveur*. Une méthode d'élaboration d'un résolveur fréquemment rencontrée consiste à mettre en œuvre une méthode à point fixe.

Un *résolveur à point fixe* pour le CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$ se construit à partir d'une fonction mathématique $\psi : \mathbb{R}^m \longrightarrow \mathbb{R}^m$ qui vérifie

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \iff \mathbf{x} = \psi(\mathbf{x}), \quad (3.18)$$

de sorte que si la suite $(\mathbf{x}_n)_{n \in \mathbb{N}}$ définie par $\mathbf{x}_{n+1} = \psi(\mathbf{x}_n)$ converge, la limite \mathbf{x}_{∞} de cette suite est une solution de \mathcal{H} . Soulignons qu'ici, la relation (3.18) exprime une équivalence et non plus seulement une implication comme dans (3.14).

Soit $[\psi] : \mathbb{IR}^m \longrightarrow \mathbb{IR}^m$ une fonction d'inclusion pour ψ . On construit un contracteur à point fixe \mathcal{C}_{PF} pour \mathcal{H} en appliquant (3.17):

$$\mathcal{C}_{\text{PF}} : [\mathbf{x}] \longmapsto [\mathbf{x}] \cap [\psi]([\mathbf{x}]).$$

Une grande partie des contracteurs présentés dans le paragraphe suivant sont des contracteurs à point fixe. Cependant, leurs domaines d'application et leurs performances diffèrent et sont complémentaires. Trois critères nous ont permis de les classer : la linéarité de la contrainte \mathbf{f} , le rapport entre les nombres des inconnues et des contraintes, et la taille $w([\mathbf{x}])$ de $[\mathbf{x}]$.

Les premiers contracteurs que nous présentons maintenant sont issus de l'analyse par intervalles. Leur principe repose à chaque fois sur l'élaboration d'un résolveur fini ϕ sur les réels, étendu ensuite aux intervalles à l'aide de (3.17). Dans cette deuxième étape, il est important d'élaborer une fonction d'inclusion $[\phi]$ de ϕ performante. Nous présentons ici brièvement quelques résolveurs très fréquemment rencontrés. D'autres contracteurs (contracteur de Krawczyk par

exemple) existent, pour plus d'informations, nous renvoyons le lecteur à [JKDW01] et [Neu90]. Nous avons choisi de distinguer deux types de CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in \mathbb{X})$. Le paragraphe 3.1.2 présente deux algorithmes très connus capables de contracter des CSP où \mathbf{f} est une fonction linéaire en \mathbf{x} . Par abus de langage, nous parlerons de CSP linéaires. Le paragraphe 3.1.3 introduit des contracteurs de CSP non linéaires. En particulier, nous y introduisons le contracteur par linéarisation parallèle, dont nous détaillerons une application au paragraphe 3.3.3.

3.1.2 Contracteurs de CSP linéaires

Nous posons alors $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$. De façon générale, il s'agit donc de contracter le CSP \mathcal{H} défini par

$$\mathcal{H} : (\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}, \mathbf{A} \in [\mathbf{A}], \mathbf{b} \in [\mathbf{b}], \mathbf{x} \in [\mathbf{x}]). \quad (3.19)$$

Nous présentons différents contracteurs du CSP \mathcal{H} selon que \mathbf{A} et \mathbf{b} sont incertains ou non.

3.1.2.1 Contracteur de Gauss-Seidel

Il s'agit d'un contracteur à point fixe, qui contracte le CSP \mathcal{H} défini par (3.19) où \mathbf{A} et \mathbf{b} peuvent être incertains : $\mathbf{A} \in [\mathbf{A}], \mathbf{b} \in [\mathbf{b}]$. Nous supposons de plus que la matrice \mathbf{A} est carrée avec tous ses éléments diagonaux non-nuls. Décomposons \mathbf{A} sous la forme

$$\mathbf{A} = \Lambda + \mathbf{M},$$

où Λ est la matrice diagonale possédant les mêmes éléments diagonaux que \mathbf{A} . La relation réelle $\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}$ s'écrit de façon équivalente $\Lambda\mathbf{x} + \mathbf{M}\mathbf{x} = \mathbf{b}$. Puisque tous les éléments diagonaux de \mathbf{A} sont supposés non-nuls, Λ est inversible, et

$$\mathbf{x} = \Lambda^{-1}(\mathbf{b} - \mathbf{M}\mathbf{x}).$$

Les solutions en \mathbf{x} du CSP \mathcal{H} sont donc des points fixes de $\psi(\cdot)$ où

$$\psi(\mathbf{x}) = \Lambda^{-1}(\mathbf{b} - \mathbf{M}\mathbf{x}).$$

Connaissant une fonction d'inclusion pour $\psi(\cdot)$, le contracteur de Gauss-Seidel peut par exemple être mis en œuvre de la façon suivante :

$$\mathcal{C}_{\text{GS}} : \begin{pmatrix} [\mathbf{A}] \\ [\mathbf{b}] \\ [\mathbf{x}] \end{pmatrix} \mapsto \begin{pmatrix} [\mathbf{A}] \\ [\mathbf{b}] \\ [\psi]([\mathbf{x}]) = [\mathbf{x}] \cap [\Lambda]^{-1}([\mathbf{b}] - [\mathbf{M}][\mathbf{x}]) \end{pmatrix}.$$

Seul le pavé $[\mathbf{x}]$ est contracté.

3.1.2.2 Contracteur par programmation linéaire

Considérons maintenant le CSP \mathcal{H} défini par

$$\mathcal{H} : (\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}, \mathbf{x} \in [\mathbf{x}], \mathbf{b} \in [\mathbf{b}]),$$

où \mathbf{A} est certaine mais pas nécessairement carrée, et \mathbf{b} est incertain : $\mathbf{b} \in [\mathbf{b}]$. Dans ce cas, il est possible de transformer ce problème en $2m$ problèmes de programmation linéaire, de façon à utiliser les performances des algorithmes existants pour résoudre ce type de problème.

Le vecteur $[\mathbf{x}]$ est consistant avec le problème \mathcal{H} si et seulement s'il existe $\mathbf{b} \in [\mathbf{b}]$ tel que $\mathbf{Ax} - \mathbf{b} = \mathbf{0}$. En d'autres termes,

$$\begin{aligned} \exists \mathbf{b} \in [\mathbf{b}] \mid \mathbf{Ax} - \mathbf{b} = \mathbf{0} \\ \Leftrightarrow \mathbf{Ax} \in [\mathbf{b}] \\ \Leftrightarrow \begin{cases} \mathbf{Ax} \geq \underline{\mathbf{b}} \\ \mathbf{Ax} \leq \overline{\mathbf{b}} \end{cases} . \end{aligned}$$

Le plus petit pavé $[\mathbf{q}]$ contenant tous les \mathbf{x} solutions peut alors être calculé en résolvant les $2m$ problèmes de programmation linéaire :

$$\begin{aligned} \underline{q}_i &= \arg \min x_i \mid \begin{pmatrix} -\mathbf{A} \\ \mathbf{A} \end{pmatrix} \mathbf{x} \leq \begin{pmatrix} -\underline{\mathbf{b}} \\ \overline{\mathbf{b}} \end{pmatrix}, \mathbf{x} \in [\mathbf{x}], i = 1, \dots, m, \\ \overline{q}_i &= \arg \max x_i \mid \begin{pmatrix} -\mathbf{A} \\ \mathbf{A} \end{pmatrix} \mathbf{x} \leq \begin{pmatrix} -\underline{\mathbf{b}} \\ \overline{\mathbf{b}} \end{pmatrix}, \mathbf{x} \in [\mathbf{x}], i = 1, \dots, m, \end{aligned}$$

de sorte que

$$\mathcal{C}_{\text{PL}} : \begin{pmatrix} \mathbf{A} \\ [\mathbf{b}] \\ [\mathbf{x}] \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{A} \\ [\mathbf{b}] \\ [\underline{\mathbf{q}}, \overline{\mathbf{q}}] \end{pmatrix} .$$

3.1.3 Contracteurs de CSP non-linéaires

Nous allons présenter dans cette partie deux types de contracteur pour des CSP non-linéaires

$$\mathcal{H}^1 : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}]) .$$

Le principe d'élaboration est le même pour les deux contracteurs. Il s'agit tout d'abord d'élaborer un CSP linéaire \mathcal{H}^2 , approximation extérieure de \mathcal{H}^1 , à l'aide d'une technique de linéarisation garantie, puis d'utiliser le contracteur pour CSP linéaire correspondant. La première technique de linéarisation présentée est à l'origine des contracteurs de Newton, et limite le domaine d'application de ces contracteurs aux cas où $\dim \mathbf{f} = m$. Nous présentons ensuite un nouveau contracteur, capable de contracter des CSP qui ne satisfont pas cette condition.

3.1.3.1 Contracteurs de Newton

Il s'appliquent lorsque le système à résoudre est carré, c'est-à-dire lorsque $\dim \mathbf{f} = \dim \mathbf{x} = m$. La première méthode consiste à élaborer un contracteur à point fixe pour le CSP \mathcal{H}^1 . On recherche donc une fonction $\psi(\cdot)$ satisfaisant :

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \Leftrightarrow \psi(\mathbf{x}) = \mathbf{x} . \quad (3.20)$$

Si \mathbf{f} était affine, on pourrait poser $\mathbf{f}(\mathbf{x}) = \mathbf{Ax} + \mathbf{b}$. A condition que \mathbf{A} soit inversible, il suffirait alors de choisir $\psi(\mathbf{x}) = \mathbf{x} - \mathbf{A}^{-1}(\mathbf{Ax} + \mathbf{b})$ pour obtenir une fonction $\psi(\cdot)$ satisfaisant (3.20). La suite $\mathbf{x}_{k+1} = \psi(\mathbf{x}_k)$ convergerait vers la solution $\mathbf{x}^* = -\mathbf{A}^{-1}\mathbf{b}$ en un pas.

Si \mathbf{f} n'est pas affine mais est au moins différentiable, de Jacobienne \mathbf{J} , on peut de façon analogue poser

$$\psi(\mathbf{x}) = \mathbf{x} - \mathbf{J}^{-1}\mathbf{f}(\mathbf{x}) ,$$

et la suite $\mathbf{x}_{k+1} = \psi(\mathbf{x}_k)$ converge (sans garantie) vers la solution obtenue par la méthode de Newton. Le contracteur résultant, noté \mathcal{C}_{N1} , s'écrit alors

$$\mathcal{C}_{N1} : [\mathbf{x}] \mapsto [\psi]([\mathbf{x}]) \cap [\mathbf{x}]$$

où $[\psi](\cdot)$ est une fonction d'inclusion pour $\psi(\cdot)$. Par exemple,

$$\mathcal{C}_{N1} : \left([\mathbf{x}] \mapsto [\mathbf{x}] - [\mathbf{J}]^{-1}[\mathbf{f}]([\mathbf{x}]) \right) \cap [\mathbf{x}].$$

La figure 3-4 à gauche illustre l'action de ce contracteur.

Lorsque \mathbf{f} est lentement variable et que le domaine $[\mathbf{x}]$ est petit, une version plus efficace de ce contracteur a été proposée. Pour cela, une approximation extérieure \mathcal{H}_2 de \mathcal{H}_1 est construite à l'aide du théorème de la valeur moyenne (chapitre 2). Soit \mathbf{x}_0 un vecteur quelconque de $[\mathbf{x}]$, choisi par l'utilisateur. Par exemple, on pourra prendre $\mathbf{x}_0 = \text{mid}([\mathbf{x}])$. Si \mathbf{x} est solution du CSP \mathcal{H}^1 , il est également solution du CSP

$$(\mathbf{f}(\mathbf{x}_0) + \mathbf{J}(\boldsymbol{\xi})(\mathbf{x} - \mathbf{x}_0) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}], \boldsymbol{\xi} \in [\mathbf{x}]),$$

qui est équivalent au CSP :

$$\mathcal{H}^2 : (\mathbf{J}(\boldsymbol{\xi})\mathbf{p} + \mathbf{f}(\mathbf{x}_0) = \mathbf{0}, \mathbf{p} = \mathbf{x} - \mathbf{x}_0, \mathbf{x} \in [\mathbf{x}], \boldsymbol{\xi} \in [\mathbf{x}]).$$

Le CSP \mathcal{H}^2 peut être contracté par le contracteur de Gauss-Seidel décrit précédemment. Le contracteur \mathcal{C}_{N2} est alors très simplement :

$$\mathcal{C}_{N2} : [\mathbf{x}] \mapsto \mathcal{C}_{GS} \begin{pmatrix} \mathbf{J}([\mathbf{x}]) \\ -\mathbf{f}(\mathbf{x}_0) \\ [\mathbf{x}] - \mathbf{x}_0 \end{pmatrix}.$$

La projection de l'ensemble solution de \mathcal{H}^2 fournit alors le pavé contracté $[\mathbf{q}]$, et on a $\mathcal{C}_{N2} \prec \mathcal{C}_{N1}$.

3.1.3.2 Contracteur par linéarisation parallèle

L'objectif visé ici est de proposer un algorithme capable de contracter une classe plus large de CSP que ne le peuvent les contracteurs décrits aux paragraphes précédents, c'est-à-dire de traiter des CSP non-linéaires et rectangulaires. Le contracteur résultant, noté \mathcal{C}_{\parallel} , a été décrit et utilisé dans [BJKW01b] et [Jau01].

La principale limite des contracteurs de Newton est que la technique de linéarisation utilisée fournit un système du type $\mathbf{Ax} - \mathbf{b} = \mathbf{0}$, où \mathbf{A} est incertaine. Il est alors impératif d'utiliser des contracteurs de type \mathcal{C}_{GS} , qui nécessitent que \mathbf{A} soit inversible. Afin de pouvoir traiter des systèmes rectangulaires, il est donc nécessaire d'utiliser des contracteurs du type \mathcal{C}_{PL} qui n'utilisent pas cette hypothèse. Afin de pouvoir utiliser \mathcal{C}_{PL} , il est donc nécessaire d'établir un encadrement de \mathbf{f} où \mathbf{A} est certaine.

Supposons \mathbf{f} différentiable, et notons \mathbf{J} la matrice Jacobienne $\frac{d\mathbf{f}}{d\mathbf{x}}(\mathbf{x})$. Soit à nouveau \mathbf{x}_0 un vecteur de $[\mathbf{x}]$. Alors le théorème de la valeur moyenne indique qu'il existe $\boldsymbol{\xi}$ dans $[\mathbf{x}]$ tel que

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_0) + \mathbf{J}(\boldsymbol{\xi})(\mathbf{x} - \mathbf{x}_0).$$

De façon équivalente,

$$\begin{aligned} \exists \xi \in [\mathbf{x}] \mid \mathbf{f}(\mathbf{x}) &= \mathbf{f}(\mathbf{x}_0) + \mathbf{J}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + (\mathbf{J}(\xi) - \mathbf{J}(\mathbf{x}_0))(\mathbf{x} - \mathbf{x}_0), \\ \exists \xi \in [\mathbf{x}] \mid \mathbf{f}(\mathbf{x}) &= \mathbf{J}(\mathbf{x}_0)\mathbf{x} + \mathbf{f}(\mathbf{x}_0) - \mathbf{J}(\mathbf{x}_0)\mathbf{x}_0 + (\mathbf{J}(\xi) - \mathbf{J}(\mathbf{x}_0))(\mathbf{x} - \mathbf{x}_0). \end{aligned} \quad (3.21)$$

En posant

$$\begin{aligned} \mathbf{A} &= \mathbf{J}(\mathbf{x}_0), \\ \text{et } [\mathbf{b}] &= [\underline{\mathbf{b}}, \bar{\mathbf{b}}] = \mathbf{f}(\mathbf{x}_0) - \mathbf{J}(\mathbf{x}_0)\mathbf{x}_0 + (\mathbf{J}([\mathbf{x}]) - \mathbf{J}(\mathbf{x}_0))(\mathbf{x} - \mathbf{x}_0), \end{aligned}$$

alors (3.21) est équivalente à

$$\mathbf{Ax} + \underline{\mathbf{b}} \leq \mathbf{f}(\mathbf{x}) \leq \mathbf{Ax} + \bar{\mathbf{b}}. \quad (3.22)$$

Nous avons donc encadré \mathbf{f} par deux hyperplans parallèles, en rejetant l'ensemble des incertitudes dans le vecteur intervalle $[\mathbf{b}]$, comme l'illustre la figure 3-4 à droite.

Le contracteur \mathcal{C}_{\parallel} résultant peut par exemple être mis en œuvre de la façon suivante :

$$\mathcal{C}_{\parallel} : [\mathbf{x}] \mapsto \mathcal{C}_{\text{PL}} \begin{pmatrix} \mathbf{J}(\mathbf{x}_0) \\ \mathbf{f}(\mathbf{x}_0) - \mathbf{J}(\mathbf{x}_0)\mathbf{x}_0 + ((\mathbf{J}([\mathbf{x}]) - \mathbf{J}(\mathbf{x}_0))([\mathbf{x}] - \mathbf{x}_0)) \\ [\mathbf{x}] \end{pmatrix}.$$

La qualité de cet encadrement par linéarisation parallèle peut être quantifiée par le rapport $\frac{w([\mathbf{b}])}{w([\mathbf{x}])}$ (voir la figure 3-4 à droite) :

$$\frac{w([\mathbf{b}])}{w([\mathbf{x}])} = \frac{w(((\mathbf{J}([\mathbf{x}]) - \mathbf{J}(\mathbf{x}_0))([\mathbf{x}] - \mathbf{x}_0))}{w([\mathbf{x}])}.$$

Si $[\mathbf{J}]$ est convergente, $w([\mathbf{b}])/w([\mathbf{x}]) \rightarrow 0$ quand $w([\mathbf{x}]) \rightarrow 0$, et la précision de l'encadrement croît lorsque $[\mathbf{x}]$ converge vers un point \mathbf{x} . Une application, traitée dans [BJKW01b], sera détaillée au chapitre 4.

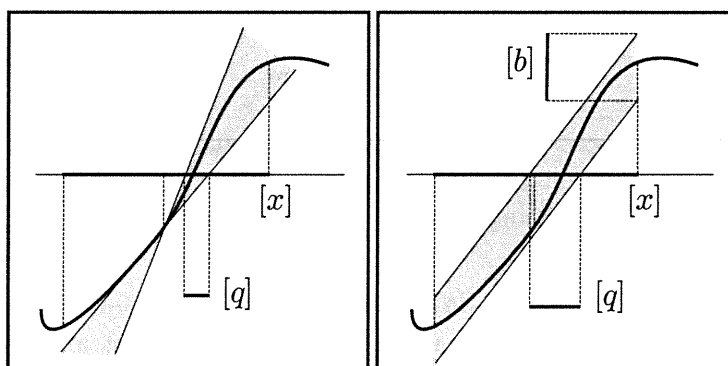


Figure 3-4: Deux techniques de linéarisation garanties utilisées dans les contracteurs de CSP non linéaires : \mathcal{C}_{N1} (à gauche) et \mathcal{C}_{\parallel} (à droite)

Les contracteurs présentés sont efficaces si la taille du pavé $[\mathbf{x}]$ à traiter est faible. Nous présentons maintenant une classe complémentaire de contracteurs qui s'appuie sur la propagation de contraintes et dont les éléments peuvent être utilisés que \mathbf{f} soit linéaire ou non. Ces contracteurs

sont issus de la combinaison de l'arithmétique des intervalles et de la propagation de contraintes, qui a été proposée indépendamment par [Cle87] et [Dav87]. L'étude de ces contracteurs pour l'estimation a donné lieu à la publication [JKBW01].

3.2 Contractions par projection

Dans un premier temps, nous considérons des fonctions f à valeurs dans \mathbb{R} , c'est-à-dire que nous nous restreignons à étudier la contrainte \mathcal{H} définie par

$$\mathcal{H} : (f(\mathbf{x}) = 0, \mathbf{x} \in [\mathbf{x}]), \quad (3.23)$$

où $[\mathbf{x}]$ est un domaine de \mathbb{R}^m .

De plus, dans tout ce paragraphe les sous-résolveurs ϕ_i considérés auront pour entrée ${}^i\mathbf{x}$ et pour sortie x_i :

$$\phi_i : ({}^i\mathbf{x}) \mapsto x_i = \phi({}^i\mathbf{x}).$$

3.2.1 Relation réelle, graphe et projection

La relation $x_i = \phi({}^i\mathbf{x})$ est une *relation réelle d'arité* $p \leq m$, qui caractérise un sous-ensemble de \mathbb{R}^p . En particulier, toutes les opérations arithmétiques et les relations d'ordre définissent des relations réelles. Nous allons montrer que f peut être exprimée sous la forme d'une relation réelle à l'aide d'une famille de sous-résolveurs $(\phi_i)_{i \in \{1, \dots, m\}}$. L'exemple suivant illustre un cas particulièrement simple d'élaboration de sous-résolveur.

Exemple 3.2.1 *Soit la fonction*

$$\begin{aligned} f : \mathbb{R}^3 &\rightarrow \mathbb{R} \\ \mathbf{x} &\mapsto x_1 + x_2 - x_3 \end{aligned}$$

Cette fonction f peut être décrite de façon équivalente par une relation réelle ternaire que nous notons ADD et qui symbolise l'addition de deux réels :

$$\text{ADD} : x_3 = x_1 + x_2. \quad (3.24)$$

Nous noterons alors $\mathcal{H} : (f(\mathbf{x}) = 0, \mathbf{x} \in [\mathbf{x}])$ sous la forme condensée $\mathcal{H} : (\mathbf{x}, \text{ADD}, [\mathbf{x}])$.

Remarque 3.2.1 *Attention, toute manipulation d'expression doit être effectuée avant l'écriture sous forme d'ensembles. Rappelons que l'expression $[x_1] + [x_2] - [x_3] = [0]$ n'est pas équivalente à $[x_3] = [x_1] + [x_2]$ (voir le chapitre 2).*

Dans la suite, nous serons fréquemment amenés à établir le graphe de représentation d'une relation. Nous le construirons de la manière suivante. Dans un premier temps, nous recensons l'ensemble des m variables x_i que nous représentons chacune par un nœud v_i . Ensuite, nous décomposons la relation p -aire en relations binaires modélisées par des arcs reliant les deux nœuds impliqués.

Exemple 3.2.1 (suite) *Le graphe de représentation de (3.24) est représenté à la figure 3-5 (à gauche).*

Remarque 3.2.2 *Par abus de langage, nous dirons qu'une contrainte est p -aire lorsque la relation réelle associée est p -aire.*

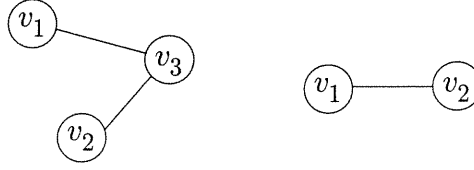


Figure 3-5: Graphes de représentation d'une contrainte ternaire (à gauche) et d'une contrainte binaire (à droite)

La contrainte ternaire ADD peut être utilisée pour contracter les domaines *a priori* de x_1 , x_2 et x_3 en éliminant très simplement des valeurs inconsistantes dans ces domaines.

Exemple 3.2.1 (suite) Soit $\mathbb{X} = [0, 5] \times [0, 4] \times [-1, 3]$. La valeur -1 pour x_3 est inconsistante avec ADD car

$$\forall x_1 \in [0, 5] \text{ et } \forall x_2 \in [0, 4], x_1 + x_2 \neq -1.$$

Soit une relation réelle c d'arité m et un entier $k \in \{1, \dots, m\}$. La $k^{\text{ième}}$ projection de c est l'ensemble $\pi_k(c)$ tel que

$$\pi_k(c) = \{x_k \in \mathbb{R} \mid \exists (x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_m) \in \mathbb{R}^{m-1}, \mathbf{x} \in c\}.$$

De même, on appelle $k^{\text{ième}}$ projection de \mathcal{H} l'ensemble $\pi_k(\mathcal{H}) = \pi_k(\mathbb{S}(\mathcal{H}))$ c'est-à-dire

$$\pi_k(\mathcal{H}) = \{x_k \in \mathbb{X}_k \mid \exists \binom{k}{\mathbf{x}} \in \binom{k}{\mathbb{X}}, x \in \mathbb{S}\}.$$

Soit $\mathcal{I}_k = \{1, \dots, k-1, k+1, \dots, m\}$. Nous notons $\rho_{\mathcal{I}_k}$ l'opérateur tel que

$$\rho_{\mathcal{I}_k} : [x_k] \mapsto \{x_k \in [x_k] \mid \exists \binom{k}{\mathbf{x}} \in \binom{k}{[x_k]}, (x_k, \binom{k}{\mathbf{x}}) \in \mathbb{S}\},$$

et ρ la *procédure de projection* [BG97] associée à la contrainte \mathcal{H} définie par

$$\rho : [\mathbf{x}] \mapsto \times_{k \in \{1, \dots, m\}} \rho_{\mathcal{I}_k}([x_k]).$$

Cet opérateur est un contracteur pour \mathcal{H} , d'après (3.17) et

$$\rho_{\mathcal{I}_k}([x_k]) = \pi_k(\mathbb{S}(\mathcal{H}) \cap [\mathbf{x}]).$$

La figure 3-6 illustre l'action de ce contracteur. D'après (3.6) la contrainte \mathcal{H} est donc *localement consistante* si aucun des domaines $[x_i]$ ne peut être réduit à l'aide des projections de la relation c définie par \mathcal{H} .

Dans la suite, nous proposerons différentes méthodes de mise en œuvre de l'opérateur de projection d'une contrainte \mathcal{H} selon la nature de celle-ci. Cet opérateur est un contracteur optimal pour une classe de contraintes spécifiques, appelées *contraintes primitives* que nous étudions au paragraphe suivant.

3.2.2 Projection d'une contrainte primitive

Une contrainte \mathcal{H} est *primitive* si la relation réelle qui la définit ne contient que des occurrences simples des variables x_i et au plus une opération arithmétique.

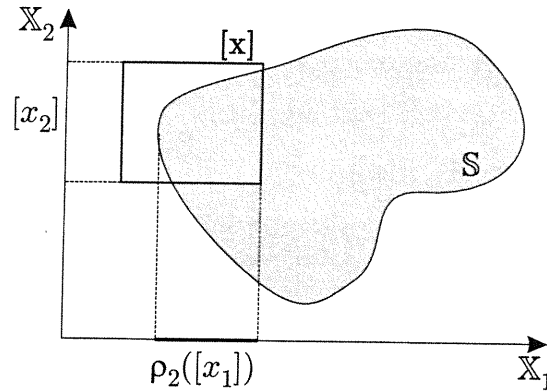


Figure 3-6: Projection de la contrainte d'ensemble solution S sur le domaine $[x_1]$

Nous distinguerons deux types de contrainte primitive :

- les contraintes ternaires ne faisant intervenir qu'un seul opérateur arithmétique : par exemple ADD ou la relation ternaire MULT : $x_3 = x_1 \cdot x_2$, qui s'écrit de façon équivalente $\text{MULT} \triangleq \{\mathbf{x} \in \mathbb{R}^3 \mid x_3 = x_1 \cdot x_2\}$,
- les fonctions élémentaires qui peuvent être interprétées comme des contraintes binaires. Par exemple, la relation réelle

$$\text{EXP} \triangleq \{\mathbf{x} \in \mathbb{R}^2 \mid x_2 = \exp(x_1)\},$$

(dont le graphe est représenté à la figure 3-5 à droite).

L'exemple suivant illustre la projection de la contrainte ADD.

Exemple 3.2.1 (fin) La projection de la contrainte $\mathcal{H} : (\mathbf{x}, \text{ADD}, \mathbb{X})$ sur x_1 est

$$\pi_1(\mathcal{H}) = \rho_{23}([x_1]) = \{x_1 \in [0, 5] \mid \exists (x_2, x_3) \in [0, 4] \times [-1, 3], x_3 = x_1 + x_2\}.$$

Dans ce cas précis, il est très facile de calculer explicitement $\pi_1(\mathcal{H})$. En effet :

$$\begin{aligned} \pi_1(\mathcal{H}) &= \{x_1 \in [0, 5] \mid \exists (x_2, x_3) \in [0, 4] \times [-1, 3], x_1 = x_3 - x_2\} \\ &= [0, 5] \cap ([-1, 3] - [0, 4]) \\ &= [0, 5] \cap [-5, 3] = [0, 3]. \end{aligned}$$

De même,

$$\begin{aligned} \pi_2(\mathcal{H}) &= \rho_{13}([x_2]) = [0, 4] \cap ([-1, 3] - [0, 3]) = [0, 4] \cap [-4, 3] = [0, 3], \\ \pi_3(\mathcal{H}) &= \rho_{12}([x_3]) = [-1, 3] \cap ([0, 3] + [0, 3]) = [-1, 3] \cap [0, 6] = [0, 3]. \end{aligned}$$

Le pavé $[q]$ défini par

$$[q] = \pi_1(\mathcal{H}) \times \pi_2(\mathcal{H}) \times \pi_3(\mathcal{H})$$

dont les projections sur les plans (x_1, x_2) et (x_1, x_3) sont représentées sur la figure 3-7, vérifie bien (3.11) et (3.12). De plus, la procédure de projection de \mathcal{H} est un contracteur optimal.

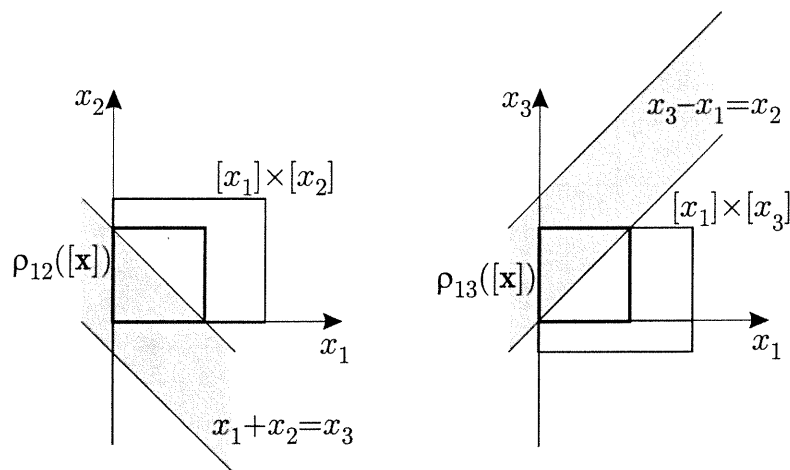


Figure 3-7: Projection de la contrainte de l'exemple 3.2.1 dans le plan (x_1, x_2) à gauche et dans le plan (x_1, x_3) à droite. Le pavé résultant de la contraction est optimal.

De façon générale, l'algorithme PADD suivant permet de contracter la contrainte ADD :

Algorithme PADD (es: $[x_1], [x_2], [x_3]$)		
1	$[x_3] := [x_3] \cap ([x_1] + [x_2]);$	(3.25)
2	$[x_1] := [x_1] \cap ([x_3] - [x_2]);$	
3	$[x_2] := [x_2] \cap ([x_3] - [x_1]).$	

De même, la mise en œuvre de la procédure de projection de la relation réelle EXP aboutit à l'algorithme PEXP suivant :

Algorithme PEXP (es: $[x_1], [x_2]$)	
1	$[x_2] := [x_2] \cap \text{exp}([x_1]);$
2	$[x_1] := [x_1] \cap \text{ln}([x_2]);$

Si la contrainte \mathcal{H} n'est pas primitive, l'optimalité ne peut pas toujours être garantie. Pour cette étude, nous allons distinguer les trois cas suivants :

1. il est possible d'élaborer un sous-résolveur ϕ sous la forme d'une fonction,
2. il est possible d'élaborer un résolveur ϕ sous la forme d'un arbre,
3. il n'est pas possible d'établir un tel résolveur.

3.2.3 Inversion explicite

Supposons qu'il soit possible d'élaborer un sous-résolveur ϕ sous la forme d'une fonction continue, que nous appellerons *fonction solution associée* à x_i . C'est le cas par exemple, si f ne fait apparaître que des occurrences simples des composantes x_i de \mathbf{x} au travers d'une composition de fonctions et d'opérations toutes inversibles. Il est alors possible d'obtenir à l'aide du calcul formel, une expression explicite $x_i = \phi_i({}^i\mathbf{x})$. Dans ce cas, nous possédons un ensemble de sous-résolveurs ϕ_i d'entrée ${}^i\mathbf{x}$ et de sortie x_i . A condition de disposer d'une fonction d'inclusion $[\phi_i]$

de chaque ϕ_i , on peut alors construire le contracteur :

$$\mathcal{C} : [\mathbf{x}] \longmapsto \left(\times_{i \in \{1, \dots, m\}} [\phi_i]({}^i[\mathbf{x}]) \right) \cap [\mathbf{x}] \quad (3.26)$$

pour tout i dans $\{1, \dots, m\}$.

Ce contracteur est optimal si $[\phi_i]$ optimale, pour tout $i \in \{1, \dots, m\}$, d'après le lemme 3.1.1.

Exemple 3.2.2 Soit la contrainte

$$\mathcal{H} : (x_1 \ln(\exp x_2 + x_3 \exp x_4) = x_5, \mathbf{x} \in [\mathbf{x}]).$$

Nous disposons pour cette contrainte des sous-résolveurs ϕ_i suivants :

$$\left\{ \begin{array}{l} \phi_1 : x_1 \mapsto \frac{x_5}{\ln(\exp x_2 + x_3 \exp x_4)} \\ \phi_2 : x_2 \mapsto \ln \left(\exp \left(\frac{x_5}{x_1} \right) - x_3 \exp x_4 \right) \\ \phi_3 : x_3 \mapsto \frac{\exp \left(\frac{x_5}{x_1} \right) - \exp x_2}{\exp x_4} \\ \phi_4 : x_4 \mapsto \ln \left(\frac{\exp \left(\frac{x_5}{x_1} \right) - \exp x_2}{x_3} \right) \\ \phi_5 : x_5 \mapsto x_1 \ln(\exp x_2 + x_3 \exp x_4) \end{array} \right.$$

Le contracteur associé défini par (3.26) utilisant les fonctions d'inclusion naturelles est optimal, puisque chaque fonction d'inclusion $[\phi_i]$ est ici optimale. Dans le cas de fonctions dans lesquelles des variables apparaissent plusieurs fois, rappelons qu'il est possible de diminuer le pessimisme de la fonction d'inclusion en utilisant par exemple, une forme centrée (au moins sur des pavés suffisamment petits).

Toute fonction f satisfaisant les conditions de ce paragraphe se comporte donc comme une contrainte primitive. Nous pourrions alors directement programmer un contracteur par projection de f .

Remarque 3.2.3 Il existe de nombreuses fonctions pour lesquelles il n'est pas possible d'établir d'expression explicite de la fonction inverse, bien que nous disposions d'une fonction d'inclusion optimale. La fonction $\text{lnc} : \begin{array}{l} [-0.5, 0.5] \longrightarrow \mathbb{R} \\ x \longmapsto \frac{\ln(1+x)}{x} \end{array}$ du chapitre précédent en est un exemple.

Dans ce cas, deux approches peuvent être envisagées. La première consiste à établir, à l'aide d'algorithmes itératifs comme ceux de Newton, une approximation extérieure des ϕ_i [Gra98]. La deuxième approche, qui consiste en la décomposition en contraintes primitives, est envisagée au paragraphe suivant.

Dans le paragraphe suivant, nous nous intéressons au cas de fonctions pour lesquelles nous ne disposons pas d'expression explicite de la fonction inverse, et où le résolveur sera élaboré sous la forme d'un arbre.

3.2.4 Contraction par décomposition

3.2.4.1 Décomposition d'une contrainte

Toute relation réelle peut se décomposer en une suite d'opérations n'impliquant que des opérateurs élémentaires et des fonctions binaires telles que celles rencontrées dans les contraintes primitives (par exemple, \cos , \sin ...) [Lho93]. Il est donc possible de décomposer la contrainte \mathcal{H} en une suite de contraintes primitives h_k en introduisant des variables intermédiaires. Signalons que l'intérêt d'introduire de telles variables avait déjà été remarqué dans la communauté de l'analyse par intervalles ([Kea96], chapitre 7). Nous noterons $\mathcal{D}(\mathcal{H})$ le CSP issu de la décomposition de \mathcal{H} .

Exemple 3.2.3 *La relation réelle du CSP \mathcal{H}*

$$\mathcal{H} : (x_1 \cos(x_2) \exp(x_3) + x_4 = 0, \mathbf{x} \in [\mathbf{x}]). \quad (3.27)$$

peut être décomposée en la série de contraintes primitives suivante, en introduisant les variables intermédiaires a_i :

$$\begin{cases} h_1 : a_1 = \cos(x_2) \\ h_2 : a_2 = x_1 a_1 \\ h_3 : a_3 = \exp(x_3) \\ h_4 : a_4 = a_2 a_3 \\ h_5 : a_5 = a_4 + x_4. \end{cases}$$

Le CSP $\mathcal{H}_d = \mathcal{D}(\mathcal{H})$ résultant de cette décomposition

$$\mathcal{H}_d : \left(\begin{cases} h_1 : a_1 = \cos(x_2) \\ h_2 : a_2 = x_1 a_1 \\ h_3 : a_3 = \exp(x_3) \\ h_4 : a_4 = a_2 a_3 \\ h_5 : a_5 = a_4 + x_4, \end{cases} , \mathbf{x} \in [\mathbf{x}], \mathbf{a} \in \mathbb{R}^{x_4} \times [0, 0] \right). \quad (3.28)$$

est une conjonction de $n_h = 5$ contraintes primitives, qui satisfait $\mathcal{H}_d \Rightarrow \mathcal{H}$.

Il s'agit alors de projeter le CSP $\mathcal{D}(\mathcal{H})$ qui est une conjonction de contraintes primitives. Lorsque le graphe de $\mathcal{D}(\mathcal{H})$ est connexe et ne comporte pas de cycle, on dit alors que ce graphe est un *arbre*, ou un *graphe acyclique*. Dans le cas contraire, on parle de *graphe cyclique*. Selon que le graphe de $\mathcal{D}(\mathcal{H})$ est ou non un arbre, l'optimalité du contracteur que nous allons présenter sera ou non assurée. Nous supposons dans le paragraphe 3.2.4.2 que le graphe de $\mathcal{D}(\mathcal{H})$ est un arbre. Le cas des graphes cycliques sera étudié au paragraphe 3.2.4.3.

3.2.4.2 Arbres

L'algorithme de Waltz ([Wal75]) est une des approches possibles. Son principe est de choisir parmi l'ensemble des contraintes primitives de $\mathcal{D}(\mathcal{H})$ une contrainte h_k quelconque et de la contracter par projection. Les domaines des variables x_i et x_j associées à cette contrainte sont alors contractés, et l'opération est itérée jusqu'à ce qu'aucune contrainte ne puisse plus être contractée. Ce contracteur vérifie donc la consistance locale de \mathcal{H} . La structure de cet algorithme est très simplement présentée à la table 3.1, où η est un critère d'arrêt à fixer par l'utilisateur. Les variables x_i et x_j sont les variables reliées par la contrainte h_k .

Algorithme WALTZ(es : $[\mathbf{x}]$, \mathcal{H})	
1	tant que $\ [\mathbf{x}]_0 - [\mathbf{x}]\ > \eta$
2	répéter
3	$[\mathbf{x}]_0 = [\mathbf{x}]$;
4	$k = \text{rand}(1, \dots, n_h)$
5	$[x_i] := \rho_j([x_i]); [x_j] := \rho_i([x_j])$;
6	fin tant que

Tableau 3.1: Contraction d'une conjonction de contraintes par projections successives

Cet algorithme constitue un contracteur optimal pour les CSP dont le graphe de représentation est un arbre. La preuve est présentée dans le paragraphe suivant, qui expose une version plus rapide de ce contracteur.

Hiérarchie des contraintes L'ordre des contraintes à projeter dans l'algorithme de Waltz est aléatoire. En définissant une stratégie d'agencement des contraintes à projeter, il est possible d'améliorer la rapidité de contraction. L'étude effectuée dans [BGGP99] a permis de choisir l'ordre des contraintes primitives h_k à contracter de manière à optimiser la taille du pavé $\mathcal{C}([\mathbf{x}])$. Le contracteur résultant, appelé *contracteur de propagation-rétropropagation* $\mathcal{C}_{\downarrow\uparrow}$ [BGGP99], constitue également un contracteur optimal pour les CSP dont les graphes sont des arbres. En particulier, il conserve les deux propriétés :

Proposition 1 *Si $\mathcal{H} : (\mathbf{x}, f, [\mathbf{x}])$ ne fait intervenir que des occurrences simples de \mathbf{x} , alors $\mathcal{C}_{\downarrow\uparrow}(\mathcal{D}(\mathcal{H}))$ est optimal.*

Proposition 2 *Si $\mathcal{H} : (\mathbf{x}, f, [\mathbf{x}])$ est arc-consistante, alors $[\mathbf{x}]$ est un point fixe pour $\mathcal{C}_{\downarrow\uparrow}$.*

Le principe de ce contracteur repose sur les deux théorèmes suivants [JKDW01].

Théorème 3.2.1 (Propagation) *Soient $\mathcal{H}_x : (\mathbf{x}, h_1, [\mathbf{x}])$ et $\mathcal{H}_y : (\mathbf{y}, h_2, [\mathbf{y}])$ deux CSP qui n'ont aucune variable commune, avec h_1 et h_2 deux contraintes primitives. Soit x_i et y_j deux variables reliées par la contrainte h_3 . Le nouveau CSP $\mathcal{H} : (\mathbf{z}, \mathbf{h}, [\mathbf{z}])$ tel que*

$$\mathbf{z} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, [\mathbf{z}] = [\mathbf{x}] \times [\mathbf{y}], \mathbf{h} = h_1 \wedge h_2 \wedge h_3,$$

est tel que si $[x_i]$ est consistant avec \mathcal{H}_x , $[y_j]$ est consistant avec \mathcal{H}_y , alors

$$\pi_i(\mathcal{H}) = \rho_{m+j}([x_i]) \quad (3.29)$$

$$\text{et } \pi_{m+j}(\mathcal{H}) = \rho_i([y_j]). \quad (3.30)$$

Preuve.

Notons tout d'abord que la procédure $\rho_{m+j}([x_i])$ concerne la projection de la contrainte liant, dans le CSP \mathcal{H} , la variable $z_i = x_i$ et la variable $z_{m+j} = y_j$. Puisque \mathcal{H}_x et \mathcal{H}_y n'ont pas de variables communes, la seule contrainte concernée est h_3 .

Par ailleurs, vue la symétrie du problème, il suffit de prouver (3.29). Puisque $[x_i]$ est consistant avec \mathcal{H}_x , alors $\pi_i(\mathcal{H}_x) = [x_i]$. De même, on a $\pi_j(\mathcal{H}_y) = [y_j]$ puisque $[y_j]$ est consistant avec \mathcal{H}_y .

Soit

$$\begin{aligned}
x_i \in \pi_i(\mathcal{H}) &\iff x_i \in [x_i] \text{ et } \begin{cases} \exists ({}^i\mathbf{x}) \in [{}^i\mathbf{x}] & (x_i, {}^i\mathbf{x}) \in \mathbb{S}(\mathcal{H}_x) \\ \exists \mathbf{y} \in [\mathbf{y}] & \mathbf{y} \in \mathbb{S}(\mathcal{H}_y) \\ & (x_i, y_j) \in \mathbb{S}(h_3) \end{cases} \\
&\iff x_i \in [x_i] \text{ et } \begin{cases} x_i \in \pi_i(\mathcal{H}_x) = [x_i] \\ \exists \mathbf{y} \in \mathbb{S}(\mathcal{H}_y) & (x_i, y_j) \in \mathbb{S}(h_3) \end{cases} \\
&\iff x_i \in [x_i] \text{ et } \exists \mathbf{y} \in \mathbb{S}(\mathcal{H}_y) \mid (x_i, y_j) \in \mathbb{S}(h_3) \\
&\iff x_i \in [x_i] \text{ et } \exists y_j \in \pi_j(\mathcal{H}_y) \mid (x_i, y_j) \in \mathbb{S}(h_3) \\
&\iff x_i \in [x_i] \text{ et } \exists y_j \in [y_j] \mid (x_i, y_j) \in \mathbb{S}(h_3) \\
&\iff x_i \in [x_i] \text{ et } x_i \in \pi_i(h_3) \\
&\iff x_i \in \rho_{m+j}([x_i])
\end{aligned}$$

■

Corollaire 3.2.1 Soit $\mathbf{x} = (\mathbf{p}, q)$ de dimension m et $\mathcal{H} : (\mathbf{p}, f, [\mathbf{p}])$. Soient la contrainte g liant $q \in [q]$ à p_i , et le CSP $\mathcal{H}^{ext} : (\mathbf{x} = (\mathbf{p}, q), f \wedge g, [\mathbf{p}] \times [q])$. Si \mathbf{p} est consistante avec \mathcal{H} , alors $\pi_m(\mathcal{H}^{ext}) = \rho_{1\dots m-1}([q])$.

Théorème 3.2.2 (Retropropagation) Soient $\mathcal{H}_x = (\mathbf{x}, h_1, [\mathbf{x}])$ et $\mathcal{H}_y = (\mathbf{y}, h_2, [\mathbf{y}])$ deux CSP qui n'ont aucune variable commune avec h_1 et h_2 deux contraintes primitives. Soit x_i et y_j deux variables reliées par la contrainte h_3 . Soit le nouveau CSP $\mathcal{H} = (\mathbf{z}, \mathbf{h}, [\mathbf{z}])$ tel que

$$\mathbf{z} = (\mathbf{x}, \mathbf{y}), [\mathbf{z}] = [\mathbf{x}] \times [\mathbf{y}], \mathbf{h} = h_1 \wedge h_2 \wedge h_3.$$

Si $[x_i]$ est consistant avec \mathcal{H} et $[y_j]$ est consistant avec \mathcal{H}_y , alors

$$\pi_{m+j}(\mathcal{H}) = \rho_i([y_j]). \quad (3.31)$$

Preuve.

Puisque $[x_i]$ est consistant avec \mathcal{H} , alors $[x_i]$ est consistant avec \mathcal{H}_x . Les hypothèses du théorème de propagation sont donc satisfaites, et on obtient $\pi_{m+j}(\mathcal{H}) = \rho_i([y_j])$. ■

Afin d'élaborer \mathcal{C}_{\uparrow} , nous allons définir deux routines, FALL et CLIMB. FALL s'appuie sur le théorème de propagation 3.2.1, et parcourt l'arbre des contraintes de ses feuilles jusqu'à sa racine. CLIMB, qui s'appuie sur le théorème de rétropropagation 3.2.2, parcourt l'arbre de sa racine jusqu'à ses feuilles. Nous allons montrer qu'une seule exécution de FALL et CLIMB permet d'atteindre la contraction optimale de \mathcal{H} . La même idée a été développée pour les domaines discrets dans [MR91].

Jusqu'à présent, aucun nœud du graphe n'avait été distingué. Sélectionnons maintenant un nœud quelconque du graphe de représentation \mathcal{T} de \mathcal{H} comme racine. Nous dirons que ce nœud racine de \mathcal{H} , noté $\mathcal{R}(\mathcal{H})$, est le nœud le plus bas de l'arbre. Un nœud v_b est un *ascendant* d'un nœud v_a si le sous-arbre \mathcal{T}_a inclus dans \mathcal{T} de racine v_a contient v_b .

Soit un arbre \mathcal{T} de racine v_i , représentant le CSP \mathcal{H} . En supprimant v_i , on obtient n sous-arbres de \mathcal{T} notés $\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, \dots, \mathcal{T}_{i_n}$. Chacun de ces n sous-arbres \mathcal{T}_{i_k} , $k = 1, \dots, n$, est appelé une *branche* de \mathcal{T} . C'est également un arbre, de racine v_k , qui correspond à un CSP \mathcal{H}_k . Un arbre sans branche est appelé une *feuille*. Si \mathcal{T} est une feuille, nous noterons $\mathcal{T} = v_i$.

Soit \mathcal{H} un CSP de racine v_i et \mathcal{H}_j le sous-CSP de \mathcal{H} de racine v_j associé avec tous les nœuds qui sont supérieurs à v_j . Nous dirons qu'une variable v_j est *consistante avec ses ascendants* dans \mathcal{H} si elle est consistante avec \mathcal{H}_j . Enfin, nous dirons que \mathcal{H} est *consistant avec ses ascendants* si tout nœud de \mathcal{H} est consistant avec ses ascendants.

Algorithme FALL(es : $[\mathbf{x}]$, \mathcal{H})	
1	si \mathcal{H} est une feuille retourner;
2	$i := \text{indice}(\mathcal{R}(\mathcal{H}))$; $n =$ nombre de branches de \mathcal{H}
3	pour chaque branche \mathcal{H}_{j_k} de \mathcal{H} , $k \in \{1, \dots, n\}$
4	FALL(\mathcal{H}_{j_k});
5	$j_k := \text{indice}(\mathcal{R}(\mathcal{H}_{j_k}))$;
6	$v_i := \rho_{j_k}(v_i)$;
7	fin pour

Tableau 3.2: Contraction ascendante d'un CSP

Algorithme Fall Cet algorithme, décrit au tableau 3.2, permet de contracter un CSP \mathcal{H} de racine v_i possédant n branches.

Théorème 3.2.3 FALL contracte tout CSP \mathcal{H} dont le graphe de représentation est un arbre de sorte qu'il soit consistant avec ses ascendants.

Preuve.

Montrons par récurrence que si FALL(\mathcal{H}_{j_k}) rend \mathcal{H}_{j_k} consistant avec ses ascendants, alors \mathcal{H} est consistant avec ses ascendants en sortie de FALL.

Supposons d'abord que toutes les branches \mathcal{H}_{j_i} de \mathcal{H} soient rendues consistantes avec leurs ascendants par FALL. Montrons que FALL(\mathcal{H}) rend alors \mathcal{H} consistant avec ses ascendants.

Soit $\mathcal{H}_i(k)$ le sous-arbre de \mathcal{H} de racine v_i avec k branches \mathcal{H}_{j_i} communes avec \mathcal{H} .

Pour $k = 1$, il découle du théorème de propagation, $\mathcal{H}_i(1)$ est consistante avec ses ascendants après le pas 6.

Supposons $\mathcal{H}_i(k)$ consistante avec ses ascendants. Du théorème de propagation, $\mathcal{H}_i(k+1)$ est consistante avec ses ascendants après $(k+1)$ exécutions du pas 6. Donc $\mathcal{H} = \mathcal{H}_i(n)$ est consistant avec ses ascendants au pas 7. La propriété est vérifiée par récurrence. Comme de plus, si \mathcal{H} est une feuille, \mathcal{H} est consistant avec ces ascendants, la propriété est vérifiée. ■

Elaborons l'algorithme FALL pour l'exemple précédent.

Exemple 3.2.3 (suite) Soit la contrainte à satisfaire définie par (3.27). Les domaines associés à chaque variable sont $[x_1]$, $[x_2]$, $[x_3]$ et $[x_4]$. La première étape consiste à écrire un sous-résolveur $x_4 = \phi(x_1, x_2, x_3)$ n'utilisant que des opérations élémentaires. On écrit l'algorithme FALL à partir de la décomposition déduite de (3.28):

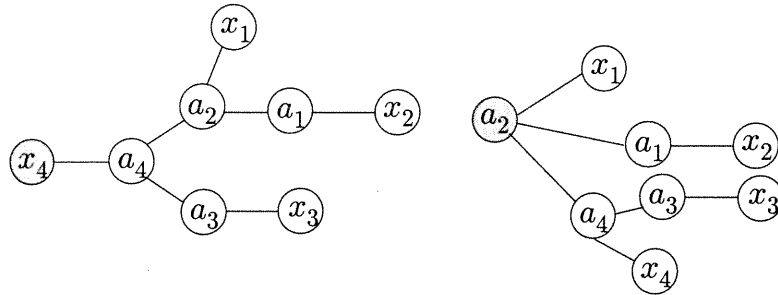
$$\begin{cases} a_1 = \cos(x_2) \\ a_2 = x_1 a_1 \\ a_3 = \exp(x_3) \\ a_4 = a_2 a_3 \\ x_4 = -a_4. \end{cases} \quad (3.32)$$

Algorithme CLIMB(es : $[\mathbf{x}], \mathcal{H}$)	
1	$i := \text{indice}(\mathcal{R}(\mathcal{H}))$;
2	pour chaque branche \mathcal{H}_j de \mathcal{H}
3	$j := \text{indice}(\mathcal{R}(\mathcal{H}_j))$;
4	$v_j := \rho_i(v_j)$;
5	CLIMB(\mathcal{H}_j)
6	fin pour

Tableau 3.3: Contracteur optimal de tout CSP consistant avec ses ascendants

Le graphe associé à cette décomposition est présenté à la figure 3-8. L'extension aux intervalles de (3.32) constitue la mise en œuvre sur ce problème illustratif de l'algorithme FALL($[\mathbf{x}], \mathcal{H}_1$) :

$$\text{FALL}([\mathbf{x}], \mathcal{H}_1) \left\{ \begin{array}{l} [a_1] := [a_1] \cap \cos([x_2]); \\ [a_2] := [a_2] \cap ([x_1] \cdot [a_1]); \\ [a_3] := [a_3] \cap \exp([x_3]); \\ [a_4] := [a_4] \cap ([a_2] \cdot [a_3]); \\ [x_4] := [x_4] \cap (-[a_4]). \end{array} \right.$$

Figure 3-8: Graphes de représentation du CSP $\mathcal{D}(\mathcal{H})$ de l'exemple 3.2.3

Algorithme ClimB Cet algorithme, décrit au tableau 3.3, contracte le CSP \mathcal{H} de racine v_i .

Théorème 3.2.4 CLIMB fournit une contraction optimale de tout CSP \mathcal{H} qui est consistant avec ses ascendants.

Preuve.

A nouveau, nous le montrons par récurrence.

Puisque \mathcal{H} de racine v_1 est consistant avec ses ascendants, alors v_1 est consistante avec ses ascendants.

Supposons que la racine v_i de \mathcal{H}_i soit consistante avec \mathcal{H} , et soit une branche \mathcal{H}_j quelconque de \mathcal{H}_i . Après le pas 4, en vertu du théorème de rétropropagation, v_j est consistante avec \mathcal{H} . En sortie de l'algorithme tous les nœuds sont donc consistants. ■

Elaborons maintenant l'algorithme CLIMB dans le cas de l'exemple précédent.

Exemple 3.2.3 Lisons l'ensemble des ces contraintes primitives de la racine de l'arbre vers ses feuilles (ce qui revient à parcourir de gauche à droite l'arbre de gauche de la figure 3-8), et en imposant que les variables des membres de droite passent à gauche :

$$\begin{cases} a_4 = -x_4; \\ a_2 = \frac{a_4}{a_3}; a_3 = \frac{a_4}{a_2}; \\ x_3 = \ln(a_3); \\ a_1 = \frac{a_2}{x_1}; x_1 = \frac{a_2}{a_1}; \\ x_2 = \cos^{-1}(a_1). \end{cases} \quad (3.33)$$

L'extension aux intervalles de (3.33) constitue la mise en œuvre sur ce problème illustratif de l'algorithme $\text{CLIMB}([\mathbf{x}], \mathcal{H}_1)$.

$$\text{CLIMB}([\mathbf{x}], \mathcal{H}_1) \begin{cases} [a_4] := [a_4] \cap (-[x_4]); \\ [a_2] := [a_2] \cap \frac{[a_4]}{[a_3]}; [a_3] := [a_3] \cap \frac{[a_4]}{[a_2]}; \\ [x_3] := [x_3] \cap \ln([a_3]); \\ [a_1] := [a_1] \cap \frac{[a_2]}{[x_1]}; [x_1] := [x_1] \cap \frac{[a_2]}{[a_1]}; \\ [x_2] := [x_2] \cap \cos^{-1}([a_1]). \end{cases}$$

Il reste maintenant à construire le contracteur $\mathcal{C}_{\downarrow\uparrow}$. Pour cela, il suffit d'insérer la contrainte sur la valeur du domaine, et de concaténer les deux algorithmes.

Exemple 3.2.3 Comme le domaine de $[x_4]$ doit se réduire à $[0, 0]$, on insère l'instruction

$$[x_4] := [x_4] \cap [0, 0],$$

et on concatène les deux algorithmes obtenus

Algorithme $\mathcal{C}_{\downarrow\uparrow}$ (es : $[\mathbf{x}], \mathcal{H}_1$)	
1	$[a_1] := [a_1] \cap \cos([x_2]);$
2	$[a_2] := [a_2] \cap ([x_1] \cdot [a_1]);$
3	$[a_3] := [a_3] \cap \exp([x_3]);$
4	$[a_4] := [a_4] \cap ([a_2] \cdot [a_3]);$
5	$[x_4] := [x_4] \cap (-[a_4]);$
6	$[x_4] := [x_4] \cap [0, 0];$
7	$[a_4] := [a_4] \cap (-[x_4]);$
8	$[a_2] := [a_2] \cap \frac{[a_4]}{[a_3]}; [a_3] := [a_3] \cap \frac{[a_4]}{[a_2]};$
9	$[x_3] := [x_3] \cap \ln([a_3]);$
10	$[a_1] := [a_1] \cap \frac{[a_2]}{[x_1]}; [x_1] := [x_1] \cap \frac{[a_2]}{[a_1]};$
11	$[x_2] := [x_2] \cap \cos^{-1}([a_1]).$

Tableau 3.4: Mise en oeuvre pour le CSP de l'exemple 3.2.3

Si $[x_4]$ est vide au pas 6, alors on a prouvé que $\mathbb{S}(\mathcal{H}_1) \subset \mathbb{S}(\mathcal{H}_2) = \emptyset$. Sinon, $[x_4]$ est remplacé par $[0, 0]$ et la rétropropagation est effectuée de manière à mettre à jour la valeur de toutes les variables.

Théorème 3.2.5 Pour tout CSP dont le graphe de représentation est un arbre, le contracteur $\mathcal{C}_{\downarrow\uparrow}$ dont la structure est présentée à la table 3.5 est optimal.

Preuve.

Soit \mathcal{H} un CSP dont le graphe de représentation est un arbre. Le CSP \mathcal{H}^1 obtenu après le pas 1 est consistant avec ses ascendants d'après le théorème 3.2.3. Il satisfait donc les hypothèses du théorème 3.2.4, et est contracté de façon optimale par CLIMB. ■

Algorithme $\mathcal{C}_{\downarrow\uparrow}$ (es: $[\mathbf{x}], \mathcal{H}$)	
1	FALL($[\mathbf{x}], \mathcal{H}$);
2	CLIMB($[\mathbf{x}], \mathcal{H}$);

Tableau 3.5: Contracteur de propagation-rétropropagation

Si l'optimalité de $\mathcal{C}_{\downarrow\uparrow}$ a pu être montré dans le cas de CSP dont le graphe de représentation est un arbre, elle n'est pas assurée lorsque ce graphe est cyclique, comme nous le montrons dans le paragraphe suivant.

3.2.4.3 Graphe cyclique

Il existe de nombreux CSP pour lesquels l'algorithme de Waltz ne constitue pas un contracteur optimal. L'exemple suivant en est une très simple illustration.

Exemple 3.2.4 *Soit le CSP*

$$\mathcal{H}_1 : (x_2 = x_1 + x_1, \mathbf{x} \in [-5, 5] \times [0, 0]).$$

L'ensemble solution $\mathbb{S}(\mathcal{H}_1)$ est évidemment réduit au singleton $[0, 0]$. Soit $\mathcal{H}_2 = \mathcal{D}(\mathcal{H}_1)$:

$$\mathcal{H}_2 : \left(\begin{cases} a = x_1; \\ x_2 = x_1 + a; \end{cases} , x_1 \in [-5, 5], x_2 \in [0, 0], a \in \mathbb{R} \right).$$

Effectuons les contractions successives :

$$\begin{aligned} [x_2] &= ([x_1] + [a]) \cap [x_2] = ([-5, 5] + \mathbb{R}) \cap [0, 0] = [0, 0]; \\ [a] &= ([x_2] - [x_1]) \cap [a] = ([0, 0] - [-5, 5]) \cap \mathbb{R} = [-5, 5]; \\ [x_1] &= ([x_2] - [a]) \cap [x_1] = ([0, 0] - [-5, 5]) \cap [-5, 5] = [-5, 5]; \end{aligned}$$

Le pavé $[\mathbf{q}] = [-5, 5] \times [0, 0] \times [-5, 5]$ constitue donc un point fixe bien que $[\mathbf{q}] \neq [\mathbb{S}]$.

Ce blocage s'explique par le fait que le graphe de représentation de \mathcal{H}_2 fait apparaître un cycle (voir la figure 3-9 à gauche). C'est en particulier le cas lorsque l'expression de la relation c associée au CSP fait intervenir plusieurs occurrences de variables.

Remarque 3.2.4 *Dans l'exemple 3.2.4, le cycle était déjà présent dans le graphe de représentation de \mathcal{H}_1 (voir la figure 3-9 à droite). L'introduction de la variable intermédiaire a dans le CSP n'a pas fait disparaître ce cycle. Dans certains cas, la décomposition du CSP fait intervenir des variables auxiliaires qui peuvent même aggraver le problème de dépendance.*

Plusieurs solutions ont été proposées pour améliorer la contraction de CSP à graphe cyclique. On peut les classer en trois catégories :

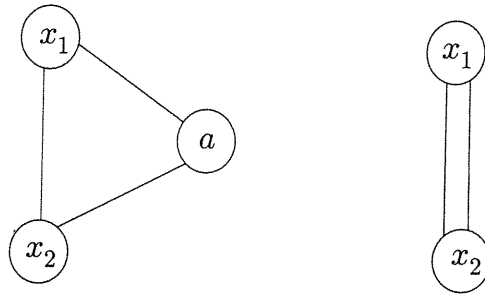


Figure 3-9: Représentation sous forme de graphe du CSP $\mathcal{H}_2 = \mathcal{D}(\mathcal{H}_1)$ à gauche, et du CSP \mathcal{H}_1 à droite. Chacun de ces graphes contient un cycle

- la vectorisation² consiste à regrouper dans un vecteur toutes les variables d'un cycle [JKBW01],
- d'autres techniques de consistances locales (box-consistance) permettent de tester d'autres types de contracteurs [Gra98]
- la bisection permet de débloquent l'algorithme très simplement, mais fait réapparaître le problème de la complexité.

Nous comparerons les résultats de ces techniques sur l'exemple 3.2.4 dans le paragraphe suivant.

Vectorisation Le principe de cette technique est de construire un CSP dont le graphe de représentation est un arbre, qui est équivalent au CSP initial. Pour cela, on choisit de ne plus manipuler des variables scalaires, mais de manipuler des vecteurs de variables. Les variables scalaires intervenant dans un cycle sont regroupées de manière à éliminer ce cycle.

Exemple 3.2.4 (suite) Posons

$$\mathbf{a} = \begin{pmatrix} x_1 \\ a \end{pmatrix}, \mathbb{A} = \{(x_1, x_1), x_1 \in [-5, 5]\} \text{ et } \mathbf{b} = x_2, \mathbb{B} = [0, 0],$$

et \mathcal{H}_3

$$\mathcal{H}_3 : \left(\beta = \mathbf{f}(\alpha), \alpha \in [-5, 5]^{\times 2}, \beta \in [0, 0] \right).$$

Alors $\mathcal{H}_3 \equiv \mathcal{H}_2 \equiv \mathcal{H}_1$, et \mathcal{H}_3 est décrit par un arbre composé de 2 nœuds et un arc. Le contracteur \mathcal{C}_{\uparrow} est optimal pour \mathcal{H}_3 .

Malheureusement, dans de nombreux problèmes de satisfaction de contraintes, les cycles sont intriqués de telle manière que le nombre de nœuds final est inférieur au nombre de variables pertinentes du problème. La vectorisation du problème permet alors d'obtenir un arbre simplifié, mais la complexité de l'algorithme est reportée dans le vecteur de variables.

² clustering dans la langue de Byron

Exemple 3.2.5 Soit le CSP :

$$\mathcal{H} : \left(\frac{\cosh x_1 \cos x_1 + \sinh x_1 \sin x_1}{x_1^2} = x_2, x_1 \in \mathbb{R}, x_2 \in \mathbb{R} \right).$$

Le graphe de représentation du CSP $\mathcal{D}(\mathcal{H})$:

$$\mathcal{D}(\mathcal{H}) : \left(\begin{array}{l} a_1 = \cosh(x_1) \\ a_2 = \cos(x_1) \\ b_1 = a_1 a_2 \\ a_3 = \sinh x_1 \\ a_4 = \sin x_1 \\ b_2 = a_3 a_4 \\ c_1 = b_1 + b_2 \\ a_5 = x_1^2 \\ x_2 = \frac{c_1}{a_5} \end{array} \right)$$

est présenté à la figure 3-10. L'arbre obtenu par vectorisation est décrit par le graphe de la

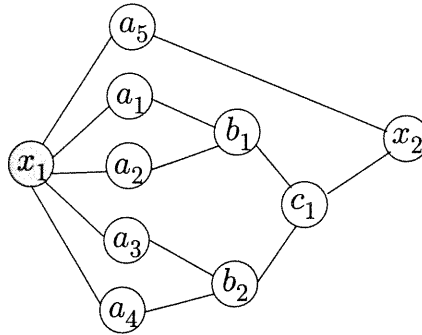


Figure 3-10: Graphe de représentation du CSP \mathcal{H} de l'exemple 3.2.5.

figure 3-11.

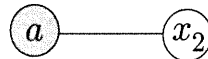


Figure 3-11: Vectorisation du CSP \mathcal{H} de l'exemple 3.2.5.

Box-consistance Comme les calculs requis lors de la projection d'une contrainte primitive h sont garantis et minimaux, la consistance d'arc est théoriquement bien testée sur chaque point du domaine de h . Aussi, lorsque le graphe de représentation du CSP est un arbre, le contracteur $\mathcal{C}_{\downarrow\uparrow}$ propage efficacement chacun des domaines contractés jusqu'à obtenir un CSP arc-consistant.

Lorsque les contraintes contiennent des variables très multi-occurentes, la décomposition en contraintes primitives peut accroître le problème de dépendance. Ainsi le CSP décomposé $\mathcal{D}(\mathcal{H})$ peut être arc-consistant même si \mathcal{H} ne l'est pas, comme le montre l'exemple suivant.

Exemple 3.2.6 Reprenons l'exemple 3.2.4. La contrainte

$$\mathcal{H}_1 : (x_1 + x_1 = x_2, \mathbf{x} \in [-5, 5] \times [0, 0])$$

n'est pas arc-consistante car elle n'est pas vérifiée aux bornes de $[x_1]$. Par contre, le CSP \mathcal{H}_2 issu de la décomposition du CSP \mathcal{H}_1

$$\mathcal{H}_2 : (x_1 + a = x_2, x_1 = a, \mathbf{x} \in [-5, 5] \times [0, 0], a \in [-5, 5])$$

est arc-consistant, ce qui explique pourquoi il n'est pas contracté par \mathcal{C}_{\uparrow} .

La *box-consistance* est une alternative à la décomposition d'un CSP en contraintes primitives [BGGP99]. Cette technique consiste à traiter les contraintes de façon globale sans les décomposer, en testant à nouveau les bornes du domaine. Elle permet ainsi d'implémenter directement des extensions sur les intervalles des contraintes réelles. En négligeant les erreurs d'arrondi, si

$$\forall p \in \{\underline{x}_i, \bar{x}_i\}, ([x_1], \dots, [x_{i-1}], p, [x_{i+1}], \dots, [x_m]) \in \mathbb{S}$$

alors la contrainte \mathcal{H}_j est dite *box-consistante* par rapport à x_i ³. Cette méthode nécessite donc une évaluation intervalles des contraintes (différentes fonctions d'inclusion peuvent être envisagées [HMK97]), et une recherche locale pour chaque variable x_i dans le domaine $[x_i]$, en utilisant par exemple, la méthode de Newton. Lorsque le nombre m de variables est important, cette méthode devient relativement coûteuse.

Les meilleurs résultats sont alors obtenus en combinant ces approches, ce qui requiert l'utilisation de logiciels dédiés aux contraintes (le logiciel REALPAVER [Webk] de L. Granvilliers a été rendu disponible en avril 2002).

Bissection Deux raisons nous ont motivé à utiliser la bissection en complément des contracteurs, bien que cet outil peut augmenter notablement la charge mémoire nécessaire. D'une part, nous avons montré au chapitre précédent que l'outil de bissection permettait de contrôler la précision de la description par encadrement, au prix bien sûr, d'une augmentation de la complexité de l'algorithme, en remplaçant l'étude de $\mathbb{S} \cap [\mathbf{x}]$ par celle de $(\mathbb{S} \cap [\mathbf{x}_1]) \cup (\mathbb{S} \cap [\mathbf{x}_2])$ où $[\mathbf{x}_1]$ et $[\mathbf{x}_2]$ sont les fils de $[\mathbf{x}]$. D'autre part, même si le contracteur \mathcal{C}_{\uparrow} est optimal, il fournit au mieux le pavé enveloppe de l'ensemble solution. Aussi, lorsque celui-ci comporte plusieurs composantes connexes, le pessimisme introduit par le phénomène d'enveloppement est à nouveau important. Aucune des méthodes précédemment citées ne permet donc seule, de décrire précisément un ensemble solution comportant plusieurs composantes connexes, puisque seule l'enveloppe $[\mathbb{S}]$ de l'ensemble peut être espérée.

Exemple 3.2.7 Soit le CSP \mathcal{H} défini par

$$\mathcal{H} : (x_1^2 + x_2^2 = 5, \mathbf{x} \in [-5, 5]^{x_2}),$$

illustré sur la figure 3-12. L'intervalle $[p_2]$ obtenu sans bissection n'est pas arc-consistant car il contient deux composantes connexes. Le pavé $[\mathbf{q}]$ est arc-consistant.

Autorisons donc l'algorithme à effectuer des bisections lorsqu'il atteint des points fixes. Sur le problème de l'exemple 3.2.4, une seule bissection est nécessaire, comme le montre la figure 3-13.

³ se reporter à l'annexe pour la mise en œuvre

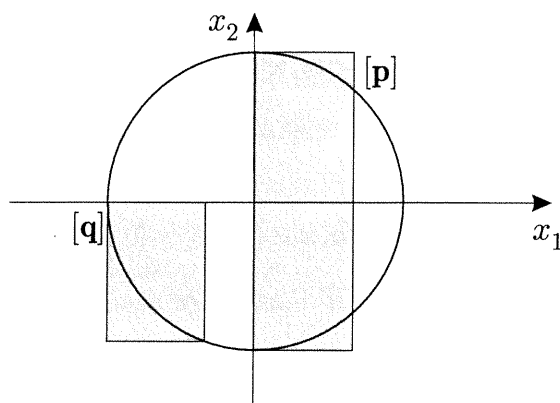


Figure 3-12: Représentation du CSP défini dans l'exemple 3.2.7

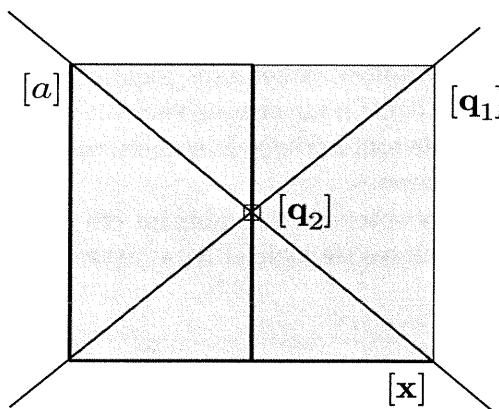


Figure 3-13: Utilisation de la bisection dans le cas du CSP \mathcal{H} défini dans l'exemple 3.2.4. $[q_1]$ est un point fixe pour $\mathcal{C}_{\downarrow\uparrow}$. Grâce à une seule bisection, on obtient deux pavés $[q_1^1]$ et $[q_1^2]$ qui ne sont pas des points fixes. L'union $[q_2] = \mathcal{C}_{\downarrow\uparrow}([q_1^1]) \cup \mathcal{C}_{\downarrow\uparrow}([q_1^2])$ est une approximation extérieure du pavé enveloppe de $\mathbb{S}(\mathcal{H})$

L'algorithme SIVIAP, qui résulte de la combinaison de contracteurs et de techniques de bisections, s'inscrit très naturellement comme prolongement des algorithmes de la famille de SIVIA. Ce résolveur est présenté au paragraphe 3.3.1.

Jusqu'à présent, nous nous sommes restreints à ne considérer qu'une seule contrainte. Dans le paragraphe suivant, nous montrons comment étendre simplement ces méthodes aux cas de plusieurs contraintes.

3.2.5 Systèmes de contraintes

Ce paragraphe vise deux objectifs. Dans un premier temps, nous montrons comment élaborer à partir d'un système de contraintes, un CSP équivalent composé d'une seule contrainte. Dans un second temps, nous montrerons sur quelques exemples les limitations du contracteur $\mathcal{C}_{\downarrow\uparrow}$ dans le cas d'un système de contraintes.

3.2.5.1 Obtention d'un CSP équivalent

Le traitement des systèmes de contraintes est plus compliqué. Afin de pouvoir utiliser le contracteur de propagation-rétropropagation, nous proposons de définir un CSP composé d'une seule contrainte, et équivalent au système de contraintes initial.

Nous appelons conjonction de contraintes le CSP \mathcal{H}_\cap défini par

$$\mathcal{H}_\cap : \left(\left(\begin{array}{l} \mathcal{H}_1 : f_1(\mathbf{x}) \leq 0 \\ \wedge \dots \\ \wedge \mathcal{H}_i : f_i(\mathbf{x}) \leq 0 \\ \wedge \dots \\ \wedge \mathcal{H}_n : f_n(\mathbf{x}) \leq 0 \end{array} \right) , \mathbf{x} \in [\mathbf{x}] \right), \quad (3.34)$$

et disjonction de contraintes le CSP du type

$$\mathcal{H}_\cup : \left(\left(\begin{array}{l} \mathcal{H}_1 : f_1(\mathbf{x}) \leq 0 \\ \vee \dots \\ \vee \mathcal{H}_i : f_i(\mathbf{x}) \leq 0 \\ \vee \dots \\ \vee \mathcal{H}_n : f_n(\mathbf{x}) \leq 0 \end{array} \right) , \mathbf{x} \in [\mathbf{x}] \right). \quad (3.35)$$

Tous les CSP \mathcal{H} que nous serons amenés à considérer sont des combinaisons de conjonctions et de disjonctions de contraintes. Nous nous restreignons donc à étudier \mathcal{H}_\cap et \mathcal{H}_\cup .

Conjonction de contraintes Comme l'ensemble solution de \mathcal{H}_\cap est donné par

$$\mathbb{S}(\mathcal{H}_\cap) = \bigcap_{j \in \{1, \dots, n\}} \mathbb{S}_j,$$

ce CSP correspond à une décomposition en contraintes (non-nécessairement primitives) \mathcal{H}_i . Il est possible d'utiliser la propagation de contraintes en projetant successivement chaque contrainte \mathcal{H}_i , jusqu'à ce que la contraction ne soit plus efficace.

Une autre approche consiste à remarquer que \mathcal{H}_\cap est équivalent au CSP \mathcal{H}'_\cap

$$\mathcal{H}'_\cap : (\text{MAX}_{i \in \{1, \dots, n\}} f_i(\mathbf{x}) \leq 0, \mathbf{x} \in [\mathbf{x}]),$$

dont la version décomposée $\mathcal{D}(\mathcal{H}'_\cap)$ est donnée (dans le cas où $n = 3$) par :

$$\mathcal{D}(\mathcal{H}'_\cap) : (\text{MAX}(\text{MAX}(f_1(\mathbf{x}), f_2(\mathbf{x})), f_3(\mathbf{x})) \leq 0, \mathbf{x} \in [\mathbf{x}]).$$

La contrainte MAX symbolise le max de deux intervalles :

$$\begin{aligned} [z] &= \max([x], [y]) \\ &= [\max(\underline{x}, \underline{y}), \max(\bar{x}, \bar{y})]. \end{aligned}$$

La projection de la contrainte MAX est mise en œuvre à l'aide de l'algorithme PMAX décrit au tableau 3.6.

Algorithme P _{MAX} (es : $[x_1], [x_2], [x_3]$)	
1	$[x_3] := [x_3] \cap \max([x_1], [x_2]);$
2	si $([x_3] \cap [x_1] = \emptyset)$ $[x_2] := [x_2] \cap [x_3];$
3	si $([x_3] \cap [x_2] = \emptyset)$ $[x_1] := [x_1] \cap [x_3];$
4	$[x_1] := [x_1] \cap [-\infty, \bar{x}_3];$
5	$[x_2] := [x_2] \cap [-\infty, \bar{x}_3];$

Tableau 3.6: Mise en œuvre de la projection de la contrainte MAX

Disjonction de contraintes Dans ce cas, l'ensemble solution est donné par

$$\mathbb{S} = \bigcup_{j \in \{1, \dots, n\}} \mathbb{S}_j.$$

De nombreux travaux ont été entrepris pour manipuler les systèmes de contraintes disjonctifs. Citons en particulier [Webi]. Nous proposons ici de remplacer tout système de contraintes sous la forme (3.35) par le CSP équivalent \mathcal{H}'_{\cup}

$$\mathcal{H}'_{\cup} : (\text{MIN}_{i \in \{1, \dots, n\}} f_i(\mathbf{x}) \leq 0, \mathbf{x} \in [\mathbf{x}]),$$

où la contrainte MIN symbolise le min de deux intervalles :

$$\begin{aligned} [z] &= \min([x], [y]) \\ &= [\min(\underline{x}, \underline{y}), \min(\bar{x}, \bar{y})]. \end{aligned}$$

La projection de la contrainte MIN est mise en œuvre à l'aide de l'algorithme P_{MIN} décrit au tableau 3.7.

Algorithme P _{MIN} (es : $[x_1], [x_2], [x_3]$)	
1	$[x_3] := [x_3] \cap \min([x_1], [x_2]);$
2	si $([x_3] \cap [x_1] = \emptyset)$ $[x_2] := [x_2] \cap [x_3];$
3	si $([x_3] \cap [x_2] = \emptyset)$ $[x_1] := [x_1] \cap [x_3];$
4	$[x_1] := [x_1] \cap [x_3, \infty];$
5	$[x_2] := [x_2] \cap [x_3, \infty];$

Tableau 3.7: Mise en œuvre de la projection de la contrainte MIN

Dans le cas d'un système hybride de contraintes, nous utiliserons alors simultanément les contraintes MIN et MAX, comme le montre l'exemple suivant.

Exemple 3.2.8 Soit le CSP

$$\mathcal{H} : \left(\begin{cases} x_1 + x_2 \leq 0 \\ \wedge x_1 x_2 \leq 0 \\ \vee -x_1 x_2 - 1 \leq 0 \end{cases}, \mathbf{x} \in [\mathbf{x}] \right).$$

A l'aide des contraintes MAX et MIN, nous construisons le CSP \mathcal{H}^1 équivalent à \mathcal{H} et défini par

$$\mathcal{H}^1 : \left(\begin{cases} \text{MAX}(x_1 + x_2, x_1 x_2) \leq 0 \\ \vee -x_1 x_2 - 1 \leq 0 \end{cases}, \mathbf{x} \in [\mathbf{x}] \right),$$

lui-même équivalent au CSP

$$\mathcal{H}^2 : (\text{MIN}(-x_1x_2 - 1, \text{MAX}(x_1 + x_2, x_1x_2)) \leq 0, \mathbf{x} \in [\mathbf{x}]).$$

Cette réécriture des CSP sous la forme d'une unique contrainte nous permet de manipuler simplement les quantificateurs existentiel et universel. Cependant, les problèmes liés à l'application de procédures de projection locale persistent.

3.2.5.2 Limitations de $\mathcal{C}_{\downarrow\uparrow}$

Comme les procédures de projection utilisées sont locales, à nouveau le pavé résultant $\mathcal{C}_{\downarrow\uparrow}([\mathbf{x}])$ peut ne pas être globalement consistant. Les exemples suivants illustrent le cas où un pavé $[\mathbf{q}]$ localement consistant avec la contrainte \mathcal{H}_j peut constituer un point fixe pour $\mathcal{C}_{\downarrow\uparrow}$. Le premier exemple illustre le cas de contraintes égalités.

Exemple 3.2.9 Soit le CSP

$$\mathcal{H} : \left(\begin{cases} \mathcal{H}_1 : x_2 - x_1 + 1 = 0 \\ \mathcal{H}_2 : x_2 - \frac{6}{x_1} = 0 \end{cases}, \mathbf{x} \in [-5, 5]^{\times 2} \right). \quad (3.36)$$

dont l'ensemble solution \mathbb{S} est une paire

$$\mathbb{S} = \left\{ \begin{pmatrix} -2 \\ -3 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right\}.$$

Les deux contraintes \mathcal{H}_1 et \mathcal{H}_2 , ainsi que le résultat du contracteur $\mathcal{C}_{\downarrow\uparrow}$ appliqué au pavé $[\mathbf{x}] = [-5, 5]^{\times 2}$, sont présentés à la figure 3-14. Le pavé $\mathcal{C}_{\downarrow\uparrow}([\mathbf{x}])$ ne correspond pas au pavé enveloppe de \mathbb{S} , mais le contracteur s'arrête car ce pavé est localement consistant avec \mathcal{H}_1 .

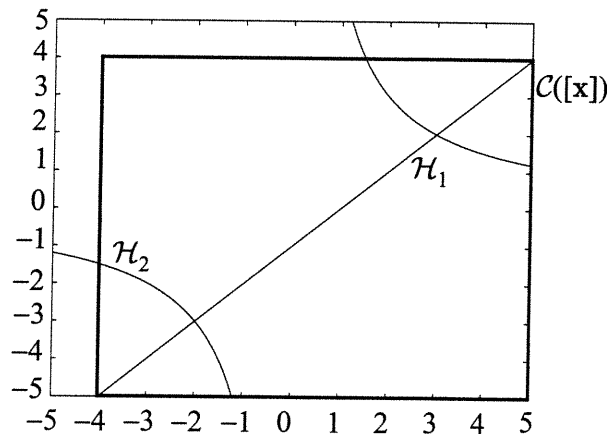


Figure 3-14: Image du pavé $[\mathbf{x}]$ par le contracteur $\mathcal{C}_{\downarrow\uparrow}$ du CSP défini dans l'exemple 3.2.9

L'exemple précédent traite le cas d'un système d'égalités. Le même problème est également rencontré dans le cas d'inégalités, comme le montre l'exemple suivant.

Exemple 3.2.10 On considère le CSP

$$\mathcal{H} : \left(\begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \mathbf{x} = \begin{pmatrix} [-1, 1] \\ [-1, 1] \end{pmatrix} ; \mathbf{x} \in [-5, 5] \times 2 \right),$$

dont l'ensemble solution \mathbb{S} est l'image par la rotation d'angle θ du pavé $[-1, 1] \times [-1, 1]$, représenté sur la figure 3-15 pour $\theta = \frac{\pi}{6}$. Le pavé $[\mathbf{q}] = \mathcal{C}_{\downarrow \uparrow}([\mathbf{x}])$ est un point fixe, puisque ses bornes satisfont la contrainte \mathcal{H}_1 .

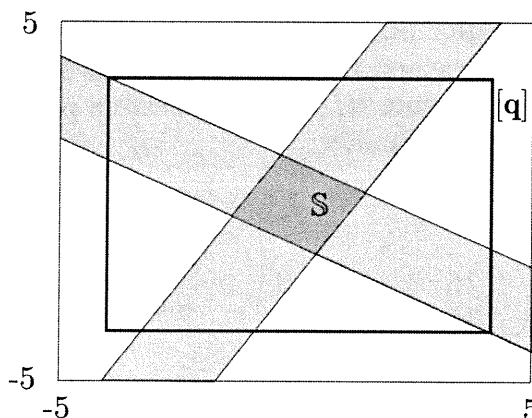


Figure 3-15: Résultat de la contraction du CSP de l'exemple 3.2.10 à l'aide de $\mathcal{C}_{\downarrow \uparrow}$. Le pavé $[\mathbf{q}]$ obtenu est un point fixe, qui n'est pas égal à $[\mathbb{S}]$

Les outils présentés dans ce paragraphe contractent des pavés. Dans le paragraphe suivant, nous montrons comment insérer ces outils dans les algorithmes ensemblistes afin de décrire, à l'aide de contractions, un ensemble \mathbb{X} quelconque. Ceci nous mènera à proposer de nouveaux algorithmes.

3.3 Nouveaux algorithmes ensemblistes

Dans cette partie, nous présentons quelques algorithmes capables de caractériser l'ensemble solution \mathbb{S} d'un CSP. En particulier, ils permettent de calculer des approximations intérieure et extérieure

- de l'ensemble \mathbb{S} de manière plus efficace que ne le permet SIVIA,
- de la projection de \mathbb{S} sur un domaine \mathbb{P} de dimension inférieure à m ,
- de la forme et du volume d'un ensemble \mathbb{S} en contrôlant la surestimation due au phénomène d'enveloppement.

Nous verrons de plus que ces algorithmes sont faciles à comprendre et à mettre en œuvre, et souvent efficaces même en dimension élevée.

Jusqu'à présent, nous avons décrit l'ensemble \mathbb{S} à l'aide d'algorithmes *branch-and-bound*, utilisant des techniques de partitionnement et un test d'inclusion. Comme le principal désavantage des contracteurs peut être compensé par l'utilisation de techniques de partitionnement, nous allons introduire les contracteurs dans ces algorithmes.

3.3.1 Caractérisation d'un ensemble

L'algorithme SIVIAP (SIVIA avec Propagation de contraintes sur les intervalles) présenté à la table 3.8, a la même structure que SIVIA, mais met en œuvre un contracteur. Il a été introduit dans [JKBW01] et [JBKW01]. Il génère deux unions de pavés $\underline{\mathbb{S}}$ et $\overline{\mathbb{S}}$ de sorte que $\underline{\mathbb{S}} \subset \mathbb{S} \subset (\underline{\mathbb{S}} \cup \Delta\mathbb{S})$. Le critère d'arrêt du partitionnement est la taille maximale ε des pavés à bissecter. Parmi les contracteurs disponibles, nous avons choisi $\mathcal{C}_{\downarrow\uparrow}$ pour sa simplicité et ses performances sur les pavés de taille importante. Cet algorithme est particulièrement performant lorsque le graphe de représentation de \mathcal{H} est un arbre et que l'ensemble solution \mathbb{S} est de volume faible, comme nous le verrons sur quelques exemples.

Algorithme SIVIAP (e: $\mathcal{C}, [\mathbf{x}], \varepsilon, s: (\underline{\mathbb{S}}, \overline{\mathbb{S}})$)	
1	$[\mathbf{x}] := \mathcal{C}([\mathbf{x}]);$
2	si $[\mathbf{x}] = \emptyset$, retourner $(\emptyset, \emptyset);$
3	sinon si $t([\mathbf{x}]) = 1$, retourner $([\mathbf{x}], [\mathbf{x}]);$
4	sinon si $w([\mathbf{x}]) < \varepsilon$, retourner $(\emptyset, [\mathbf{x}]);$
5	$(\underline{\mathbb{S}}, \overline{\mathbb{S}}) := \text{SIVIAP}(\mathcal{C}, L[\mathbf{x}], \varepsilon) \cup \text{SIVIAP}(\mathcal{C}, R[\mathbf{x}], \varepsilon);$

Tableau 3.8: Mise en œuvre de la propagation de contraintes sur les intervalles

Dans le paragraphe suivant, nous comparons les performances de SIVIA et de SIVIAP sur quelques exemples issus d'applications que nous traiterons par la suite.

3.3.1.1 Sivia-SiviaP

L'algorithme SIVIA utilise un test d'inclusion $t(\cdot)$ tandis que l'algorithme SIVIAP utilise un contracteur \mathcal{C} . Dans ce qui suit, le contracteur utilisé est $\mathcal{C}_{\downarrow\uparrow}$, mais rappelons que d'autres contracteurs pourraient être mis en œuvre afin d'améliorer encore les performances de SIVIAP.

Pour justifier l'utilité des contracteurs, nous considérons dans un premier temps le cas de CSP où il est possible d'obtenir des fonctions d'inclusion optimales. Nous considérerons dans ce cas les performances de SIVIAP et de SIVIA dans leurs domaines d'application optimaux.

Ensemble solution non-connexe Reprenons dans un premier temps le CSP de l'exemple 3.2.9, et fixons $\varepsilon = 10^{-3}$. SIVIAP caractérise la solution en 2 *ms* car il utilise de nombreuses contractions mais seulement 13 bisections (voir la figure 3-16 à droite). SIVIA utilise pour le même problème 208 bisections et fournit le résultat représenté à la figure 3-16 à gauche en 20 *ms*. Le rapport des volumes des couches d'incertitudes $\text{Vol}(\Delta\mathbb{S})$ obtenus dans chaque cas est de 0.02 en faveur de SIVIAP.

Ensembles solutions de volume non nul Accroissons maintenant la taille de l'ensemble solution du CSP de l'exemple 3.2.9. Pour cela, nous considérons le CSP

$$\mathcal{H}' : \left(\begin{cases} x_2 - x_1 + 1 = [-0.5, 0.5] \\ x_2 - \frac{6}{x_1} = [-0.5, 0.5] \end{cases} \quad \mathbf{x} \in [-10, 10]^{\times 2} \right)$$

Pour $\varepsilon = 10^{-3}$, SIVIAP fournit un encadrement de $\mathbb{S}(\mathcal{H}')$ en 2.4 *s* avec une couche d'incertitude $\text{Vol}(\Delta\mathbb{S}) = 3.10^{-3}$. Pour obtenir la même valeur de la couche d'incertitude avec SIVIA, il faut imposer $\varepsilon = 2.10^{-4}$, ce qui fait monter le temps de calcul à 10.7 *s*. Même en faible dimension, l'avantage de SIVIAP se fait donc déjà ressentir.

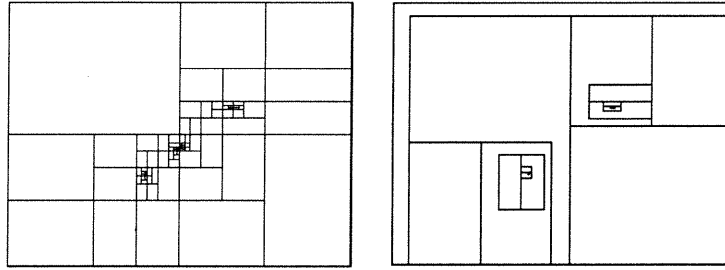


Figure 3-16: Résultat de SIVIA (à gauche) et de SIVIAP (à droite) pour le CSP de l'exemple 3.2.9. Les bisections et contractions sont représentées par des cadres noirs

Exemple 3.3.0 Reprenons le CSP défini dans l'exemple 3.2.10. Le nombre de bisections effectuées par SIVIAP est égal au nombre de bisections effectuées par SIVIA, puisqu'elles sont dues au phénomène d'enveloppement que n'améliore pas le contracteur.

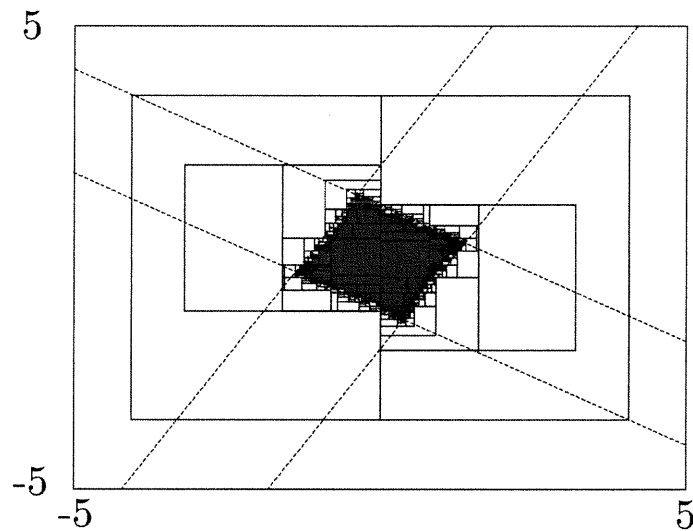


Figure 3-17: Sous-pavage obtenu par SIVIAP en 0.1s pour $\varepsilon = 0.1$ pour le CSP de l'exemple 3.2.10

Etudions maintenant les performances de SIVIAP selon le volume de l'ensemble solution et le critère d'arrêt ε .

Influence de Vol(S) On considère le CSP

$$\mathcal{H} : (x_1 \exp(-x_2 t_i) \in [y_i], \mathbf{x} \in [10^{-6}, 10^{-6}] \times [0, 1], i = 1, \dots, 56) \quad (3.37)$$

où

$$t_i = -2 + \frac{15}{56}i,$$

$$\text{et } [y_i] = 100 \exp(-0.4t_i) + b + e[-1, 1],$$

et b est une grandeur aléatoire prise dans $e[-1, 1]$. La figure 3-19 illustre les performances de SIVIA et SIVIAP sur cet exemple pour différentes valeurs de e .

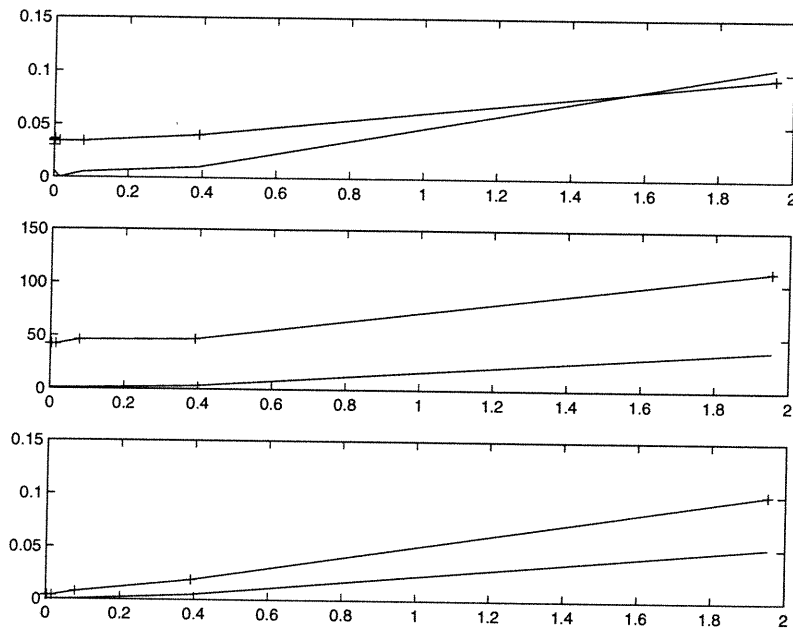


Figure 3-18: Comparaison des comportements de SIVIA (+) et de SIVIAP (-) en fonction du volume de $\mathbb{S}(\mathcal{H})$ défini par (3.37). En haut : temps de calcul t_c , Au milieu nombre de bisections, en bas, $Vol(\Delta\mathbb{S})$ en fonction de e

Influence de ε On fixe maintenant $e = 0.01$. La figure 3-19 illustre les performances de SIVIA et SIVIAP sur cet exemple pour différentes valeurs du coefficient ε .

L'intérêt de l'algorithme pour des pavés de taille relativement élevée est mis en évidence. Par contre, comme l'on pouvait s'y attendre, le contracteur $\mathcal{C}_{\downarrow\uparrow}$ n'apporte que peu d'avantages pour des pavés de taille très faible.

3.3.1.2 Améliorations possibles

Collaborations de solveurs Notre stratégie va consister à faire collaborer les outils de l'analyse par intervalles (tests d'inclusion, contracteurs) avec les contracteurs issus de la propagation de contraintes, puisque leurs performances sont complémentaires. La collaboration de solveurs est un domaine de l'informatique en plein essor. Dans ce document, nous ferons généralement collaborer le contracteur $\mathcal{C}_{\downarrow\uparrow}$ élaboré sur une expression du système faisant intervenir le moins d'occurrences possibles des variables x_i , avec différentes fonctions d'inclusion menant à différents tests d'inclusion.

Recherche de points admissibles Il est également possible d'améliorer les performances des contracteurs en utilisant des algorithmes de recherche intérieure ponctuel [JKDW01]. Supposons qu'on connaisse un ensemble de points admissibles, obtenus par tirage aléatoire ou optimisation locale. Soit $[\mathbf{r}]$ le pavé enveloppe de cet ensemble de points. Nécessairement $[\mathbf{r}]$ est un point fixe pour tout contracteur \mathcal{C} . Soit $\neg[\mathbf{r}]$ l'ensemble complémentaire de $[\mathbf{r}]$ dans le domaine de

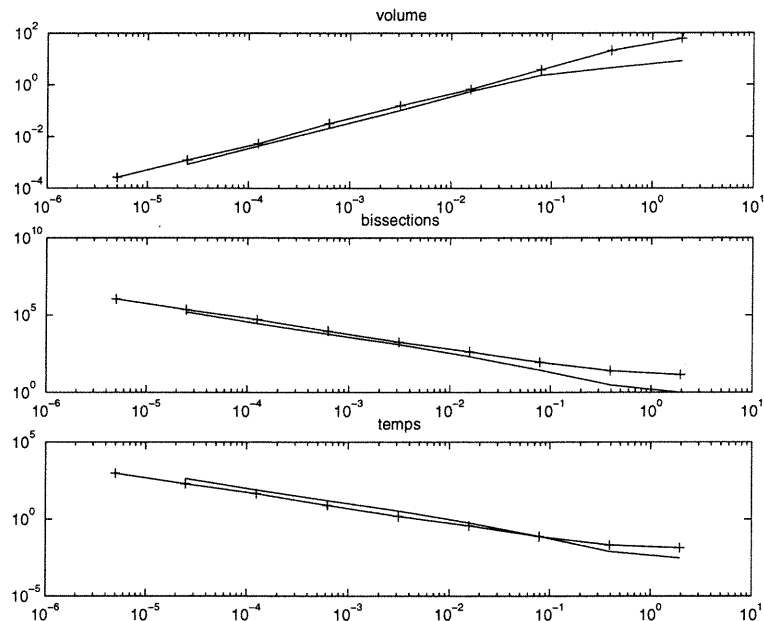


Figure 3-19: Evaluation des comportements de SIVIA(+/-) et SIVIAP selon la valeur de ε . Ces comportements sont identiques lorsque ε est petit, mais $\mathcal{C}_{\downarrow\uparrow}$ améliore les performances de SIVIAP pour des valeurs de ε plus grandes

recherche \mathbb{X} . Comme

$$\mathbb{X} = [\mathbf{r}] \cup \neg[\mathbf{r}],$$

on a

$$\begin{aligned} \mathcal{C}(\mathbb{X}) &= \mathcal{C}([\mathbf{r}] \cup \neg[\mathbf{r}]) \\ &= \mathcal{C}([\mathbf{r}]) \sqcup \mathcal{C}(\neg[\mathbf{r}]) \\ &= [\mathbf{r}] \sqcup \mathcal{C}(\neg[\mathbf{r}]). \end{aligned}$$

Il est donc possible de contracter \mathbb{X} en ne cherchant à contracter que chacun des pavés constitutifs de l'ensemble $\neg[\mathbf{r}]$. Cette approche a été développée dans l'algorithme HULL [Jau00a], qui utilise la technique d'épluchage [JKDW01]

Les contracteurs que nous avons présentés jusqu'à présent permettent d'améliorer l'approximation extérieure $\overline{\mathbb{S}}$ de l'ensemble \mathbb{S} . Par la suite, nous serons amenés à améliorer son approximation intérieure $\underline{\mathbb{S}}$. Pour cela, nous utiliserons également les algorithmes de recherche locale présentés ci-dessous.

Recherche intérieure Nous présentons deux techniques de recherche intérieure.

Cross L'algorithme CROSS [Jau00a] est un algorithme de recherche locale développé par L. Jaulin. Il consiste à tirer un faisceau à partir d'un point de $\underline{\mathbb{S}}$ dans une direction. En contractant ce faisceau, \mathcal{C} permet alors d'en délimiter la longueur. A partir de la nouvelle extrémité du faisceau, dont on vérifie qu'elle est élément de \mathbb{S} , une nouvelle direction est privilégiée, et l'opération itérée.

Contracteur complémentaire Soit $\neg\mathbb{S}$ l'ensemble complémentaire de \mathbb{S} dans \mathbb{X} , et \mathcal{C}_- un contracteur associé à $\neg\mathbb{S}$. Soit $[y] \in \mathbb{X}$. Si $\mathcal{C}_-([y]) = \emptyset$, alors $[y] \cap \neg\mathbb{S} = \emptyset$ d'après (3.12). Puisque $\mathbb{S} \cap \neg\mathbb{S} = \emptyset$ et $\mathbb{S} \cup \neg\mathbb{S} = \mathbb{X}$, $[y] \in \mathbb{S}$. Il est donc possible de tester l'appartenance d'un pavé à un ensemble \mathbb{S} à l'aide d'un contracteur associé à l'ensemble complémentaire de \mathbb{S} . La Figure 3-20 illustre l'action d'un contracteur \mathcal{C} pour \mathbb{S} agissant sur $[x]$ et un contracteur \mathcal{C}_- pour son ensemble complémentaire $\neg\mathbb{S}$ agissant sur $[y]$.

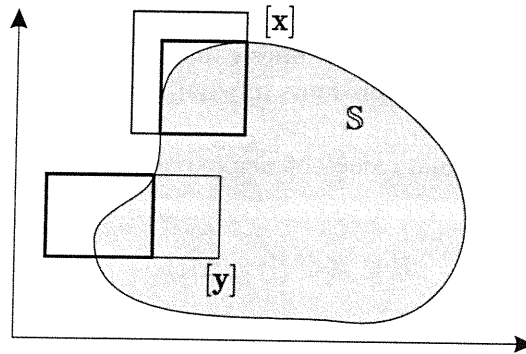


Figure 3-20: Contraction de $[x]$ par \mathcal{C} et de $[y]$ par \mathcal{C}_-

Par exemple le complémentaire $\neg\mathbb{S}$ de l'ensemble \mathbb{S} défini par

$$\mathbb{S} = \{x \in \mathbb{R}^m \mid x_1 \exp x_2 + x_2 \leq 1 \wedge \sin(x_1 + x_2) \leq 0\},$$

est donné par

$$\neg\mathbb{S} = \{x \in \mathbb{R}^m \mid x_1 \exp x_2 + x_2 > 1 \vee \sin(x_1 + x_2) > 0\} \quad (3.38)$$

$$= \{x \in \mathbb{R}^m \mid \max(x_1 \exp x_2 + x_2 - 1, \sin(x_1 + x_2)) > 0\}. \quad (3.39)$$

Un contracteur pour $\neg\mathbb{S}$ est donné par

Algorithme \mathcal{C}_- (es : $[x]$)	
1	$[a_1] := [a_2] := [a_3] := [a_4] := [a_5] := [a_6] := -\infty, \infty[; [a_7] :=]0, \infty[;$
2	tant que la contraction est significative
3	PEXP($[x_2], [a_1]$);
4	PMULT($[x_1], [a_1], [a_2]$);
5	PADD($[a_2], [-1, -1], [a_3]$);
6	PADD($[a_3], [x_2], [a_4]$);
7	PADD($[x_1], [x_2], [a_5]$);
8	PSIN($[a_5], [a_6]$);
9	PMAX($[a_6], [a_4], [a_7]$);
10	fin tant que

La performance de cette approche sera particulièrement soulignée dans le paragraphe suivant, qui propose des algorithmes nouveaux de caractérisation de la projection d'ensembles. Cette étude a donné lieu à l'article soumis [JBW02].

3.3.2 Projection d'un ensemble

Soit l'ensemble

$$\mathbb{S} = \{(\mathbf{p}, \mathbf{q}) \in \mathbb{P} \times \mathbb{Q} \mid \mathbf{f}(\mathbf{p}, \mathbf{q}) \leq \mathbf{0}\}$$

où \mathbb{P} est un pavé de \mathbb{R}^{n_p} , \mathbb{Q} est un pavé de \mathbb{R}^{n_q} et \mathbf{f} est une fonction de $\mathbb{R}^{n_p} \times \mathbb{R}^{n_q}$ dans \mathbb{R} . La projection de \mathbb{S} sur \mathbb{P} est définie par

$$\Pi = \{\mathbf{p} \in \mathbb{P} \mid \exists \mathbf{q} \in \mathbb{Q}, \mathbf{f}(\mathbf{p}, \mathbf{q}) \leq \mathbf{0}\}. \quad (3.41)$$

Dans cette partie, nous proposons des algorithmes capables de fournir des approximations intérieure et extérieure de l'ensemble Π . Nous nous appuyerons pour cela sur la notion de contracteurs.

De façon générale, ces algorithmes calculent deux sous-pavages $\underline{\Pi}$ et $\Delta\Pi$ tels que l'ensemble Π définis par (3.41) satisfasse

$$\underline{\Pi} \subset \Pi \subset (\underline{\Pi} \cup \Delta\Pi). \quad (3.42)$$

Leur principe est de mettre à jour trois listes :

- (i) la pile \mathcal{L} contenant la zone de $\mathbb{P} \times \mathbb{Q}$ qui doit encore être étudiée,
 - (ii) la liste \mathcal{L}^- contenant les pavés de \mathbb{R}^{n_p} dont on a prouvé qu'ils étaient inclus dans Π
 - (iii) la liste $\Delta\mathcal{L}$ contenant les pavés de \mathbb{R}^{n_p} pour lesquels aucune décision n'a été prise mais dont la taille est inférieure à ε ,
- de sorte que $\Delta\mathcal{L}$ décrit $\Delta\Pi$ et \mathcal{L}^- décrit $\underline{\Pi}$.

La structure générale des algorithmes de la famille PROJECT est présentée à la table 3.9. Elle est tout-à-fait comparable à celle des algorithmes de partitionnement, et en particulier à SIVIAP. Ainsi, \mathcal{L} est une liste de pavés de $\mathbb{R}^{n_p} \times \mathbb{R}^{n_q}$ d'intérieur disjoints. Sa spécificité réside dans l'utilisation de deux sous-routines, INSIDE et OUTSIDE. OUTSIDE s'appuie sur un contracteur de l'ensemble $[\mathbf{p}] \times [\mathbf{q}]$ qui fournit donc une approximation extérieure de $\mathbb{S} \cap [\mathbf{p}] \times [\mathbf{q}]$. La routine INSIDE permet de déterminer une approximation intérieure de $\mathbb{S} \cap [\mathbf{p}] \times [\mathbf{q}]$. La mise à jour des listes permet d'assurer que la relation

$$\Pi \subset \pi_p(\mathcal{L}) \cup \mathcal{L}^- \cup \Delta\mathcal{L} \quad (3.43)$$

est satisfaite chaque fois que l'algorithme atteint le pas 2. Lorsqu'il termine, la liste \mathcal{L} est vide et

$$\mathcal{L}^- \subset \Pi \subset \mathcal{L}^- \cup \Delta\mathcal{L}.$$

Selon le contexte et les routines OUTSIDE et INSIDE utilisées, nous obtenons des performances très différentes. Pour illustrer ce propos, nous présentons dans la suite deux algorithmes que nous dénommerons PROJECT1 et PROJECT2, aux caractéristiques très différentes. Nous les appliquerons sur plusieurs CSP.

3.3.2.1 Project1

Dans cette version, la routine OUTSIDE utilisée est elle-même un simple contracteur de \mathbb{S} . Comme il s'agit d'obtenir un encadrement extérieur de $[\mathbf{p}] \times [\mathbf{q}]$ sur un domaine relativement grand, nous avons choisi d'utiliser \mathcal{C}_{\uparrow} . Bien entendu, là aussi une collaboration de contracteurs serait plus efficace. La routine INSIDE1 permet de déterminer s'il existe une valeur de \mathbf{q} admissible pour une

Algorithme PROJECT ($e: [p], [q], s: \mathcal{L}^-, \Delta\mathcal{L}$)	
1	$\mathcal{L} := ([p], [q]); \mathcal{L}^- := \emptyset; \Delta\mathcal{L} := \emptyset;$
2	tant que $\mathcal{L} \neq \emptyset,$
3	dépiler une paire $([p], [q])$ de $\mathcal{L};$
4	OUTSIDE($[p], [q]$);
5	si $[p] = \emptyset$ aller au pas 2;
6	si (INSIDE ($[p], [q]$)) alors $\mathcal{L}^- := \mathcal{L}^- \cup \{[p]\};$ aller au pas 2;
7	si $w([p]) > \varepsilon,$
8	bissecter $[p]$ en $[p_1]$ et $[p_2];$
9	$\mathcal{L} := \mathcal{L} \cup \{[p_1], [q]\}; \mathcal{L} := \mathcal{L} \cup \{[p_2], [q]\};$
10	sinon $\Delta\mathcal{L} := \Delta\mathcal{L} \cup \{[p]\};$
11	fin tant que.

Tableau 3.9: Structure générale de la famille d'algorithmes PROJECT

valeur de p fixée. Elle s'appuie sur une stratégie de test inspirée de [JW96]. Sa mise en œuvre est décrite à la table 3.10. Le coefficient α est ajusté par l'utilisateur. Il caractérise le rapport entre la taille maximale des pavés $[p]$ bissectés et la taille maximale des pavés $[q]$ bissectés.

Algorithme INSIDE1 ($e: [p], [q], s: r$)	
1	$[p'] := [p]; [q'] := [q];$
2	$\mathcal{C}([p'] \times [q']);$
3	$[q_0] := \text{mid}([q']); [p_0] := \text{mid}([p']);$
4	si $([t]([p_0], [q']) = [0])$ renvoyer 0;
5	si $([t]([p'], [q_0]) = [0])$ renvoyer 1;
6	si $(w([q']) < \alpha\varepsilon)$ renvoyer 2;
7	$r_1 := \text{INSIDE1}([p], L[q']);$ si (r_1) renvoyer $r_1;$
8	$r_2 := \text{INSIDE1}([p], R[q']);$ si (r_2) renvoyer $r_2;$

Tableau 3.10: Algorithme CSC2

Elle consiste à contracter le pavé $[q]$ par \mathcal{C} et à tester si un point q_0 de $\mathcal{C}([q])$ est un point admissible. Le point q_0 choisi est le milieu de $\mathcal{C}([q])$. Comme nous l'avons déjà signalé, il est possible d'utiliser d'autres techniques (tirage aléatoire ou optimisation locale de critère) pour sélectionner un point q_0 . L'algorithme PROJECT1 est présenté au tableau 3.11.

Algorithme PROJECT1 ($e: [p], [q], \varepsilon, s: (\underline{\Pi}, \overline{\Pi})$)	
1	$[q'] := [q]; \mathcal{C}([p] \times [q']);$
2	si $([p] = \emptyset \vee [q'] = \emptyset)$ renvoyer $(\emptyset, \emptyset);$
3	si (INSIDE1($[p], [q]$)) renvoyer $([p], [p]);$
4	si $(w([p]) < \varepsilon)$ renvoyer $(\emptyset, [p]);$
5	$(\underline{\Pi}, \overline{\Pi}) := \text{PROJECT1}(L[p], [q], \varepsilon) \cup \text{PROJECT1}(R[p], [q], \varepsilon);$

Tableau 3.11: Algorithme PROJECT1

L'exemple 3.3.1 illustre les performances de PROJECT1 lorsque le graphe de représentation du CSP \mathcal{H} est un arbre, et qu'on dispose d'une fonction d'inclusion optimale.

Exemple 3.3.1 Soient S_1 et S_2 deux ensembles de \mathbb{R}^7 définis par :

$$\begin{aligned} S_1 &= \{ \mathbf{x} \in \mathbb{R}^7 \mid f(\mathbf{x}) \in [0, 1] \} \\ S_2 &= \{ \mathbf{x} \in \mathbb{R}^7 \mid f(\mathbf{x}) \in [0.95, 1] \} \\ f(\mathbf{x}) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2. \end{aligned}$$

Soit \mathbf{p} le sous-vecteur de \mathbf{x} défini par $\mathbf{p} = (x_1, x_2)$ et soit \mathbf{q} le sous-vecteur complémentaire de \mathbf{p} . On cherche à déterminer les projections Π_1 et Π_2 de S_1 et S_2 sur le domaine $\mathbb{P} = [-5, 5]^{\times 2}$ de \mathbf{p} . Les couples de sous-pavages $(\underline{\Pi}_1, \overline{\Pi}_1)$ et $(\underline{\Pi}_2, \overline{\Pi}_2)$ obtenus par PROJECT1 sur l'exemple 3.3.1 ainsi que les temps de calcul correspondants sont présentés sur la figure 3-21.

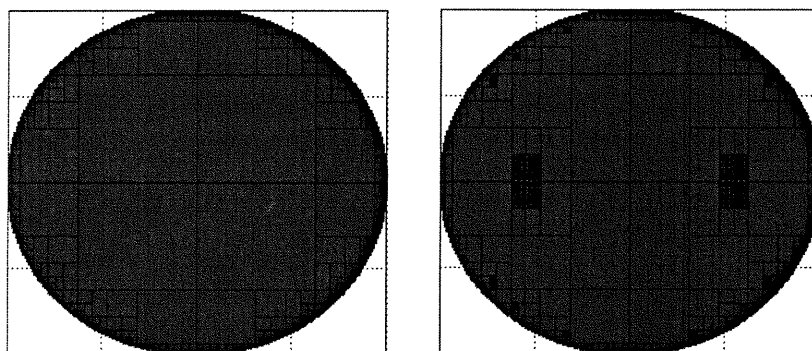


Figure 3-21: Approximations intérieure et extérieure $(\underline{\Pi}_1, \overline{\Pi}_1)$ (à gauche) obtenues en 4.6 s et approximations intérieure et extérieure $(\underline{\Pi}_2, \overline{\Pi}_2)$ obtenues en 51.2 s avec PROJECT1 pour l'exemple 3.3.1. Le cadre noir délimite la pavé extérieur $[-1, 1]^{\times 2}$.

Comme PROJECT1 s'appuie sur un test d'inclusion, ses performances sont limitées par l'efficacité du contracteur et de la fonction d'inclusion utilisés. Par ailleurs, comme l'algorithme de recherche intérieure INSIDE1 utilisé effectue des tests en recherchant des points admissibles, il est efficace lorsque le volume de l'ensemble solution est important. C'est pourquoi le temps de calcul de $(\underline{\Pi}_1, \overline{\Pi}_1)$ est beaucoup plus faible que celui de $(\underline{\Pi}_2, \overline{\Pi}_2)$. Enfin, nous avons introduit un coefficient α qui permet de limiter la profondeur de bisection dans l'espace \mathbb{Q} .

Le choix de α résulte d'une heuristique, et de l'information dont nous disposons *a priori* sur le volume de S . Lorsque le volume de S est important et que l'on dispose d'une fonction d'inclusion optimale, accroître la valeur de α améliore notablement les temps de calcul sans diminuer la qualité de l'encadrement, comme le montre le tableau 3.12 illustrant les résultats obtenus avec PROJECT1 pour la détermination de Π_1 pour différentes valeurs de α .

Bien entendu, si la fonction d'inclusion utilisée n'est pas optimale, augmenter la valeur de α augmente le volume de la couche d'incertitude. Il faut alors réaliser un compromis entre la qualité de l'encadrement obtenu, et le temps de calcul t_c .

Si dans ce cas, l'algorithme PROJECT1 est peu performant, c'est parce que la routine INSIDE1 s'appuie sur une technique de partitionnement systématique pour fournir une description de l'approximation intérieure $\underline{\Pi}$. L'algorithme PROJECT2 présenté dans le paragraphe suivant permet d'obtenir cette approximation intérieure en limitant le nombre de bisections.

ε	$\text{Vol}(\Delta\Pi)$	$\text{Vol}(\overline{\Pi})$	$\frac{\text{Vol}(\Delta\Pi)}{\text{Vol}(\overline{\Pi})}$	$t_c(\alpha = 10)$	$t_c(\alpha = 100)$	$t_c(\alpha = 10^{10})$
1	4.68	6.25	0.75	0.18	0.02	0.02
0.5	2.73	5.08	0.54	4.47	0.054	0.053
0.25	1.17	3.71	0.31	18.62	0.11	0.12
0.1	0.58	3.39	0.17	90.69	3.32	0.32
0.05	0.29	3.29	0.09	321.53	52.6	0.835
0.01	0.07	3.18	0.02	4003.46	1421	9.98

Tableau 3.12: Résultats obtenus par PROJECT1 sur l'exemple de la sphère pleine pour différentes valeurs de α

3.3.2.2 Project2

Sa mise en œuvre est présentée au tableau 3.13. La routine OUTSIDE2 appelée au pas 4 élabore un encadrement extérieur de $\mathbb{S} \cap ([\mathbf{p}] \times [\mathbf{q}])$. Elle est donnée au tableau 3.14. Elle met en œuvre un contracteur de l'ensemble \mathbb{S} . A chaque fois que l'algorithme atteint le pas 2, nous avons l'inclusion

$$([\mathbf{p}_0^{\text{init}}] \times [\mathbf{q}_0^{\text{init}}]) \cap \mathbb{S} = (\mathcal{L} \cup ([\mathbf{p}_0] \times [\mathbf{q}_0])) \cap \mathbb{S},$$

où $[\mathbf{p}_0^{\text{init}}]$ et $[\mathbf{q}_0^{\text{init}}]$ représentent les pavés de recherche initiaux. Lorsque l'algorithme OUTSIDE2 termine, la pile \mathcal{L} est vide et la propriété

$$([\mathbf{p}_0^{\text{init}}] \times [\mathbf{q}_0^{\text{init}}]) \cap \mathbb{S} = ([\mathbf{p}_0] \times [\mathbf{q}_0]) \cap \mathbb{S}$$

est vérifiée.

Algorithme PROJECT2(e: $[\mathbf{p}], [\mathbf{q}]; s: \mathcal{L}^-, \Delta\mathcal{L}$)	
1	$\mathcal{L} := ([\mathbf{p}], [\mathbf{q}]); \mathcal{L}^- := \emptyset; \Delta\mathcal{L} := \emptyset;$
2	tant que $\mathcal{L} \neq \emptyset$,
3	d piler $([\mathbf{p}], [\mathbf{q}])$ de \mathcal{L} ;
4	OUTSIDE2($[\mathbf{p}], [\mathbf{q}]$);
5	si $[\mathbf{p}] = \emptyset$, aller au pas 2;
6	$[\overline{\mathbf{p}}] := \text{INSIDE2}([\mathbf{p}], [\mathbf{q}]);$
7	si $[\overline{\mathbf{p}}] = \emptyset$ alors $\mathcal{L}^- := \mathcal{L}^- \cup [\mathbf{p}];$ aller au pas 2;
8	si $w([\overline{\mathbf{p}}]) > \varepsilon$,
9	bissecter $[\mathbf{p}]$ en $[\mathbf{p}_1]$ et $[\mathbf{p}_2]$,
10	$\mathcal{L} := \mathcal{L} \cup \{([\mathbf{p}_1], [\mathbf{q}]), ([\mathbf{p}_2], [\mathbf{q}])\};$
11	sinon $\Delta\mathcal{L} := \Delta\mathcal{L} \cup \{[\mathbf{p}]\};$
12	fin tant que.

Tableau 3.13: Algorithme PROJECT2

La routine INSIDE2 appelée au pas 6 au tableau 3.13 renvoie un pavé $[\overline{\mathbf{p}}]$ inclus dans $[\mathbf{p}]$ tel que $[\mathbf{p}] \cap \overline{\Pi} = [\overline{\mathbf{p}}] \cap \overline{\Pi}$. Contrairement à OUTSIDE2, cette routine ne modifie pas $[\mathbf{p}]$ et $[\mathbf{q}]$. La routine INSIDE2 renvoie $[\overline{\mathbf{p}}]$ de $[\mathbf{p}]$ tel que $[\mathbf{p}] \cap \Pi = [\overline{\mathbf{p}}] \cap \Pi$, c'est-à-dire que INSIDE2 met en œuvre un contracteur pour le complémentaire de l'ensemble Π . Toutes les valeurs de $[\overline{\mathbf{p}}]$ qui ont été éliminées dans la boucle du pas 4 appartiennent à Π . La figure 3-22 illustre ceci. Dans cet algorithme, seul $[\mathbf{q}]$ est bissecté, tous les pavés intermédiaires sont stockés dans la liste \mathcal{L} .

Cet algorithme s'appuie sur la propagation de contraintes dans les deux routines OUTSIDE2

Algorithme OUTSIDE2(es : $[p_0], [q_0]$)	
1	$\mathcal{L} := \{[p_0], [q_0]\}; \varepsilon := w([p_0]);$ $[p_0] := \emptyset; [q_0] := \emptyset;$
2	tant que $\mathcal{L} \neq \emptyset,$
3	d piler $([p], [q])$ de $\mathcal{L};$
4	$\mathcal{C}([p] \times [q]);$
5	si $w([q]) > \varepsilon,$
6	bissecter $[q]$ en $[q_1]$ et $[q_2];$
7	$\mathcal{L} := \mathcal{L} \cup \{([p], [q_1]), ([p], [q_2])\};$
8	sinon $[p_0] := [p_0] \sqcup [p]; [q_0] := [q_0] \sqcup [q];$
9	fin tant que.

Tableau 3.14: Algorithme OUTSIDE2

Algorithme INSIDE2(e : $[p_0], [q_0], s : \overline{[p]}$)	
1	$\mathcal{L} := \{[q_0]\}; \overline{[p]} := [p_0];$
2	tant que $\mathcal{L} \neq \emptyset,$
3	prendre le premier pavé $[q]$ de $\mathcal{L};$
4	répéter
5	$[m] := \text{mid}([q]); \mathcal{C}_\neg(\overline{[p]} \times [m]);$
6	$[p_1] := \overline{[p]}; \mathcal{C}([p_1] \times [q]);$
7	tant qu'une contraction significative de $\overline{[p]}$ est obtenue;
8	si $\overline{[p]} = \emptyset,$ retourner;
9	si $w([q]) > \alpha w([p_0]),$
10	bissecter $[q]$ en $[q_1]$ et $[q_2];$
11	stocker $[q_1]$ et $[q_2]$ dans $\mathcal{L};$
12	fin tant que.

Tableau 3.15: Routine INSIDE2

et INSIDE2. Cet algorithme est donc peu efficace si la fonction d'inclusion est optimale et le volume de l'ensemble solution élevé, comme le montre l'exemple suivant.

Exemple 3.3.1 *La détermination d'approximations intérieure et extérieure pour Π_1 et Π_2 à l'aide de PROJECT2 est très coûteuse en charge mémoire, puisque seul le pavé initial est réduit. Le temps de calcul est exponentiel.*

Dans le cas où l'on ne dispose pas d'une fonction d'inclusion optimale, la routine INSIDE2 permet d'obtenir une meilleure approximation intérieure que INSIDE1. Nous en verrons un exemple au chapitre 4.

3.3.2.3 Quantificateur universel

Les algorithmes de projection fournissent des approximations intérieure et extérieure d'ensembles solutions Π de CSP qui comportent un ou plusieurs quantificateurs existentiels \exists . Cependant, de nombreux problèmes fréquents en automatique peuvent se formuler comme des CSP comportant le quantificateur universel \forall . Il s'agit par exemple de caractériser l'ensemble Γ défini par

$$\Gamma = \{p \in \mathbb{P} \mid \forall q \in \mathbb{Q}, f(p, q) \leq 0\}. \quad (3.44)$$

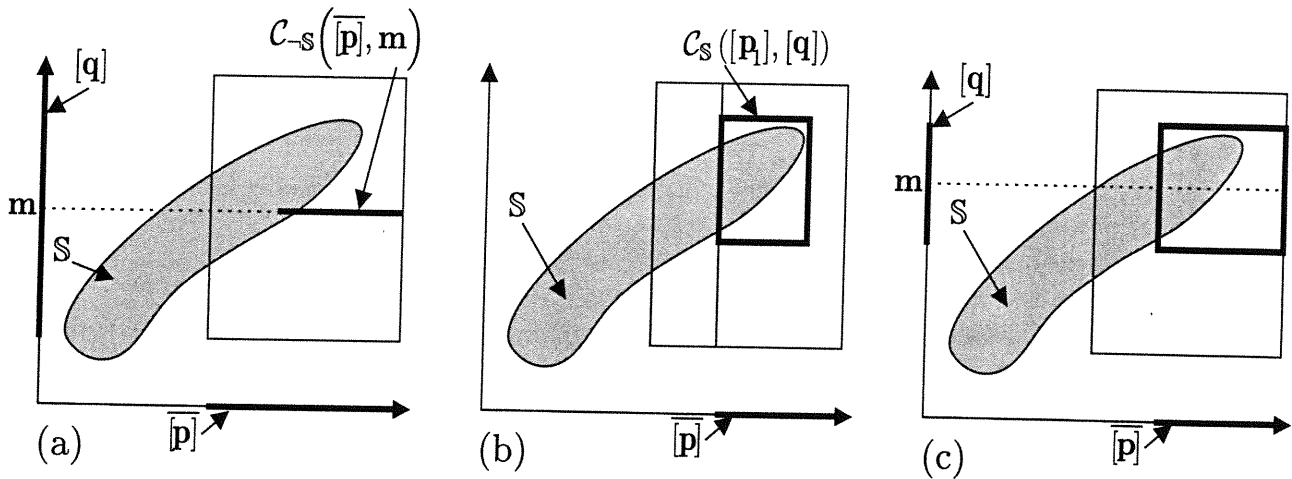


Figure 3-22: Illustration de INSIDE2; (a) $[\bar{p}]$ est contracté sans perdre un seul point de $]S$; (b) $[p]$ est contracté; (c) Nouvelle itération

Si une approximation extérieure –éventuellement très pessimiste– peut être facilement obtenue, il est indispensable de pouvoir aussi fournir une approximation intérieure de l'ensemble Γ , ce qui fait l'objet de nombreux travaux, dans les deux communautés de l'analyse par intervalles et de l'informatique.

Une des approches envisagées dans la communauté de l'analyse par intervalles est l'*analyse modale*. Elle étend la notion classique d'intervalle $[x] = [\underline{x}, \bar{x}] = \{x \mid \underline{x} \leq x \leq \bar{x}\}$, qui suppose que $\underline{x} \leq \bar{x}$ à la notion d'intervalle modal où la contrainte $\underline{x} \leq \bar{x}$ est relâchée. Cette extension permet d'enrichir la sémantique de l'analyse par intervalles et de définir des opérateurs capables de prendre en compte directement l'opérateur \forall . Il existe d'autres directions de recherche dans la communauté informatique. Malheureusement, ces techniques ne sont performantes que lorsque le système ne dispose que d'un seul quantificateur universel, c'est-à-dire lorsque q est scalaire. Nous proposons ici une méthode très simple de caractérisation de l'ensemble Γ défini par (3.44) à partir des algorithmes de projection.

Soit l'ensemble complémentaire $\Pi = \neg\Gamma$ de Γ . Par définition,

$$\Pi = \{p \in \mathbb{P} \mid \exists q \in \mathbb{Q}, f(p, q) > 0\},$$

et donc Π est la projection sur \mathbb{P} de l'ensemble solution du CSP

$$\mathcal{H} : (p \in \mathbb{P}, q \in \mathbb{Q}, f(p, q) > 0).$$

Les algorithmes de projection présentés ci-dessus permettent donc de fournir des approximations intérieure $\underline{\Pi}$ et extérieure $\bar{\Pi}$ de Π , avec $\bar{\Pi} = \underline{\Pi} \cup \Delta\Pi$. En complétant l'équation (3.42), il vient

$$\neg\bar{\Pi} \subset \Gamma \subset \neg\underline{\Pi}.$$

Les ensembles $\neg\bar{\Pi}$ et $\neg\underline{\Pi}$ constituent donc des approximations intérieure et extérieure de Γ . L'approximation intérieure de Γ est obtenue à partir de l'approximation extérieure de Π , dont

nous pouvons contrôler la précision. En remarquant par ailleurs que

$$\begin{aligned}
\neg\overline{\Pi} \cup \Delta\Pi &= \neg(\underline{\Pi} \cup \Delta\Pi) \cup \Delta\Pi \\
&= (\neg\underline{\Pi} \cap \neg\Delta\Pi) \cup \Delta\Pi \\
&= (\neg\underline{\Pi} \cup \Delta\Pi) \cap (\neg\Delta\Pi \cup \Delta\Pi) \\
&= \neg\underline{\Pi} \cup \Delta\Pi = \neg\underline{\Pi},
\end{aligned}$$

l'encadrement obtenu pour Γ possède la même couche d'incertitude que celui obtenu pour Π . En particulier, il devient possible de comparer la qualité de deux encadrements de Γ , l'un calculé directement et l'autre obtenu à partir de cette remarque. L'applicabilité de cette méthode est testée sur l'exemple suivant, proposé par F. Goualard et M. Christie de l'IRIN.

Exemple 3.3.2 On cherche à déterminer l'ensemble des positions (x, y, z) d'une caméra telles qu'à tout instant cette caméra reste distante d'au moins 0.5 unités d'un objet mobile dont la trajectoire plane est décrite par le système

$$\begin{cases} X(\theta) = 3 \sin \theta \cos \theta (\sin \theta - \cos \theta) \\ Y(\theta) = 3 \sin \theta \cos \theta (\sin \theta + \cos \theta) \end{cases},$$

avec bien entendu $\theta \in [0, 2\pi]$. Il s'agit donc de déterminer l'ensemble Γ

$$\Gamma = \left\{ (x, y, z) \in [-3, 3]^{\times 3} \mid \forall \theta \in [0, 2\pi] \sqrt{(X(\theta) - x)^2 + (Y(\theta) - y)^2 + z^2} \geq 0.5 \right\}.$$

Nous choisissons de caractériser de façon équivalente l'ensemble $\Pi = \neg\Gamma$ qui vérifie

$$\Pi = \left\{ (x, y, z) \in [-3, 3]^{\times 3} \mid \exists \theta \in [0, 2\pi] \sqrt{(X(\theta) - x)^2 + (Y(\theta) - y)^2 + z^2} < 0.5 \right\}$$

qui est la projection $\pi_{123}(\mathbb{S})$ où

$$\mathbb{S} = \left\{ \mathbf{x} \in [-3, 3]^{\times 3} \times [0, 2\pi] \mid \sqrt{(3 \sin x_4 \cos x_4 (\sin x_4 - \cos x_4) - x_1)^2 + (3 \sin x_4 \cos x_4 (\sin x_4 + \cos x_4) - x_2)^2 + x_3^2} < 0.5 \right\}.$$

Les approximations intérieure et extérieure de Π obtenues avec PROJECT1 pour $\varepsilon = 0.1$ sont représentées sur la figure 3-23. L'approximation intérieure de Γ est donc représentée en blanc, sa couche d'incertitude est représentée en gris. La comparaison des performances des deux algorithmes testés sur un Pentium II, 350 MHz, est présentée au tableau 3.16.

ε	Vol($\Delta\Pi$)	Vol($\overline{\Pi}$)	η	Vol($\underline{\Gamma}$)	$t_c(s)$
1	17.69	17.69	1	198.31	2.1
0.5	7.48	10.09	0.74	205.91	12.8
0.25	2.82	8.45	0.33	207.55	45.6
0.1	1.16	7.38	0.15	208.62	172.3

Tableau 3.16: Performances de Project1 pour le CSP de l'exemple 3.3.2

Ces résultats, obtenus sur un Pentium II (300MHz), sont à comparer avec ceux présentés au tableau 3.18, qui sont obtenus à l'aide de l'algorithme dédié à l'élaborer d'approximations

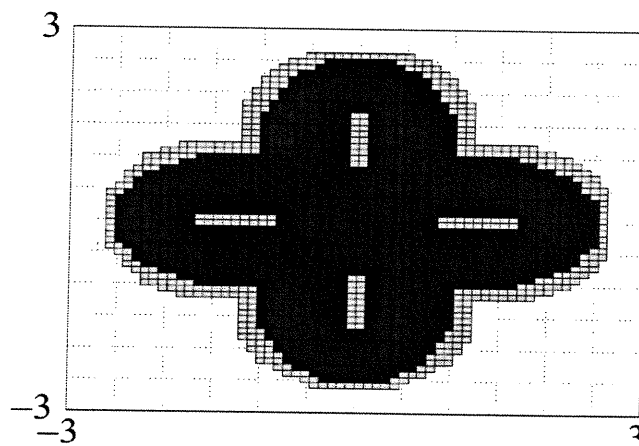


Figure 3-23: Sous-pavages tridimensionnels Π (en noir) et $\Delta\Pi$ (en gris) obtenus par PROJECT1 pour $\varepsilon = 0.1$ pour le CSP de l'exemple 3.3.2, projetés dans le plan (x_1, x_2)

ε	$Vol(\Delta\Pi)$	$Vol(\Pi)$	η	$Vol(\Gamma)$	$t_c(s)$
1	26.69	26.69	1	189.31	1.3
0.5	10.95	10.95	1	205.05	11.6
0.25	8.99	8.99	1	207.01	69.8

Tableau 3.17: Performances de Project2 pour le CSP de l'exemple 3.3.2

intérieures ICAB5 sur un Pentium III (800MHz) par M. Christie [BGLC99] [BG00] [Chr02], que je tiens à remercier.

Les algorithmes PROJECT1 et PROJECT2 permettent donc d'obtenir une approximation intérieure de l'ensemble Γ en des temps de calcul intéressants. Nous allons maintenant nous intéresser à déterminer la forme et le volume d'un ensemble \mathbb{S} . Pour cela, il est nécessaire de contrôler le pessimisme dû au phénomène d'enveloppement. Ce travail a donné lieu à la publication [BJKW01b].

3.3.3 Contrôle du pessimisme lié au phénomène d'enveloppement

Lorsque \mathbb{S} représente un ensemble solution d'un système d'équations ou d'inégalités, sa forme peut révéler le mauvais conditionnement du système numérique, comme nous le verrons au chapitre 4. Lorsque \mathbb{S} représente le support d'une variable incertaine \mathbf{x} , son volume est directement lié à l'incertitude d'estimation de \mathbf{x} . Il est donc très souvent nécessaire de pouvoir fournir une estimation précise du volume et de la forme de \mathbb{S} . Une technique de recouvrement à l'aide de pavés

ε	$Vol(\Gamma)$	$t_c(s)$
1	193.58	0.910
0.5	197.74	5.230
0.25	199.62	53.870
0.1	201.66	1480.630

Tableau 3.18: Performances de l'algorithme ICAB5 pour le CSP de l'exemple 3.3.2

peut alors se révéler inadaptée, en raison du phénomène d'enveloppement. Prenons l'exemple du calcul du volume. La description de \mathbb{S} par deux sous-pavages intérieur $\underline{\mathbb{S}}$ et extérieur $\overline{\mathbb{S}}$ permet d'écrire

$$\begin{aligned}\text{Vol}(\mathbb{S}) &\in [\text{Vol}(\underline{\mathbb{S}}), \text{Vol}(\overline{\mathbb{S}})], \\ \text{Vol}(\mathbb{S}) &\in \text{Vol}(\underline{\mathbb{S}}) + [0, \text{Vol}(\Delta\mathbb{S})],\end{aligned}$$

de sorte que la précision de l'encadrement du volume $\text{Vol}(\mathbb{S})$ dépend directement de $\text{Vol}(\Delta\mathbb{S})$. Soit \bar{n} le nombre maximal de pavés dans la couche $\Delta\mathbb{S}$. Puisque cette couche est calculée à partir d'un algorithme de partitionnement dont le paramètre d'arrêt ε est la taille maximale des pavés de la couche $\Delta\mathbb{S}$, nous avons $\text{Vol}(\Delta\mathbb{S}) \leq \bar{n}\varepsilon^m$. Si ε décroît, l'encadrement devient plus précis. En quantifiant cette précision par $\delta = \frac{1}{\text{Vol}(\Delta\mathbb{S})}$, on dispose d'une borne minimale $\delta \geq \frac{\varepsilon^{-m}}{\bar{n}}$, dans le cas d'une description par sous-pavages réguliers. Sur un même problème, on cherche généralement à améliorer la précision en diminuant ε , ce qui peut très rapidement mener à des problèmes d'explosion de \bar{n} . Un exemple extrême est présenté sur la figure 3-24. Sur cet exemple, qui correspond à un cas linéaire, on a $\text{Vol}(\Delta\mathbb{S}) = 2^n\varepsilon^m$, où n est la profondeur de bissection.

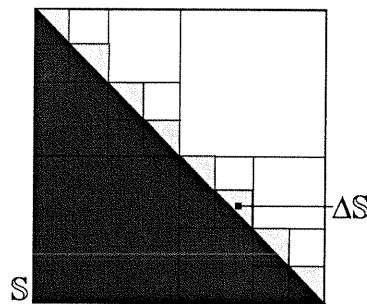


Figure 3-24: Illustration du phénomène d'enveloppe

Pour éviter ces problèmes d'explosion numérique, nous proposons ici un nouvel algorithme de recouvrement, qui ne s'appuie pas sur un pavage régulier. En réduisant la taille de la couche d'incertitude sans augmenter le nombre de bisections, nous envisageons alors d'obtenir une estimation du volume de \mathbb{S} plus précise, en réduisant le volume de $\Delta\mathbb{S}$. Les objets stockés dans la liste sont de nature différente.

Exemple 3.3.3 *Considérons le problème bidimensionnel suivant :*

$$\mathbb{S} = \left\{ \mathbf{x} \in [0, 5]^{\times 2} \mid \mathbf{f}(\mathbf{x}) \in [\mathbf{y}] \right\},$$

où

$$\begin{aligned}f : \mathbb{R}^2 &\rightarrow \mathbb{R}^3 \\ \mathbf{x} &\mapsto \begin{pmatrix} x_2 - 2x_1 \\ x_2 - \frac{6}{x_1} \\ x_2 - 0.5\sqrt{x_1} \end{pmatrix}\end{aligned}$$

et

$$[\mathbf{y}] = [-1, 1] \times [-2, 2] \times [-2, 2].$$

Les ensembles $\underline{\mathbb{S}}$ and $\overline{\mathbb{S}}$ obtenus par SIVIA sont représentés sur la figure 3-25. Lorsque ε décroît,

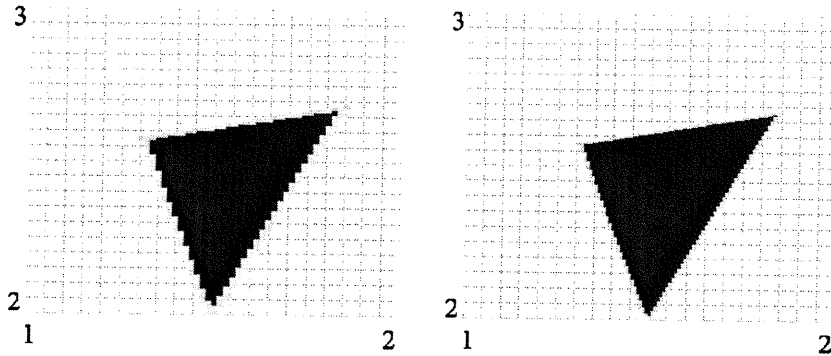


Figure 3-25: Sous-pavages $\underline{\mathbb{S}}$ (noir) et $\overline{\mathbb{S}}$ (gris clair) de l'ensemble solution \mathbb{S} de l'exemple 3.3.3 calculé avec SIVIA. A gauche $\varepsilon = 0.01$, 638 bisections, 0.394 s, $\delta = 294$. A droite $\varepsilon = 0.005$, 1269 bisections, 0.771 s, $\delta = 833$

le nombre de bisections augmente, de même que la charge de calcul, sans toutefois résoudre le problème d'enveloppement. La situation empire bien évidemment lorsque m augmente.

Puisque le temps de calcul t_c augmente de façon exponentielle lorsque SIVIA bissecte des pavés pour des problèmes où m est élevé, nous cherchons à réduire la taille des pavés indéterminés $[\mathbf{x}]$ stockés dans $\Delta\mathcal{L}$ sans les bissecter davantage. On peut alors formuler la question en ces termes:

Supposons qu'un ensemble \mathbb{S} soit encadré entre deux unions de pavés $\underline{\mathbb{S}}$ et $\overline{\mathbb{S}}$ tels que $\underline{\mathbb{S}} \subset \mathbb{S} \subset \overline{\mathbb{S}}$, avec $\overline{\mathbb{S}} = \underline{\mathbb{S}} \cup \Delta\mathbb{S}$. Peut-on réduire la taille de $\Delta\mathbb{S}$ sans bisection, sous l'hypothèse que les pavés de $\Delta\mathbb{S}$ sont déjà de taille faible?

Soit $[\mathbf{x}]$ un pavé de $\Delta\mathbb{S}$, et notons $\mathbb{S}_{[\mathbf{x}]} = \mathbb{S} \cap [\mathbf{x}]$. Réduire la taille de $\Delta\mathbb{S}$ signifie trouver un encadrement $(\underline{\mathcal{X}}, \overline{\mathcal{X}})$ de $\mathbb{S}_{[\mathbf{x}]}$ dans $[\mathbf{x}]$, tel que $\underline{\mathcal{X}} \subset \mathbb{S}_{[\mathbf{x}]} \subset \overline{\mathcal{X}} \subset [\mathbf{x}]$. Lorsque seul $\overline{\mathcal{X}}$ est recherché, sous la contrainte que $\overline{\mathcal{X}}$ est un pavé, l'ensemble des contracteurs (en particulier $\mathcal{C}_{1\uparrow}$) peuvent répondre à ce problème. Malheureusement, si la frontière $\partial\mathbb{S}$ de \mathbb{S} est loin d'être parallèle aux axes de l'espace des paramètres, en raison du phénomène d'enveloppement, tout accroissement de la précision requièrera un nombre important de bisections. De plus, cette approche ne fournit qu'une approximation extérieure et ne permet pas d'améliorer la qualité de la représentation de $\underline{\mathbb{S}}$. Afin de prendre en compte l'information fournie par le gradient de $\partial\mathbb{S}$ dans la procédure de contraction, nous proposons de linéariser les fonctions $f_i(\mathbf{x})$ sur chaque pavé $[\mathbf{x}]$ de $\Delta\mathbb{S}$. Le paragraphe suivant indique alors différentes techniques de calcul des récipients $\overline{\mathcal{X}}$ et $\underline{\mathcal{X}}$ pour l'ensemble \mathbb{S} .

3.3.3.1 Linéarisations garanties

Nous avons présenté au paragraphe 3.1.3 quelques techniques de linéarisation garanties, que nous avons utilisées dans les contracteurs de CSP non linéaires. Nous nous sommes intéressés aux techniques de linéarisation de Newton et au contracteur par linéarisation parallèle, mais nous pourrions également citer [BM98] et [BB01]. Ces contracteurs fournissent en sortie une approximation extérieure éventuellement exacte du pavé enveloppe du CSP linéarisé. Dans ce qui suit, nous proposons de réduire le phénomène d'enveloppement en fournissant une approximation extérieure de l'ensemble solution du CSP linéarisé. Afin de pouvoir étudier le cas le plus général

possible, l'approche choisie est la linéarisation parallèle, mise en œuvre dans le contracteur \mathcal{C}_{\parallel} . Soit \mathbb{S} l'ensemble solution de volume éventuellement non-nul, que nous écrirons génériquement sous la forme

$$\mathbb{S} = \{\mathbf{x} \in \mathbb{X} \mid \mathbf{f}(\mathbf{x}) \in [\mathbf{y}]\}.$$

Soit $[\mathbf{x}] \in \mathbb{X}$. Nous allons définir un encadrement extérieur pour $\mathbb{S} \cap [\mathbf{x}]$ à l'aide de la linéarisation parallèle. Encadrons chaque $f_i(\mathbf{x})$ sur $[\mathbf{x}]$ entre deux hyperplans parallèles, de manière à pouvoir écrire

$$\mathbf{Ax} + \underline{\mathbf{b}} \leq \mathbf{f}(\mathbf{x}) \leq \mathbf{Ax} + \overline{\mathbf{b}},$$

et calculons une approximation extérieure $\overline{\mathcal{X}}$ de $\mathbb{S} \cap [\mathbf{x}]$. En remarquant que

$$\begin{aligned} \mathbf{f}(\mathbf{x}) \in [\mathbf{y}] &\Rightarrow \exists \mathbf{b} \in [\underline{\mathbf{b}}], \exists \mathbf{y} \in [\mathbf{y}] \mid \mathbf{Ax} + \mathbf{b} = \mathbf{y} \\ &\Leftrightarrow \mathbf{Ax} \in [\mathbf{y}] - [\underline{\mathbf{b}}] = [\underline{\mathbf{y}} - \overline{\mathbf{b}}, \overline{\mathbf{y}} - \underline{\mathbf{b}}], \end{aligned}$$

alors l'ensemble solution $\langle \overline{\mathbf{x}} \rangle$ de $\begin{cases} \mathbf{Ax} \in [\mathbf{c}_e] \\ \mathbf{x} \in [\mathbf{x}] \end{cases}$ où $[\mathbf{c}_e] = [\underline{\mathbf{y}} - \overline{\mathbf{b}}, \overline{\mathbf{y}} - \underline{\mathbf{b}}]$ est une approximation extérieure de $\mathbb{S} \cap [\mathbf{x}]$. Comme $\langle \overline{\mathbf{x}} \rangle$ est l'intersection finie de demi-espaces fermés, c'est un polyèdre convexe et compact, c'est-à-dire un *polytope* (voir la figure 3-26, gauche). Remarquons que $[\mathbf{c}_e]$ a été obtenu en incorporant les non-linéarités délaissées par le système linéarisé dans le pavé $[\mathbf{y}]$.

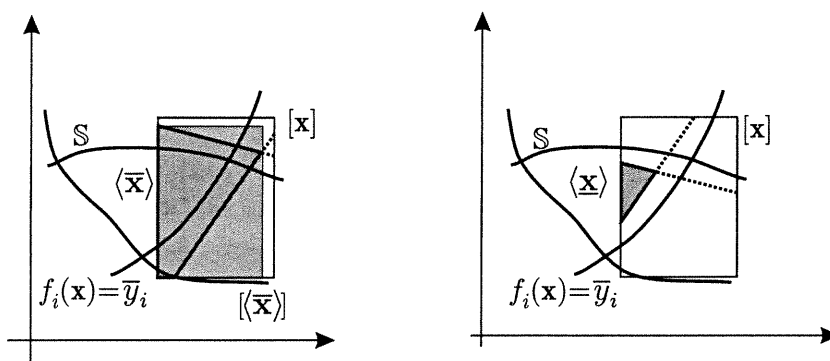


Figure 3-26: Objets obtenus après application de deux techniques de contraction: $\langle \overline{\mathbf{x}} \rangle$ est obtenu par le contracteur \mathcal{C}_{\uparrow} , l'ensemble $\langle \overline{\mathbf{x}} \rangle$ en gris foncé est obtenu par linéarisation extérieure. A droite, l'ensemble $\langle \underline{\mathbf{x}} \rangle$ est obtenu par linéarisation intérieure

La détermination de l'approximation intérieure $\underline{\mathcal{X}}$ de \mathbb{S} sur le pavé $[\mathbf{x}]$ suit la même procédure. Soit un élément $\mathbf{x} \in [\mathbf{x}]$ tel que $\mathbf{Ax} \in [\mathbf{c}_i]$ où

$$[\mathbf{c}_i] = [\underline{\mathbf{y}} - \underline{\mathbf{b}}, \overline{\mathbf{y}} - \overline{\mathbf{b}}]. \quad (3.45)$$

Alors

$$\begin{aligned} \mathbf{Ax} &\geq \underline{\mathbf{y}} - \underline{\mathbf{b}} \text{ et } \mathbf{Ax} \leq \overline{\mathbf{y}} - \overline{\mathbf{b}} \\ &\Leftrightarrow [\mathbf{Ax} + \underline{\mathbf{b}}, \mathbf{Ax} + \overline{\mathbf{b}}] \subset [\underline{\mathbf{y}}, \overline{\mathbf{y}}] \\ &\Rightarrow \mathbf{f}(\mathbf{x}) \in [\mathbf{y}], \end{aligned}$$

où \mathbf{A} et $[\mathbf{b}]$ sont calculés par le contracteur \mathcal{C}_{\parallel} . Tout \mathbf{x} vérifiant

$$\begin{cases} \mathbf{x} \in [\mathbf{x}] \\ \mathbf{Ax} \in [\mathbf{c}_i] \end{cases} \quad (3.46)$$

est dans $\mathbb{S} \cap [\mathbf{x}]$. L'ensemble défini par (3.46) est à nouveau un polytope, noté $\langle \underline{\mathbf{x}} \rangle$, inclus dans $\langle \bar{\mathbf{x}} \rangle$ (voir la Figure 3-26, droite).

Finalement à partir du pavé $[\mathbf{x}]$, les linéarisations intérieure et extérieure fournissent deux polytopes inclus dans $[\mathbf{x}]$, appelés $\langle \underline{\mathbf{x}} \rangle$ et $\langle \bar{\mathbf{x}} \rangle$, tels que

$$\langle \underline{\mathbf{x}} \rangle \subset \mathbb{S} \cap [\mathbf{x}] \subset \langle \bar{\mathbf{x}} \rangle,$$

où $\langle \underline{\mathbf{x}} \rangle$ est défini par $(\mathbf{A}, [\mathbf{c}_i])$ et $\langle \bar{\mathbf{x}} \rangle$ est défini par $(\mathbf{A}, [\mathbf{c}_e])$. le calcul des volumes des polytopes $\langle \underline{\mathbf{x}} \rangle$ et $\langle \bar{\mathbf{x}} \rangle$ est un thème d'étude très important en programmation [WEBc]. Plusieurs logiciels gratuits et téléchargeables existent, parmi lesquels l'algorithme *lrs* [WEBb]. Ceux-ci peuvent alors être utilisés dans SIVIA1 de manière à calculer $\langle \underline{\mathbf{x}} \rangle$ et $\langle \bar{\mathbf{x}} \rangle$ pour chaque pavé indéterminé $[\mathbf{x}]$ de $\Delta\mathcal{L}$.

3.3.3.2 Sivia1

L'idée principale de cet algorithme et d'obtenir une plus grande précision dans la description de l'ensemble solution \mathbb{S} et de son volume tout en gardant la structure de SIVIA, sans accroître le nombre de bisections, à l'aide des techniques de linéarisations garanties du paragraphe précédent. De même que SIVIA, SIVIA1 est un algorithme récursif visant à caractériser un ensemble décrit par des inégalités, et plus spécialement à calculer son volume. Les ensembles $\underline{\mathbb{S}}$ et $\bar{\mathbb{S}}$ sont alors hybridés: ils sont constitués de pavés intérieurs $[\mathbf{x}]$ et de polytopes $\langle \mathbf{x} \rangle$ caractérisant la frontière $\Delta\mathbb{S}$.

Remarque 3.3.1 *Il se peut que $\langle \underline{\mathbf{x}} \rangle$ soit vide, lorsque $\underline{\mathbf{y}} - \underline{\mathbf{b}} > \bar{\mathbf{y}} - \bar{\mathbf{b}}$. Pour éviter un calcul alors inutile, un test d'existence est inclus dans SIVIA1. De plus, nous avons maintenant un encadrement pour les inégalités définies sur $[\mathbf{x}]$, plus précis que le pavé initial $[\mathbf{x}]$, si $[\mathbf{x}]$ est suffisamment petit pour que les contraintes y soient quasiment linéaires.*

Contrairement à SIVIA, SIVIA1 ne fournit pas un pavage régulier. Le paramètre ε commande encore directement le nombre de bisections: plus il est petit, plus le nombre de bisections et la charge de calcul sont importants. Ceci est alors en partie compensé par le fait que les pavés de $\Delta\mathcal{L}$ seront petits, de telle sorte que peu de contraintes y seront actives. Ainsi en général, seule une contrainte f_i sera linéarisée sur le pavé, ce qui facilitera grandement les calculs des polytopes $\langle \underline{\mathbf{x}} \rangle$ et $\langle \bar{\mathbf{x}} \rangle$. Afin de se focaliser sur les performances de cette technique de linéarisation, tous les résultats suivants sont présentés sur des problèmes d'estimation bidimensionnels simples.

Exemple 3.3.4 *On considère le CSP \mathcal{H} défini par*

$$\mathcal{H} : (\forall i \in \{1, \dots, 5\} x_1 \sin(2\pi x_2 t_i) \in [y_i], \mathbf{x} \in [1, 10] \times [0, 1]),$$

où

$$\begin{aligned} [y_i] &= 5 \sin(2\pi \cdot 0.4 t_i) + [-0.5, 0.5] \quad \forall i \in \{1, \dots, 5\}, \\ \text{et } \mathbf{t} &= (0 \ 2 \ 4 \ 6 \ 8)^T. \end{aligned}$$

La figure 3-27 représente les ensembles $\underline{\mathbb{S}}$ et $\overline{\mathbb{S}}$ définissant l'encadrement garanti de \mathbb{S} . Pour une même valeur de $\varepsilon = 0.05$, l'encadrement fourni par SIVIA1 est plus fin que celui fourni par SIVIA, puisque $\delta_0 = 7.19 < \delta_1 = 500$.

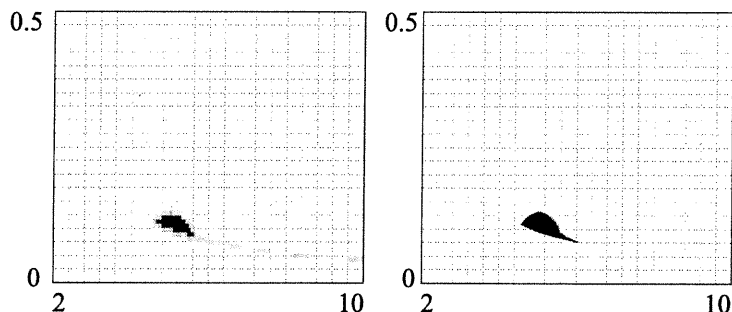


Figure 3-27: Approximations intérieure et extérieure de l'ensemble solution \mathbb{S} défini dans l'exemple, pour $\varepsilon = 0.05$. A gauche, SIVIA obtient en 166 bisections et $0.05s$, $\text{Vol}(\mathbb{S}) \in [0.099, 0.238]$. A droite, SIVIA1 fournit en 166 bisections et $0.395s$, $\text{Vol}(\mathbb{S}) \in [0.163, 0.165]$

Une application de SIVIA1 à la détection de données dangereuse est proposée au chapitre 4.

3.4 Conclusion

Nous avons détaillé différents algorithmes capables de contracter des problèmes de satisfaction de contraintes \mathcal{H} . Ainsi les contracteurs issus de l'analyse par intervalles sont performants pour des pavés $[\mathbf{x}]$ de petite taille. Le contracteur de propagation-rétropropagation $C_{\downarrow\uparrow}$, qui s'appuie sur la propagation de contraintes est quant à lui performant sur des pavés de taille importante. En particulier, il est optimal lorsque le graphe de représentation du CSP \mathcal{H} est un arbre. L'utilisation de ces outils permet d'une part, d'améliorer les performances de l'algorithme de partitionnement SIVIA, mais également de développer de nouveaux algorithmes ensemblistes. A cette occasion, nous avons proposé deux algorithmes de projection aux performances complémentaires. Nous avons enfin illustré la possibilité de réduire le phénomène d'enveloppement en manipulant des unions d'objets mixtes.

Ces outils nous permettront de traiter de manière efficace différentes applications issues de problèmes réels rencontrés tout au long de la démarche de construction de modèles. Nous soulignerons ainsi que les techniques garanties peuvent constituer des outils intéressants pour diverses étapes du processus de construction d'un modèle, du choix du capteur à l'identification des paramètres.

Chapitre 4

Identification

L'identification de paramètres est un domaine où les méthodes ensemblistes garanties ont pu fournir de nombreux résultats depuis quelques années, en particulier dans le contexte de l'inversion ensembliste. Deux grandes parties composent ce chapitre. La première illustre les performances de ces techniques sur des modèles très simples de cinétique réactionnelle. Ces techniques sont à notre connaissance, nouvelles dans le domaine de l'électrochimie. L'intérêt qu'elles ont suscité est souligné par la parution récente de [MCR02] dans la continuité de nos travaux [BBJ⁺01b]. Dans cette application, qui est proposée au lecteur soucieux de se familiariser aux méthodes ensemblistes garanties, les résultats obtenus par ces méthodes sont comparés aux estimateurs ponctuels, plus classiquement utilisés. La deuxième partie de ce chapitre contient le résultat principal de ce chapitre. En présence d'autres grandeurs incertaines, l'estimation ensembliste à erreurs bornées consiste à projeter un ensemble solution, que nous pouvons caractériser à l'aide des algorithmes exposés dans le chapitre précédent. Grâce à cette méthode nouvelle, nous pouvons alors caractériser de façon garantie les performances d'un dispositif de mesure indirecte en tenant compte des incertitudes intrinsèques au dispositif.

4.1 Introduction

Dans ce chapitre, nous considérons le problème de l'identification du vecteur \mathbf{x} des paramètres d'un modèle mathématique \mathcal{M} à partir de données expérimentales. Ces données \mathbf{y} sont recueillies en sortie d'un système \mathcal{S} regroupant le système physique étudié soumis à l'excitation \mathbf{u} et le dispositif de mesure. Ceci peut être envisagé en vue d'atteindre deux objectifs :

- la construction d'un modèle décrivant le comportement de \mathcal{S} , dans des objectifs de connaissance, de prédiction ou de commande,
- l'attribution de valeurs numériques aux composantes de \mathbf{x} qui représentent des grandeurs physiques incertaines : nous parlerons dans ce cas de *mesure indirecte* de \mathbf{x} .

Nous supposons dans tout ce chapitre que le vecteur de paramètres \mathbf{x} est constant. Nous supposons le modèle mathématique $\mathcal{M}(\cdot)$ paramétré en \mathbf{x} disponible *a priori* et tel que le vecteur $\boldsymbol{\eta}(\mathbf{x})$ de ses sorties soit homogène à \mathbf{y} (voir la figure 4-1). On définit l'*erreur de sortie* $\mathbf{e}(\mathbf{x})$ comme l'écart entre les deux vecteurs de sortie :

$$\mathbf{e}(\mathbf{x}) = \mathbf{y} - \boldsymbol{\eta}(\mathbf{x}). \quad (4.1)$$

Idéalement, on souhaiterait obtenir un modèle qui décrive parfaitement (tout au moins dans les conditions de l'expérience qui a fourni \mathbf{y}) le comportement de \mathcal{S} , c'est-à-dire trouver une

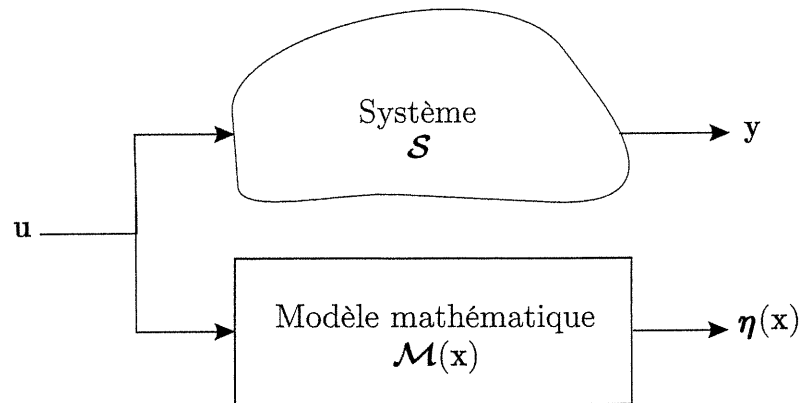


Figure 4-1: Identification du vecteur de paramètres \mathbf{x} à partir de données expérimentales

estimée $\hat{\mathbf{x}}$ qui annule $\mathbf{e}(\cdot)$. Cependant, deux types d'erreurs rendent la chose généralement impossible :

- l'*erreur de modélisation*, de caractérisation ou encore de modèle \mathbf{e}_η rendant compte du caractère nécessairement imparfait du modèle mathématique utilisé, difficilement quantifiable,
- l'*erreur de mesure* \mathbf{e}_y entachant la collecte des données, sur laquelle l'expérimentateur peut disposer d'informations.

Par souci de simplicité, nous choisissons de reporter ces deux erreurs en sortie, en introduisant le vecteur de données expérimentales utiles \mathbf{y}^* , correspondant aux données qui seraient recueillies sur le système \mathcal{S} en l'absence d'erreur de mesure :

$$\mathbf{y} = \mathbf{y}^* + \mathbf{e}_y.$$

Les données expérimentales \mathbf{y}^* seraient également obtenues, dans les mêmes conditions expérimentales, à la sortie du modèle idéal \mathcal{M}^* , de sorte que l'on peut également écrire :

$$\boldsymbol{\eta}(\mathbf{x}) = \mathbf{y}^* - \mathbf{e}_\eta.$$

Nous pouvons alors simplement écrire :

$$\mathbf{e}(\mathbf{x}) = \mathbf{e}_\eta + \mathbf{e}_y.$$

La figure 4-2 illustre ces définitions.

Puisqu'il n'est généralement pas possible d'annuler $\mathbf{e}(\mathbf{x})$, il s'agit donc d'estimer \mathbf{x} en prenant en compte l'information disponible sur \mathbf{e} . La démarche s'articule alors en deux étapes :

1. en fonction de l'information disponible sur \mathbf{e} , établir un *critère* d'ajustement de $\boldsymbol{\eta}(\mathbf{x})$ au vecteur \mathbf{y} ,
2. déterminer toutes les valeurs numériques de \mathbf{x} satisfaisant ce critère.

Le choix du critère d'ajustement incombe à l'expérimentateur. Pour ce faire, celui-ci doit, dans un premier temps, analyser le type d'information dont il dispose sur les grandeurs incertaines \mathbf{e}_η et \mathbf{e}_y , puis spécifier le type de résultat qu'il souhaite obtenir.

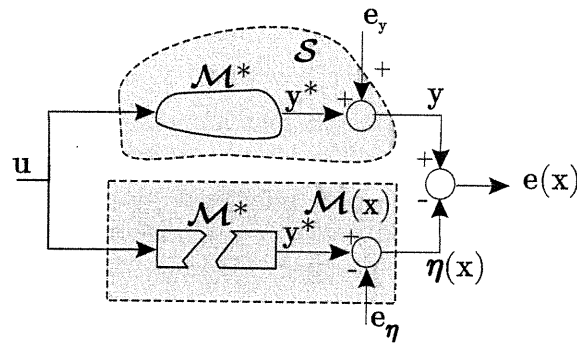


Figure 4-2: Représentation de l'erreur de sortie e , de l'erreur de modélisation e_η et de l'erreur de mesure e_y

4.1.1 Erreur de mesure

L'erreur de mesure e_y est une grandeur incertaine de nature expérimentale. Le choix d'une modélisation pertinente de e_y s'appuie donc sur les possibilités offertes par l'expérience et les contraintes expérimentales. Les organismes officiels de métrologie distinguent deux types d'erreurs de mesure [CET98] :

- celles pour lesquelles des propriétés statistiques peuvent être établies (dites de *type A*), qu'on choisira de décrire par une variable aléatoire (approche stochastique)
- celles dont on n'a pas pu établir de propriétés statistiques (dites de *type B*), pour lesquelles nous montrerons l'avantage d'une description par intervalles.

4.1.1.1 Variable stochastique

Cette modélisation est adaptée dans trois cas : quand l'expérience est répétable, quand on dispose d'information *a priori* sur la nature des perturbations qui la justifie, ou quand la notice technique de l'appareil de mesure fournit une information appropriée.

Expérience répétable Lorsque l'expérience est *répétable*, l'expérimentateur peut, s'il dispose du temps et des moyens nécessaires, effectuer des séries de mesure dans des conditions expérimentales identiques (mêmes entrées u , mêmes conditions initiales). Il lui est alors possible d'établir, à partir des séries de mesure obtenues, des propriétés statistiques de la variabilité de la grandeur y , et de choisir une densité de probabilité $\mathcal{P}_y(\cdot)$ pour y . Il obtient alors la densité de probabilité $\mathcal{P}(\cdot)$ de e_y par une simple translation de $\mathcal{P}_y(\cdot)$, puisque le vecteur de sortie y^* d'un système déterministe invariant dans le temps \mathcal{M}^* est déterministe :

$$\mathcal{P}(e_y) = \mathcal{P}_y(y - y^*).$$

On écrira alors généralement e_y sous la forme de la somme de $m(e_y)$ (*erreur systématique*) et d'une variable aléatoire centrée de moments supérieurs identiques à ceux de y .

Nature de la perturbation Une perturbation liée aux phénomènes physiques internes dans les composants électroniques du capteur participe à l'erreur de mesure. Cette perturbation, appelée *bruit de fond*, est souvent décrite par un modèle faisant intervenir des générateurs pour

lesquels on calcule une valeur efficace. Parmi les bruits les plus étudiés, nous pouvons citer par exemple, le bruit thermique ou bruit de Johnson, et le bruit de grenaille ou de scintillation (bruit en $\frac{1}{f}$).

Incertitudes de type A Lorsque le constructeur a choisi de modéliser chaque composante e_{yi} de l'erreur de mesure de l'appareil à partir d'une variable aléatoire, il spécifie l'écart-type de e_{yi} dans la notice technique. Lorsque e_{yi} est modélisée par une variable gaussienne, comme c'est fréquemment le cas, il est alors possible de reconstruire sa densité de probabilité à partir de la notice technique de l'appareil. Mais de façon générale, la règle consiste ensuite à sommer les variances des diverses erreurs de mesure inhérentes à la chaîne de mesure, ce qui revient à supposer – sans toujours le justifier – qu'elles suivent toutes des lois normales indépendantes.

Les organismes de métrologie recommandent de spécifier une plage d'appartenance de l'erreur de mesure, à l'aide d'une grandeur appelée *incertitude de mesure*. Nous montrons dans le paragraphe suivant qu'un autre type de modélisation peut alors être plus adapté.

4.1.1.2 Variable intervalle

On appelle *incertitude de mesure* δe_y la valeur absolue de l'erreur de mesure maximale commise lors d'une mesure :

$$-\delta e_y \leq e_y \leq \delta e_y, \quad (4.2)$$

et c'est souvent la seule information disponible sur l'erreur de mesure. Remarquons que

$$y - \delta e_y \leq y^* \leq y + \delta e_y.$$

La connaissance de δe_y permet de décrire la grandeur incertaine e_y à l'aide d'un support borné. Pour cette raison, nous parlerons de *contexte à erreurs bornées* (voir [Sch68], [PLW90], [Nor94], [Nor95] et [MNPLW96] ainsi que les nombreuses références qu'ils contiennent). Cette formulation apparaît naturellement dans de nombreux cas. Ainsi, l'*erreur de relevé* sur une abaque ou de lecture sur un cadran est une grandeur naturellement bornée puisque l'erreur maximale commise correspond à la moitié de l'écart entre deux graduations successives. De même, il est généralement possible de proposer un majorant réaliste des bornes des erreurs de mesure décrites par une incertitude de mesure de type B (erreur de manipulation, erreur de l'expérimentateur...).

Remarquons que l'incertitude de mesure de type A, définie en métrologie comme un écart-type, ne satisfait pas l'équation (4.2). Un exemple flagrant est celui de l'*erreur de quantification*, fréquemment modélisée à partir d'une variable aléatoire de densité de probabilité uniforme, comme l'illustre l'exemple suivant.

Exemple 4.1.1 (Erreur de quantification) *Considérons un dispositif de mesure incluant un convertisseur analogique-numérique de pas de quantification q , et calculons la composante δe_q de l'incertitude de mesure due à l'erreur de quantification e_q commise. Par définition du pas de quantification q , e_q est une grandeur incertaine qui prend ses valeurs dans l'intervalle $[-\frac{q}{2}, \frac{q}{2}]$. Modéliser cette grandeur par une variable intervalle semble alors très naturel et nous obtenons alors une incertitude $\delta^B e_q$ de type B telle que*

$$\delta^B e_q = \frac{q}{2}$$

de sorte que (4.2) est satisfaite. Une modélisation sous forme de variable aléatoire de densité de

probabilité uniforme à support borné (voir la figure 4-3) est cependant très fréquemment utilisée. Dans ce cas, on calcule $\delta^A e_q$ comme l'écart-type de la distribution de e_q , d'où

$$\delta^A e_q = \frac{q}{\sqrt{12}},$$

ce qui ne satisfait plus (4.2).

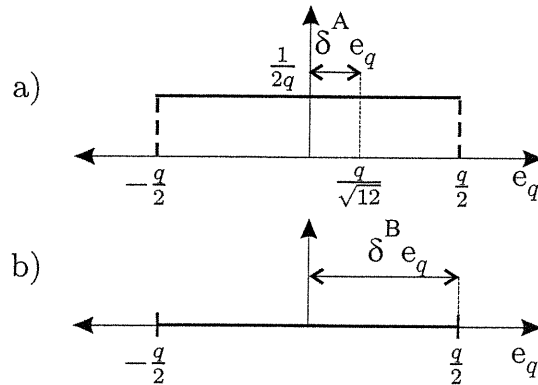


Figure 4-3: Deux modélisations possibles de l'erreur de quantification, dans l'approche stochastique (cas a), l'incertitude de mesure est calculée comme l'écart-type de la distribution uniforme, tandis que l'incertitude de type B vaut le rayon de son support (cas b).

Nous ferons par la suite l'hypothèse que nous savons transformer toute incertitude de mesure δe_y fournie par la notice technique en une grandeur satisfaisant (4.2). Dans le cas d'une incertitude $\delta^A e_y$ de type A, cela peut conduire à considérer le support de la distribution ou un support convenablement tronqué dans le cas d'une distribution à support telle que la distribution gaussienne.

4.1.2 Erreur de modèle

L'identification de paramètres d'un système physique peut se décomposer en deux étapes :

- la construction d'un modèle, qui suppose la connaissance d'un ensemble de lois qui décrivent les phénomènes siégeant au sein du système, à exprimer sous la forme d'équations mathématiques ou d'algorithmes,
- la détermination de ses caractéristiques au travers d'expériences pertinentes au sens où elles mettront en évidence leur influence.

Ces deux étapes soulignent combien le modèle est une "représentation structurée de notre connaissance du système réel dans un cadre d'étude spécifique" [Kim98]. L'erreur de modèle exprime donc notre connaissance nécessairement imparfaite du système réel qu'aucun modèle mathématique fini ne peut décrire exactement, et cela quelle que soit la quantité de données disponible. En raison de l'existence intrinsèque de cette incertitude structurelle dans tout schéma de modélisation, la quête d'un modèle unique a été rapidement abandonnée, au profit d'une nouvelle approche qui propose de décrire l'estimée comme un ensemble. Cette approche est au cœur de la commande robuste, qui depuis les années 1970, cherche à synthétiser des correcteurs dont

les performances satisfont un cahier des charges même en présence d'incertitudes de modélisation [DYA97], [GNS92]. De l'extension de cette approche aux problèmes de l'identification, a émergé la notion d'estimateur à erreurs bornées (voir [MNPLW96], [Nor94], [Nor95] et les références incluses).

Les deux types de modélisation de l'erreur de sortie \mathbf{e} les plus couramment envisagés sont donc la modélisation stochastique et la modélisation intervalle. Nous choisissons le même type de modélisation pour \mathbf{x} .

4.2 Modélisation d'une grandeur incertaine

Suivant le type d'information dont dispose l'expérimentateur, celui-ci choisit un type de description de l'erreur de sortie \mathbf{e} et donc du vecteur de paramètres \mathbf{x} qu'il cherche à estimer. Très fréquemment, une description stochastique est sélectionnée, car elle permet de ramener le problème d'estimation à un problème d'optimisation d'une fonction coût. Malheureusement, ceci ne va pas sans conséquence, comme nous le montrons dans le paragraphe suivant; non seulement des limitations numériques dégradent la qualité du résultat et entravent le processus de décision, mais d'autres types de problèmes, tels que la non-identifiabilité du modèle mathématique en construction, peuvent pousser l'expérimentateur à préférer une description de l'estimée sous forme ensembliste plutôt que ponctuelle.

4.2.1 Approche stochastique

Dans ce cas, on choisit de décrire également \mathbf{x} sous la forme d'une variable stochastique, dont on cherche les caractéristiques statistiques.

4.2.1.1 Optimisation d'une fonction coût

Le problème d'estimation de \mathbf{x} se traduit alors par la donnée d'une fonction mathématique, dite fonction coût $j(\mathbf{x})$, à optimiser (techniques du maximum de vraisemblance, du maximum *a posteriori*...). Cette optimisation permet d'associer une valeur numérique à \mathbf{x} , que nous appellerons son estimée $\hat{\mathbf{x}}$, et très souvent un intervalle de confiance associé à cette estimée. On espère que l'estimée $\hat{\mathbf{x}}$ de \mathbf{x} correspond à un optimiseur global de $j(\cdot)$, qu'on espère de plus unique.

Nous précisons les limites de cette approche dans le paragraphe suivant.

4.2.1.2 Algorithmes d'optimisation

Une fois l'expression de la fonction coût $j(\cdot)$ choisie, encore reste-t-il à sélectionner un algorithme d'optimisation. Suivant les particularités de j , les algorithmes disponibles se révèlent d'efficacités différentes, et malheureusement laissent encore souvent à désirer. Afin de montrer leur domaine d'utilisation, nous les décrivons par degré d'ambition croissant.

Les *méthodes ponctuelles locales* sont de loin les plus connues. Leur principal inconvénient est le caractère local des algorithmes qui en découlent (Gauss-Newton, quasi-Newton, gradients conjugués...). La valeur numérique obtenue pour $\hat{\mathbf{x}}$ peut donc dépendre fortement de la valeur initiale choisie pour l'algorithme. De plus, ces algorithmes itératifs, peuvent s'arrêter sans même avoir atteint la solution locale, en raison d'un mauvais conditionnement numérique.

Les méthodes d'*optimisation globale* par exploration aléatoire de l'espace de recherche \mathbb{X} [Zhi91] évitent en partie ces inconvénients. Le recuit simulé [PT95], les algorithmes génétiques [Whi93] ou la recherche aléatoire adaptative [WP97], ont ainsi pour ambition de contourner les problèmes d'initialisation. Cependant, il n'y a aucune garantie que ces méthodes fourniront de

bonnes approximations du minimum global \hat{j} et du minimiseur global associé après un nombre fini d'itérations.

Enfin, le problème lié à la description de l'estimée sous une forme ponctuelle n'est pas des moindres. Puisque ces deux types de méthodes n'autorisent pour tout résultat qu'une seule valeur numérique, elles ne fournissent des résultats satisfaisants que lorsque j possède un seul minimiseur global. Ainsi, si le modèle, de par sa structure, admet deux valeurs de \mathbf{x} équivalentes, au mieux l'estimée fournie correspondra à l'une de ces deux valeurs¹.

L'ensemble de ces problèmes est résumé dans [IV99], sans toutefois qu'une solution soit proposée. Afin d'obtenir une caractérisation garantie de tous les optimiseurs globaux d'une fonction coût j , il faut avoir recours à l'*optimisation globale déterministe* (voir [HT90], [Han79]) et l'application à un problème d'estimation de paramètres sur des modèles comparables à ceux traités ici par maximum de vraisemblance [KW98]).

Même si grâce à ces dernières techniques, on peut théoriquement obtenir l'ensemble des minimiseurs globaux d'une fonction coût, la réponse fournie au problème d'estimation reste incomplète, puisque certaines valeurs numériques du vecteur de paramètres qui ne correspondent pas à des optimiseurs globaux peuvent être jugés acceptables compte tenu de l'incertitude de la mesure expérimentale. Ceci nous ramène tout naturellement à la famille des *estimateurs à erreurs bornées*.

4.2.2 Estimation à erreurs bornées

Ici l'estimée n'est plus décrite sous la forme d'une valeur numérique et éventuellement d'un intervalle de confiance, mais directement sous la forme d'un ensemble (voir en particulier [MNPLW96], [Wal90], [Nor94], [Nor95] et les références incluses). Nous présentons ici la formulation ensembliste de cette approche, et situons les techniques par intervalles que nous utiliserons dans tout ce document.

Si l'erreur \mathbf{e} appartient au vecteur intervalle $[\mathbf{e}] = [-\delta\mathbf{e}, \delta\mathbf{e}] \subset \mathbb{R}^n$, alors le vecteur $\boldsymbol{\eta}(\mathbf{x})$ doit appartenir au vecteur intervalle

$$[\mathbf{y}] = \mathbf{y} + [-\delta\mathbf{e}, \delta\mathbf{e}].$$

Une estimée $\hat{\mathbf{x}}$ du vecteur de paramètres \mathbf{x} est alors dite *acceptable* si et seulement si $\boldsymbol{\eta}(\hat{\mathbf{x}}) \in [\mathbf{y}]$. Il s'agit ainsi de déterminer l'ensemble \mathbb{S} de tous les $\hat{\mathbf{x}}$ acceptables appartenant à un espace de recherche connu *a priori* \mathbb{X}

$$\mathbb{S} = \{\mathbf{x} \in \mathbb{X} \mid \boldsymbol{\eta}(\mathbf{x}) \in [\mathbf{y}]\}. \quad (4.3)$$

Comme (4.3) se réécrit très facilement, à l'aide des notations ensemblistes du chapitre 1, sous la forme

$$\mathbb{S} = \boldsymbol{\eta}^{-1}([\mathbf{y}]) \cap \mathbb{X}, \quad (4.4)$$

on parle alors de problème d'*inversion ensembliste*. Remarquons qu'en posant, pour i variant de 1 à n :

$$\begin{cases} f_{2i-1}(\mathbf{x}) = y_i - \eta_i(\mathbf{x}), \\ f_{2i}(\mathbf{x}) = \eta_i(\mathbf{x}) - \bar{y}_i, \end{cases}$$

¹Voir le modèle à deux exponentielles traité au paragraphe 4.3.3

il est également possible de réécrire (4.4) sous la forme

$$\mathbb{S} = \{\mathbf{x} \in \mathbb{X} \mid \mathbf{f}(\mathbf{x}) \leq \mathbf{0}\} = \mathbf{f}^{-1}(]-\infty, 0]^{2n}) \cap \mathbb{X}.$$

On peut donc encadrer l'ensemble \mathbb{S} défini par (4.4) de façon garantie, à l'aide des outils de la famille de SIVIA présentés dans le chapitre 2.

Comparativement aux approches statistiques classiques, l'estimation à erreurs bornées présente plusieurs avantages, en plus de prendre naturellement en compte l'erreur de modélisation.

Les outils de l'analyse par intervalles sont plus efficaces dans ce contexte que dans les problèmes d'optimisation de fonction coût, en raison d'un nombre nettement inférieur d'occurrences des paramètres (voir l'exemple 4.2.1).

Exemple 4.2.1 *Soit à estimer le paramètre scalaire x à partir du modèle dont la sortie vectorielle s'écrit $\eta_i(x) = x \exp x$, pour $i = 1, \dots, n$. La fonction d'inclusion naturelle de η s'écrit $[\eta_i](x) = [x] \exp[x]$ et fait intervenir $N_0 = 2$ occurrences de x , tandis que la fonction d'inclusion naturelle $[j]$ de la fonction de coût quadratique présente nN_0 occurrences de x puisqu'elle s'écrit $[j](x) = \sum_{i=1, \dots, n} ([x] \exp[x] - y_i)^2$.*

Enfin, à partir des deux approximations intérieure et extérieure $\underline{\mathbb{S}}$ et $\overline{\mathbb{S}}$ de \mathbb{S} , nous pourrions conclure de façon garantie dans les trois cas suivants.

Propriété 4.2.1 *Si $\overline{\mathbb{S}} = \emptyset$: \mathbb{S} est alors vide; le modèle proposé et les données ne sont pas compatibles (modèle invalide ou présence d'au moins une donnée aberrante...).*

Propriété 4.2.2 *Si $\overline{\mathbb{S}} \neq \emptyset$: alors la projection $\pi_i(\overline{\mathbb{S}})$ est une borne supérieure de l'incertitude d'estimation associée au paramètre x_i .*

Propriété 4.2.3 *Si $\underline{\mathbb{S}} \neq \emptyset$: alors \mathbb{S} n'est pas vide; il existe au moins une valeur du vecteur de paramètres qui rend le modèle compatible avec les données recueillies.*

La suite de ce chapitre s'articule autour de deux applications, d'objectif et de contexte expérimental différents. Nous avons choisi de souligner les avantages et les limites de l'identification ensembliste garantie par rapport aux techniques classiques sur la première application, traitée à la section 4.3. C'est pourquoi nous la proposons sous une forme très détaillée afin de familiariser le lecteur avec l'utilisation des techniques ensemblistes. Elle correspond à une collaboration avec F. Berthier du Laboratoire d'Etude des Matériaux Hors-Equilibre (LEMHE), à l'Université de Paris-Sud, qui a donné lieu à la publication [BBJ⁺01b]. Il s'agit d'estimer les constantes cinétiques de mécanismes de transfert électronique dans le but de caractériser le matériau soumis à l'étude. Dans ce domaine où la construction de mécanismes s'appuie sur la *non-invalidation*² expérimentale de modèles, l'intérêt des techniques garanties pourra être clairement mis en évidence. Puisque l'environnement de mesure y est complètement maîtrisé, seules les constantes cinétiques et les données expérimentales sont décrites par des grandeurs incertaines. Nous dirons que l'environnement de mesure est gouverné par des *facteurs expérimentaux connus*. Nous comparerons les résultats obtenus avec ceux des méthodes d'optimisation ponctuelles classiques, très fréquemment rencontrées dans la littérature [Bou86], [OA96], afin de souligner que dans ce cas

- le choix d'un contexte à erreurs bornées s'impose naturellement,

²Nous préférons ce terme barbare à celui de validation, puisque selon nous l'identification n'est en aucune mesure capable de valider un modèle, elle ne peut qu'en invalider.

- il est possible d'obtenir l'ensemble des valeurs acceptables,
- seule l'approche ensembliste permet d'obtenir l'ensemble des mécanismes réactionnels, l'utilisation d'un algorithme ponctuel classique étudie des mécanismes concurrents éventuellement acceptables.

Dans un second temps (section 4.4), nous considérerons l'estimation de paramètres thermo-physiques d'un matériau. Là, l'objectif est différent : il s'agit d'évaluer les performances d'une chaîne de mesure prototype, et en particulier de caractériser son incertitude de mesure. Ce processus de mesure de paramètres thermo-physiques a été proposé par l'équipe du Centre d'Etudes et de Recherche en Thermique, Energétique et Systèmes (CERTES), à l'Université de Paris XII – Val de Marne. Pour étudier ces performances, nous cherchons à identifier de façon garantie les grandeurs caractéristiques d'un échantillon de PVC dont nous savons *a priori* qu'elles appartiennent à un domaine \mathbb{X} . Comme il s'agit de caractériser cette chaîne de mesure, l'ensemble des facteurs expérimentaux est incertain, et il s'agit de prendre en compte cette incertitude perturbatrice dans la caractérisation. Ce problème s'exprime très bien dans un contexte à erreurs bornées. Nous montrerons que les techniques précédemment décrites permettent d'identifier de façon garantie les paramètres recherchés dans ce contexte incertain, et de fournir une estimation garantie de l'incertitude de mesure.

Enfin, nous présenterons une technique simple d'analyse des données recueillies au paragraphe 4.5.

4.3 Facteurs expérimentaux connus

Les deux objectifs de l'identification paramétrique apparaissent très nettement en électrochimie, où il est à la fois nécessaire de valider (ou plutôt de ne pas invalider) un mécanisme réactionnel proposé, et d'identifier des paramètres physiques caractéristiques.

La construction de modèles s'articule en deux temps. Une étude théorique propose des mécanismes réactionnels faisant intervenir des entités pouvant exister durant la réaction. On en déduit alors des modèles de comportement du système. Dans un second temps, une étude expérimentale incluant une démarche d'identification permet, parmi l'ensemble des modèles proposés, d'en invalider certains. L'électrochimiste choisit alors de conserver l'ensemble des modèles restants ou d'en sélectionner un particulier, qu'il trouvera par exemple plus simple ou plus aisément manipulable.

Une fois ce modèle sélectionné, il s'agit d'estimer les valeurs prises par les grandeurs physiques intervenant dans ce modèle, afin de caractériser le matériau soumis à l'étude.

Nous envisageons ici l'étude de deux modèles très fréquents en électrochimie qui mettent en évidence l'ensemble des problèmes cités au paragraphe précédent. Pour cela, nous comparons les résultats ponctuels obtenus par les méthodes classiques et ceux issus des techniques garanties d'inversion ensembliste. Nous allons montrer tout particulièrement l'intérêt de manipuler des ensembles dans ce contexte, puisque nous pourrions déduire des conclusions théoriques fiables là où des estimateurs classiques échouent et faussent le processus de décision.

4.3.1 Identification paramétrique en électrochimie

La caractérisation électrochimique des métaux s'effectue par l'estimation de constantes cinétiques caractéristiques de mécanismes réactionnels décrivant des phénomènes d'oxydo-réduction dans lesquels ces métaux sont impliqués. Comme ces constantes cinétiques ne sont pas directement accessibles à la mesure, elles sont généralement estimées à partir des observées que sont

la tension d'électrode E et la densité de courant i établi dans le circuit fermé par les électrodes (voir la figure 4-4). La relation entre E et i est modélisée à partir des lois de la cinétique réactionnelle appliquées au mécanisme sélectionné, et fait alors intervenir ces constantes sous forme de paramètres. Parmi tous les mécanismes envisageables, les transferts mono- ou bi-électroniques sont de loin les plus fréquents, et les plus abondamment étudiés dans la littérature [DGM96]. En effet, le mécanisme est alors particulièrement simple, et il est très facile d'obtenir la solution quasi-stationnaire décrivant le processus. Ce type de solution, en forme de somme d'exponentielles, est très largement rencontré dans tous les domaines où des modèles mathématiques sont construits à partir d'observations expérimentales. De plus, l'identification des paramètres impliqués dans une somme d'exponentielles est un problème dont la difficulté est bien connue [IV99]. Toutes ces raisons nous ont motivés à les choisir en tant que banc-test de SIVIA.

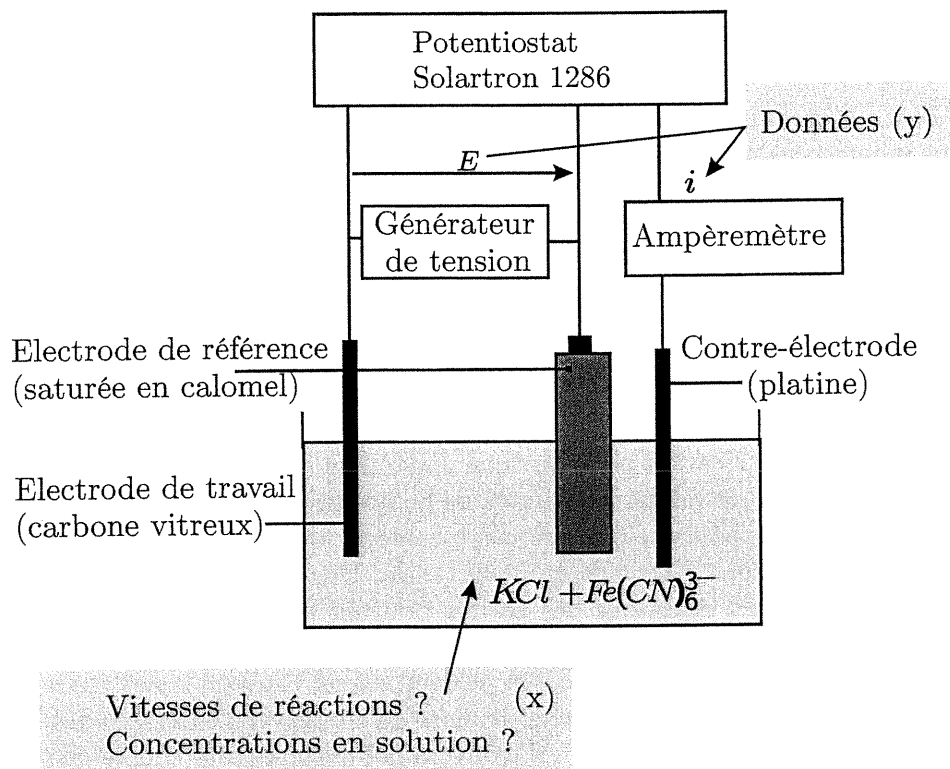


Figure 4-4: Dispositif expérimental utilisé en électrochimie (cas de la réaction d'oxydo-réduction du fer)

En raison

- d'un manque d'information pour définir des propriétés statistiques au bruit entachant les mesures,
- ou du comportement éventuellement gaussien de l'erreur de mesure, qui rend équivalentes l'estimation au sens du maximum de vraisemblance et la minimisation d'une fonction coût quadratique,
- ou d'une utilisation abusive du théorème central limite,

l'approche la plus fréquemment rencontrée en électrochimie est la minimisation de l'erreur quadratique $\|y - \eta(x)\|^2$ ([Bou86], [OA96], [DB99], [VA98] et [Sal98]). Cependant, les électrodes

de mesure ne sont pas chimiquement inertes mais subissent une usure au cours de la réaction. Ceci a deux conséquences. D'une part, l'expérience n'est pas répétable, puisqu'il est nécessaire de changer d'électrodes entre deux séries de mesures. Comme de plus aucun *a priori* (fourni éventuellement par la notice technique du constructeur) ne nous permet d'affirmer le caractère gaussien ou non de l'erreur de mesure e_y , nous ne disposons pas d'informations statistiques sur e_y . D'autre part, durant la même série de mesures, le système n'est pas stationnaire; il faudrait donc tenir compte de l'évolution de l'état de surface de l'électrode de travail (pour plus d'informations, nous renvoyons le lecteur à [Webm].) Comme aucun modèle n'est à notre disposition pour quantifier cette dérive, l'erreur de modélisation commise peut être importante. L'ensemble des informations disponibles se résume donc à :

- la notice technique des appareils SOLARTRON 1286 fournie par le constructeur, qui donne l'ensemble des incertitudes de mesure associées à chaque élément constitutif de la chaîne de mesure, sans préciser la nature des erreurs de mesure [SAM85].
- les données recueillies au cours de l'expérience.

Nous cherchons alors à caractériser l'ensemble \mathbb{S} satisfaisant (4.3), où l'espace de recherche \mathbb{X} est déterminé à partir de la connaissance de l'expert et de la définition du modèle.

4.3.2 Mécanisme rédox : une exponentielle

Considérons le cas d'un échange mono-électronique décrit par le mécanisme rédox:



qui peut par exemple, décrire la réduction sans modification de géométrie de coordination de l'ion complexe cyané du Fer $Fe(CN)_6^{3-}$. Dans (4.5), O et R représentent les espèces oxydante et réductrice d'un couple rédox dissoutes dans un solvant. En régime quasi-statique, l'évolution de la densité de courant de Faraday j_f est égale à celle de la densité du courant i qui traverse l'électrode, qui est une grandeur observable. Elle s'écrit, en fonction de la tension d'électrode normalisée ξ , également observable [DGM96], [Bra99] :

$$j_f = Fk^\circ \frac{c_{R,\text{init}} \exp(\alpha_o \xi) - c_{O,\text{init}} \exp(-\alpha_r \xi)}{1 + k^\circ \left(\frac{\exp(\alpha_o \xi)}{m_R} + \frac{\exp(-\alpha_r \xi)}{m_O} \right)}, \quad (4.6)$$

$$\text{avec } k^\circ = k_o^{\alpha_o} k_r^{\alpha_r} \text{ et } \xi = f(E - E^\circ). \quad (4.7)$$

Dans ces équations, k° est la constante cinétique de transfert électronique standard et E° le potentiel standard; α_o et α_r sont les facteurs de symétrie de la réaction, et k_o et k_r les constantes cinétiques associées; m_O et m_R sont les constantes de transfert de masse, F est la constante de Faraday, $f = \frac{F}{RT}$, où T est la température absolue du système considéré, et R est la constante des gaz parfaits; enfin, $c_{O,\text{init}}$ et $c_{R,\text{init}}$ sont les concentrations initiales en O et R.

En supposant m_O , m_R et E° connus, nous définissons le vecteur de paramètres

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{k^\circ} \\ \alpha_o \end{pmatrix}, \quad (4.8)$$

et la sortie du système

$$y^1(t) = \frac{F}{j_f}(c_{R,\text{init}} - c_{O,\text{init}} \exp(-t)) - \frac{1}{m_R} - \frac{\exp(-t)}{m_O}, \quad (4.9)$$

$$\text{avec } t = \xi. \quad (4.10)$$

obtenue à partir de (4.6) en regroupant tous les termes ne faisant intervenir que des grandeurs certaines. Dans le but de minimiser le nombre d'occurrences des composantes de \mathbf{x} , nous écrivons dans une première étape le modèle η^1 sous la forme

$$\eta^1(\mathbf{x}, t) = x_1 \exp(-x_2 t). \quad (4.11)$$

Il apparaît très nettement qu'à condition de disposer d'au moins deux données, dans un contexte non bruité, une seule valeur du vecteur de paramètres \mathbf{x} peut fournir le vecteur de données escompté.

Comme $\eta^1(\cdot, t)$ est une combinaison de fonctions monotones, la fonction d'inclusion naturelle $[\eta^1]$ définie par

$$[\eta^1]([x], t) = [x_1] \exp(-[x_2] t)$$

est minimale (cf théorème 2.2.3), et le contracteur $\mathcal{C}_{1\uparrow}$ utilisé dans SIVIAP sera optimal. Dans un premier temps, nous étudions le comportement de l'ensemble \mathbb{S} en fonction de l'incertitude de mesure sur des données simulées. Nous comparons ensuite les résultats obtenus par SIVIA et SIVIAP. Enfin, nous présentons les résultats obtenus dans le cas de données réelles.

4.3.2.1 Données simulées

Afin de souligner l'influence de l'incertitude sur la solution, étudions d'abord le comportement de \mathbb{S} en fonction du volume des données intervalles $[\mathbf{y}]$ sur des données simulées. L'erreur de sortie se réduit ici à l'erreur de mesure, que nous simulons par un bruit blanc additif de densité de probabilité uniforme sur $b[-1, 1]$ où $b = 0.01 \text{ mA/cm}^{-2}$. La vraie valeur du vecteur de paramètres est choisie égale à

$$\mathbf{x}^* = (100 \ 0.4)^T.$$

L'espace de recherche fixé par l'expérimentateur vaut $\mathbb{X} = [10^{-6}, 10^6] \times [0, 1]$. La figure 4-5 présente les ensembles $\underline{\mathbb{S}}$ et $\bar{\mathbb{S}}$ obtenus lorsque l'on suppose que l'erreur de mesure sur j_f est une grandeur incertaine de support égal à: a) $10b[-1, 1]$ et b) $b[-1, 1]$.

La valeur simulée \mathbf{x}^* du vecteur de paramètres \mathbf{x} est bien un élément de $\underline{\mathbb{S}} \cup \Delta\mathbb{S}$, et même $\mathbf{x}^* \in \underline{\mathbb{S}}$. La version de SIVIA utilisée ici est dans sa forme la plus simple, sans contracteur, et ne cherche donc pas à minimiser le temps de calcul t_c , qui dépend de la tolérance ε . Le nombre de bisections n_b et le temps de calcul t_c croissent lorsque ε décroît, selon une loi en ε^{-1} [Jau94], comme le montre la figure 4-6.

L'utilisation de contracteurs permet d'améliorer t_c en réduisant n_b . Ainsi, il devient possible d'obtenir une bonne estimée ($\varepsilon = 0.001$), en moins de 5 secondes sur un Pentium II 350 MHz, en utilisant le contracteur $\mathcal{C}_{1\uparrow}$ dans SIVIAP, qui est ici optimal.

D'après la propriété 4.2.2, l'intervalle d'incertitude de chaque paramètre est obtenu en projetant $\bar{\mathbb{S}}$ sur les axes de l'espace des paramètres:

$$\text{a) } x_1 \in [63.489, 141.457], \quad x_2 \in [0.085, 0.668].$$

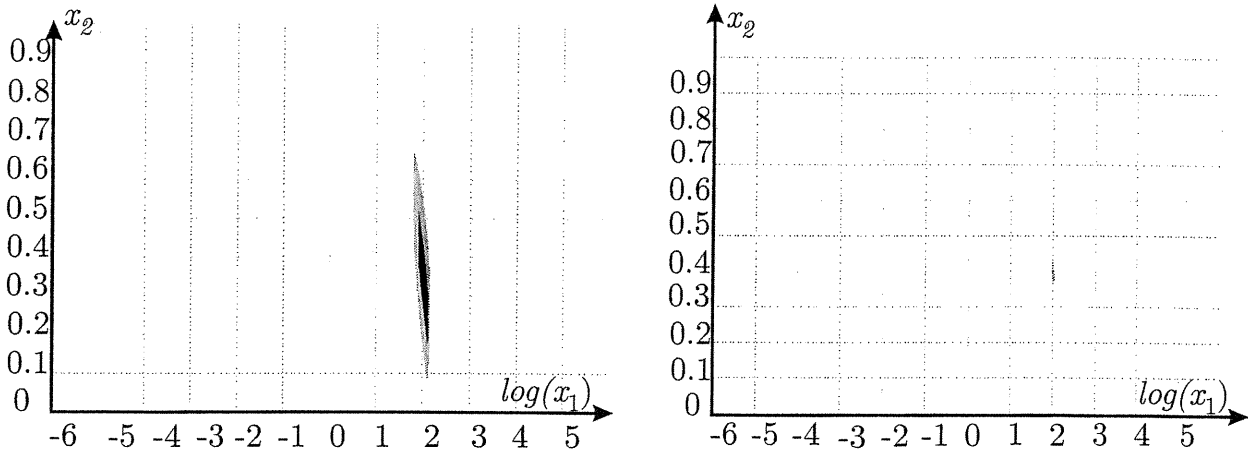


Figure 4-5: Approximation intérieure \underline{S} (en noir) et couche d'incertitude ΔS (en gris) obtenues par SIVIA pour les données simulées du modèle rédox. Le cas a) (à gauche) correspond à des bornes de l'incertitude sur j_f surévaluées d'un facteur 10; le cas b) (à droite) correspond à une incertitude sur j_f correctement évaluée.

b) $x_1 \in [96.141, 103.908]$, $x_2 \in [0.370, 0.429]$.

Pour des bornes sous-évaluées (erreur de mesure supposée de support $[-1, 1] \cdot 10^{-3}$), \bar{S} est vide: SIVIA a éliminé l'ensemble des \mathbf{x} de \mathbb{X} , ce qui permet de prouver qu'il n'existe aucune estimée compatible avec l'ensemble des hypothèses et des données (propriété 4.2.1). Il faut souligner l'importance de ce résultat, que ne peut obtenir aucun estimateur ponctuel traditionnel.

4.3.2.2 Données réelles

Les données réelles ont été collectées sur le système $\text{Fe}(\text{CN})_6^{3-}/\text{Fe}(\text{CN})_6^{4-}$ dans une solution KCl (0.5M), $\text{Fe}(\text{CN})_6^{3-}$ (10^{-2}M) de coefficient de viscosité ν , dans laquelle est plongée une électrode tournant à la vitesse ω . La chaîne de mesure était constituée d'un potentiostat SOLARTRON Schlumberger 1286, géré par le logiciel d'acquisition SAMDIE [DGM91]. L'ensemble des grandeurs expérimentales connues est rassemblé ci-dessous :

$$\begin{aligned} c_{\text{O,init}} &= 10^{-5} \text{ mol.cm}^{-3}, \quad c_{\text{R,init}} = 0 \text{ mol.cm}^{-3}, \quad \nu = 0.977 \times 10^{-2} \text{ cm}^2/\text{sec}, \\ \omega &= 2000 \times \frac{\pi}{30} \text{ rad/s}, \quad E^\circ = 0.223 \text{ V/SHE}. \end{aligned}$$

Les estimées sont classiquement obtenues en électrochimie par le tracé des lignes de Tafel, ce qui revient à ajuster les asymptotes. Cependant, on obtient ainsi un modèle qui peut sortir du gabarit imposé par les contraintes expérimentales puisque cette méthode n'en tient pas compte. De plus, seule l'information fournie par les asymptotes est prise en compte, au détriment de celle contenue dans la zone intermédiaire. L'utilisation de techniques garanties nous permet d'éviter ces ajustements locaux. L'ensemble \bar{S} obtenu par SIVIA est vide: une des hypothèses est donc invalidée (propriété 4.2.1). En effet, pour des grandes valeurs du potentiel d'électrode, la densité de courant de Faraday relevée est faible mais positive (0.46 mA.cm^{-2}), ce qui est en contradiction avec l'hypothèse $c_{\text{R,init}} = 0$. Nous avons donc reformulé le modèle en incorporant $c_{\text{R,init}}$ dans le

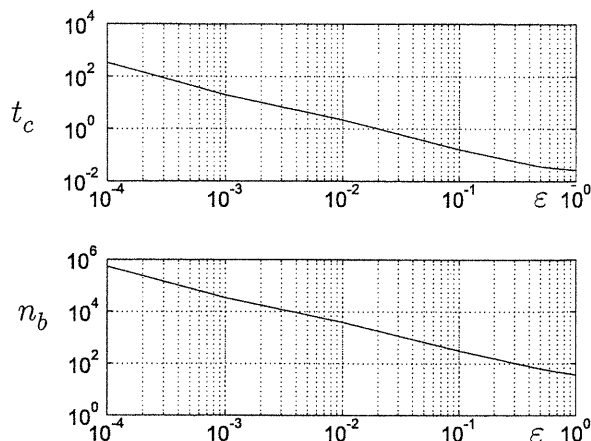


Figure 4-6: Comportement de SIVIA en fonction du paramètre ε à régler par l'utilisateur; en haut, temps de calcul t_c (en secondes), en bas nombre de bisections effectuées n_b

vecteur de paramètres à estimer:

$$\mathbf{x} = \left(\frac{1}{k^\circ} \quad \alpha_o \quad c_{R,\text{init}} \right)^T.$$

La sortie du modèle devient

$$\eta^2(\mathbf{x}, t) = \frac{x_3 - c_{O,\text{init}} \exp(-t)}{x_1 \exp(-x_2 t) + \frac{1}{m_o} + \frac{1}{m_R} \exp(-t)}. \quad (4.12)$$

On obtient alors un ensemble solution non vide, inclus dans le pavé:

$$[263.652, 299.212] \times [0.581, 0.632] \times [2.039, 2.191]10^{-7},$$

ce qui entraîne, d'après la propriété 4.2.2

$$k^\circ \in [3.3, 3.8] 10^{-3}.$$

Par ailleurs, l'ensemble intérieur $\underline{\mathbb{S}}$ n'est pas vide, donc \mathbb{S} n'est pas vide (cf propriété 4.2.3). Enfin, nous pouvons confirmer que l'hypothèse $c_{R,\text{init}} = 0$ était incompatible avec nos autres hypothèses, puisque $c_{R,\text{init}} \in [2.039, 2.191]10^{-7}$. Soulignons bien que les méthodes asymptotiques ne permettent pas cette estimation de $c_{R,\text{init}}$ sur de telles valeurs expérimentales. En particulier la solution obtenue par moindres carrés à partir d'une procédure utilisant la fonction *NonLinearRegress* de Mathematica (fondée sur la résolution de Newton) est nettement en dehors du gabarit, surtout dans la zone de régime mixte (figure 4-7).

Nous avons supposé jusqu'à présent que la valeur numérique des coefficients de diffusion et du potentiel standard était relativement bien connue. Supposons maintenant ces coefficients incertains. Le vecteur de paramètres étendu devient alors:

$$\mathbf{x} = \left(\frac{1}{k^\circ} \quad \alpha_o \quad c_{R,\text{init}} \quad \frac{1}{m_o} \quad \frac{1}{m_R} \right)^T.$$

Les résultats obtenus avec SIVIA dans ce cas sont présentés ci-dessous:

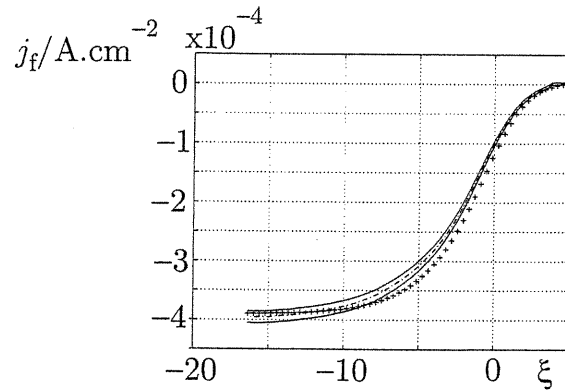


Figure 4-7: Représentation de la sortie de modèle $\eta^2(\hat{\mathbf{x}}, t)$ obtenue à partir d'un estimateur ponctuel (+), gabarit $[\mathbf{y}](t)$ (trait continu) imposé par les données expérimentales \mathbf{y} (- -); l'estimateur ponctuel ne respecte pas le gabarit

i	x_i	\mathbb{X}_i	$[\hat{x}_i]$
1	$\frac{1}{k^0}$	$[0.01, 10^6]$	$[199.522, 314.580]$
2	α_0	$[0, 1]$	$[0.562, 0.743]$
3	$c_{R,init}$	$[0, 10^5]$	$[1.937, 2.361] \cdot 10^{-7}$
4	$\frac{1}{m_O}$	$[0.1, 10^5]$	$[132.517, 235.526]$
5	$\frac{1}{m_R}$	$[0.1, 10^5]$	$[147.340, 157.596]$

Les valeurs numériques prises précédemment pour m_O et m_R sont cohérents avec ces résultats. De plus, la relation présupposée dans le modèle $m_O = m_R$ n'est pas invalidée, puisque les intervalles obtenus pour $[\hat{x}_4]$ et $[\hat{x}_5]$ ne sont pas disjoints. On peut cependant relâcher cette hypothèse puisque ces deux paramètres peuvent être estimés séparément. Les résultats fournissent alors

$$\begin{aligned}\hat{m}_O &\in [4.24, 7.55] \times 10^{-3}, \\ \hat{m}_R &\in [6.34, 6.79] \times 10^{-3}.\end{aligned}$$

On constate alors que m_R est obtenu de façon plus précise que m_O , ce qui n'est en rien surprenant puisque la majorité des données concerne la réduction de l'élément cyanure de fer. Il n'est alors possible d'améliorer la précision sur m_O qu'en explorant plus en avant le comportement anodique, et donc en redéfinissant le plan d'expériences.

L'équation (4.11) est un cas particulier du modèle correspondant au mécanisme de Volmer-Heyrovsky, décrit ci-dessous.

4.3.3 Mécanisme de Volmer-Heyrovsky: somme de deux exponentielles

De nombreuses réactions chimiques, telles que la production électrolytique de dichlore ou de dihydrogène, ou encore l'oxydation anodique du fer, mettent en jeu le transfert de deux électrons. Le mécanisme suivant résume ainsi la dissolution-passivation de métaux en deux étapes :



La première étape concerne l'oxydation d'un métal M_s par électrosorption d'espèces A^- pour former MA_s . Ensuite, l'électrodésorption régénère un site de surface libre. Chacune de ces deux étapes produit un électron. Sous les hypothèses que la déplétion interfaciale est négligeable, c'est-à-dire que la concentration à l'interface en A^- reste égale à $c_{A,init}$, et que les deux étapes sont irréversibles, la cinétique de la réaction suivant ce mécanisme est alors décrite, en régime quasistatique, par :

$$j_f(E) = 2F\Gamma \frac{c_{A,init} k_{o1} \exp(\alpha_{o1} f E) k_{o2} \exp(\alpha_{o2} f E)}{c_{A,init} k_{o1} \exp(\alpha_{o1} f E) + k_{o2} \exp(\alpha_{o2} f E)}. \quad (4.15)$$

où Γ représente le nombre total de sites (libres ou occupés) par unité de surface. En posant $t = f E$ et en choisissant comme observée

$$y^3(t) = \frac{2F\Gamma}{j_f(E)},$$

la sortie de modèle correspondante peut alors se réécrire sous la forme

$$\eta^3(\mathbf{x}, t) = \frac{1}{k_{o1} \exp(\alpha_{o1} t)} + \frac{c_{A,init}}{k_{o2} \exp(\alpha_{o2} t)} \quad (4.16)$$

$$= x_1 \exp(-x_3 t) + x_2 \exp(-x_4 t), \quad (4.17)$$

où

$$\mathbf{x} = \left(\frac{1}{k_{o1}} \quad \frac{c_{A,init}}{k_{o2}} \quad \alpha_{o1} \quad \alpha_{o2} \right)^T. \quad (4.18)$$

De (4.17), on déduit facilement qu'interchanger les rôles de (x_1, x_3) et (x_2, x_4) ne modifie en rien η^3 . Les vecteurs

$$\mathbf{x}^a = \left(\frac{1}{k_{o1}} \quad \frac{c_{A,init}}{k_{o2}} \quad \alpha_{o1} \quad \alpha_{o2} \right)^T \text{ et } \mathbf{x}^b = \left(\frac{c_{A,init}}{k_{o2}} \quad \frac{1}{k_{o1}} \quad \alpha_{o2} \quad \alpha_{o1} \right)^T$$

sont donc tels que $\eta(\mathbf{x}^a, t) = \eta(\mathbf{x}^b, t)$ pour tout instant de mesure t . L'étude menée sur la structure du modèle utilisé dans le cas d'un mécanisme à deux étapes montre ainsi qu'il n'est pas possible, en régime quasistatique, de déterminer à quelle étape correspond chaque vitesse estimée (voir [BDM⁺93], [BDM⁺95], et [BDPW96] pour plus de détails). Il n'est donc pas possible de privilégier l'une de ces deux valeurs du vecteur de paramètres \mathbf{x} à partir de cette expérience en régime quasi-statique, et en faisant une estimation numérique, au moins deux solutions devraient être obtenues.

Considérons tout d'abord des données simulées sans bruit

$$\mathbf{y}(t_i) = 10^{-5} \exp(-0.4 t_i) + 10^{-9} \exp(-0.8 t_i); \quad i = 1, \dots, 81, \quad (4.19)$$

générées à partir de (4.17) avec

$$\mathbf{x}^* = (10^{-5} \quad 10^{-9} \quad 0.4 \quad 0.8)^T$$

pour $c_{A,init} = 10^{-3} \text{ mol/cm}^{-3}$. Les t_i sont compris entre -11.61 et 19.35 puisque

$$E \in [-0.3V, 0.5V].$$

Nous fixons également

$$\mathbb{X} = \mathbb{X}^1 = [10^{-9}, 10^6] \times [10^{-9}, 10^6] \times [0, 1] \times [0, 1]. \quad (4.20)$$

La projection de l'ensemble $\bar{\mathbb{S}}$ obtenu par SIVIA sur le plan (x_3, x_4) est présentée sur la figure 4-8.

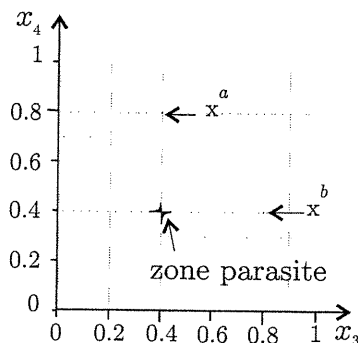


Figure 4-8: Projection $\pi_{34}(\bar{\mathbb{S}})$ de $\bar{\mathbb{S}}$ sur le plan (x_3, x_4) en l'absence de bruit de mesure

Comme l'on s'y attendait, $\pi_{34}(\bar{\mathbb{S}})$ contient deux points symétriques, images des deux vecteurs $\mathbf{x}^a = \mathbf{x}^*$ et \mathbf{x}^b . Une troisième composante, qui n'était pas prévue par cette étude, est également présente. Elle est composée de pavés indéterminés, centrés sur $(x_1 \in [10^{-9}, 10^{-5}]; x_2 \in [10^{-9}, 10^{-5}]; x_3 = 0.4; x_4 = 0.4)$. Cette zone correspond à un modèle à une exponentielle $(x_3 = x_4)$. Elle est due au fait que le rapport entre les amplitudes et la grande différence entre les constantes de temps imposent la prépondérance d'une exponentielle sur l'autre [IV99]: le système est ici mal conditionné. Plus nous diminuons la valeur du paramètre ε de SIVIA, plus le volume de cette "zone parasite" diminue. Notons que toutes les méthodes itératives locales essayées sur cet exemple avec la fonction coût j quadratique (parmi elles le simplexe, Gauss-Newton, Levenberg-Marquardt, les gradients conjugués, quasi-Newton, et la programmation quadratique séquentielle) ont tendance à converger vers cette région parasite. A titre d'exemple, les paramètres estimés \hat{x}_i fournis par l'algorithme de Gauss-Newton sont

$$\hat{\mathbf{x}} = \begin{pmatrix} 3.357 \cdot 10^{-7} \\ 9.622 \cdot 10^{-6} \\ 0.4655 \\ 0.3978 \end{pmatrix}.$$

et la sortie du modèle obtenu pour ces estimées est représentée sur la figure 4-9. Bien que la valeur de la fonction coût obtenue $j(\hat{\mathbf{x}})$ soit très faible (1.00619×10^{-6}) et que le modèle semble satisfaisant, $\hat{\mathbf{x}}$ ne correspond pas au minimiseur global $\mathbf{x}^* = (10^{-5} \ 10^{-9} \ 0.4 \ 0.8)^T$ qui satisfait $j(\mathbf{x}^*) = 0$, puisque les données sont simulées avec du bruit. L'estimée $\hat{\mathbf{x}}$ est en réalité très éloignée de \mathbf{x}^* , tandis que \mathbf{x}^* est lui bien inclus dans $\bar{\mathbb{S}}$.

Remarque 4.3.1 SIVIA permet également d'éliminer des structures de modèles incompatibles avec les données recueillies et leur incertitude. Dans le cas où cette incertitude est négligeable, comme par exemple dans le cas de données non bruitées, cela permet d'étudier la discernabilité de modèles de façon numérique. Prenons le modèle à une exponentielle du mécanisme rédox décrit par (4.11), et estimons son vecteur de paramètres sur les données simulées à partir du

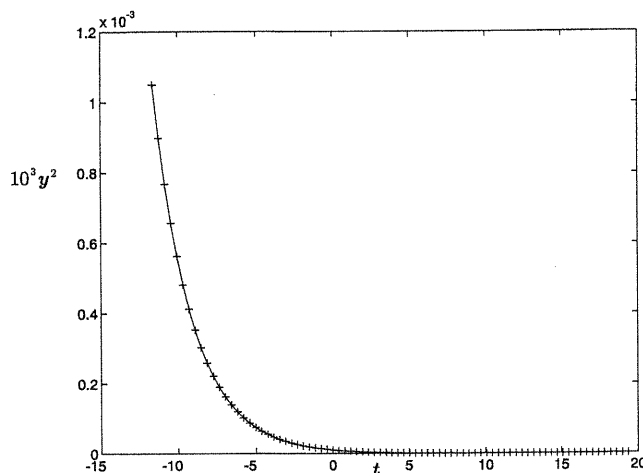


Figure 4-9: Sortie $\eta^3(\hat{\mathbf{x}})$ du modèle obtenue pour la valeur $\hat{\mathbf{x}}$ estimée par l'algorithme de Gauss-Newton (trait continu) et données expérimentales (+)

modèle de Volmer-Heyrovsky dans un contexte sans bruit. Les estimateurs ponctuels classiques trouveront une solution, puisque nous avons montré que ces données peuvent facilement être assimilées à celles obtenues par un modèle à une exponentielle, et donc conduire à accepter le mécanisme rédox: numériquement, on conclurait alors que le modèle rédox n'est pas discernable du modèle de Volmer-Heyrovsky en régime quasi-statique. Par contre, à condition que ε soit pris suffisamment petit, SIVIA permet de prouver numériquement que le modèle rédox est incompatible avec les données générées par le mécanisme de Volmer-Heyrovsky.

Prenons maintenant $\mathbb{X} = \mathbb{X}^2 = [10^{-6}, 10^6] \times [10^{-9}, 10^3] \times [0, 1] \times [0, 1]$, ce qui permet d'éliminer simplement la solution équivalente \mathbf{x}^b . Ceci revient à supposer que

$$\left(\begin{array}{l} k_{01} \in [10^{-6}, 10^6] \\ k_{02} \in [10^{-6}, 10^6] \\ \alpha_{01} \in [0, 1] \\ \alpha_{02} \in [0, 1] \end{array} \right).$$

Considérons également des données entachées d'une erreur de mesure telle que

$$[y_i] = |y_i| \times [0.95, 1.05] \quad \text{pour } i = 1, \dots, n.$$

L'estimation de x_4 devient très imprécise (voir la figure 4-10).

Une méthode des moindres carrés donnerait alors de résultats pires encore que dans le cas non bruité. La mauvaise qualité du résultat n'est pas due à la méthode d'estimation, mais aux données. Pour des problèmes trop mal conditionnés, aucune méthode d'estimation aussi sophistiquée soit-elle ne peut améliorer les résultats. La solution réside alors dans le choix d'un plan d'expériences qui maximise l'information fournie par les données, par exemple en diminuant $c_{A,\text{init}}$ puisqu'il suffit de diminuer cette concentration pour améliorer le conditionnement du système, ou en ayant recours aux mesure d'impédance. L'estimation à erreurs bornées permet alors d'aider à mieux concevoir une expérience dans le but d'améliorer la qualité des estimées.

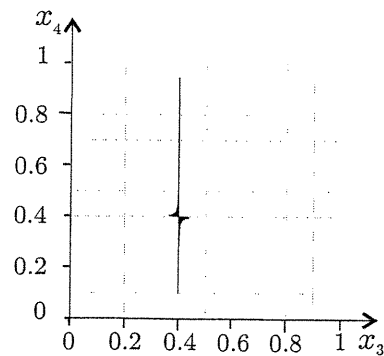


Figure 4-10: Projection $\pi_{34}(\bar{\mathbb{S}})$ de $\bar{\mathbb{S}}$ sur le plan (x_3, x_4) pour une incertitude de mesure de 5%, pour l'espace *a priori* \mathbb{X}^2

4.3.4 Conclusions

Nous avons illustré l'utilisation des méthodes garanties d'inversion ensembliste en identification paramétrique. Les algorithmes issus de SIVIA permettent de caractériser de façon garantie l'ensemble des vecteurs de paramètres compatibles avec les données expérimentales recueillies, c'est-à-dire qu'aucune valeur de l'espace des paramètres compatible avec les données intervalles n'a été perdue. Nous avons souligné sur des exemples simples mais représentatifs de situations fréquemment rencontrées en physique, chimie ou biologie, l'avantage de ces techniques là où on ne peut pas prouver que les valeurs numériques de paramètres calculées par optimisation itérative locale d'une fonction coût correspondent à un optimiseur global du critère. Le fruit de ce travail, qui a fourni la publication [BBJ⁺01b], a été suivi dans [MCR02] par l'étude du même modèle dans le régime petits signaux.

Soulignons encore l'importance d'une spécification réaliste du gabarit qui définit les barres d'incertitudes associées à chaque donnée. Dans ce cas, nous nous sommes appuyés sur la notice technique des appareils de mesure, que nous avons supposés parfaitement connus. Au paragraphe suivant, il s'agit d'étudier les performances d'une chaîne de mesure expérimentale, lorsque certains paramètres de ce dispositif ne sont qu'imparfaitement connus.

4.4 Facteurs expérimentaux inconnus

L'objectif de cette partie est d'évaluer les caractéristiques d'un nouveau dispositif de mesure proposé par le CERTES de l'Université Paris XII - Val - de - Marne.

En thermique du bâtiment, la modélisation des échanges thermiques au sein d'un habitat nécessite la détermination de la diffusivité a et la conductivité λ du matériau constitutif de ses parois. Dans les dispositifs de mesure classiques, a est mesurée par des méthodes impulsionnelles, tandis que la mesure de λ nécessite l'établissement d'un régime permanent. En plus de la lourdeur et de l'inconfort du dispositif, l'expérience souffre donc de la durée conséquente du protocole expérimental. Le dispositif proposé par le CERTES est de taille modeste et léger. Il a pour ambition de permettre la mesure *conjointe* des deux paramètres thermo-physiques en des temps raisonnables à l'échelle de la thermique. Pour cela, le matériau étudié est soumis à un régime multi-harmonique, ce qui impose un dispositif spécifique. Dans cette configuration, les grandeurs a et λ ne sont pas directement accessibles à la mesure, mais interviennent dans les phénomènes de transport thermique au sein du matériau. Le modèle de connaissance \mathcal{M} établi par le CERTES

fait ainsi apparaître un ensemble de relations mathématiques liant a et λ à la température de surface du matériau [TK98], qui est mesurable à l'aide de thermocouples. Il a alors été envisagé d'identifier a et λ en tant que paramètres du modèle de connaissance \mathcal{M} .

L'évaluation de ce dispositif de mesure passe préalablement par la détermination des caractéristiques d'un échantillon étalon. Le matériau testé est un PVC, dont les caractéristiques a_p et λ_p sont connues de façon très imprécise car elles dépendent fortement de l'échantillon utilisé. Par ailleurs, de nombreuses sources d'incertitudes intrinsèques au dispositif entachent la mesure. Chacune peut être décrite par une grandeur incertaine mais dont le support borné est connu des thermiciens. L'estimation des caractéristiques a_p et λ_p doit tenir compte de l'ensemble de ces sources d'incertitudes. Pour cela, nous allons décomposer notre démarche en plusieurs étapes.

Le paragraphe 4.4.1 aborde succinctement le dispositif expérimental mis en œuvre. S'en suit une description détaillée de la structure du modèle de connaissance $\mathcal{M}(\cdot)$ au paragraphe 4.4.2. Les spécificités de ce problème perçu dans un contexte ensembliste à l'aide d'une modélisation intervalles seront soulignées au paragraphe 4.4.3. La première étape de notre démarche d'identification va consister à estimer a_p et λ_p en ignorant la totalité des sources d'incertitudes intrinsèques du dispositif (paragraphe 4.4.4). Bien que peu réaliste, cette hypothèse va nous permettre de comparer la qualité des résultats obtenus par estimation ensembliste garantie avec ceux fournis lors d'une étude antérieure, effectuée dans un contexte statistique [TK98]. Nous soulignerons pourquoi il est nécessaire que cette hypothèse simplificatrice puisse être relâchée. L'étape suivante consiste à réaliser une nouvelle estimation des paramètres de l'échantillon, mais en prenant en compte les incertitudes inhérentes au dispositif (paragraphe 4.4.5). Dans un premier temps, nous proposons de quantifier l'influence de chaque source d'incertitude sur la qualité de la mesure. Cette approche nous permettra de mettre en évidence le comportement particulier de certaines de ces sources, comportement ultérieurement justifié par une étude théorique de sensibilité. Cette étape pourra à cette occasion s'inscrire dans une démarche de planification d'expériences en vue d'améliorer les performances du dispositif. Enfin, nous montrerons que l'identification de a_p et λ_p revient à projeter un ensemble solution sur un espace de dimension inférieure. Nous présentons alors les résultats obtenus par les algorithmes PROJECT du chapitre 2 suivant la quantité de sources d'incertitudes prises en compte. L'ensemble des résultats obtenus nous permettra enfin de proposer un modèle simplifié du dispositif, traité au paragraphe 4.4.6. Le dernier paragraphe regroupe nos conclusions, tant du point de vue de l'identificateur, que du thermicien, ainsi que les perspectives que cette étude nous a permis d'envisager.

4.4.1 Dispositif expérimental

L'élément chauffant est un bloc Peltier qui impose une excitation périodique de température. Les capteurs de température disponibles sont des thermocouples. Comme ceux-ci ne peuvent pas être directement apposés sur le PVC, l'échantillon étudié est placé entre deux plaques métalliques qui forment le porte-échantillon (voir la figure 4-11). Ces plaques ont donc le rôle d'interface entre la source d'excitation et les capteurs. L'excitation périodique de température à laquelle est soumise la plaque avant en laiton est notée T_{avant} . La plaque arrière, en cuivre, est en contact avec l'air ambiant à la température T_0 . Sa face arrière est donc le siège d'un phénomène de convection³. Les températures notées T_{avant} et $T_{\text{arrière}}$ sont mesurées à l'aide des deux thermocouples disposés au sein des deux plaques.

³ A température ambiante, le phénomène de transfert radiatif peut également être modélisé par une relation linéaire entre T_0 et φ_0 . Le coefficient h qui tient compte de ce phénomène est alors appelé coefficient d'échange globalisé.

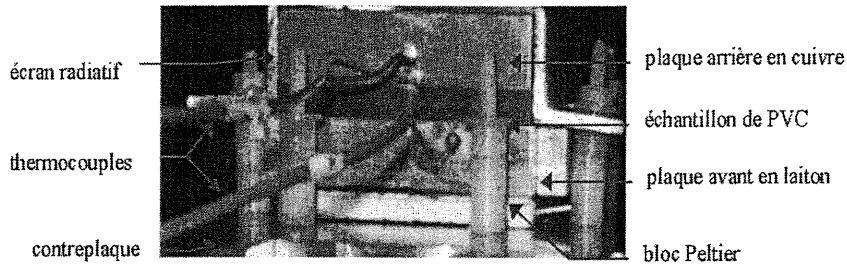


Figure 4-11: Dispositif expérimental

4.4.2 Modèle à base de connaissance

Le modèle que nous présentons ici a été étudié dans [TK98].

4.4.2.1 Notations

Les grandeurs thermo-physiques et géométriques utilisées dans la suite de ce document sont répertoriées dans les tableaux 4.1 et 4.2.

	Laiton	Colle	PVC	Cuivre	CTC ⁴
épaisseur	e_l	e_{co}	e_p	e_c	ξe_c
diffusivité	a_l	a_{co}	a_p	a_c	a_c
conductivité	λ_l	λ_{co}	λ_p	λ_c	λ_c
temps de Fourier	τ_l	τ_{co}	τ_p	τ_c	$\xi^2 \tau_c$
résistance	R_l	R_{co}	R_p	R_c	ξR_c
matrice de transfert	Z_l	Z_{co}	Z_p	Z_c	Z_{ctc}

Tableau 4.1: Grandeurs caractéristiques des matériaux

coefficient d'échange thermique	h
température de la source excitatrice	T_{avant}
température de la face arrière	T_0
température relevée par le thermocouple	$T_{\text{arrière}}$
flux thermique travers la face avant	φ_{avant}
flux thermique travers la face arrière	$\varphi_{\text{arrière}}$

Tableau 4.2: Grandeurs thermiques utilisées

4.4.2.2 Quadripôle thermique

Nous supposons dans la suite que toutes les couches des matériaux sont homogènes, et nous considérerons des transports de chaleur monodimensionnels, suivant la direction x (voir la figure 4-12).

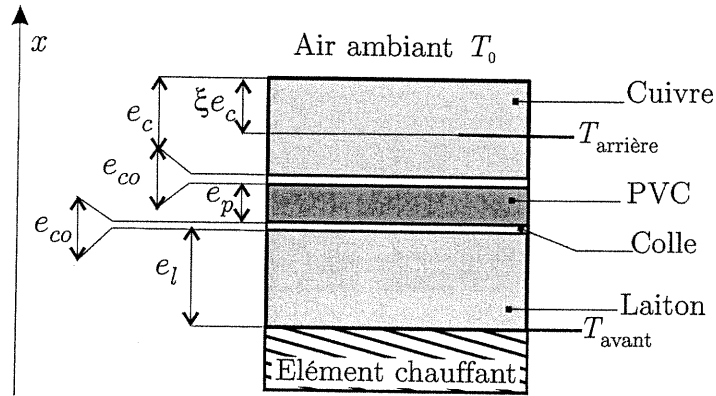


Figure 4-12: Schéma du dispositif expérimental et grandeurs géométriques utilisées

Isolons une couche homogène quelconque d'épaisseur e_i . Soit (T_0^i, φ_0^i) la température et le flux thermique à l'entrée de la couche et (T_1^i, φ_1^i) la température et le flux thermique à la sortie. En régime sinusoïdal, les équations de transport thermique par conduction au sein de la couche nous permettent de décrire les phénomènes d'échanges thermiques à l'aide d'une matrice de transfert $Z_i(s)$ fonction de la variable de Laplace s , définie comme suit :

$$\begin{pmatrix} T_0(s) \\ \varphi_0(s) \end{pmatrix} = Z_i(s) \begin{pmatrix} T_1(s) \\ \varphi_1(s) \end{pmatrix}, \quad (4.21)$$

où

$$Z_i(s) = \begin{pmatrix} \cosh(\sqrt{\tau_i s}) & \frac{R_i}{\sqrt{\tau_i s}} \sinh(\sqrt{\tau_i s}) \\ \frac{\sqrt{\tau_i s}}{R_i} \sinh(\sqrt{\tau_i s}) & \cosh(\sqrt{\tau_i s}) \end{pmatrix}, \quad (4.22)$$

avec

$$\tau_i = \frac{e_i^2}{a_i}; R_i = \frac{e_i}{\lambda_i}. \quad (4.23)$$

La matrice de transfert $Z_i(s)$ est donc uniquement paramétrée par le temps caractéristique de Fourier τ_i et la résistance thermique R_i de la couche i . Nous choisissons donc d'utiliser cette paramétrisation, en gardant à l'esprit la remarque suivante.

Remarque 4.4.1 Les paramètres a_i, e_i et λ_i étant indépendants, et a_i et λ_i inconnus, la reparamétrisation

$$\gamma: \begin{pmatrix} a_i \\ e_i \\ \lambda_i \end{pmatrix} \rightarrow \begin{pmatrix} R_i \\ \sqrt{\tau_i} \end{pmatrix} = \begin{pmatrix} \frac{e_i}{\lambda_i} \\ \frac{e_i}{\sqrt{a_i}} \end{pmatrix} \quad (4.24)$$

est bijective si et seulement si la valeur de e_i est connue de façon exacte. Lorsque la mesure de e_p est fournie avec une erreur bornée, nous réaliserons dans un premier temps l'identification de R_p et τ_p , puis celle de a_p et λ_p , compte tenu de l'incertitude sur e_p .

La colle ayant une inertie négligeable, nous posons $\tau_{co} = 0$, de sorte que

$$Z_{co} = \begin{pmatrix} 1 & R_{co} \\ 0 & 1 \end{pmatrix}. \quad (4.25)$$

Enfin, le phénomène de convection (et de radiation) est modélisé par

$$\varphi_0(s) = hT_0(s), \quad (4.26)$$

où h est le *coefficient d'échange globalisé*.

A l'aide des équations de conservation de l'énergie et des relations de continuité de la température aux interfaces, on modélise le comportement entrée-sortie du dispositif par une chaîne de quadripôles de fonction de transfert

$$H(s) = \frac{T_{arri\ re}(s)}{T_{avant}(s)}, \quad (4.27)$$

paramétrée par les coefficients recherchés.

4.4.2.3 Modélisation du dispositif expérimental

La modélisation proposée fait intervenir, par le phénomène de convection, la température T_0 . Il est donc nécessaire d'introduire deux chaînes de quadripôles. La chaîne principale \mathcal{A} , d'entrée T_0 et de sortie T_{avant} , et la chaîne de retour \mathcal{B} , d'entrée T_0 et de sortie $T_{arri\ re}$, telles que

$$\mathcal{A} : \begin{pmatrix} T_{avant}(s) \\ \varphi_{avant}(s) \end{pmatrix} = Z_l(s) \cdot Z_{co}(s) \cdot Z_p(s) \cdot Z_{co}(s) \cdot Z_c(s) \begin{pmatrix} T_0(s) \\ \varphi_0(s) \end{pmatrix}, \quad (4.28)$$

$$\mathcal{B} : \begin{pmatrix} T_{arri\ re}(s) \\ \varphi_{arri\ re}(s) \end{pmatrix} = Z_{ctc}(s) \begin{pmatrix} T_0(s) \\ \varphi_0(s) \end{pmatrix}. \quad (4.29)$$

$Z_{ctc}(s)$ est la matrice de transfert de la couche de cuivre d'épaisseur ξe_c (voir la figure 4-12) qui est paramétrée par

$$\begin{cases} R_{ctc} = \xi R_c \\ \sqrt{\tau_{ctc}} = \xi \sqrt{\tau_c} \end{cases}.$$

La prise en compte de la chaîne de retour introduit donc l'unique paramètre supplémentaire ξ . En remarquant que $\varphi_0(s) = hT_0(s)$, la chaîne de mesure globale est alors obtenue par association de ces deux chaînes en éliminant (T_0, φ_0) (voir la figure 4-13).

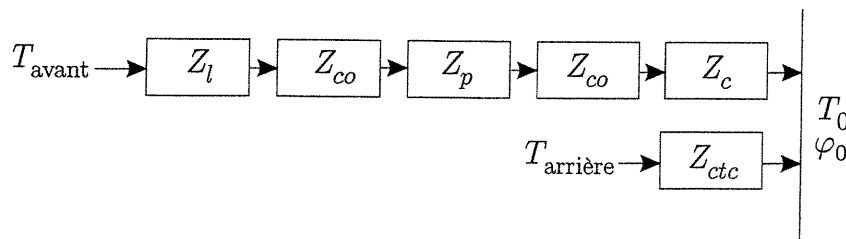


Figure 4-13: Modélisation du dispositif expérimental par association de quadripôles, en faisant intervenir les grandeurs thermiques (T_0, φ_0)

La fonction de transfert de cette chaîne s'écrit

$$H(s) = \frac{T_{\text{arri re}}(s)}{T_{\text{avant}}(s)} = \frac{B(s)}{A(s)}, \quad (4.30)$$

où

$$B(s) = \cosh(\xi\sqrt{\tau_c s}) + h\xi R_c \frac{\sinh(\xi\sqrt{\tau_c s})}{\xi\sqrt{\tau_c s}}, \quad (4.31)$$

et

$$\begin{aligned} A(s) = & [1 + 2hR_{co}] \cosh(\sqrt{\tau_c s}) \cosh(\sqrt{\tau_l s}) \cosh(\sqrt{\tau_p s}) \\ & + [(1 + hR_{co}) \frac{R_{co}}{R_p} \sqrt{\tau_p s} + h \frac{R_p}{\sqrt{\tau_p s}}] \cosh(\sqrt{\tau_c s}) \cosh(\sqrt{\tau_l s}) \sinh(\sqrt{\tau_p s}) \\ & + [h \frac{R_l}{\sqrt{\tau_l s}}] \cosh(\sqrt{\tau_c s}) \sinh(\sqrt{\tau_l s}) \cosh(\sqrt{\tau_p s}) \\ & + [2R_{co} \frac{\sqrt{\tau_c s}}{R_c} + \frac{hR_c}{\sqrt{\tau_c s}}] \sinh(\sqrt{\tau_c s}) \cosh(\sqrt{\tau_l s}) \cosh(\sqrt{\tau_p s}) \\ & + \left[(1 + hR_{co}) \frac{R_l}{R_p} \sqrt{\frac{\tau_p}{\tau_l}} \right] \cosh(\sqrt{\tau_c s}) \sinh(\sqrt{\tau_l s}) \sinh(\sqrt{\tau_p s}) \\ & + \left[\frac{R_{co}}{R_p} \sqrt{\tau_p s} \left(\frac{R_{co}}{R_c} \sqrt{\tau_c s} + h \frac{R_c}{\sqrt{\tau_c s}} \right) + \frac{R_p}{R_c} \sqrt{\frac{\tau_c}{\tau_p}} \right] \sinh(\sqrt{\tau_c s}) \cosh(\sqrt{\tau_l s}) \sinh(\sqrt{\tau_p s}) \\ & + \left[\frac{R_l}{R_c} \sqrt{\frac{\tau_c}{\tau_l}} \right] \sinh(\sqrt{\tau_c s}) \sinh(\sqrt{\tau_l s}) \cosh(\sqrt{\tau_p s}) \\ & + \left[\frac{R_l}{R_p} \sqrt{\frac{\tau_p}{\tau_l}} \left(\frac{R_{co}}{R_c} \sqrt{\tau_c s} + h \frac{R_c}{\sqrt{\tau_l s}} \right) \right] \sinh(\sqrt{\tau_c s}) \sinh(\sqrt{\tau_l s}) \sinh(\sqrt{\tau_p s}). \end{aligned} \quad (4.32)$$

$A(s)$ et $B(s)$ font intervenir deux types de grandeurs :

- les grandeurs thermo-physiques du PVC R_p et $\sqrt{\tau_p}$, que nous cherchons à identifier; nous les regroupons donc dans le vecteur de *paramètres*

$$\mathbf{p} = \left(R_p \quad \sqrt{\tau_p} \right)^T, \quad (4.33)$$

- les grandeurs caractéristiques des autres matériaux (porte-échantillon et colle), et le coefficient de convection h , que nous regroupons dans le vecteur des *facteurs expérimentaux*⁵

$$\mathbf{q} = \left(R_l \quad \sqrt{\tau_l} \quad R_{co} \quad R_c \quad \sqrt{\tau_c} \quad \xi \quad h \right)^T. \quad (4.34)$$

Par la suite, nous serons amenés à considérer \mathbf{p} et \mathbf{q} comme deux sous-vecteurs complémentaires du vecteur de *paramètres étendu*

$$\mathbf{x} = \left(R_p \quad \sqrt{\tau_p} \quad R_l \quad \sqrt{\tau_l} \quad R_{co} \quad R_c \quad \sqrt{\tau_c} \quad \xi \quad h \right)^T. \quad (4.35)$$

Chacune de ces grandeurs est incertaine, comme nous le précisons au paragraphe suivant.

⁵Nous avons choisi cette terminologie particulière au lieu du terme "paramètres de nuisance" que nous préférons réserver au contexte probabiliste.

4.4.2.4 Sources d'incertitudes

La fonction de transfert $H(s)$ définie par (4.30), dépend de plusieurs grandeurs physiques incertaines :

- les caractéristiques thermo-physiques des matériaux thermiquement non-inertes (cuivre, laiton et PVC) : le dispositif de mesure est collé, ce qui empêche d'effectuer une identification séparée non-destructive de chaque matériau⁶; l'intervalle qui leur est associé correspond alors à la plage de dispersion des valeurs numériques tirées de la littérature;
- les caractéristiques thermo-physiques de la colle : la conductivité λ_{co} est fournie sous forme d'intervalle par le constructeur, la diffusivité a_{co} a été supposée infinie dans les hypothèses de modélisation (4.25);
- les grandeurs géométriques mesurées : l'intervalle associé correspond donc au support de l'erreur de mesure;
- le coefficient d'échange convectif à la surface h , supposé constant par hypothèse⁷, est très mal quantifié pour une paroi dont la température est modulée. L'incertitude associée à h , qui correspond à une incertitude de modélisation, est fournie par l'expérimentateur.

Il devient alors possible d'attribuer aux vecteurs \mathbf{p} et \mathbf{q} des espaces de recherche bornés \mathbb{P} et \mathbb{Q} .

Caractéristiques de l'échantillon Comme nous l'avons déjà signalé, les caractéristiques du PVC varient notablement avec l'échantillon utilisé. Un regroupement des différentes valeurs trouvées dans la littérature permet de définir une plage qui pourrait éventuellement contenir les valeurs de a_p et $\sqrt{\tau_p}$ (voir le tableau 4.3). Chaque grandeur X est donc modélisée sous la forme d'un intervalle $[X]$ dont nous avons calculé le centre X_0 , et l'incertitude relative $\Delta_r X$ par

$$\Delta_r X = \frac{\bar{X} - \underline{X}}{\bar{X} + \underline{X}}.$$

X	$[X]$	$X_0 \pm \Delta_r X$
a_p ($10^{-6} \text{ m}^2 \cdot \text{sec}^{-1}$)	[0.11, 0.12]	$0.115 \pm 4.3\%$
λ_p ($\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$)	[0.16, 0.18]	$0.17 \pm 5.9\%$
e_p (10^{-3} m)	[4.98, 5.02]	$5.00 \pm 0.4\%$
R_p (SI)	[0.0276, 0.031]	$0.0293 \pm 5.8\%$
$\sqrt{\tau_p}$ (SI)	[14, 15.2]	$14.6 \pm 4.1\%$

Tableau 4.3: Valeurs numériques des caractéristiques du PVC

⁶Ce point soulevé, il a été proposé, dans des versions ultérieures du dispositif, de remplacer la colle par de la graisse.

⁷L'utilisation d'un coefficient d'échange h est fréquente en thermique du bâtiment. Les coefficients utilisés sont soit constants, soit variables et dépendant des flux, de la vitesse du fluide, de l'orientation de la paroi... A l'heure actuelle, on ne sait pas quantifier ce coefficient pour une paroi dont la température de surface est modulée. Nous le supposons donc incertain.

Afin d'obtenir l'ensemble des valeurs possibles pour les caractéristiques du PVC, nous choisissons de définir un espace de recherche *a priori* \mathbb{P} élargi, centré sur cette plage⁸ :

$$\mathbb{P} = [0.014, 0.047] \times [7.2, 23].$$

Incertitudes intrinsèques du dispositif Il s'agit de l'incertitude entachant les composantes du vecteur des facteurs expérimentaux \mathbf{q} . Le tableau 4.4 résume l'information disponible *a priori*. Le domaine du coefficient d'échange h est fixé à $[5, 10]$ $\text{W.m}^{-2}.\text{K}^{-1}$.

	Laiton	Cuivre	Colle	CTC
a ($10^{-6} \text{ m}^2.\text{sec}^{-1}$)	[33, 35]	[114, 117]	∞	[114, 117]
λ ($\text{W.m}^{-1}.\text{K}^{-1}$)	[100, 110]	[386, 402]	[1.4, 1.6]	[386, 402]
e (10^{-3}m)	[2, 4]	[9.95, 10.05]	$[30, 70].10^{-3}$	$[3, 7].10^{-3}$
R (10^{-5}SI)	[1.8, 4]	[2.4, 2.6]	[1.8, 4.3]	[0.733, 1.81]
$\sqrt{\tau}$ (SI)	$[3.38, 7]10^{-4}$	$[9, 9.4]10^{-1}$	0	[0.27, 0.67]

Tableau 4.4: Incertitudes intrinsèques au dispositif

Le tableau 4.5 met en évidence l'incertitude associée à chaque composante q_i pour laquelle nous rappelons la grandeur physique associée.

	X	X_0	$\Delta_r X$
q_1	$R_l(10^{-5}\text{SI})$	2.90	38%
q_2	$\sqrt{\tau_l}$	5.19×10^{-4}	39%
q_3	$R_{co}(10^{-5}\text{SI})$	3.05	41%
q_4	$R_c(10^{-5}\text{SI})$	2.5	4%
q_5	$\sqrt{\tau_c}$	0.92	2%
q_6	ξ	0.5005×10^{-3}	40%
q_7	h	7.5	33%

Tableau 4.5: Incertitudes des facteurs expérimentaux

L'incertitude relative des facteurs expérimentaux va jouer un rôle important dans la qualité de l'estimation. Remarquons ainsi, qu'à part les caractéristiques du cuivre, qui sont relativement bien connues, l'incertitude relative associée à chacun de ces facteurs expérimentaux est de l'ordre de 40%. Cet exemple illustre bien la variabilité des diamètres des intervalles manipulés sur certaines applications réelles, et justifiera tout particulièrement l'utilisation de contracteurs efficaces pour des intervalles grands, comme ceux utilisant la propagation de contraintes (voir le chapitre 3).

Remarque 4.4.2 *Notons à ce propos que la plage fournie par la littérature pour les paramètres du PVC est déjà plus précise que celle fournie pour de nombreux matériaux, dont le laiton en particulier; il semble donc opportun, dans des versions ultérieures de dispositif, de caractériser le laiton plus précisément si cela est possible, ou de choisir un autre type d'alliage, pour lequel une information plus précise sera à disposition.*

⁸Lorsque les unités ne sont pas spécifiées, il s'agit toujours des unités S.I.

Le domaine de recherche \mathbb{Q} est pris égal au produit cartésien des intervalles associés à chaque facteur expérimental q_i :

$$\mathbb{Q} = \left(\begin{array}{l} [1.8, 4].10^{-5} \\ \times [3.38, 7].10^{-4} \\ \times [1.8, 4.3].10^{-5} \\ \times [2.4, 2.6].10^{-5} \\ \times [9, 9.4].10^{-1} \\ \times [0.3003, 0.7007] \\ \times [5, 10] \end{array} \right).$$

Erreurs de mesure bornées Les grandes constantes de temps thermiques du système imposent une contrainte sur la durée des échantillons obtenus. Comme une vérification expérimentale justifie que le système se comporte de façon linéaire, il a été choisi, pour des raisons de gain de temps, d'exciter le dispositif par un signal temporel $T_{\text{avant}}(t)$ composé de la somme de cinq sinusoïdes de pulsation ω_k , $k = 1, \dots, 5$. Le thermocouple recueille alors en sortie un signal temporel $T_{\text{arrière}}(t)$ sur un horizon de 85 min. L'expérience, effectuée trente fois, nous permet de disposer de trente réalisations du signal $T_{\text{arrière}}(t)$, soit trente mesures, qui sont alors soumises à une série de traitements :

- amplification,
- échantillonnage à 100 Hz,
- filtrage passe-bas (fréquence de coupure 5 Hz)
- sous-échantillonnage à 0.4 Hz.

En calculant la transformée de Fourier de chaque échantillon, on obtient 30 estimations des spectres $\hat{T}_{\text{avant}}(j\omega)$ et $\hat{T}_{\text{arrière}}(j\omega)$ des températures T_{avant} et $T_{\text{arrière}}$. Pour chacune, on relève les amplitudes $\hat{T}_{\text{avant}}(j\omega_k)$ et $\hat{T}_{\text{arrière}}(j\omega_k)$ pour $k = 1, \dots, 5$, correspondant aux cinq pulsations de mesure. Les trente rapports $y_k = \frac{\hat{T}_{\text{arrière}}(j\omega_k)}{\hat{T}_{\text{avant}}(j\omega_k)}$ fournissent ainsi les parties réelle et imaginaire de la réponse fréquentielle du système étudié aux cinq fréquences de mesure. Les erreurs de mesure peuvent ainsi être recensées:

- erreurs dues aux appareils électroniques,
- erreurs dues au relevé du spectre,
- erreurs de lecture.

Puisque seules trente mesures ont été effectuées, aucune information d'ordre statistique n'a pu être établie. Nous nous plaçons donc dans un contexte à erreurs bornées. Contrairement au cas de l'appareil SOLARTRON (paragraphe 4.3.1), nous ne disposons pas de résultats d'une étude antérieure de ces diverses erreurs de mesure, aussi est-il décidé de délimiter le support de la variable aléatoire modélisant la sortie du système réel à partir des trente relevés ponctuels, sous la forme

$$[y_k] = \left[\inf_{i=1..30} y_k^i, \sup_{i=1..30} y_k^i \right].$$

Nous disposons alors de cinq données intervalles complexes, que nous regroupons dans un vecteur intervalles $[\mathbf{H}]$. Pour plus de simplicité, nous noterons également la sortie de modèle à la pulsation ω_k par

$$\eta_k(\mathbf{p}, \mathbf{q}) = \eta_k(\mathbf{x}) = H(s = j\omega_k).$$

Remarque 4.4.3 *L'erreur de lecture des pulsations de mesure $\omega_k, k = 1, \dots, 5$ a été négligée dans cette première étude de faisabilité. Une étude ultérieure⁹ visant à la prendre en compte pourra en particulier mettre en œuvre l'algorithme proposé dans [JW99].*

4.4.3 Approche ensembliste

Notre démarche va s'articuler en deux étapes :

- choisir une représentation des intervalles complexes,
- établir une fonction d'inclusion.

4.4.3.1 Représentation des intervalles complexes

Il existe plusieurs types de représentations des intervalles complexes $[z] = \{z = x + iy\}$, dont les trois plus répandues sont les représentations rectangulaire, polaire et circulaire [PP98]. Chacune d'entre elles définit un type spécifique d'arithmétique d'intervalles complexes. Il s'agit donc de déterminer la représentation la plus adaptée à notre problème. Pour cela, nous allons étudier le pessimisme des opérations classiques étendues et la complexité de mise en œuvre des opérations ensemblistes pures pour des intervalles complexes.

Représentation rectangulaire Assez intuitive, elle consiste à représenter l'élément courant z d'un intervalle complexe $[z]$ à partir de sa partie réelle et de sa partie imaginaire. L'ensemble

$$[z]_R = \{z = z_r + iz_i \mid z_r \in [z_r], z_i \in [z_i]\},$$

noté $([z_r], [z_i])$ est un pavé $[z_r] \times [z_i]$ dans le plan complexe, pour lequel les opérations de base (additions, soustractions, multiplications et divisions) sont aisément définies. Par exemple l'addition de deux intervalles complexes rectangulaires vaut

$$[z_1]_R + [z_2]_R = ([z_{1r}] [z_{2r}], [z_{1i}] [z_{2i}]).$$

Cependant, la multiplication et la division introduisent naturellement du pessimisme, en raison, d'une part, de la multiplication des occurrences :

$$\frac{[z_1]_R}{[z_2]_R} = \frac{[z_{1r}] [z_{2r}] + [z_{1i}] [z_{2i}]}{[z_{2r}]^2 + [z_{2i}]^2} + i \frac{[z_{1i}] [z_{2r}] - [z_{1r}] [z_{2i}]}{[z_{2r}]^2 + [z_{2i}]^2},$$

mais également en raison du phénomène d'enveloppement. Ainsi par exemple, l'inverse d'un pavé contenant l'origine n'est pas un pavé. Les opérations ensemblistes pures, telles que l'inclusion et

⁹ Une thèse sur ce sujet a débuté au CERTES en septembre 2001.

la disjonction, sont très naturellement définies :

$$\begin{aligned} [z_1]_R \subset [z_2]_R &\iff \left\{ \begin{array}{l} [z_{1r}] \subset [z_{2r}] \\ \wedge [z_{1i}] \subset [z_{2i}] \end{array} \right. , \\ [z_1]_R \cap [z_2]_R = \emptyset &\iff \left\{ \begin{array}{l} [x_1] \cap [x_2] = \emptyset \\ \vee [y_1] \cap [y_2] = \emptyset \end{array} \right. . \end{aligned}$$

Représentation polaire L'élément courant z est représenté à partir de son module et de son argument

$$[z]_P = \{z = z_\rho \exp(i z_\theta) \mid z_\rho \in [z_\rho], z_\theta \in [z_\theta]\}.$$

Les intervalles complexes $[z]_P$, notés $([z_\rho], [z_\theta])$ sont donc représentés par des portions de couronne dans le plan complexe. Les avantages et défauts de cette représentation sont complémentaires de ceux de la représentation rectangulaire, puisqu'ici les opérations pessimistes sont l'addition et la soustraction :

$$[z_1]_P + [z_2]_P = \left(\begin{array}{l} \sqrt{[z_{\rho 1}]^2 + [z_{\rho 2}]^2 + 2 [z_{\rho 1}] [z_{\rho 2}] \cos([z_{\theta 1}] - [z_{\theta 2}])}, \\ \arctan \frac{[z_{\rho 1}] \sin[z_{\theta 1}] + [z_{\rho 2}] \sin[z_{\theta 2}]}{[z_{\rho 1}] \cos[z_{\theta 1}] + [z_{\rho 2}] \cos[z_{\theta 2}]} \end{array} \right).$$

Les multiplications et divisions sont ici optimales :

$$\begin{aligned} [z_1]_P [z_2]_P &= ([z_{\rho 1}] [z_{\rho 2}], [z_{\theta 1}] + [z_{\theta 2}]), \\ \frac{[z_1]_P}{[z_2]_P} &= \left(\frac{[z_{\rho 1}]}{[z_{\rho 2}]}, [z_{\theta 1}] - [z_{\theta 2}] \right). \end{aligned}$$

Il faut de plus souligner que la comparaison de deux intervalles complexes $[z_1]$ et $[z_2]$ dans cette représentation suppose de représenter leurs arguments $z_{\theta 1}$ et $z_{\theta 2}$ dans la même détermination. Dans de nombreux cas, cela suppose donc de manipuler des intervalles disjoints.

Exemple 4.4.1 Nous souhaitons déterminer si les intervalles

$$\begin{aligned} [z_1] &= [0, 1] \exp\left(i \left[\frac{\pi}{2}\right]\right) \\ [z_2] &= [0, 1] \exp\left(i \left[\frac{7\pi}{4}, \frac{9\pi}{4}\right]\right) \end{aligned}$$

sont disjoints. Pour cela, nous allons donc exprimer l'argument $[z_{\theta 2}]$ de $[z_2]$ dans la détermination de $[z_{\theta 1}]$. On obtient alors la représentation suivante :

$$[z_2] = [0, 1] \exp\left(i \left[0, \frac{\pi}{4}\right]\right) \cup [0, 1] \exp\left(i \left[\frac{7\pi}{4}, 2\pi\right]\right).$$

Représentation circulaire Cette représentation, préconisée dans [PP98], consiste à représenter un intervalle complexe sous la forme d'un disque. Soit c un nombre complexe et r un réel positif. L'ensemble

$$[z] = \{z \in \mathbb{C} \mid |z - c| \leq r\}$$

noté (c, r) est appelé un *intervalle circulaire*. Il est alors nécessaire de définir l'ensemble des opérations élémentaires. A nouveau, l'addition et la soustraction sont des opérations non-pessimistes,

contrairement à la multiplication pour laquelle il existe à nouveau un phénomène d'enveloppe. En définissant la multiplication par

$$[z_1][z_2] = \{c_1 c_2 \mid |c_1| r_2 + |c_2| r_1 + r_1 r_2\},$$

on peut montrer que

$$[z_1][z_2] \supseteq \{z_1 z_2 \mid z_1 \in [z_1], z_2 \in [z_2]\},$$

puisque (voir [PP98])

$$\begin{aligned} |z_1 z_2 - c_1 c_2| &\leq |c_1(z_2 - c_2) + c_2(z_1 - c_1) + (z_1 - c_1)(z_2 - c_2)| \\ &\leq |c_1||z_2 - c_2| + |c_2||z_1 - c_1| + |z_1 - c_1||z_2 - c_2| \\ &\leq |c_1|r_2 + |c_2|r_1 + r_1 r_2. \end{aligned}$$

Enfin, on a les propriétés suivantes

$$[z_1] \cap [z_2] = \emptyset \iff |c_1 - c_2| > r_1 + r_2,$$

et

$$[z_1] \subset [z_2] \iff |c_1 - c_2| > r_2 - r_1.$$

Il s'agit maintenant de déterminer la représentation la plus adaptée. Les fonctions que nous allons manipuler sont principalement des fonctions trigonométriques. Elles apparaissent dans le modèle sous la forme $\cosh(\sqrt{x}s)$, où la variable de Laplace s vérifie $s = j2\pi f$. En notant la parité de toutes les fonctions utilisées ($\cosh(z)$, $z \sinh(z)$ et $\frac{\sinh(z)}{z}$), nous avons à manipuler des fonctions sous la forme

$$\begin{aligned} \cos([z]) \circ [z] &= \{z = x(1+i), x \in \mathbb{IR}\} \\ &= \left\{z = \sqrt{2}x \exp\left(i\frac{\pi}{4}\right), x \in \mathbb{IR}\right\}. \end{aligned}$$

La première remarque que nous pouvons formuler est que la représentation la plus adaptée des nombres complexes du type $[x](1+i)$ est la représentation polaire, puisqu'elle introduit le minimum de pessimisme (voir la Figure 4-14).

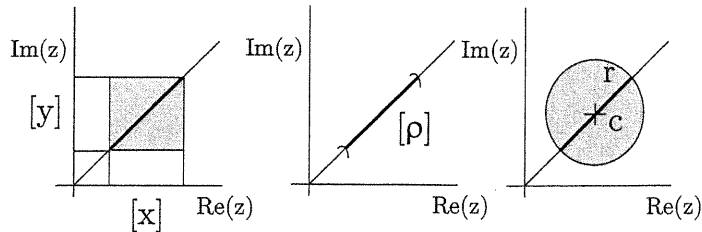


Figure 4-14: Représentation de l'intervalle complexe manipulé $[z] = [x](1+i)$

Il s'agit maintenant d'évaluer l'image par les fonctions trigonométriques présentes dans le modèle de cet intervalle complexe. A titre d'exemple, nous allons calculer les composantes de la matrice de transfert d'un quadripôle thermique pour les représentations rectangulaires et

polaires.

Exemple 4.4.2 On considère un quadripôle thermique de résistance connue $R = 1$ et de temps de Fourier incertain $\sqrt{\tau} \in [0, 3]$. Soient

$$\begin{aligned} a_{11}(R, \sqrt{\tau}, f) &= a_{22}(R, \sqrt{\tau}, f) = \cosh\left(\sqrt{\tau j 2\pi f}\right) \\ a_{12}(R, \sqrt{\tau}, f) &= R \frac{\sinh\left(\sqrt{\tau j 2\pi f}\right)}{\sqrt{\tau j 2\pi f}} \\ a_{21}(R, \sqrt{\tau}, f) &= \frac{\sqrt{\tau j 2\pi f}}{R} \sinh\left(\sqrt{\tau j 2\pi f}\right) \end{aligned}$$

les trois fonctions qui définissent les composantes de la matrice de transfert de ce quadripôle thermique. La figure 4-15 présente, pour les deux types de représentation considérées, le pessimisme de deux fonctions d'inclusion. Les images de $[0, 3]$ par les trois fonctions d'inclusion naturelle sont représentées par des cadres en pointillés, qui correspondent à l'évaluation, dans le cas de la représentation rectangulaire :

$$\begin{aligned} [a_{11}]_R([x], 1) &= (\cos([x]) \cosh([x]), \sin([x]) \sinh([x])). \\ [a_{12}]_R([x], 1) &= \left(\frac{\cos([x]) \sinh([x]) + \cosh([x]) \sin([x])}{2[x]}, \frac{\cosh([x]) \sin([x]) - \sinh([x]) \cos([x])}{2[x]} \right) \\ [a_{21}]_R([x], 1) &= ((\sinh([x]) \cos([x]) - \cosh([x]) \sin([x])) [x], [x] (\sinh([x]) \cos([x]) + \cosh([x]) \sin([x]))) \end{aligned}$$

et pour la représentation polaire :

$$\begin{aligned} [\alpha] &= \frac{-\sinh([x]) \cos([x]) + \cosh([x]) \sin([x])}{\sinh([x]) \cos([x]) + \cosh([x]) \sin([x])} \\ [a_{11}]_P([x], 1) &= \left(\sqrt{\frac{\cos(2[x]) + \cosh(2[x])}{2}}, \arctan(\tan([x]) \tanh([x])) \right) \\ [a_{12}]_P([x], 1) &= \left(\frac{1}{2} \frac{1}{[x]} \sqrt{\cosh(2[x]) - \cos(2[x])}, \arctan([\alpha]) \right) \\ [a_{21}]_P([x], 1) &= \left([x] \sqrt{\cosh(2[x]) - \cos(2[x])}, \arctan\left(\frac{\pi}{2} - [\alpha]\right) \right). \end{aligned}$$

En gras, nous avons représenté les images de l'intervalle $[0, 3]$ obtenues par des fonctions d'inclusion améliorées, obtenues à partir d'une étude de la monotonie des fonctions, en s'appuyant sur le théorème 2.2.1. Comme l'illustre la figure 4-15, ces fonctions sont optimales : le problème de dépendance est bien contrôlé, au détriment de l'effet d'enveloppement dû à la manipulation de pavés.

Remarque 4.4.4 Remarquons cependant que l'approche choisie décrit le module comme une fonction toujours positive, ce qui induit une discontinuité dans l'expression de l'argument. Cette discontinuité peut être réduite en autorisant le module à être négatif, d'où une évaluation pessimiste du module.

Nous allons utiliser principalement la représentation rectangulaire. Il est bien entendu possible d'effectuer la même étude à l'aide d'une représentation polaire. La combinaison de ces deux représentations peut, au détriment de la complexité du calcul, améliorer la qualité du résultat.

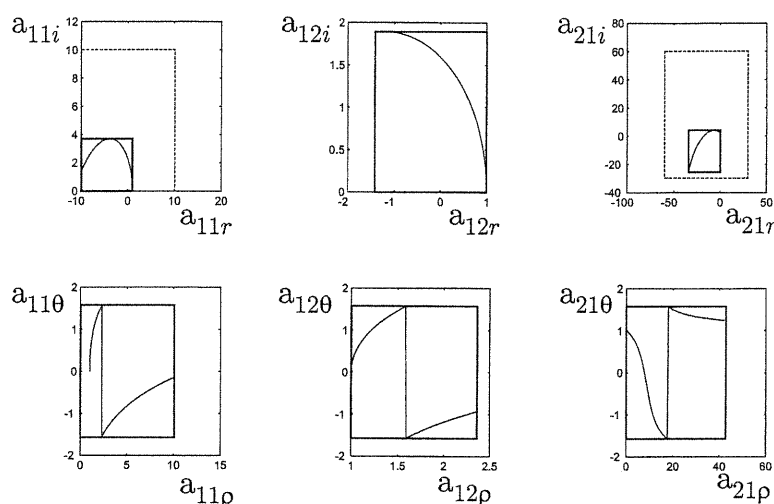


Figure 4-15: Phénomène d’enveloppement dans les représentations rectangulaire (en haut) et polaire (en bas) des composantes a_{11} , a_{12} et a_{21} de la matrice du quadripôle thermique de l’exemple 4.4.2. Le cadre en pointillés correspond à l’évaluation naturelle, le cadre en gras à notre évaluation, qui est optimale. Pour a_{12i} , $a_{11\theta}$, $a_{12\theta}$ et $a_{21\theta}$, les deux évaluations sont égales.

4.4.3.2 Fonctions d’inclusion

Il s’agit maintenant d’établir une fonction d’inclusion pour le modèle complet H représentant la chaîne de quadripôles. La première étape de la démarche d’élaboration d’une fonction d’inclusion consiste à réduire le pessimisme dû au phénomène de dépendance. Les pavés que nous allons considérer sont de taille initiale importante. Afin d’obtenir un contracteur performant pour des pavés de taille faible et élevée, nous allons mettre en œuvre, outre les fonctions d’inclusions naturelles, la forme centrée et le contracteur $C_{1\uparrow}$, que nous allons faire collaborer. Il s’agit donc de sélectionner une expression formelle de H minimisant les occurrences du vecteur de paramètres étendu \mathbf{x} .

Pour cela, nous allons comparer diverses expressions formelles de H . Deux approches se distinguent, que nous dénommerons par l’approche par fonction de transfert et l’approche par chaîne de quadripôles.

Approche par fonction de transfert Elle consiste à développer complètement le modèle H et à minimiser les occurrences de chacune des neuf grandeurs. Ceci nous fournit donc théoriquement déjà $9! = 362880$ expressions. Nous sélectionnons donc neuf expressions, chacune d’elles minimisant l’une des neuf grandeurs. Par exemple, l’expression du modèle qui minimise les occurrences du temps de Fourier du PVC est donnée par

$$\eta_k = \frac{B(f_k)}{A(f_k)},$$

où $B(f_k)$ est exprimé sous la forme (4.31), et

$$A(f_k) = m_{11}(\mathbf{q}) a_{11}(p_1, p_2, f_k) + m_{12}(\mathbf{q}) a_{12}(p_1, p_2, f_k) + m_{21}(\mathbf{q}) a_{21}(p_1, p_2, f_k), \quad (4.36)$$

où les coefficients $m_{11}(\mathbf{q})$, $m_{12}(\mathbf{q})$ et $m_{21}(\mathbf{q})$ sont indépendants de \mathbf{p} et ont pour expression, donnée à titre indicatif:

$$m_{11}(\mathbf{q}) = (1 + 2q_3q_7) a_{11}(q_1, q_2, f_k) a_{11}(q_4, q_5, f_k) \quad (4.37)$$

$$+ a_{11}(q_1, q_2, f_k) (2q_3a_{21}(q_4, q_5, f_k) + q_7a_{12}(q_4, q_5, f_k)) \\ + a_{12}(q_1, q_2, f_k) (a_{21}(q_4, q_5, f_k) + q_7a_{11}(q_4, q_5, f_k))$$

$$m_{12}(\mathbf{q}) = a_{11}(q_1, q_2, f_k) (a_{21}(q_4, q_5, f_k) + q_7a_{11}(q_4, q_5, f_k)) \quad (4.38)$$

$$m_{21}(\mathbf{q}) = (a_{11}(q_1, q_2, f_k) q_3 + a_{12}(q_1, q_2, f_k)) \quad (4.39)$$

$$(a_{11}(q_4, q_5, f_k) (1 + q_3q_7) + q_3a_{21}(q_4, q_5, f_k) + q_7a_{12}(q_4, q_5, f_k)) \quad (4.40)$$

Cette approche permet, par développement et factorisation, de diminuer énormément le nombre d'occurrences des paramètres, mais elle n'est pas du tout adaptée à une modification éventuelle du système, et est relativement coûteuse en termes de temps de développement. Par contre, l'expression explicite de H permet de mettre en œuvre simplement la forme centrée, et le graphe des contraintes.

Approche par chaîne de quadripôles Elle consiste à garder la structure du système, ce qui d'une part, permet de l'adapter simplement à une modification éventuelle du modèle, et d'autre part de caractériser l'influence de chaque quadripôle. Elle peut être envisagée de deux façons, selon que le produit matriciel est effectué de la gauche vers la droite, ou le contraire. Dans le premier cas, les occurrences du cuivre sont minimisées, dans le second cas on minimise les occurrences du laiton. Les occurrences des caractéristiques du PVC, au centre du dispositif, sont approximativement constantes. Cependant, l'élaboration du graphe des contraintes fait intervenir de nombreux cycles.

Les atouts des deux approches étant complémentaires, nous avons choisi de combiner ces 11 expressions formelles. Ces fonctions d'inclusion, ainsi que les formes centrées et le contracteur \mathcal{C}_{\uparrow} seront utilisés dans SIVIAP.

Notre démarche s'articule autour de plusieurs étapes. Pour commencer, nous allons déterminer l'ensemble des valeurs de \mathbf{p} acceptables lorsque les facteurs expérimentaux sont égaux à leur valeur nominale. Ceci nous permet de comparer les résultats obtenus sous l'hypothèse d'erreurs bornées et ceux obtenus dans un contexte statistique.

4.4.4 Connaissance exacte des facteurs expérimentaux

Supposer que chaque facteur expérimental q_i est connu de façon exacte revient à lui attribuer une valeur numérique constante \hat{q}_i . Dans l'expérience, deux types de facteurs expérimentaux peuvent être distingués.

Certains facteurs représentent une grandeur physique ou géométrique dont on peut affirmer qu'elle prend une valeur numérique constante mais inconnue au cours de l'expérience. L'intervalle *a priori* associé à cette grandeur caractérise alors l'incertitude entachant leur détermination. Il s'agit par exemple des facteurs q_i , $i = 1, 2, 4, 5, 6$. Dans [TK98], il a été choisi de les fixer à leur valeur nominale, calculée à partir des valeurs des centres des intervalles associés aux grandeurs physiques et géométriques.

D'autres facteurs sont par nature incertains, et il est difficile de leur attribuer une valeur nominale. Par exemple, ils peuvent varier au cours de l'expérience dans une plage qui correspond à l'intervalle associé, ou caractériser une incertitude structurelle. Leur attribuer une valeur numérique exacte constitue donc une hypothèse de modélisation. Dans [TK98], le coefficient d'échange q_7 et la résistance de la colle q_3 ont ainsi été fixés à des valeurs arbitraires: la valeur

de \hat{q}_7 est choisie comme la borne inférieure \underline{Q}_7 de l'intervalle Q_7 associé, et la valeur $\hat{q}_3 = \frac{e_{co}}{\lambda_{co}}$ a été choisi nulle, alors que l'épaisseur de la colle e_{co} est bien entendu, strictement positive.

L'estimation garantie va nous permettre d'étudier l'influence de ces choix sur le résultat de l'estimation. Dans chaque cas, la fonction d'inclusion $[\eta_k]$ de η_k choisie est obtenue comme

$$[\eta_k]([\mathbf{p}], \hat{\mathbf{q}}) = \frac{B(\hat{\mathbf{q}}, f_k)}{[A]([\mathbf{p}], \hat{\mathbf{q}}, f_k)},$$

où $B(f_k)$ est exprimé sous la forme (4.31), et

$$\begin{aligned} [A]([\mathbf{p}], \hat{\mathbf{q}}, f_k) &= m_{11}(\hat{\mathbf{q}}, f_k) [a_{11}]([p_1], [p_2], f_k) \\ &\quad + m_{12}(\hat{\mathbf{q}}, f_k) [a_{12}]([p_1], [p_2], f_k) \\ &\quad + m_{21}(\hat{\mathbf{q}}, f_k) [a_{21}]([p_1], [p_2], f_k). \end{aligned} \quad (4.41)$$

Les coefficients $m_{11}(\hat{\mathbf{q}}, f_k)$, $m_{12}(\hat{\mathbf{q}}, f_k)$ et $m_{21}(\hat{\mathbf{q}}, f_k)$ sont ponctuels et donnés par (4.37), (4.38) et (4.39). D'après le paragraphe précédent, nous disposons de fonctions d'inclusion $[a_{11}](\cdot)$, $[a_{12}](\cdot)$ et $[a_{21}](\cdot)$ optimales. La fonction d'inclusion résulte donc de la combinaison linéaire de trois fonctions d'inclusion optimales, et seul le phénomène d'enveloppement introduit du pessimisme.

La première partie de cette étude analyse l'influence du choix de \hat{q}_7 .

4.4.4.1 Coefficient d'échange

Dans un premier temps, nous allons définir deux valeurs numériques distinctes pour le vecteur des facteurs expérimentaux, correspondant aux deux valeurs extrêmes du coefficient d'échange. La première valeur correspond au contexte de [TK98]:

$$\hat{\mathbf{q}}^1 = \left(2.10^{-5} \quad \frac{2}{\sqrt{33}} \quad 0 \quad \frac{0.010}{389} \quad \frac{10}{\sqrt{114}} \quad 0.005 \quad 5 \right)^T,$$

qui a fourni l'estimée suivante¹⁰ du vecteur de paramètres \mathbf{p} , par minimisation d'un critère quadratique:

$$\hat{\mathbf{p}} = \begin{pmatrix} 0.030 \\ 14.451 \end{pmatrix} \in \begin{pmatrix} \frac{0.005}{0.1667 \pm 0.0005} \\ \frac{5}{0.346 \pm 0.029} \end{pmatrix} = \begin{pmatrix} [0.0299, 0.03009] \\ [13.33, 15.77] \end{pmatrix}. \quad (4.42)$$

La seconde valeur que nous considérerons est définie comme suit:

$$\hat{\mathbf{q}}^2 = \left(2.10^{-5} \quad \frac{2}{\sqrt{33}} \quad 0 \quad \frac{0.010}{389} \quad \frac{10}{\sqrt{114}} \quad 0.005 \quad 10 \right)^T.$$

Cas $\mathbf{q} = \hat{\mathbf{q}}^1$ L'estimée (4.42) est obtenue dans [TK98] par optimisation de critère de distance moyenne sur l'ensemble des trente mesures, sous l'hypothèse $\hat{\mathbf{q}} = \hat{\mathbf{q}}^1$. Vérifions dans un premier temps si le vecteur de sorties du modèle $\boldsymbol{\eta}(\hat{\mathbf{p}}, \hat{\mathbf{q}}^1)$ est compatible avec le vecteur de données intervalles $[\mathbf{y}]$. Sur la figure 4-16 sont représentées par des ronds (o) les parties réelle et imaginaire des composantes $\eta_k(\hat{\mathbf{p}}, \hat{\mathbf{q}}^1)$ en fonction de ω_k , pour k variant de 1 à 5, tandis que les traits continus fins délimitent le gabarit imposé par $[\mathbf{y}]$.

¹⁰Suivant la technique utilisée, les estimées obtenues dans [TK98] sont différentes. Comme les résultats sont complètement analogues pour les deux estimées obtenues, nous ne présentons ici les résultats que pour une des deux estimées.

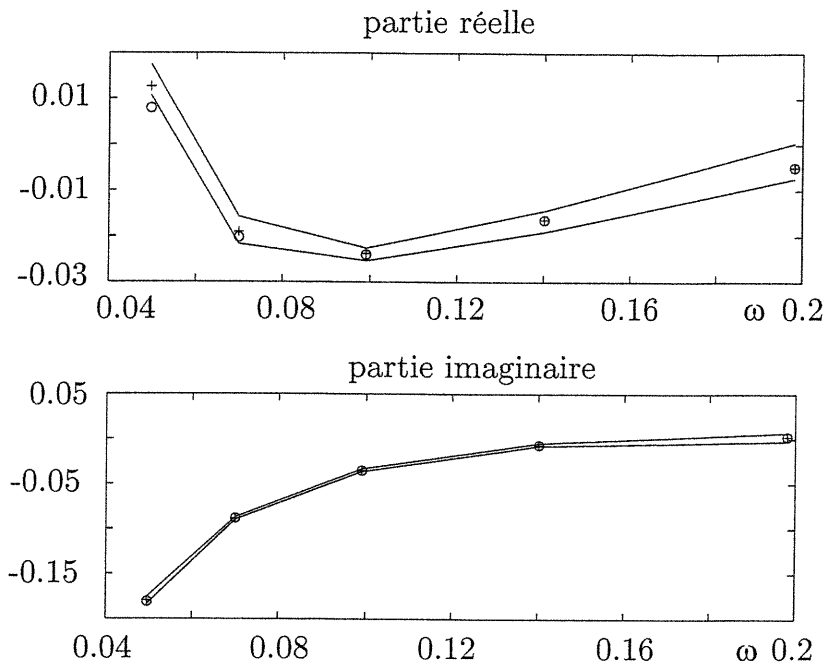


Figure 4-16: Parties réelles et imaginaires des sorties de modèle $\eta_k(\hat{\mathbf{x}}^1)$ (o) et $\eta_k(\hat{\mathbf{x}}^2)$ (+) du gabarit imposé par les données intervalles $[y_k]$ (-) en fonction de la pulsation ω_k .

Le vecteur étendu $\hat{\mathbf{x}}^1 = (\hat{\mathbf{p}}, \hat{\mathbf{q}}^1)$ correspondant au cadre de [TK98] est compatible pour les données intervalles $[y_k]$ avec $k = 2 \dots 5$, mais est incompatible avec $[y_1]$. L'estimée $\hat{\mathbf{p}}$ obtenue dans [TK98] ne permet pas de satisfaire le gabarit imposé. Deux hypothèses peuvent être formulées.

- L'intervalle $[y_1]$ est trop optimiste, il n'existe alors aucune valeur du vecteur de paramètres \mathbf{p} telle que $\boldsymbol{\eta}(\mathbf{p}, \hat{\mathbf{q}}^1) \in [y]$;
- Les hypothèses conduisant à poser $\mathbf{q} = \hat{\mathbf{q}}^1$ ne sont pas satisfaites. En particulier, il n'est pas réaliste d'affecter à q_7 la valeur 5 dans ce contexte expérimental.

L'approche ensembliste va nous permettre de répondre à cette question.

Soit \mathbb{S}_2^1 l'ensemble des valeurs de \mathbf{p} compatibles avec $[y]$:

$$\mathbb{S}_2^1 = \{\mathbf{p} \in \mathbb{P} \mid \boldsymbol{\eta}(\mathbf{p}, \hat{\mathbf{q}}^1) \in [y]\}.$$

Si \mathbb{S}_2^1 est vide, nous aurons prouvé que nos hypothèses sont incompatibles, ou qu'une donnée est aberrante, ce que confirme SIVIAP en prouvant en 2.5 secondes pour $\varepsilon = 0.02$ que $\bar{\mathbb{S}}_2^1$ est vide. Il n'existe aucune valeur de \mathbf{p} dans \mathbb{P} compatible avec le vecteur de données intervalles $[y]$, et les hypothèses conduisant au choix de $\hat{\mathbf{q}}^1$ sont incompatibles avec le vecteur de données expérimentales $[y]$.

Deux solutions sont alors possibles, qui consistent à éliminer des composantes de $[y]$, ou à changer de valeur numérique pour \mathbf{q} . Nous décidons de changer la valeur de \hat{q}_7 .

Cas $\mathbf{q} = \hat{\mathbf{q}}^2$ Nous nous proposons alors d'étudier le comportement du modèle obtenu dans le même contexte expérimental mais en imposant $q_7 = \bar{Q}_7 = 10$. Simulons les sorties de modèle obtenues avec le vecteur de paramètres $\hat{\mathbf{x}}^2 = (\hat{\mathbf{p}}, \hat{\mathbf{q}}^2)$. Ces sorties $\eta_k(\hat{\mathbf{x}}^2)$ obtenues sont représentées

sur la figure 4-16. Le vecteur étendu $\widehat{\mathbf{x}}^2$ est compatible avec le vecteur de données intervalles $[\mathbf{y}]$. Nous pouvons alors conclure que l'estimée $\widehat{\mathbf{p}}$, obtenue dans un cadre statistique, par optimisation de l'erreur moyenne quadratique, est également acceptable dans le contexte à erreurs bornées à condition de remplacer $h = 5$ par $h = 10$.

Le domaine d'incertitude associé à $\widehat{\mathbf{p}}$ calculé dans le cadre statistique est représenté par l'ellipse sur la figure 4-17. Il correspond à une région de confiance pour un niveau de confiance fixé à 95%. Dans le contexte à erreurs bornées, lorsqu'aucune information d'ordre statistique n'est disponible, aucune valeur ponctuelle de \mathbf{p} parmi l'ensemble des valeurs compatibles avec l'expérience n'est privilégiée. Nous cherchons alors s'il existe d'autres valeurs de \mathbf{p} compatibles avec le vecteur de données intervalles $[\mathbf{y}]$. En posant

$$\mathbb{S}_2^2 = \{ \mathbf{p} \in \mathbb{P} \mid \boldsymbol{\eta}(\mathbf{p}, \widehat{\mathbf{q}}^2) \in [\mathbf{y}] \},$$

le domaine d'incertitude associé à \mathbf{p} correspond alors directement à l'enveloppe de \mathbb{S}_2^2 .

En 17.8 secondes pour $\varepsilon = 0.01$, SIVIAP fournit deux sous-pavages $\underline{\mathbb{S}}_2^2$ et $\overline{\mathbb{S}}_2^2$ non-vides, représentés sur la figure 4-17 pour l'espace de recherche *a priori* \mathbb{P} . Le cadre en pointillés représente la plage de dispersion fournie par la littérature. Remarquons que $\widehat{\mathbf{p}}$ est bien solution puisqu'il appartient à $\underline{\mathbb{S}}_2^2$, et est un proche voisin du centre de $\underline{\mathbb{S}}_2^2$. La projection de $\overline{\mathbb{S}}_2^2$ sur les axes p_1 et p_2 fournit un encadrement extérieur de l'incertitude associée à chaque paramètre :

$$\begin{aligned} p_1 &\in [0.0290, 0.0308] \\ p_2 &\in [14.0468, 15.0469] \end{aligned}$$

d'où

$$\begin{aligned} a_p &= 0.1186 \times 10^{-6} \pm 7.7\%, \\ \lambda_p &= 0.1672 \pm 3.2\%. \end{aligned}$$

4.4.4.2 Résistance thermique de la colle

L'ensemble \mathbb{S}_2^2 , obtenu dans le cas où le vecteur des facteurs expérimentaux \mathbf{q} prend sa valeur nominale, contient l'estimée obtenue par une approche statistique, qui a fait abstraction de l'influence de la colle (q_3 est fixé à zéro). Cette hypothèse n'est *a priori* pas justifiée, car

$$R_{co} \in \frac{[e_{co}]}{[\lambda_{co}]},$$

et e_{co} , mesurée au microscope, est supérieure à $30.10^{-6}m$.

Comme \mathbb{S}_2^2 n'est pas vide, ce modèle simplifié n'est pas invalidé, et permet de décrire de façon satisfaisante les données obtenues. Afin d'étudier l'ensemble des valeurs que peut prendre R_{co} , nous cherchons à estimer l'ensemble \mathbb{S}_3 défini par

$$\mathbb{S}_3 = \left\{ (\mathbf{p}, q_3) \in \mathbb{P} \times \mathbb{Q}_3 \mid \begin{array}{l} \boldsymbol{\eta}(\mathbf{x}) \in [\mathbf{y}], \\ \mathbf{x} = (p_1, p_2, q_1^*, q_2^*, q_3, q_4^*, q_5^*, q_6^*, q_7^*) \end{array} \right\}.$$

SIVIAP fournit en 54 secondes pour $\varepsilon = 0.05$ les sous-pavages $\underline{\mathbb{S}}_3$ et $\overline{\mathbb{S}}_3$ représentés sur la figure 4-18, pour un espace de recherche supérieur au domaine de recherche *a priori* fourni. Il apparaît que l'ensemble des valeurs de R_{co} dans l'intervalle $[0, 5.10^{-5}]$ est compatible avec les données

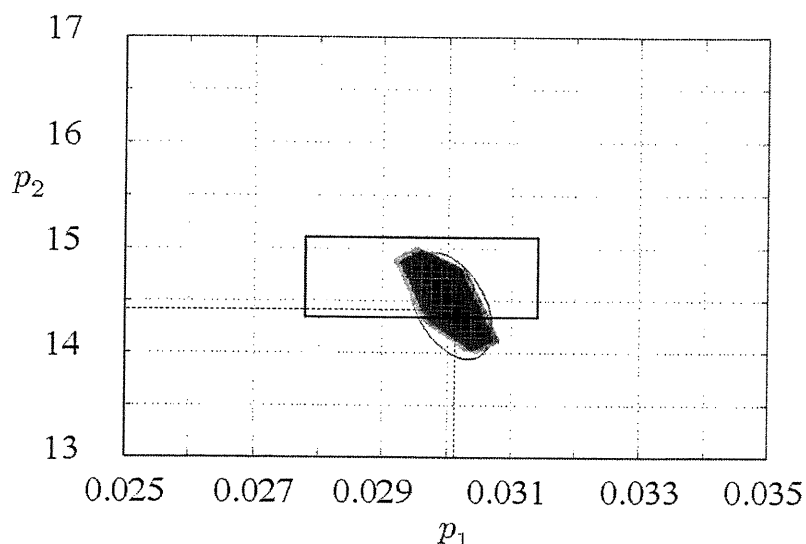


Figure 4-17: Sous-pavages $\underline{\mathbb{S}}_2^2$ (en noir) et $\overline{\mathbb{S}}_2^2$ (en gris) obtenus par SIVIAP dans le cas où les facteurs expérimentaux sont supposés connus de façon exacte. L'estimée $\hat{\mathbf{p}}$ et la zone de confiance obtenue par l'approche statistique sont repérées. Le cadre correspond à la plage de dispersion fournie dans la littérature

expérimentales recueillies.

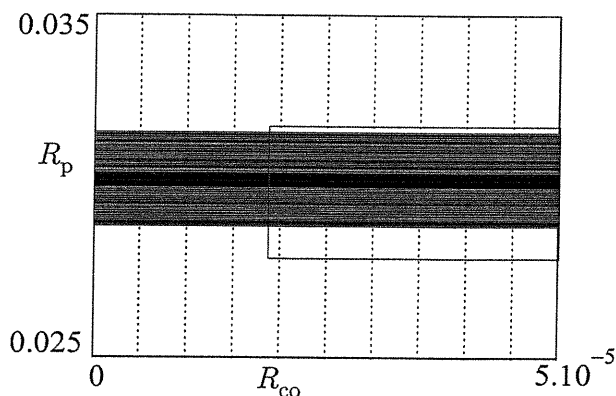


Figure 4-18: Projection dans le plan (q_3, p_1) des sous-pavages $\underline{\mathbb{S}}_3$ et $\overline{\mathbb{S}}_3$ obtenus par SIVIA. L'espace de recherche *a priori* $\mathbb{Q}_3 \times \mathbb{P}_1$ est représenté par un rectangle en pointillés.

L'hypothèse simplificatrice $R_{co} = 0$, qui consiste à négliger la présence des couches de colle, sera effectuée au paragraphe 4.4.6.

Jusqu'à présent, nous avons supposé sans justification que les facteurs expérimentaux étaient certains et connus exactement. Nous allons maintenant réaliser la même étude pour des facteurs expérimentaux \mathbf{q} incertains.

4.4.5 Prise en compte des incertitudes intrinsèques

Prendre en compte l'incertitude associée aux facteurs expérimentaux q_i fait passer le nombre de grandeurs incertaines présentes dans le modèle de 2 à 9. Il est alors fortement conseillé de s'assurer de l'identifiabilité de \mathbf{x} (voir le chapitre 5).

4.4.5.1 Identifiabilité de \mathbf{x}

L'étude de l'identifiabilité structurelle des paramètres thermo-physiques de deux couches de matériaux homogènes en contact a été effectuée dans [RTFJC01]. Elle a abouti pour deux contextes expérimentaux différents, aux conclusions suivantes :

1. si les températures de surface entrée-sortie sont connues, les paramètres identifiables sont les temps caractéristiques de Fourier τ de chaque couche, les rapports des résistances surfaciques aux différentes interfaces, et le nombre de Biot $B_i = hR_{\text{arrière}}$.
2. si une température de surface et un flux surfacique sont connus, alors on peut identifier les temps caractéristiques, les résistances surfaciques et le coefficient d'échange surfacique h .

La généralisation au cas multi-couches, qui n'est pas achevée à l'heure actuelle, fait l'objet de nombreux débats. Grâce à l'approche ensembliste garantie, qui permet de calculer l'ensemble de toutes les valeurs des paramètres compatibles avec l'expérience, il est possible de contourner cette étape d'identifiabilité, et d'identifier ces paramètres, qu'ils soient identifiables ou non. Remarquons cependant que même si la généralisation au cas multi-couches, qui pour l'instant est une conjecture, est vérifiée, alors dans notre modélisation à deux chaînes \mathcal{A} et \mathcal{B} , le vecteur de paramètres \mathbf{x} ne peut pas être structurellement identifiable. En effet, si cette conjecture est vérifiée, seules les grandeurs

$$\tau_c, \tau_p, \tau_l, h\xi R_c, \frac{\xi R_c}{R_c} = \xi, \frac{R_c}{R_{co}}, \frac{R_p}{R_p}, \frac{R_p}{R_{co}} \text{ et } \frac{R_{co}}{R_l}$$

peuvent être structurellement identifiables.

Ceci permet de souligner l'intérêt d'utiliser une méthode ensembliste garantie, puisque les résultats seront valides malgré cet éventuel défaut d'identifiabilité, et de remarquer que le vecteur des paramètres étendu \mathbf{x} ne peut devenir structurellement identifiable sans introduction d'information supplémentaire. Par exemple, on peut essayer de connaître plus précisément une résistance thermique. Nous rencontrerons ce dernier cas au cours du paragraphe 4.4.5.3.

4.4.5.2 Estimation ensembliste

Deux approches sont alors envisageables.

La première consiste à estimer conjointement le vecteur de facteurs expérimentaux \mathbf{q} et le vecteur de paramètres \mathbf{p} , en déterminant l'ensemble \mathbb{S} des vecteurs de paramètres étendus \mathbf{x} compatibles avec les données

$$\mathbb{S} = \{\mathbf{x} \in \mathbb{X}, \boldsymbol{\eta}(\mathbf{x}) \in [\mathbf{y}]\}. \quad (4.43)$$

La caractérisation de ce sous-ensemble de \mathbb{R}^9 , qui de plus peut être très volumineux en raison de la non-identifiabilité de \mathbf{x} , est gourmande en temps de calcul et en mémoire. De plus, elle produit une information qui n'est pas demandée par le cahier des charges, à savoir une identification des facteurs expérimentaux.

La deuxième approche est plus modeste. Elle consiste à caractériser directement l'ensemble Π des valeurs de (R_p, τ_p) dans leur domaine de recherche *a priori* compatibles avec une valeur de \mathbf{q} dans \mathbb{Q} :

$$\Pi = \{\mathbf{p} \in \mathbb{P}, \exists \mathbf{q} \in \mathbb{Q}, \boldsymbol{\eta}(\mathbf{p}, \mathbf{q}) \in [\mathbf{y}]\}. \quad (4.44)$$

Cet ensemble Π est la projection de l'ensemble \mathbb{S} . Une représentation géométrique des ensembles Π et \mathbb{S} , dans le cas où $\dim \mathbf{p} = 2$ et $\dim \mathbf{q} = 1$, est proposée à la figure 4-19. Comme nous l'avons déjà signalé, caractériser directement Π revient à réaliser un compromis : les facteurs expérimentaux ne sont plus estimés, au profit d'un temps de calcul et d'une charge en mémoire beaucoup plus faible, puisque la zone d'accumulation du sous-pavage décrivant Π est un ensemble de dimension 1. C'est cette approche que nous avons suivie dans ce document.

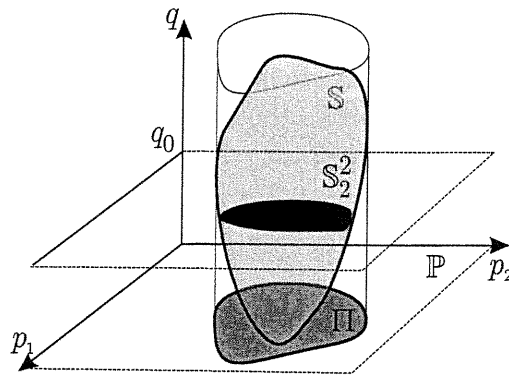


Figure 4-19: Ensembles \mathbb{S} , \mathbb{S}_2^2 et Π . Par souci de simplicité, on a posé $\dim \mathbf{p} = 2$ et $\dim \mathbf{q} = 1$.

Nous avons également fait figurer sur la figure 4-19, l'ensemble \mathbb{S}_2^2 calculé dans le cas où \mathbf{q} est supposé certain. Cet ensemble, qui correspond à une coupe de \mathbb{S} , vérifie

$$\mathbb{S}_2^2 \subset \Pi. \quad (4.45)$$

L'équation (4.45) traduit le fait que l'incertitude \mathbb{Q}_i dégrade la qualité de la mesure de \mathbf{p} . On cherche alors l'ensemble Π défini par (4.44), c'est-à-dire la projection de \mathbb{S} sur le plan \mathbb{P} . Pour cela, nous utilisons les algorithmes de partitionnement de la famille PROJECT présentés au chapitre précédent. Comme l'évaluation de la fonction d'inclusion associée à la chaîne est très pessimiste, nous utiliserons l'algorithme PROJECT2. Cependant, l'existence de Π est assurée, puisque d'après (4.45), \mathbb{S}_2^2 est une approximation intérieure non vide de Π . Enfin, seule l'approximation extérieure $\overline{\Pi}$ est nécessaire au calcul de l'incertitude associée à \mathbf{p} (propriété 4.2.2). Le pavé enveloppe obtenu est

$$[\Pi] = [0.0265, 0.0336] \times [13.125, 15.5625]$$

soit un volume de 0.016, ce qui fournit

$$a_p = [1.024, 1.463] 10^{-7} = 1.2435 \cdot 10^{-7} \pm 17.7\% \quad (4.46)$$

$$\lambda_p = [0.1482, 0.1895] = 0.16885 \pm 12.3\% \quad (4.47)$$

Cette augmentation de volume correspond à une dégradation de la mesure indirecte. Bien entendu, elle dépend du facteur expérimental q_i concerné, mais aussi de la taille de \mathbb{Q}_i . Nous

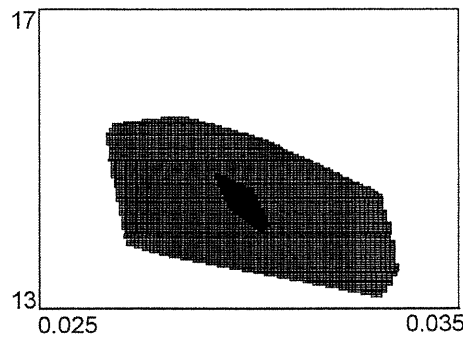


Figure 4-20: Approximations intérieure $\underline{\Pi}$ et extérieure $\overline{\Pi}$ de l'ensemble Π obtenues pour $\varepsilon = 0.05$ en $1h54mn$ sur un Pentium IV à $1.7 GHz$

proposons dans le paragraphe suivant, une étude très simple afin de quantifier cette influence. Ceci sera alors proposé comme une première étape en vue d'une planification d'expériences ou d'une réduction du modèle ultérieure (paragraphe 4.4.6).

4.4.5.3 Influence des incertitudes

La prise en compte de l'incertitude du facteur expérimental q_i dégrade la qualité de la mesure de \mathbf{p} , de deux manières :

- physique : il est nécessaire de posséder une connaissance très précise du paramètre q_i pour pouvoir mesurer les caractéristiques du PVC (facteur expérimental très influent). De plus, si le vecteur des paramètres étendu \mathbf{x} n'est pas identifiable, il est possible qu'il existe des groupes de symétries liant les composantes de \mathbf{p} à celles de \mathbf{q} ;
- purement mathématique : si q_i est multi-occurent, la fonction d'inclusion peut devenir pessimiste, et dégrader la taille de l'approximation intérieure de l'ensemble solution.

Dans un contexte statistique, l'évaluation de la chaîne consiste à la soumettre à une batterie de tests préétablis, et à caractériser sa performance par la dispersion des résultats obtenus. Il est alors possible de déterminer les facteurs expérimentaux pour lesquels une mesure précise de \mathbf{x} requiert une incertitude plus faible. Nous présentons ici quelques outils simples adaptés au contexte d'erreurs bornées, dans le cas précis où les données ont déjà été recueillies (l'évaluation *a priori* est traitée au chapitre 6).

Supposons momentanément qu'il existe une vraie valeur $\mathbf{q}^* \in \mathbb{Q}$ du vecteur des facteurs expérimentaux \mathbf{q} , et notons

$$\mathbb{S}_2^* = \{\mathbf{p} \in \mathbb{P} \mid \eta(\mathbf{p}, \mathbf{q}^*) \subset [\mathbf{y}]\},$$

l'ensemble des valeurs de \mathbf{p} compatibles avec ce contexte.

Soient les fonctions $f_i, i = 1, \dots, 7$ définies par

$$\begin{aligned} f_i & : \quad \mathbb{R}^7 \rightarrow \mathbb{I}\mathbb{R}^7 \\ \mathbf{q} & \longmapsto (q_1^*, \dots, q_{i-1}^*, [q_i], q_{i+1}^*, \dots, q_7^*)^T. \end{aligned}$$

et notons

$$\Pi^i = \{\mathbf{p} \in \mathbb{P} \mid \boldsymbol{\eta}(\mathbf{p}, f_i(\mathbf{q})) \subset [\mathbf{y}]\},$$

les ensembles obtenus en substituant l'expression $q_i = q_i^*$ par $q_i = [q_i]$ dans l'hypothèse $\mathbf{q} = \mathbf{q}^*$.

L'ensemble Π^i caractérise le support de \mathbf{p} dans l'hypothèse où seule la valeur du facteur expérimental q_i est incertaine. On a donc

$$\mathbb{S}_2^* \subset \Pi^i.$$

Cet ensemble va nous permettre de quantifier l'influence de l'incertitude de q_i , puisque

- quand la taille $w([q_i])$ augmente, $\text{Vol}(\Pi^i)$ augmente,
- si $\text{Vol}(\Pi^i) = \text{Vol}(\mathbb{S}_2^*)$, alors la prise en compte de l'incertitude sur q_i n'entraîne pas de modification notable de l'ensemble des paramètres compatibles avec l'expérience.

Nous choisissons alors comme critère la variation de volume

$$s_i^* = \frac{\text{Vol}(\Pi^i)}{\text{Vol}(\mathbb{S}_2^*)} \geq 1.$$

Si $s_i^* > s_j^*$, alors la prise en compte de l'incertitude associée au facteur q_i dégrade de façon plus conséquente la mesure indirecte de \mathbf{p} à l'aide du dispositif de mesure que la prise en compte de l'incertitude associée au paramètre q_j .

Comme la valeur \mathbf{q}^* est inconnue, nous allons approximer le calcul de s_i^* en sélectionnant une valeur de référence pour \mathbf{q} . Bien entendu, si cette valeur n'est pas compatible avec l'expérience, alors \mathbb{S}_2 est vide et $s_i^* \rightarrow \infty$. Comme nous avons montré que $\hat{\mathbf{q}}_2$ était compatible avec l'expérience, nous approximations s_i par

$$s_i = \frac{\text{Vol}(\Pi^i)}{\text{Vol}(\mathbb{S}_2^*)} \geq 1.$$

Les volumes sont particulièrement sensibles à la valeur de ε , et au nombre d'occurrences de q_i dans l'expression de la fonction d'inclusion $[\boldsymbol{\eta}]$. Afin de désensibiliser les résultats à ces deux perturbations, nous utilisons la même valeur de ε et la même fonction d'inclusion pour l'estimation des différents ensembles. Les approximations intérieure et extérieure des Π_i sont représentées à la figure 4-21.

i	1	2	3	4	5	6	7
q_i	R_l	$\sqrt{\tau_l}$	R_{co}	R_c	$\sqrt{\tau_c}$	ξ	h
$[s_i]$	[0.86, 1.17]	[0.87, 1.48]	[0.91, 1.45]	[1.89, 5.31]	[1.83, 7.18]	[0.88, 1.33]	[0.86, 1.55]

Tableau 4.6: Influence de la prise en compte de l'incertitude associée à chaque facteur q_i

Il ressort que la prise en compte de l'incertitude associée aux caractéristiques du cuivre est plus importante que celle associée aux autres facteurs, alors que l'incertitude relative associée aux caractéristiques du cuivre était la plus faible (voir le tableau 4.5). Il est donc particulièrement important, pour la qualité de l'estimation, de connaître le plus précisément possible, les paramètres thermo-physiques et géométriques du cuivre. Par ailleurs, le comportement similaire de s_4 et s_5 peut être confirmé par une étude de sensibilité locale. En effet, il apparaît qu'aux

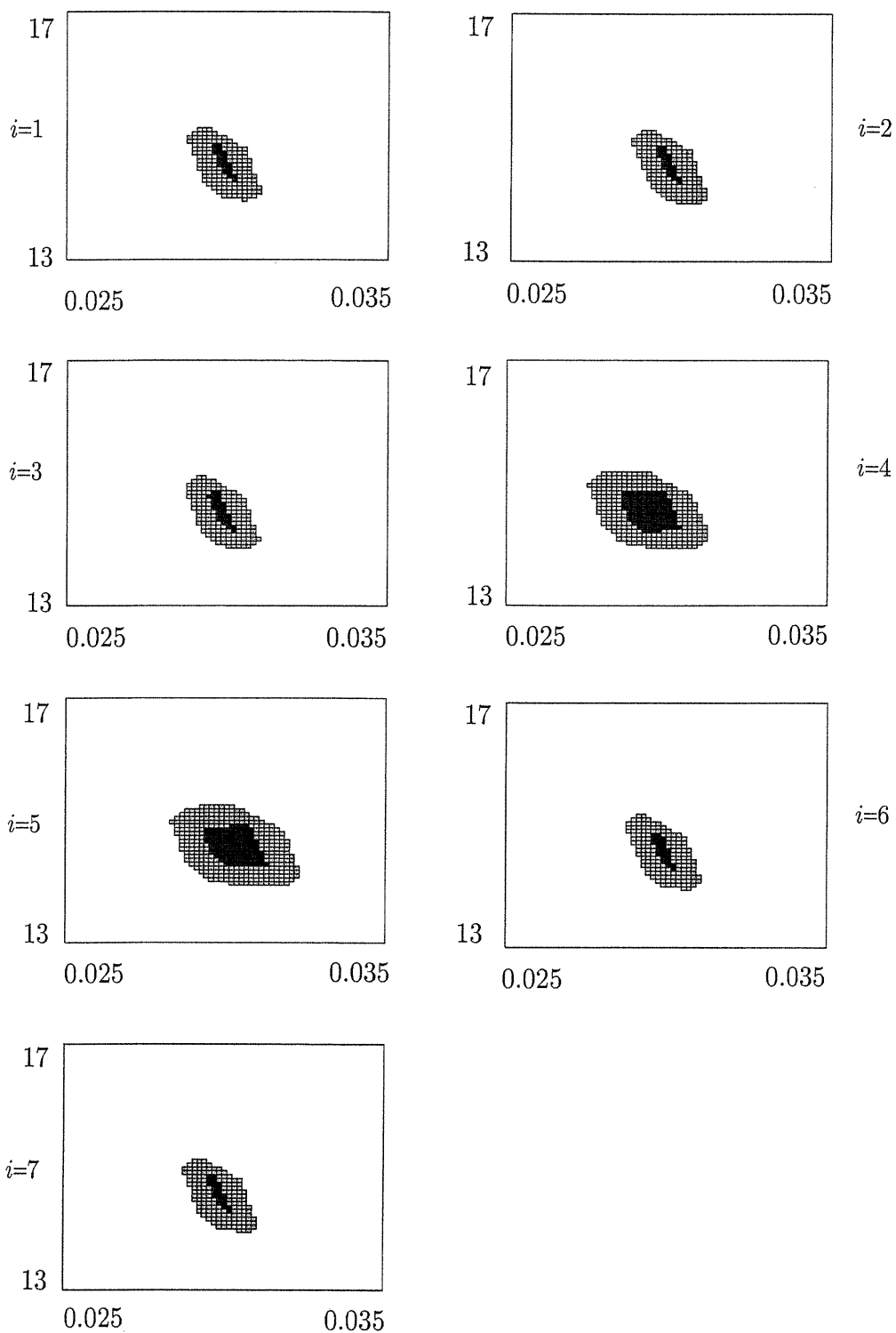


Figure 4-21: Approximations intérieure et extérieure de chaque ensemble $\Pi_i, i = 1, \dots, 7$: les ensembles Π_4 et Π_5 , correspondant aux caractéristiques du cuivre sont les plus volumineux

fréquences d'excitation, le cuivre est fortement capacitif : la grandeur pertinente est alors la capacité du cuivre

$$C_c = \frac{R_c}{\tau_c},$$

et il vient

$$\frac{d \ln H}{d R_c} \approx \frac{d \ln H}{d \tau_c} \approx 0.5 \frac{d \ln H}{d \sqrt{\tau_c}}.$$

En effet, la caractérisation de la projection de \mathbb{S} sur l'espace des paramètres du cuivre, notée $\pi_{67}(\mathbb{S})$ montre que

$$[\pi_{67}(\mathbb{S})] = [0.0267, 0, 0.333] \times [13.375, 15.5625],$$

et le rapport des volumes $\frac{Vol(\pi_{67}(\mathbb{S}))}{Vol(\Pi)} = \frac{0.009}{0.016} \approx 0.6$. Puisque les deux grandeurs introduites dans le modèle sont $(R_c, \tau_c) = \left(R_c, \frac{R_c}{\tau_c}\right)$, il apparaît que la connaissance exacte de R_c permettrait de limiter les variations de volume de l'ensemble solution. Faute de pouvoir réaliser les expériences qui permettraient d'affiner la connaissance de R_c , il a été convenu de fixer la valeur de R_c au milieu du domaine de recherche associé. De plus, nous choisissons la fonction d'inclusion minimisant les occurrences des caractéristiques du PVC et des facteurs expérimentaux autres que la résistance du cuivre, afin d'exploiter au maximum l'information supposée certaine dont on dispose sur le cuivre. Notons enfin que les temps caractéristiques de Fourier, arguments des fonctions hyperboliques, restent des facteurs limitants de l'identifiabilité numérique du modèle. En résumé, cette étape nous a permis de :

- classer les facteurs expérimentaux selon l'influence de leur incertitude associée,
- sélectionner une fonction d'inclusion à partir de ce classement,
- proposer une démarche d'identification des caractéristiques du cuivre afin d'améliorer les performances du dispositif.

4.4.6 Réduction du modèle

Suite aux premiers résultats d'estimation par inversion ensembliste, nous proposons d'apporter les simplifications au modèle du banc d'essai.

4.4.6.1 Comportement de la colle

En accord avec les résultats du paragraphe 4.4.4.2, la présence des deux couches de colle est négligée. Nous supposons donc

$$R_{co} = 0.$$

4.4.6.2 Comportement du cuivre

Le comportement très capacitif du cuivre a été mis en évidence au paragraphe 4.4.5.3. Une application numérique rapide souligne que la plus petite période de la sollicitation multiharmonique est 100 fois plus grande que la constante de temps de Fourier de la plaque de cuivre. On suppose

que le porte-échantillon en cuivre est à une température homogène. En particulier, nous y avons supposé connue la valeur de R_c , et reporté l'ensemble de l'incertitude sur la capacité du cuivre.

Ceci suppose alors que l'on peut utiliser le modèle simplifié suivant :

$$Z_c = \begin{pmatrix} \cosh(\sqrt{\tau_c s}) & \frac{R_c}{\sqrt{\tau_c s}} \sinh(\sqrt{\tau_c s}) \\ \frac{\sqrt{\tau_c s}}{R_c} \sinh(\sqrt{\tau_c s}) & \cosh(\sqrt{\tau_c s}) \end{pmatrix} \simeq \begin{pmatrix} 1 & R_c \\ \frac{\tau_c s}{R_c} & 1 \end{pmatrix}$$

De plus, une application numérique rapide fournit que $R_c \approx 2.10^{-5}$, ce qui est négligeable devant $\frac{\tau_c s}{R_c} \approx 0.1\pi$. Outre le fait de supposer que R_c est connue, nous allons supposer que sa valeur est nulle, ce qui revient à considérer que le cuivre a un comportement purement capacitif, qui se modélise à l'aide de l'impédance thermique

$$Z'_c = \begin{pmatrix} 1 & 0 \\ \rho_c c_c e_c s & 1 \end{pmatrix},$$

où $\rho_c c_c$ est la chaleur volumique au sein de la couche de cuivre, dont la valeur appartient à l'intervalle $[3.32, 3.52] \times 10^6 \text{ W.s.m}^{-3} \cdot \text{K}^{-1}$. Le dispositif est donc désormais modélisé par

$$T_{\text{avant}} = \begin{pmatrix} 1 & 0 \end{pmatrix} Z_l Z_p Z_c \begin{pmatrix} 1 \\ h \end{pmatrix} T_0.$$

4.4.6.3 Identification

L'ensemble obtenu à l'aide de l'algorithme PROJECT2 en 26mn 23 s (pour $\varepsilon = 0.01$) est présenté à la figure 4-22. On obtient

$$\begin{aligned} R_p &\in [0.0276, 0.0323] \\ \sqrt{\tau_p} &\in [13.5, 15.375] \end{aligned}$$

et $\text{Vol}(\overline{\Pi}) = 6.3.10^{-3}$. On obtient alors

$$a_p = [1.049, 1.3827] \cdot 10^{-7} = 1.21585 \pm 13.72\% \quad (4.48)$$

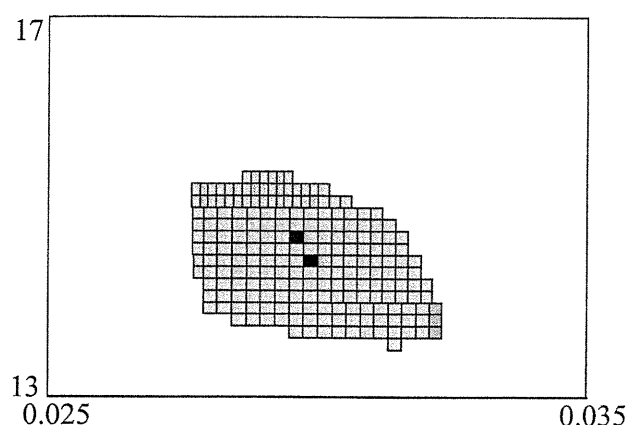
$$\lambda_p = [0.1541, 0.1819] = 0.168 \pm 8.27\% \quad (4.49)$$

à comparer avec les valeurs obtenues avec le modèle précédent (voir (4.46) et (4.47)). D'une part, ces valeurs sont compatibles avec celles obtenues dans le paragraphe précédent, et l'incertitude de mesure obtenue est plus faible. Nous obtenons avec ce modèle, une évaluation plus optimiste de l'incertitude associée à la mesure indirecte de a_p et λ_p .

4.4.7 Conclusions et perspectives

Dans une première étape l'identification ensembliste garantie du vecteur des paramètres \mathbf{p} en supposant les facteurs expérimentaux connus a permis de prouver qu'il existe des valeurs de \mathbf{p} compatibles avec les données intervalles recueillies. L'ensemble \mathbb{S}_2^2 déterminé contient la valeur obtenue par l'approche statistique, et permet d'obtenir simplement l'incertitude de cette mesure indirecte à partir de son volume. Bien que la forme de \mathbb{S}_2^2 soit voisine de l'ellipse de confiance obtenue par l'approche statistique, les hypothèses effectuées sont différentes, et en particulier aucune hypothèse sur le caractère gaussien des données recueillies n'a été effectuée.

Cet ensemble apporte également des informations supplémentaires : le volume de \mathbb{S}_2^2 constitue une borne inférieure garantie de l'ensemble des valeurs de \mathbf{p} compatibles avec l'expérience, dans le

Figure 4-22: Ensemble $\bar{\Pi}$ obtenu par PROJECT2

contexte où les facteurs expérimentaux sont supposés incertains. Dans ce contexte, nous avons montré que la mesure indirecte de \mathbf{p} revient à projeter l'ensemble des grandeurs incertaines compatibles avec l'expérience sur l'espace de recherche *a priori* \mathbb{P} . Nous avons alors caractérisé cette projection Π à l'aide de l'algorithme PROJECT2. Le volume de cet ensemble est supérieur à celui de \mathbb{S}_2^2 , car la prise en compte des incertitudes entachant l'expérience dégrade le résultat de la mesure. Nous avons alors caractérisé chaque facteur expérimental en étudiant l'influence qu'a la prise en compte de son incertitude sur le résultat final de la mesure indirecte. Nous avons ainsi pu poser des jalons pour l'étude d'un nouveau dispositif plus performant.

Dans un premier temps, nous avons souligné l'importance de la prise en compte des incertitudes du cuivre sur le résultat de la mesure indirecte. Il est donc recommandé, dans le but d'améliorer les performances du dispositif, de réaliser une identification antérieure à la mesure indirecte proprement dite des caractéristiques thermo-physiques du cuivre. En effet, bien que ces caractéristiques sont parmi les mieux connues dans la littérature, la prise en compte d'une incertitude de 4% sur R_p et $\sqrt{\tau_p}$ entraîne une dégradation de la mesure indirecte des caractéristiques du PVC, dans ce protocole expérimental, supérieure à la prise en compte d'une incertitude de 40% sur les caractéristiques du laiton. Si cette identification ne peut améliorer la connaissance de a_c et λ_c , il peut être envisagé de remplacer le cuivre par un matériau moins influent comme le laiton. D'autre part, cette étude a souligné l'intérêt des techniques de propagation de contraintes, car il n'est pas possible d'obtenir un encadrement satisfaisant de ces ensembles de dimension élevée à partir des algorithmes classiques tels que SIVIA en des temps de calcul (et des charges en mémoire) raisonnables.

Il reste encore, bien entendu, à développer des fonctions d'inclusions mieux adaptées à ce problème, ce qui fait l'objet d'une nouvelle thèse débutée récemment, dans la continuité de ce travail.

4.5 Analyse critique de résultats

Au paragraphe précédent, nous avons utilisé la variation de volume de l'ensemble solution pour décrire l'influence de la prise en compte de l'incertitude associée à un facteur expérimental. Dans cette partie, nous proposons de développer une technique similaire très simple pour qualifier les données recueillies.

4.5.1 Introduction

Une fois les données recueillies et l'estimation effectuée, l'expérimentateur éventuellement insatisfait peut souhaiter modifier le processus et le relancer. Il peut, par exemple, soupçonner la présence de données corrompues, voire aberrantes, parmi les données recueillies. La détection de données aberrantes est un domaine très important du diagnostic, qui utilise de plus en plus les techniques intervalles. Dans cette partie, nous souhaitons montrer qu'il est possible d'utiliser directement les estimateurs ensemblistes garantis afin de qualifier les données recueillies; en particulier, il est possible de détecter les données qui doivent être recueillies avec le plus grand soin, une fois le processus relancé.

Nous définissons dans un premier temps la fiabilité d'une donnée dans un contexte à erreurs bornées. Nous exprimons ce critère sous la forme d'un rapport de volumes. Afin d'obtenir une évaluation précise de la fiabilité, nous soulignons l'intérêt de l'algorithme SIVIA1, et nous illustrons cette approche sur un exemple d'application très simple.

4.5.2 Classification des données

4.5.2.1 Données aberrantes

Dans le contexte à erreurs bornées, nous appelons *donnée aberrante* toute donnée intervalle $[y_i]$ telle que

$$y_i^* \notin [y_i],$$

où y_i^* est la donnée expérimentale qui serait obtenue en l'absence de bruit. Dans le cas général, celle-ci est évidemment inconnue. La détection des données aberrantes ne peut donc être effectuée qu'à partir des données expérimentales bruitées. Diverses méthodes ont été recensées dans le domaine du diagnostic. Citons en particulier les approches qui s'appuient sur la redondance matérielle ou analytique, les traitements statistiques et les systèmes experts... Comme nous supposons que le modèle analytique η est disponible, et en l'absence de toute redondance matérielle, nous privilégions les approches par redondance analytique.

Deux cas se présentent alors, selon que le support \mathbb{S} de l'estimée $\hat{\mathbf{x}}$ est ou non vide.

Supposons dans un premier temps que \mathbb{S} est vide. Si le modèle est correct, il est alors certain qu'une donnée au moins est une donnée aberrante. L'algorithme GOMNE (*Guaranteed Outlier Minimal Number Estimator*) [JWD96] permet alors de calculer un ensemble non vide \mathbb{S} compatible avec le plus grand nombre de données possible. S'il s'avère que \mathbb{S} est vide, nous utiliserons donc cette étape de pré-traitement afin de nous ramener au cas suivant.

Si \mathbb{S} n'est pas vide, deux possibilités peuvent être envisagées :

- pour chaque donnée recueillie, la propriété $y_i^* \in [y_i]$ est vérifiée, donc la vraie valeur du vecteur de paramètres \mathbf{x}^* appartient à \mathbb{S} ,
- il existe au moins une donnée aberrante, donc $\mathbf{x}^* \notin \mathbb{S}$, mais cette donnée n'est pas incompatible avec l'expérience.

Comme la valeur de \mathbf{x}^* est inconnue, il n'est pas possible de déterminer dans ce cas s'il existe ou non une valeur aberrante. Le critère que nous définissons dans le paragraphe suivant permet d'évaluer la compatibilité des données, et ainsi de déterminer s'il est possible qu'au moins une donnée aberrante soit présente.

4.5.2.2 Fiabilité

Dans cette partie, nous supposons que \mathbb{S} n'est pas vide.

Soit ${}^i\mathbb{S}$ l'ensemble des \mathbf{x} compatibles avec toutes les données sauf la $i^{\text{ème}}$:

$${}^i\mathbb{S} = \{\mathbf{x} \mid \eta_i(\mathbf{x}) \in [y_i], \quad i \in \{1, \dots, i-1, i+1, \dots, n\}\}$$

Le volume de ${}^i\mathbb{S}$ quantifie l'incertitude associée à la grandeur physique \mathbf{x} . Comme $\mathbb{S} = \bigcap_{i=1}^n ({}^i\mathbb{S})$, la prise en compte de la $i^{\text{ème}}$ donnée expérimentale réduit le domaine des valeurs admissibles selon la quantité d'information apportée par cette valeur. Ainsi, une valeur peu informative réduit peu le domaine. Elle est donc redondante et cohérente avec les autres données. Par contre, la prise en compte d'une donnée très informative réduit fortement l'ensemble \mathbb{S} . Nous allons détecter la fiabilité d'une donnée à l'aide du critère suivant.

Définissons alors la *fiabilité* [JGW⁺97] d'une donnée par le rapport

$$\gamma_i = \frac{\text{Vol}(\mathbb{S})}{\text{Vol}({}^i\mathbb{S})}.$$

Ainsi, plus la fiabilité γ_i est proche de 1, moins la donnée $[y_i]$ est jugée informative. Une donnée expérimentale ayant une fiabilité très faible doit être vérifiée de manière particulièrement minutieuse, car s'il s'agit d'une donnée aberrante, elle peut entièrement ruiner le résultat du capteur logiciel – même si \mathbb{S} reste non vide.

Plus l'évaluation du volume des ensembles \mathbb{S} et ${}^i\mathbb{S}$ sera précise, plus ce critère sera sensible. Nous choisissons donc d'évaluer la fiabilité des données à l'aide de l'algorithme SIVIA1 (voir le chapitre 3). Nous obtenons alors un encadrement garanti de γ_i noté $[\gamma_i]$.

Étudions le comportement de ce critère sur un exemple simulé.

Exemple 4.5.1 *On considère huit données $y(t_i)$ représentées sur la figure 4-23 à gauche avec leurs barres d'incertitude. Ces données sont générées par le système*

$$\eta(t_i) = x_1 \sin(2\pi x_2 t_i)$$

avec $\mathbf{x} = (5 \ 0.4)^T$, pour des temps régulièrement espacés entre 0.3 s et 2.4 s. Elles sont entachées d'un bruit d'observation b borné. Deux cas ont été considérés. Le premier (+) correspond à un fonctionnement normal de la chaîne de mesure et ne comporte pas de données aberrantes. Dans le second cas, représenté par (o), nous avons simulé la défaillance locale du capteur : la donnée y_5 a été remplacée par 0. Remarquons que cette défaillance survient à un moment où la vraie valeur de y est proche de zéro, ce qui rend ce défaut particulièrement difficile à détecter. D'ailleurs, les barres d'incertitudes associées à la cinquième valeur dans les deux cas ont une intersection non vide : l'ensemble solution \mathbb{S} obtenu à l'aide des données représentées par des (o) est non-vide. La donnée y_5 ne peut donc pas être détectée par GOMNE. Les résultats obtenus $[\gamma_i]$ pour chaque mesure i sont représentés sur la figure 4-23 à droite. En l'absence de donnée aberrante toutes les fiabilités sont élevées (+). En présence de la donnée "faiblement" aberrante introduite à $t = t_5$, on constate que γ_5 devient très faible (o). La donnée y_5 a donc bien été détectée comme la donnée la moins fiable.

4.5.3 Conclusions

Dans le cas de capteurs à erreur de sortie bornée, les estimateurs garantis développés en inversion ensembliste grâce à l'analyse par intervalles permettent de caractériser les performances d'une

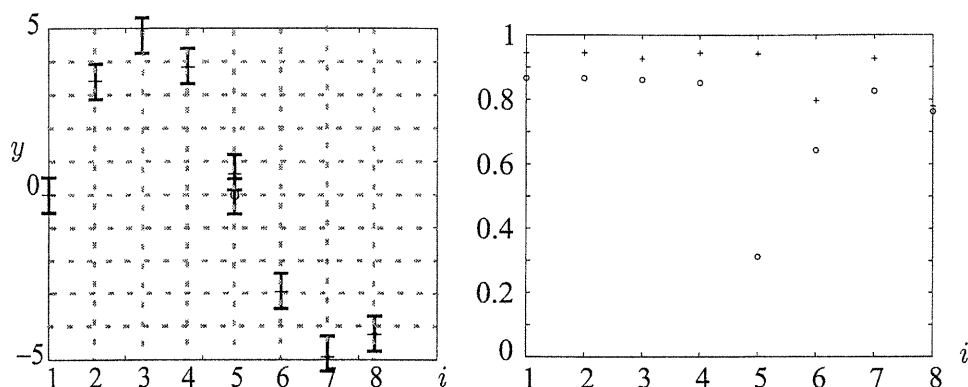


Figure 4-23: Etude de la dangerosité des données pour l'exemple 4.5.1: à gauche : données recueillies et barres d'incertitude, à droite : évaluation de la fiabilité des données dans les deux cas

chaîne de mesure en étudiant le comportement de l'ensemble solution \mathbb{S} estimé. Le degré de connexité de \mathbb{S} renseigne ainsi sur l'identifiabilité de \mathbf{x} , tandis que la forme et le volume de \mathbb{S} , déduits de la forme et du volume de ses approximations intérieure $\underline{\mathbb{S}}$ et extérieure $\overline{\mathbb{S}}$ permettent d'invalider un modèle, d'évaluer la sensibilité des composantes de \mathbf{x} en fonction du bruit, ou de montrer qu'un protocole expérimental ne permettra pas d'atteindre la précision désirée. Nous avons utilisé l'algorithme SIVIA1 afin de mettre en œuvre un critère de classification des données. La notion de fiabilité d'une mesure permet ainsi de détecter des mesures très informatives et donc potentiellement dangereuses pour la qualité de l'estimation, même dans des cas où un algorithme de type GOMNE n'aurait rien pu détecter. Les techniques employées peuvent être utilisées avec tout capteur pour lequel des bornes crédibles pour les erreurs de mesure sont disponibles, à condition de disposer aussi d'un modèle analytique liant les grandeurs à estimer \mathbf{x} aux mesures \mathbf{y} .

4.6 Conclusions

Nous avons illustré l'apport des techniques ensemblistes garanties dans un contexte d'estimation à erreurs bornées. D'une part, ces techniques nous permettent de nous affranchir de l'étape d'identifiabilité, puisqu'il devient possible d'estimer des paramètres jugés non-identifiables (voir le chapitre 5). D'autre part, cette approche permet de calculer directement le support de l'estimée $\hat{\mathbf{x}}$, et donc d'obtenir une évaluation simple de l'incertitude associée à cette estimation.

Deux applications réelles ont été envisagées. La première, issue de l'électrochimie, consiste à étudier les constantes cinétiques de mécanismes réactionnels très simples. Dans cette application, nous avons pu souligner les avantages des techniques ensemblistes garanties par rapport aux approches ponctuelles. La deuxième application consiste à évaluer les caractéristiques d'un dispositif de mesure indirect au travers de données recueillies sur un échantillon test. Nous avons recensé l'ensemble des grandeurs incertaines : les paramètres que nous cherchons à identifier, et les facteurs expérimentaux, dont l'incertitude dégrade la qualité de l'estimation. Grâce aux algorithmes de projection présentés au chapitre 3, nous avons pu estimer le vecteur de paramètres tout en prenant en compte l'incertitude associée à ces facteurs expérimentaux.

Enfin, nous avons présenté une méthode très simple d'évaluation des données. Cette approche permet de détecter des données très informatives, et donc potentiellement dangereuses pour la

qualité de l'estimation, même lorsque le support de l'estimée \hat{x} n'est pas vide.

Nous nous sommes intéressés jusqu'à présent aux étapes *a posteriori*. Les ensembles que nous avons manipulés représentaient donc les supports de variables incertaines. Dans les chapitres suivants, nous montrons que ces techniques garanties peuvent fournir des réponses utiles à des problèmes posés avant tout recueil de données.

Chapitre 5

Caractérisation de structures de modèle

La *caractérisation* est une étape délicate dans la démarche d'élaboration d'un modèle paramétrique à partir d'une connaissance expérimentale. Elle consiste à déterminer, parmi une classe de structures de modèles, une structure $\mathcal{M}(\cdot)$ dont on peut penser qu'elle sera pertinente compte tenu des données expérimentales recueillies –ou à recueillir– sur le système \mathcal{S} . Si aucune autre source d'information n'est disponible, la structure de modèle $\mathcal{M}(\cdot)$ est généralement de type comportemental, c'est-à-dire qu'elle vise à reproduire le comportement du système \mathcal{S} sans qu'un sens physique ne puisse être attribué aux paramètres de ce modèle. Mais s'il est possible d'établir un ensemble de relations (lois ou hypothèses) décrivant le phénomène dont le système \mathcal{S} est le siège, la structure résultante $\mathcal{M}(\cdot)$ définira plutôt un modèle de connaissance, dont les paramètres ont alors un sens physique concret. Dans ce cas, l'étape d'identification de paramètres est équivalente à la mesure indirecte des grandeurs physiques correspondantes, comme nous l'avons précisé dans le chapitre 1. Il est alors légitime de se demander si on peut espérer identifier de manière *unique* les paramètres de cette structure $\mathcal{M}(\cdot)$, ce qui correspond à étudier l'*identifiabilité* de cette structure de modèle.

Par ailleurs, le choix même de la structure $\mathcal{M}(\cdot)$ est souvent très difficile et il arrive fréquemment qu'on hésite entre deux structures de modèles mathématiques $\mathcal{M}(\cdot)$ et $\mathcal{N}(\cdot)$. On espère alors qu'une de ces deux structures se révélera plus à-même de décrire le comportement du système \mathcal{S} . Cette seconde question correspond à la notion de *discernabilité* entre modèles. Il s'agit alors de mettre en compétition deux structures et de déterminer s'il *existe* des valeurs de vecteurs de paramètres pour lesquelles les comportements des deux modèles sont identiques.

La caractérisation de structures de modèle, qui regroupe les tests d'identifiabilité et de discernabilité, s'avère donc souvent nécessaire, et peut prendre place à deux stades dans le processus de modélisation.

Si les données expérimentales ont déjà été recueillies, l'étape de caractérisation est alors directement liée au processus d'identification. Il s'agit de déterminer l'ensemble des valeurs numériques admissibles du vecteur de paramètres à estimer, en prenant en compte le contexte de l'expérience. On parle alors de test d'identifiabilité et de discernabilité *a posteriori*. Si les paramètres se révèlent non-identifiables, il est alors possible de modifier le modèle proposé, et de relancer la démarche. Dans ce contexte, il est indispensable de pouvoir garantir l'estimée obtenue à partir de laquelle le diagnostic sera tiré. Nous avons rencontré ce cas lors de l'étude du modèle de Volmer-Heyrovsky au chapitre 4.

Par souci d'économie de temps et de moyens, il est souvent préconisé d'effectuer ces tests avant que les mesures ne soient recueillies. On parle alors d'identifiabilité et de discernabilité *a*

priori. Dans ce cas, aucune valeur du vecteur de paramètres n'a encore pu être estimée. L'approche classique a alors consisté à étudier ces caractéristiques en tant que *propriétés structurelles*, dans un cadre idéalisé, affranchi des contraintes relatives à l'expérience et de l'information *a priori* dont dispose l'expérimentateur. Nous proposons ici une nouvelle approche non-structurale de ces notions, qui permet de rester dans un cadre plus proche de celui de l'expérience qui sera réalisée.

Ce chapitre, volontairement court, concerne principalement l'analyse d'identifiabilité *a priori*. Un bref rappel de l'approche classique, ainsi que des méthodes de tests utilisées est d'abord proposé au lecteur. Nous introduisons ensuite une nouvelle définition de l'identifiabilité *a priori*, et soulignons les spécificités de l'approche associée. Nous utilisons alors les outils ensemblistes garantis étudiés aux chapitres 2 et 3 pour développer de nouveaux tests numériques, simples et efficaces, d'identifiabilité globale de modèles inspirés en majeure partie de problèmes réels. L'étude de la discernabilité *a priori* de deux modèles par un traitement numérique peut être obtenue à parti d'outils similaires. Nous présentons alors brièvement ces définitions et un exemple d'illustration.

Pour ces deux tests, les applications mettront particulièrement en évidence la spécificité des méthodes numériques garanties. Cette étude a fait l'objet des deux publications [BJKW01a] et [BJKW02] pour l'étude de discernabilité.

5.1 Identifiabilité *a priori*

Nous considérons dans cette partie une unique structure de modèle mathématique $\mathcal{M}(\cdot)$ dont le vecteur de paramètres est noté \mathbf{p} . Il s'agit alors d'ajuster \mathbf{p} de telle sorte que le modèle $\mathcal{M}(\mathbf{p})$ puisse rendre compte du comportement du système \mathcal{S} à l'étude. Soit \mathbb{S} l'ensemble des valeurs numériques $\hat{\mathbf{p}}$ telles que le modèle $\mathcal{M}(\hat{\mathbf{p}})$ obtenu rende compte de façon satisfaisante du comportement du système \mathcal{S} . On s'attend à obtenir soit un ensemble réduit à un point, soit à un ensemble continu fini. Mais s'il est prouvé que cet ensemble est disjoint, alors différentes valeurs du vecteur de paramètres conduisent à des comportements voisins du comportement observé du modèle. Une telle ambiguïté est évidemment à proscrire dans le cas d'un modèle à base de connaissance puisque la signification physique des paramètres laisse espérer qu'on ne peut leur attribuer qu'une "vraie" valeur unique. Elle est également à éviter si des décisions doivent ensuite être prises sur la base de la valeur numérique de ces paramètres. Il pourra être envisagé de modifier le protocole expérimental de manière à lever cette ambiguïté, en particulier par des techniques de planification d'expérience qualitative [WP90].

Bien que l'importance de cette notion ait été soulignée il y a plus de 50 ans (voir [KR50]), le test d'identifiabilité ne s'est toujours pas imposé dans tous les domaines où les modèles mathématiques sont construits à partir de données expérimentales. S'il reste beaucoup à faire pour convaincre ses utilisateurs potentiels, il faut également pouvoir leur fournir des outils efficaces pour tester l'identifiabilité de leurs modèles. Une littérature abondante développe ce sujet [VR94], [GCV94], [DVJBNP01], [dH95], [Noi00], mais relativement peu de logiciels sont mis à la disposition de l'utilisateur, en particulier dans le domaine des modèles compartimentaux, très utilisés en chimie, en biologie et en pharmacocinétique. Citons par exemple le logiciel IDENT3 [JP90], implémenté en C ou en FORTRAN, et développé pour le logiciel SAAM II [Webj], ou encore le logiciel ADAPT [DS92].

Le test de l'identifiabilité *a priori* peut être effectué avant toute mesure. Nous nous placerons donc dans toute cette partie dans un cadre idéalisé. Ainsi supposerons-nous tout d'abord que l'erreur de modèle est nulle; les données sont donc générées par un modèle de structure $\mathcal{M}(\cdot)$ paramétré par une valeur \mathbf{p}^* inconnue du vecteur de paramètres. Cette valeur appartient à

un sous-ensemble \mathbb{P} de $\mathbb{R}^{\dim \mathbf{p}}$ fourni *a priori*. Nous supposons également que ces données ne sont pas bruitées et que les entrées et les instants de mesure peuvent être choisis librement. A partir des données générées par $\mathcal{M}(\mathbf{p}^*)$, on veut estimer les paramètres du modèle $\mathcal{M}(\hat{\mathbf{p}})$ (voir la figure 5-1).

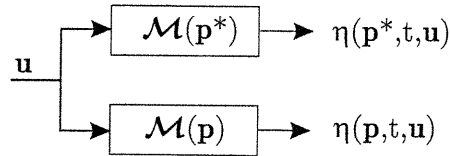


Figure 5-1: Cadre idéalisé de l'étude d'identifiabilité *a priori*

Dans ces conditions idéalisées, il est toujours possible de régler $\hat{\mathbf{p}}$ de manière à assurer que les sorties générées par le "modèle" $\mathcal{M}(\hat{\mathbf{p}})$ soient identiques à celles générées par le "système" $\mathcal{M}(\mathbf{p}^*)$ quels que soient l'entrée u et l'instant de mesure t . Nous dirons alors que les comportements entrée-sortie ([Wal82], [Oll90]) de $\mathcal{M}(\hat{\mathbf{p}})$ et $\mathcal{M}(\mathbf{p}^*)$ sont identiques. Pour plus de simplicité, nous noterons cette identité simplement par

$$\mathcal{M}(\hat{\mathbf{p}}) = \mathcal{M}(\mathbf{p}^*). \quad (5.1)$$

L'identifiabilité de $\hat{\mathbf{p}}$ ne dépend que du nombre de solutions de (5.1) pour $\hat{\mathbf{p}}$ dans \mathbb{P} . Ce nombre est strictement positif puisque le cas $\hat{\mathbf{p}} = \mathbf{p}^*$ fournit déjà une solution triviale de (5.1).

Supposons momentanément \mathbf{p}^* numériquement connu. Si l'ensemble \mathbb{S} de toutes les solutions de (5.1) est un singleton, alors $\hat{\mathbf{p}}$ est *globalement* (ou *uniquement*) *identifiable en \mathbf{p}^** . Si \mathbb{S} est fini ou dénombrable, alors $\hat{\mathbf{p}}$ est *localement identifiable en \mathbf{p}^** . Si \mathbb{S} n'est pas dénombrable, alors $\hat{\mathbf{p}}$ est dit *non-identifiable en \mathbf{p}^** . Notons que certaines des composantes de $\hat{\mathbf{p}}$ peuvent être globalement ou localement identifiables même si \mathbf{p} est non-identifiable.

Dans un contexte *a priori*, il n'est pas possible d'attribuer une valeur numérique à \mathbf{p}^* . On souhaite donc que les conclusions soient valides quelle que soit la valeur de \mathbf{p}^* dans le domaine de recherche *a priori* \mathbb{P} . Comme il peut exister, dans \mathbb{P} , des valeurs de \mathbf{p}^* pour lesquelles le cardinal de \mathbb{S} diffère de sa valeur typique, l'approche classique a consisté à étudier l'identifiabilité en tant que propriété structurelle.

5.1.1 Identifiabilité structurelle et approches formelles

Une propriété est dite *structurelle* (ou *générique*) si elle est vraie pour presque toute valeur des paramètres; elle peut donc éventuellement être fautive sur un sous-espace \mathbb{A} de mesure nulle de l'espace des paramètres, appelé *ensemble atypique*.

Ainsi, un paramètre p_i est *structurellement globalement identifiable (s.g.i.)* si, pour presque tout \mathbf{p}^* dans le domaine *a priori* \mathbb{P} , l'ensemble des solutions en \hat{p}_i de (5.1) se réduit au singleton p_i^* . Il est *structurellement localement identifiable (s.l.i.)* si pour presque tout \mathbf{p}^* dans \mathbb{P} , l'ensemble des solutions en \hat{p}_i de (5.1) est dénombrable ou fini. Il est *structurellement non-identifiable (s.n.i.)* si pour presque tout \mathbf{p}^* dans \mathbb{P} l'ensemble des solutions de (5.1) en \hat{p}_i n'est pas dénombrable. Le vecteur \mathbf{p} est *s.g.i.* ou *s.l.i.* si toutes ses composantes sont *s.g.i.* ou *s.l.i.*

Manipuler l'équation (5.1) est ardu. La première étape de toute étude d'identifiabilité structurelle consiste à établir les équations que doit satisfaire $\hat{\mathbf{p}}$ pour que $\mathcal{M}(\hat{\mathbf{p}})$ et $\mathcal{M}(\mathbf{p}^*)$ aient le même comportement entrée-sortie.

Le résumé de $\mathcal{M}(\cdot)$ est une application $\mathbf{r}_{\mathcal{M}}$ de \mathbb{P} dans \mathbb{R}^k telle que

$$\forall(\widehat{\mathbf{p}}, \mathbf{p}^*) \in \mathbb{P}^{\times 2}, \mathcal{M}(\widehat{\mathbf{p}}) = \mathcal{M}(\mathbf{p}^*) \implies \mathbf{r}_{\mathcal{M}}(\widehat{\mathbf{p}}) = \mathbf{r}_{\mathcal{M}}(\mathbf{p}^*). \quad (5.2)$$

Il est parfois possible de remplacer l'implication (5.2) par l'équivalence

$$\forall(\widehat{\mathbf{p}}, \mathbf{p}^*) \in \mathbb{P}^{\times 2}, \mathcal{M}(\widehat{\mathbf{p}}) = \mathcal{M}(\mathbf{p}^*) \iff \mathbf{r}_{\mathcal{M}}(\widehat{\mathbf{p}}) = \mathbf{r}_{\mathcal{M}}(\mathbf{p}^*), \quad (5.3)$$

nous parlerons alors de *résumé exhaustif* [WL81]. L'ensemble des solutions en \widehat{p}_i de (5.1) est alors égal à l'ensemble des solutions en \widehat{p}_i de

$$\mathbf{r}_{\mathcal{M}}(\widehat{\mathbf{p}}) = \mathbf{r}_{\mathcal{M}}(\mathbf{p}^*). \quad (5.4)$$

Suivant les caractéristiques de la structure $\mathcal{M}(\cdot)$, différentes approches permettent d'obtenir une application \mathbf{r} satisfaisant (5.2) ou (5.3) [WL82][WL81][RLWV85]. Pour les modèles linéaires, l'approche par les paramètres de Markov, l'approche par la transformée de Laplace, ou encore l'approche par "changement de base des états" [Wal82] sont les plus usitées. Pour les modèles non-linéaires, on citera en particulier les approches par linéarisation autour d'un point d'équilibre, par série de Taylor, la méthode des similarités [VGR89], ou enfin les méthodes d'algèbre différentielle [Oll90], [Sed01]. De nombreux algorithmes de calcul formel facilitent grandement cette étape de manipulation symbolique (par exemple DIFFALG de F. Boulier [Webe]).

Dans toute la suite, nous nous restreindrons au cas où le système d'équations (5.4) obtenu est purement algébrique. L'approche par transformée de Laplace est illustrée sur l'exemple très simple ci-dessous.

Exemple 5.1.1 Soit la structure de modèle compartimental $\mathcal{M}(\cdot)$ représentée sur la figure 5-2.

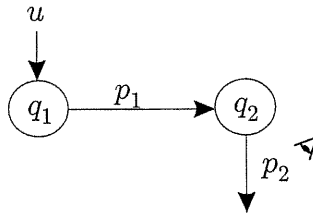


Figure 5-2: Modèle à deux compartiments de l'exemple 5.1.1.

Le comportement entrée-sortie se traduit par la fonction de transfert suivante

$$F(\mathbf{p}, s) = \frac{p_1}{(s + p_1)(s + p_2)},$$

dont la forme canonique est

$$F(\mathbf{p}, s) = \frac{p_1}{s^2 + (p_1 + p_2)s + p_1 p_2}.$$

Le résumé exhaustif est alors l'application qui associe à \mathbf{p} les coefficients des polynômes de F

écrites sous forme canonique :

$$\mathbf{r}_{\mathcal{M}} : \mathbf{p} \longmapsto \begin{pmatrix} p_1 \\ p_1 + p_2 \\ p_1 p_2 \end{pmatrix}.$$

L'égalité (5.1) impose

$$\begin{cases} \hat{p}_1 = p_1^* \\ \hat{p}_2 = p_2^* \end{cases} \text{ ou } \begin{cases} \hat{p}_1 = p_2^* \\ \hat{p}_2 = p_1^* \end{cases}$$

sous la condition que $p_1^* \neq 0$. Dans le cas contraire, \mathbf{p}^* appartient à un ensemble atypique, et \mathbf{p} n'est pas identifiable. Le vecteur de paramètres \mathbf{p} est donc structurellement localement identifiable.

Une fois l'application $\mathbf{r}_{\mathcal{M}}$ obtenue, l'ensemble \mathbb{S} de tous les vecteurs de paramètres $\hat{\mathbf{p}}$ satisfaisant (5.4) est recherché. Afin de s'assurer qu'aucune solution ne soit perdue, il est fréquent d'avoir recours aux techniques de calcul formel. En particulier, lorsque chaque composante de $\mathbf{r}_{\mathcal{M}}(\mathbf{p})$ est une fonction polynômiale des composantes de \mathbf{p} , la théorie de l'élimination permet de triangulariser ce système

$$\mathbf{t}(\hat{\mathbf{p}}) = \mathbf{t}(\mathbf{p}^*), \quad (5.5)$$

où à nouveau chacune des composantes de $\mathbf{t}(\mathbf{p})$ est une fonction polynômiale des composantes de \mathbf{p} , mais où le dernier polynôme ne dépend que d'une seule composante de \mathbf{p} , le précédent fait intervenir une composante de \mathbf{p} supplémentaire, et ainsi de suite. Il est alors possible, au moins en principe, de résoudre (5.5) en résolvant successivement des équations polynômiales à une seule inconnue, de façon similaire à celle employée pour résoudre par triangularisation des systèmes linéaires. Les logiciels de calcul formel sont souvent sollicités, car les calculs requis par cette deuxième étape peuvent être très compliqués même pour des modèles à faible dimension. La méthode employée pour construire (5.5) à partir (5.4) garantit qu'aucune solution n'a été perdue. Si le degré de chaque équation polynômiale à résoudre est génériquement un, il est alors prouvé que \mathbf{p} est *s.g.i.* Même si cela n'est pas le cas, il est souvent possible de générer l'expression explicite de l'ensemble de toutes les solutions de (5.4) en $\hat{\mathbf{p}}$ en fonction de \mathbf{p}^* . Pourvu que le cardinal de cet ensemble ait la même valeur pour presque toutes les valeurs de \mathbf{p}^* , cette approche formelle permet de conclure sur l'identifiabilité structurelle du modèle étudié.

Comme il s'agit de fournir une réponse simple à une question qualitative : *le modèle est-il s.g.i.?*, cette étude structurelle de modèles mathématiques est un domaine d'application très important des méthodes de calcul formel. Il y a cependant plusieurs raisons pour lesquelles cette approche peut se révéler inadaptée. Les premières relèvent de la lourdeur et la complexité des calculs mis en jeu :

- il n'est pas toujours possible de traduire l'équation (5.1) à l'aide d'une fonction $\mathbf{r}_{\mathcal{M}}$ polynômiale, c'est-à-dire manipulable par les algorithmes formels, ou lorsque cette traduction est possible, elle peut aboutir à un problème excessivement compliqué à résoudre,
- les manipulations formelles requises pour obtenir une conclusion peuvent être trop compliquées,
- le degré de certaines des équations polynômiales monovariabiles à résoudre lors du traitement de (5.5) peut être trop élevé pour qu'une solution analytique existe. Il faut alors se

résoudre à accepter de perdre la nature formelle de la solution, et à avoir recours à des méthodes de calcul numérique, ce qui oblige à spécifier une (ou des) valeurs numériques à \mathbf{p}^* et fait perdre sa garantie au résultat.

Le second type de raison pour souhaiter une approche alternative est le caractère structurel du test. Le nombre de solutions réelles en \hat{p}_i (c'est-à-dire les seules d'intérêt) peut en effet dépendre de la valeur numérique de \mathbf{p}^* de telle sorte qu'aucune conclusion de nature structurelle ne puisse être établie. L'exemple simple à un paramètre qui suit illustre cette idée.

Exemple 5.1.2 Soit le modèle à un paramètre p (dont on sait qu'il appartient à \mathbb{P}), sans entrée et donc l'unique sortie est donnée par

$$\eta(p) = p(p-1)(p+1), \quad \mathbb{P} = [-2, 2].$$

$\mathcal{M}(\hat{p})$ et $\mathcal{M}(p^*)$ auront le même comportement entrée-sortie si et seulement si

$$p^*(p^* - 1)(p^* + 1) - \hat{p}(\hat{p} - 1)(\hat{p} + 1) = 0, \quad p^* \in \mathbb{P}, \quad \hat{p} \in \mathbb{P}.$$

Suivant la valeur numérique de p^* , l'ensemble des valeurs possibles pour \hat{p} sera un singleton si $p^* \in]-2, -a[\cup]a, 2[$, un couple si $p^* = a$ ou $p^* = -a$, ou un triplet pour p^* dans $] -a, a[$, où $a = \frac{2}{\sqrt{3}}$ (voir la figure 5-3). L'identifiabilité globale n'est donc pas une propriété structurelle sur \mathbb{P} .

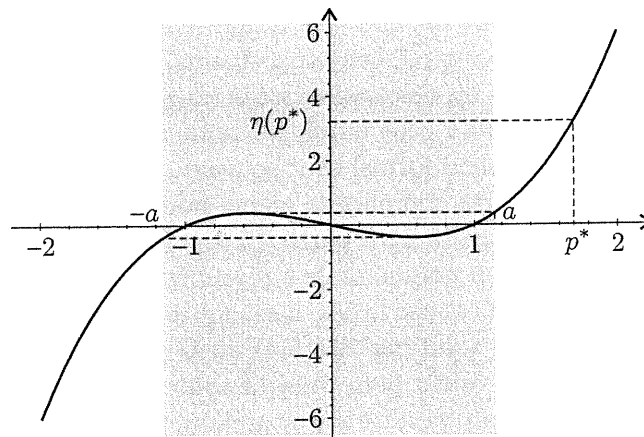


Figure 5-3: Sortie η du modèle de l'exemple 5.1.2 en fonction du paramètre p^*

Une approche alternative permettant de dépasser cette difficulté est présentée dans le paragraphe suivant.

5.1.2 Approches numériques garanties

La méthodologie employée dépend de l'information disponible sur \mathbf{p}^* . Nous présentons ici les résultats obtenus pour deux algorithmes issus de l'analyse par intervalles. Leur domaine d'utilisation et leur principe diffèrent : le premier s'applique au cas où une valeur numérique a déjà été attribuée à \mathbf{p}^* , tandis que le second traite le cas où on ne dispose que d'un domaine de recherche *a priori*. Dans ces deux contextes, nous supposons avoir à disposition les équations algébriques

(5.4) traduisant l'identité de comportement entrée-sortie des deux modèles, obtenues à partir de l'une ou l'autre des méthodes citées plus haut.

5.1.2.1 Une valeur numérique a déjà été attribuée à \mathbf{p}^*

Parfois l'expérimentateur dispose déjà d'une valeur numérique pour le vecteur de paramètres $\hat{\mathbf{p}}$. Reprenons l'exemple du dispositif de mesure de paramètres thermo-physiques issu du chapitre 4. Trois types d'information avaient alors pu être dégagés :

- certains paramètres correspondent à des grandeurs pour lesquelles il existe une valeur tabulée dans la littérature ou par un expert,
- l'expérimentateur formule une hypothèse qu'il souhaite valider par son expérience,
- une première estimation du paramètre p_i a déjà été effectuée à partir d'une méthode quelconque.

Dans ces trois cas, on remplace \mathbf{p}^* par la valeur disponible $\hat{\mathbf{p}}$, et on souhaite déterminer s'il y a d'autres $\hat{\mathbf{p}}$ équivalents pour le modèle de structure $\mathcal{M}(\cdot)$. Cela revient à estimer le nombre de vecteurs $\hat{\mathbf{p}}$ solutions de (5.4).

La résolution de (5.4) en $\hat{\mathbf{p}}$ lorsque la valeur numérique de \mathbf{p}^* est connue revient à trouver l'ensemble des racines de la fonction \mathbf{f} définie sur \mathbb{P} par

$$\mathbf{f}(\hat{\mathbf{p}}) = \mathbf{r}_{\mathcal{M}}(\hat{\mathbf{p}}) - \mathbf{r}_{\mathcal{M}}(\mathbf{p}^*). \quad (5.6)$$

Si $\dim \mathbf{p} = \dim \mathbf{f}$, et sous l'hypothèse que \mathbf{f} est de classe \mathcal{C}^1 , le résolveur garanti NEWTON INTERVAL permet d'encadrer *toutes* ces racines dans une union \mathbb{L} de pavés $[\hat{\mathbf{p}}]$ ([HHKR95b] et références incluses). La taille de ces pavés dépend d'un paramètre réglable qui influe directement sur la charge de calcul. En complément, un test permet de prouver l'existence et l'unicité d'un zéro inclus dans un pavé $[\hat{\mathbf{p}}]$. Il s'appuie sur les propriétés suivantes de l'opérateur de Newton-Gauss-Seidel $\mathcal{C}_{GS}([\mathbf{p}])$ [HHKR95b] :

- (i) chaque zéro $\hat{\mathbf{p}}$ de (5.6) d'un pavé $[\mathbf{p}]$ vérifie $\hat{\mathbf{p}} \in \mathcal{C}_{GS}([\mathbf{p}])$;
- (ii) si $\mathcal{C}_{GS}([\mathbf{p}]) = \emptyset$, alors il n'existe pas de zéro de (5.6) dans le pavé $[\mathbf{p}]$
- (iii) si $\mathcal{C}_{GS}(\mathbf{p})$ est strictement inclus dans $[\mathbf{p}]$, alors $[\mathbf{p}]$ contient exactement un zéro de \mathbf{f} .

Les propriétés (i) et (ii) indiquent que le résultat est garanti au sens où aucun zéro n'est perdu. La propriété (iii) permet d'évaluer le nombre exact de zéros. Si \mathbb{L} est réduit à un simple pavé $[\hat{\mathbf{p}}]$ et (iii) est satisfaite sur $[\hat{\mathbf{p}}]$, alors le modèle est globalement identifiable en \mathbf{p}^* . Sinon, $\hat{\mathbf{p}}$ peut contenir un ou plusieurs zéros : \mathcal{C}_{GS} est alors appliqué de façon récursive aux deux pavés $[\mathbf{p}_1]$ et $[\mathbf{p}_2]$ résultant de la bisection de $[\hat{\mathbf{p}}]$. Si toutes les boîtes de \mathbb{L} sont telles que la propriété (iii) est satisfaite, alors le modèle est localement identifiable en \mathbf{p}^* .

5.1.2.2 Un espace de recherche \mathbb{P} est disponible

Comme déjà mentionné, l'un des intérêts de l'étude de l'identifiabilité *a priori* est qu'elle peut être effectuée avant toute mesure et peut donc servir de base à une planification d'expériences qualitative [WP90]. D'un autre côté, comme l'identifiabilité globale n'est pas nécessairement une propriété structurelle, cette seule étude peut se révéler insuffisante face aux attentes d'un utilisateur potentiel, qui peut disposer d'informations *a priori* sur l'ordre de grandeur de ses paramètres qu'il souhaiterait prendre en compte, et qui attend un résultat garanti adapté au contexte de son expérience. Il s'agit alors d'effectuer une étude d'identifiabilité sous contraintes, sans pour autant imposer une valeur numérique précise à \mathbf{p}^* . Ceci nous conduit à proposer la définition suivante :

Définition 5.1.1 (Identifiabilité globale sur \mathbb{P}) *Le paramètre p_i est globalement identifiable sur \mathbb{P} (g.i.i. \mathbb{P}) si*

$$\forall(\mathbf{p}^*, \hat{\mathbf{p}}) \in \mathbb{P}, \mathcal{M}(\mathbf{p}^*) = \mathcal{M}(\hat{\mathbf{p}}) \Rightarrow p_i^* = \hat{p}_i$$

Le vecteur de paramètres \mathbf{p} sera dit *g.i.i. \mathbb{P}* si toutes ses composantes $p_i, i = 1, \dots, \dim \mathbf{p}$ sont *g.i.i. \mathbb{P}* .

Ces définitions n'autorisent plus l'existence de régions atypiques et soulignent l'importance de l'espace de recherche \mathbb{P} . Le lemme suivant nous sera utile pour reformuler le test d'identifiabilité globale sur \mathbb{P} sous la forme d'un problème de satisfaction de contraintes.

Lemme 5.1.1 *Si la proposition*

$$\exists(\hat{\mathbf{p}}, \mathbf{p}^*) \in \mathbb{P}^{\times 2}, \mathbf{r}_{\mathcal{M}}(\hat{\mathbf{p}}) = \mathbf{r}_{\mathcal{M}}(\mathbf{p}^*), \|\hat{\mathbf{p}} - \mathbf{p}^*\|_{\infty} > 0 \quad (5.7)$$

est fausse, alors \mathbf{p} est g.i.i. \mathbb{P} .

Remarque 5.1.1 *Si (5.7) est fausse, alors l'application $\mathbf{r}_{\mathcal{M}}$ est injective sur \mathbb{P} .*

La contrainte inégalité de (5.7) est forte puisqu'elle n'autorise pas deux valeurs numériques équivalentes de \mathbf{p} même dans un voisinage très petit. En pratique, l'expérimentateur peut relâcher cette contrainte en définissant une distance δ en-dessous de laquelle il ne peut pas – ou ne souhaite pas – séparer deux valeurs numériques. La quantité δ représente alors l'incertitude minimale admise par l'expérimentateur. Elle peut, par exemple, correspondre à la tolérance du cahier des charges : l'expérimentateur peut ne pas souhaiter séparer deux valeurs numériques proches à 10^{-6} d'une constante de vitesse de réaction lorsque son cahier des charges n'impose qu'une précision de 10^{-2} ! Ceci nous suggère la définition suivante.

Définition 5.1.2 *Soit δ une constante positive fixée par l'utilisateur. Si la proposition*

$$\exists(\hat{\mathbf{p}}, \mathbf{p}^*) \in \mathbb{P}^{\times 2}, \mathbf{r}_{\mathcal{M}}(\hat{\mathbf{p}}) = \mathbf{r}_{\mathcal{M}}(\mathbf{p}^*), \|\hat{\mathbf{p}} - \mathbf{p}^*\|_{\infty} \geq \delta, \quad (5.8)$$

n'a pas de solution, alors le modèle $\mathcal{M}(\cdot)$ est dit δ -g.i.i. \mathbb{P} .

Afin de tester de façon garantie si la proposition (5.8) est fausse ou non, réinterprétons ce problème sous la forme d'un problème de satisfaction de contraintes. Selon la dimension de \mathbf{p} , nous obtiendrons ensuite une caractérisation garantie de l'ensemble solution de ce CSP à l'aide des algorithmes SIVIAP ou PROJECT décrits au chapitre 3.

Soient \mathbf{p} et \mathbf{q} deux vecteurs de \mathbb{P} et \mathbf{x} le vecteur étendu

$$\mathbf{x} = \begin{pmatrix} \mathbf{p} \\ \mathbf{q} \end{pmatrix}. \quad (5.9)$$

Soit la fonction

$$\mathbf{f} : \begin{array}{ccc} \mathbb{R}^{2 \dim \mathbf{p}} & \longrightarrow & \mathbb{R}^k \\ \mathbf{x} & \longmapsto & \mathbf{r}_{\mathcal{M}}(\mathbf{p}) - \mathbf{r}_{\mathcal{M}}(\mathbf{q}) \end{array}$$

pour laquelle nous disposons d'une fonction d'inclusion $[\mathbf{f}]$.

L'ensemble solution \mathbb{S} du CSP

$$\mathcal{H} : \left(\begin{array}{l} \mathbf{f}(\mathbf{x}) = \mathbf{0} \\ \wedge \|\mathbf{p} - \mathbf{q}\|_{\infty} \geq \delta \end{array}, \mathbf{x} = (\mathbf{p}, \mathbf{q})^T \in [\mathbf{x}] \right), \quad (5.10)$$

Algorithme GIIP ($e : [\mathbf{r}], [\mathbf{x}], \varepsilon, \delta; s : \bar{\mathbb{S}}$)	
1	$\mathcal{C}([\mathbf{x}]);$
2	si ($w([\mathbf{x}]) < \varepsilon$) $\bar{\mathbb{S}} = \bar{\mathbb{S}} \cup [\mathbf{x}]; \bar{\mathbb{S}} = \bar{\mathbb{S}} \cup [\mathbf{x}]^\diamond$; renvoyer;
3	Bissecter($[\mathbf{x}], [\mathbf{x}_1], [\mathbf{x}_2]$);
4	GIIP($[\mathbf{r}], [\mathbf{x}_1], \varepsilon, \delta, \bar{\mathbb{S}}$); GIIP($[\mathbf{r}], [\mathbf{x}_2], \varepsilon, \delta, \bar{\mathbb{S}}$);

Tableau 5.1: Algorithme GIIP

qui se réécrit comme

$$\mathcal{H} : \left(\left\{ \begin{array}{l} \mathbf{f}(\mathbf{x}) = \mathbf{0} \\ \wedge \max_{i=1, \dots, n} |p_i - q_i| \geq \delta \end{array} \right. , \mathbf{x} = (\mathbf{p}, \mathbf{q})^T \in [\mathbf{x}] \right),$$

regroupe l'ensemble des couples $(\mathbf{p}^*, \hat{\mathbf{p}})$ solutions de (5.8). On a alors l'implication :

$$\bar{\mathbb{S}} = \emptyset \implies \mathbf{p} \text{ est } \delta\text{-g.i.i.}\mathbb{P}.$$

L'algorithme GIIP présenté à la table 5.1, fournit une approximation extérieure $\bar{\mathbb{S}}$ de \mathbb{S} . Il est directement dérivé de SIVIAP, et traite donc des pavés de dimension 2 $\dim \mathbf{p}$. Afin de limiter la complexité, GIIP exploite le fait que si $\mathbf{x} \in \mathbb{S}$, alors $\mathbf{x}^\diamond = (\mathbf{q}, \mathbf{p})^T \in \mathbb{S}$. Des approximations obtenues, nous pouvons déduire que si $\bar{\mathbb{S}}$ est vide, alors $\mathbb{S} = \emptyset$ et le vecteur \mathbf{p} est $\delta\text{-g.i.i.}\mathbb{P}$. Sinon, la projection $\pi_{\mathbb{P}}(\bar{\mathbb{S}})$ de $\bar{\mathbb{S}}$ sur \mathbb{P} fournit une approximation extérieure de l'ensemble des pavés $[\mathbf{p}]$ pour lesquels il n'a pas pu être prouvé que \mathbf{p} était $\delta\text{-g.i.i.}[\mathbf{p}]$.

Remarque 5.1.2 La δ_i -identifiabilité d'une composante p_i s'obtient directement en remplaçant la contrainte de distance $|\hat{\mathbf{p}} - \mathbf{p}^*| \geq \delta$ par $|\hat{p}_i - p_i^*| \geq \delta_i$.

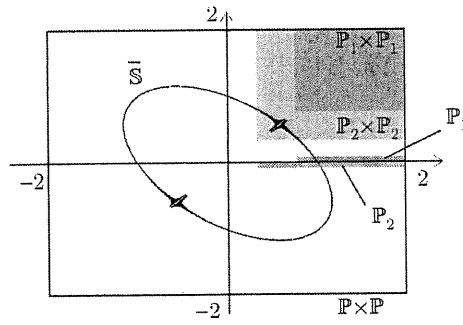
Exemple 5.1.3 Considérons à nouveau le modèle de l'exemple 5.1.2. En 10^{-3} s sur un Pentium 350, GIIP a prouvé que \mathcal{M} est $\delta\text{-g.i.i.}\mathbb{P}_1$, pour $\delta = 0.02$ (voir la figure 5-4), mais ne peut pas tirer de conclusions quant à savoir si \mathcal{M} est $\delta\text{-g.i.i.}\mathbb{P}_2$. Le sous-pavage $\bar{\mathbb{S}}$ fourni par l'algorithme, lorsque le nombre maximal de bisections est fixé à 14, est représenté en noir. Il se compose donc de tous les pavés $[\mathbf{x}] = [p] \times [q]$ de $\mathbb{P}^{\times 2}$ sur lesquels on n'a pas pu prouvé que la proposition (5.7) était fausse. Il s'agit donc d'une approximation extérieure de l'ensemble \mathbb{S} des vecteurs de paramètres ponctuels $(\hat{\mathbf{p}}, \mathbf{p}^*)^T$ pour lesquels \mathcal{M} n'est pas globalement identifiable. Bien évidemment, tout logiciel de calcul formel permet d'obtenir très rapidement l'expression analytique de \mathbb{S} , qui définit une ellipse :

$$(\hat{p} + p^*)^2 - \hat{p}p^* = 1.$$

La projection de $\bar{\mathbb{S}}$ sur \mathbb{P} fournit un encadrement extérieur de la projection de \mathbb{S} sur \mathbb{P} , que l'on peut affiner en autorisant un nombre de bisections plus élevé. Notons que l'espace de recherche $\mathbb{P}_2 \times \mathbb{P}_2$ intersecte bien cet ensemble $\bar{\mathbb{S}}$.

L'hyperplan $\mathbf{p} = \mathbf{q}$ est une zone d'accumulation de l'algorithme GIIP. Ainsi, la contrainte de distance, en limitant cette accumulation, limite le temps de calcul.

Fréquemment \mathbb{P} et $\mathbb{R}^{\dim \mathbf{p}}$ ne sont pas isomorphes en raison de contraintes dues à la nature physique de la grandeur \mathbf{p} issue du modèle de connaissance. Par exemple, on peut imposer pour

Figure 5-4: Sous-pavage $\bar{\mathcal{S}}$ obtenu par GIIP pour l'exemple 5.1.2

des modèles compartimentaux, que tous les coefficients d'échange soient positifs. Ces contraintes (ou d'autres, proposées par l'expérimentateur) sont aisément prises en compte dans l'algorithme. Elles permettent d'en déduire des résultats a priori, sans avoir à recueillir des données. Ces contraintes peuvent éventuellement servir à définir un plan d'expériences. Les exemples traités ci-dessous, en particulier l'exemple 5.1.4, illustrent ce propos.

Exemple 5.1.4 *Considérons le modèle compartimental suivant, qui décrit le comportement de la Glafénine administrée oralement:*

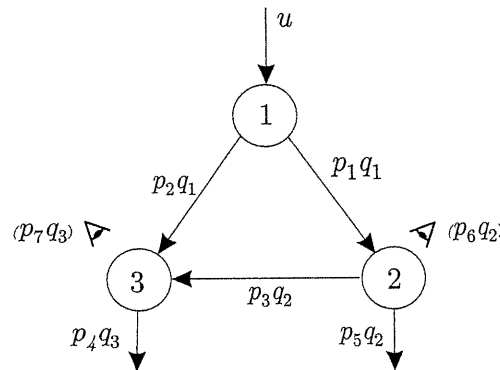


Figure 5-5: Modèle compartimental de la distribution de la Glafénine dans un organisme

$$\mathcal{M}(\mathbf{p}) : \begin{cases} \dot{q}_1 &= -(p_1 + p_2)q_1 + u, \\ \dot{q}_2 &= p_1 q_1 - (p_3 + p_5)q_2, \\ \dot{q}_3 &= p_2 q_1 + p_3 q_2 - p_4 q_3, \\ \eta_1 &= p_6 q_2, \\ \eta_2 &= p_7 q_3, \end{cases} \quad (5.11)$$

où les conditions initiales sur les variables d'état q_i sont toutes égales à zéro. Puisque $\dim \mathbf{p} = 7$, la dimension du CSP est 14. L'identité des comportements de $\mathcal{M}(\hat{\mathbf{p}})$ et $\mathcal{M}(\mathbf{p}^*)$ se réécrit sous la forme de (5.4) où les composantes de $\mathbf{r}_{\mathcal{M}}(\mathbf{p})$ sont les coefficients de la matrice de transfert associée à (5.11). Rappelons que pour que cette traduction soit valide, il faut que les éléments de cette matrice soient exprimés sous une forme canonique. Pour $\mathbb{P} = [0.6, 1]^{x7}$ et $\delta = 10^{-9}$, GIIP trouve en 0.01 s que le modèle est δ -g.i.i.P. Remarquons que pour $\mathbb{P} = \mathbb{R}^7$, le modèle est structurellement non-identifiable [WP90].

Par utilisation des contraintes, GIIP peut parfois fournir une conclusion sans aucune bisection, ce qui permet d'envisager des modèles à nombre de paramètres important. L'exemple suivant illustre cette approche dans un cadre contraint.

Exemple 5.1.5 La distribution de la Vincamine dans le corps humain est couramment modélisée par des mécanismes de transfert entre quatre compartiments, représentés sur la figure 5-6. La quantité q_i de Vincamine dans le compartiment 1 et le débit sortant du compartiment 2 vers

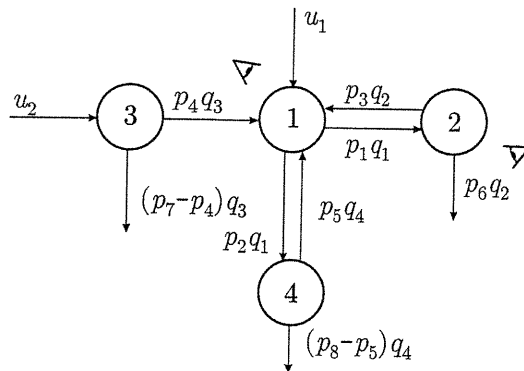


Figure 5-6: Modèle compartimental de la distribution de la Vincamine

l'extérieur sont mesurés. L'équation d'état du modèle s'écrit :

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{A}\mathbf{q} + \mathbf{B}\mathbf{u}, \\ \boldsymbol{\eta} &= \mathbf{C}\mathbf{q},\end{aligned}$$

avec

$$\mathbf{A} = \begin{pmatrix} -(p_1 + p_2) & p_3 & p_4 & p_5 \\ p_1 & -(p_6 + p_3) & 0 & 0 \\ 0 & 0 & p_7 & 0 \\ 0 & 0 & 0 & p_8 \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & p_6 & 0 & 0 \end{pmatrix}.$$

Plusieurs méthodes permettent d'établir un résumé exhaustif du comportement entrée-sortie du modèle. Avec l'approche par transformée de Laplace, il est possible d'établir que l'union des hyperplans définis par

$$\mathbb{U} : \begin{cases} \forall i \in \{1, \dots, 8\} & p_i = 0 \\ p_7 = p_8 \end{cases}$$

appartient à l'ensemble atypique \mathbb{A} des valeurs de \mathbf{p} pour lesquelles les conclusions structurelles ne sont pas valides. L'égalité des coefficients des matrices de transfert écrites sous forme canonique fournit un système de 25 équations à 16 inconnues. A condition d'inclure la contrainte

supplémentaire $p_7 \neq p_8$, SIVIA \mathbb{P} montre en 25.10^{-3} s pour $\delta = 0.01$ que l'ensemble

$$\mathbb{S} = \left\{ \begin{array}{l} (\mathbf{p}^*, \widehat{\mathbf{p}}) \mid (\mathbf{r}_{\mathcal{M}}(\mathbf{p}^*) = \mathbf{r}_{\mathcal{M}}(\widehat{\mathbf{p}})) \\ \wedge (p_7^* \neq p_8^*) \wedge (\widehat{p}_7 \neq \widehat{p}_8) \\ \wedge (\|\mathbf{p}^* - \widehat{\mathbf{p}}\|_{\infty} > \delta) \end{array} \right\}$$

est vide, pour \mathbb{P} quelconque tel que $0 \notin \mathbb{P}_i$, $i = 1, \dots, 8$. Les résultats obtenus permettent de conclure que pour tout \mathbb{P} tel que $\mathbb{P} \cap \mathbb{U} = \emptyset$, le modèle étudié est δ -g.i.i. \mathbb{P} . Ici, on a donc $\mathbb{U} = \mathbb{A}$.

Les études décrites dans ce paragraphe, effectuées avant toute saisie des données, se placent dans un cadre idéalisé, où l'erreur de modèle, l'incertitude de mesure et les contraintes du protocole expérimental ne sont pas encore prises en compte. Si les données ont déjà été recueillies, on peut s'affranchir de ce cadre idéalisé.

5.2 Discernabilité a priori

Supposons qu'on dispose maintenant de deux structures de modèles et d'un contexte expérimental fixé. C'est par exemple le cas lorsqu'on cherche à déterminer un mécanisme réactionnel en chimie. L'expérimentateur peut être amené à comparer deux structures qui correspondent à des mécanismes concurrents et espère parvenir à les discerner en étudiant, par exemple, la cinétique de la réaction qu'il étudie. Cette discrimination s'appuie sur le désir de déterminer la structure la plus à-même de décrire les données qu'il aura recueillies. C'est à cette question que se propose de répondre l'étude de discernabilité [CGV94] [CG96].

Considérons deux structures de modèles $\mathcal{M}(\cdot)$ et $\mathcal{N}(\cdot)$, différentes, associées respectivement aux vecteurs de paramètres \mathbf{p} et \mathbf{q} . Bien évidemment, \mathbf{p} et \mathbf{q} peuvent ne pas être de même dimension. Ces deux structures entrent en compétition pour la description d'un même système. Supposons à nouveau que l'erreur de modèle est nulle lorsque le modèle est décrit par une des deux structures, soit $\mathcal{M}(\cdot)$. Il existe donc une valeur inconnue \mathbf{p}^* telle que $\mathcal{M}(\mathbf{p}^*)$ décrive parfaitement le comportement du système dans le cadre de l'étude, mais, puisque les deux structures sont différentes, il se peut désormais qu'il n'existe aucune valeur de paramètre \mathbf{q} pour laquelle le modèle $\mathcal{M}(\mathbf{p}^*)$ et $\mathcal{N}(\mathbf{q})$ aient des comportements identiques. Ceci aboutit à la notion de discernabilité.

On dira que $\mathcal{N}(\cdot)$ est structurellement discernable de $\mathcal{M}(\cdot)$, si pour presque toute valeur de \mathbf{p} dans \mathbb{P} , il n'existe aucune valeur \mathbf{q} dans \mathbb{Q} satisfaisant

$$\mathcal{M}(\mathbf{p}) = \mathcal{N}(\mathbf{q}). \quad (5.12)$$

Notons que contrairement à celle de l'identifiabilité, cette définition n'est pas symétrique : la proposition “ $\mathcal{N}(\cdot)$ est structurellement discernable de $\mathcal{M}(\cdot)$ ” n'entraîne pas nécessairement “ $\mathcal{M}(\cdot)$ est structurellement discernable de $\mathcal{N}(\cdot)$ ”. Lorsque les deux assertions sont vérifiées, alors on dira que $\mathcal{M}(\cdot)$ et $\mathcal{N}(\cdot)$ sont structurellement discernables.

Les algorithmes de tests de discernabilité structurelle sont analogues à ceux employés dans les tests d'identifiabilité. L'exemple suivant illustre à nouveau un cas où aucune conclusion d'ordre structurel ne peut être obtenue.

Exemple 5.2.1 *Considérons les deux fonctions de transfert du deuxième ordre à coefficients réels*

$$\mathcal{M}(\mathbf{p}) : F(\mathbf{p}, s) = \frac{1}{s^2 + p_1 s + p_2} \quad \text{et} \quad \mathcal{N}(\mathbf{q}) : G(\mathbf{q}, s) = \frac{1}{(s + q_1)(s + q_2)}.$$

$G(\mathbf{q}, s)$ est discernable de $F(\mathbf{p}, s)$ si et seulement si $F(\mathbf{p}, s)$ ne possède pas deux pôles réels, c'est-à-dire si et seulement si la condition

$$p_1^2 - 4p_2 < 0 \quad (5.13)$$

est satisfaite. Suivant que la valeur numérique de \mathbf{p} satisfait ou non cette condition, $\mathcal{N}(\mathbf{q})$ sera ou non discernable de $\mathcal{M}(\mathbf{p})$.

Lorsque les deux structures sont issues de modèles de connaissance, il est possible de donner un espace de recherche *a priori* pour l'ensemble des paramètres à estimer. Dès lors, on peut fournir une nouvelle définition de la discernabilité dans le cadre de l'expérience :

Définition 5.2.1 *On dira que $\mathcal{M}(\cdot)$ et $\mathcal{N}(\cdot)$ sont globalement discernables sur $\mathbb{P} \times \mathbb{Q}$, s'il n'existe aucune valeur de \mathbf{p} dans \mathbb{P} et \mathbf{q} dans \mathbb{Q} satisfaisant*

$$\mathcal{M}(\mathbf{p}) = \mathcal{N}(\mathbf{q}). \quad (5.14)$$

Dans ce contexte, il n'est donc plus nécessaire d'introduire une dissymétrie dans la définition. A nouveau, cette propriété peut être testée simplement à l'aide d'algorithmes de propagation de contraintes.

Soit \mathbb{S} l'ensemble solution du CSP \mathcal{H}

$$\mathcal{H} : (\mathbf{r}_{\mathcal{N}}(\mathbf{q}) - \mathbf{r}_{\mathcal{M}}(\mathbf{p}) = \mathbf{0}, \mathbf{p} \in \mathbb{P}, \mathbf{q} \in \mathbb{Q}).$$

$\mathcal{M}(\cdot)$ et $\mathcal{N}(\cdot)$ sont globalement discernables sur $\mathbb{P} \times \mathbb{Q}$ si et seulement si $\mathbb{S} = \emptyset$. Les sous-pavages $\underline{\mathbb{S}}$ et $\overline{\mathbb{S}}$ obtenus par SIVIAP permettent de conclure dans les cas suivants :

- $\overline{\mathbb{S}} = \emptyset \Rightarrow \mathbb{S} = \emptyset \Rightarrow \mathcal{M}(\cdot)$ et $\mathcal{N}(\cdot)$ sont globalement discernables sur $\mathbb{P} \times \mathbb{Q}$;
- $\underline{\mathbb{S}} \neq \emptyset \Rightarrow \exists (\mathbf{p}, \mathbf{q}) \in \mathbb{P} \times \mathbb{Q} \mid \mathcal{M}(\mathbf{p}) = \mathcal{N}(\mathbf{q}) \Rightarrow \mathcal{M}(\cdot)$ et $\mathcal{N}(\cdot)$ ne sont pas globalement discernables sur $\mathbb{P} \times \mathbb{Q}$.

L'exemple 5.2.1 illustre ces résultats.

Exemple 5.2.1 (suite) *Considérons à nouveau les deux fonctions de transfert de l'exemple précédent. Dans un premier temps établissons les conditions pour lesquelles les structures de modèle $\mathcal{M}(\cdot)$ et $\mathcal{N}(\cdot)$ ne sont pas discernables sur $\mathbb{P} \times \mathbb{Q}$. Pour $\mathbb{P} = \mathbb{Q} = [-20, 20] \times [-100, 100]$, ces conditions sont*

$$\left\{ \begin{array}{l} (i) \quad p_1^2 - 4p_2 > 0 \\ (ii) \quad q_1 + q_2 \in [-20, 20] \\ (iii) \quad q_1 q_2 \in [-100, 100] \\ (iv) \quad p_1 = q_1 + q_2 \\ (v) \quad p_2 = q_1 q_2 \end{array} \right. .$$

Les projections $\pi_{\mathbb{P}}(\overline{\mathbb{S}})$ et $\pi_{\mathbb{Q}}(\overline{\mathbb{S}})$ obtenues par SIVIAP en 2 s sont représentées à la figure 5-7. Elles permettent de faire apparaître ces 5 contraintes. La projection $\pi_{\mathbb{P}}(\overline{\mathbb{S}})$ de $\overline{\mathbb{S}}$ sur \mathbb{P} , représentée à la figure 5-7 b), est un ensemble délimité par la parabole

$$p_1^2 - 4p_2 = 0,$$

tandis que la projection $\pi_{\mathbb{Q}}(\bar{\mathbb{S}})$ de $\bar{\mathbb{S}}$ sur \mathbb{Q} , représentée à la figure 5-7 a), est l'ensemble délimité par les droites

$$\begin{cases} q_1 + q_2 = 20 \\ q_1 + q_2 = -20 \end{cases} ,$$

et les branches d'hyperboles

$$\begin{cases} q_1 q_2 = 100 \\ q_1 q_2 = -100 \end{cases} .$$

Comme ni \mathbb{S} ni son complémentaire dans $\mathbb{P} \times \mathbb{Q}$ ne sont de mesure nulle, la discernabilité n'est pas une propriété structurelle pour ces modèles.

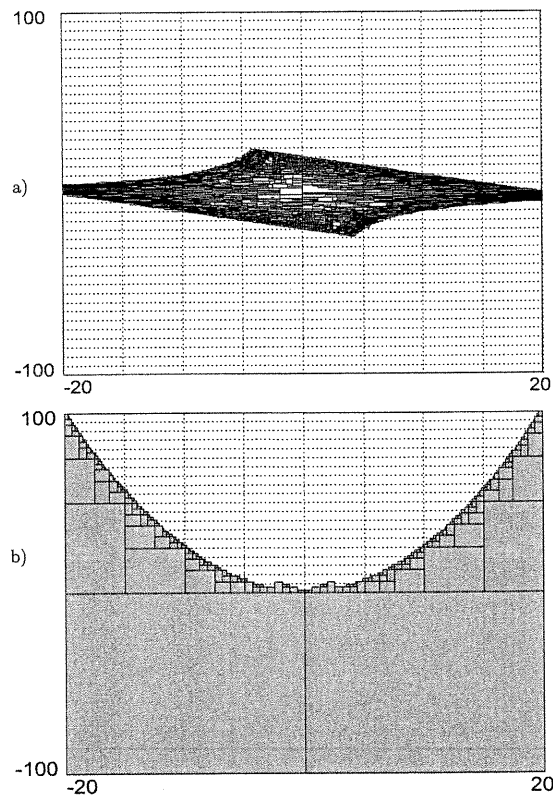


Figure 5-7: Approximation extérieure $\bar{\mathbb{S}}$ obtenue par SIVIA1P en 2 s, projetée sur \mathbb{Q} (a) et sur \mathbb{P} (b). Les espaces de recherche \mathbb{P} et \mathbb{Q} sont pris égaux à $[-20, 20] \times [-100, 100]$

Remarque 5.2.1 L'ensemble solution \mathbb{S} décrit l'intersection dans un compact $\mathbb{P} \times \mathbb{Q}$ de deux nappes paramétrées $\mathbf{r}_{\mathcal{M}}$ et $\mathbf{r}_{\mathcal{N}}$. L'étude de la discernabilité globale s'inscrit donc dans le contexte de l'intersection de deux nappes paramétrées en infographie. En ce sens, il est possible d'améliorer les performances de l'algorithme à l'aide de techniques de bisections plus adaptées, ou d'utilisation d'un autre type d'arithmétique [BB01].

Il est en particulier possible d'utiliser la description par pavage hybride proposée dans SIVIA1 pour améliorer la précision de description de cette intersection.

5.3 Conclusions

L'étape de caractérisation, et en particulier le test d'identifiabilité, est incontournable dans au moins trois types de situations:

- lorsque les paramètres dont on veut estimer une valeur numérique ont un sens physique (modèles de connaissance),
- lorsqu'il s'agit d'estimer à partir d'un vecteur de données des variables d'état non directement accessibles à la mesure et possédant un sens physique,
- lorsque les valeurs numériques des paramètres ou des variables d'état à estimer participent à une prise de décision.

On distingue alors deux méthodologies, suivant que les données ont ou non déjà été recueillies. En testant l'identifiabilité *a priori* des paramètres, on espère obtenir une conclusion qui soit aussi indépendante que possible de la valeur \mathbf{p}^* du vecteur des paramètres du modèle supposé avoir généré les données. A partir de cette conclusion, on modifiera ou non le plan d'expériences (en déplaçant les capteurs par exemple) de façon à améliorer l'identifiabilité [WP90].

Alors que l'approche standard consiste à espérer obtenir des conclusions qui ne soient fausses que pour certaines valeurs du vecteur de paramètres jugées atypiques, la notion d'identifiabilité globale sur un domaine \mathbb{P} permet de fournir une conclusion garantie, en particulier lorsqu'aucune conclusion d'ordre structurel ne peut être obtenue. Nous avons montré qu'à l'aide des techniques de calcul garanti, il est désormais possible de proposer des outils fiables capables de tester des propriétés pour *toutes* les valeurs du vecteur de paramètres prises dans un ensemble fourni *a priori*. Nous avons proposé un algorithme complètement numérique capable de fournir une réponse garantie à ce problème et de prendre en compte facilement des informations *a priori* sur les paramètres. L'avantage de ces notions dans un contexte expérimental a été souligné. Notons que cette approche est particulièrement performante lorsque l'ensemble \mathbb{S} à caractériser est vide, c'est-à-dire lorsque le modèle est globalement identifiable sur \mathbb{P} ou lorsque la paire de modèles est globalement discernable sur $\mathbb{P} \times \mathbb{Q}$. Rappelons cependant que cette étude s'effectue une fois le résumé établi à l'aide des méthodes classiques.

L'étude envisagée, effectuée avant toute saisie de données, se place dans un cadre idéalisé. Si les données ont déjà été recueillies, il n'est plus nécessaire d'obtenir des conclusions valides pour toutes ou presque toutes les valeurs de \mathbf{p}^* . En particulier, les instants de mesure et les entrées ne sont plus choisis librement, les données sont bruitées et l'erreur de caractérisation n'est pas nulle. Le cadre idéalisé de l'identifiabilité *a priori* est alors délaissé au profit de celui, naturel, de l'identification. La question d'actualité est alors de déterminer l'ensemble de toutes les valeurs du vecteur de paramètres compatibles avec les données et l'information disponible sur le bruit entachant ces données. Dans ce cas, les algorithmes ensemblistes garantis de l'analyse par intervalles fournissent également des outils intéressants, en permettant en particulier d'identifier des paramètres qui ne sont pas globalement identifiables [JKDW01] [BJKW01a] (chapitre 4). On préférera en effet directement caractériser l'ensemble \mathbb{S} de toutes les valeurs du vecteur de paramètres estimé $\hat{\mathbf{p}}$ qui sont compatibles avec les données, dans un sens à spécifier.

Chapitre 6

Incertitude de mesure d'un capteur logiciel

Nous avons considéré au chapitre 4 le problème de l'estimation paramétrique dans un contexte d'erreurs bornées. Pour un modèle donné, il est possible de caractériser de manière approchée mais garantie l'ensemble des valeurs des paramètres compatibles avec les données expérimentales recueillies et les bornes sur les erreurs de sortie du modèles. Comme cet ensemble constitue le support des valeurs estimées, sa taille permet d'avoir une idée de l'incertitude d'estimation associée à ces paramètres.

Cependant, un expérimentateur peut souhaiter connaître *a priori*, c'est-à-dire avant d'effectuer une quelconque mesure, les performances de la chaîne de mesure (capteur et estimateur) qu'il s'apprête à utiliser. Les raisons qui le motivent pour obtenir une telle information peuvent être liées :

- au coût de l'expérience (prix des capteurs),
- au manque de temps disponible (durée de l'expérience importante),
- au fait qu'il n'est pas possible d'intervenir sur l'expérience en cours pour ajuster ou changer certains capteurs.

Ceci sera en particulier intéressant lorsqu'un cahier des charges impose une borne à l'erreur maximale tolérable sur les paramètres à estimer. Ainsi, si l'incertitude déterminée *a priori* ne permet pas de satisfaire cette contrainte, l'expérimentateur saura que même s'il menait l'expérience à terme, les objectifs du cahier des charges pourraient ne pas être atteints. En conséquence, il devra modifier la chaîne de mesure qu'il envisageait d'utiliser. Ce chapitre cherche à apporter une solution à ce problème d'évaluation *a priori* des performances d'une chaîne de mesure, et d'aide à l'amélioration de ces performances si nécessaire.

La procédure d'estimation d'une grandeur x à partir de données expérimentales s'articule en deux étapes. La première consiste à prélever les données y à l'aide de capteurs physiques. La seconde met en œuvre un traitement de ces données de manière à obtenir une estimée \hat{x} de x . Lorsque ce traitement est numérique, ces deux étapes peuvent être assimilées à la construction d'un *capteur logiciel* délivrant une *mesure* \hat{x} de x . Déterminer *a priori* l'incertitude d'estimation de x revient ainsi à quantifier *a priori* les performances de ce capteur logiciel.

Cette démarche est classique en métrologie dans le cas de capteurs physiques. Pour cela, les organismes officiels tels que le Bureau National de Métrologie (BNM) préconisent de soumettre ce type de capteurs à une *évaluation*, dont le but est de "*constater à l'aide de méthodes d'essais appropriées, les performances et les caractéristiques du capteur objet de l'essai*" [RMP93]. Le

cadre de l'évaluation est alors purement probabiliste : les grandeurs considérées sont modélisées par des variables aléatoires, et un grand nombre d'essais est nécessaire avant de parvenir à des conclusions.

Cependant, cette procédure ne s'applique qu'à une évaluation *a posteriori*, c'est-à-dire une fois que des mesures sont disponibles. Ce type de modélisation est inadapté dans le cadre de ce chapitre, car nous souhaitons tirer des conclusions avant d'effectuer une quelconque mesure. Seules des informations *a priori* sur le modèle des capteurs physiques et le support de l'erreur de mesure seront supposées disponibles. Plus précisément, les erreurs de mesure et de modèle seront considérées bornées, avec des bornes connues.

Dans un premier temps, nous rappelons les définitions usuellement employées en métrologie pour caractériser un capteur physique, définitions que nous étendrons ensuite à notre contexte d'erreurs bornées (paragraphe 6.1). L'ensemble des hypothèses de l'étude sera rassemblé au paragraphe 6.2. Cela permettra de poser le problème de l'estimation *a priori* des performances d'un capteur au paragraphe 6.3. Afin d'étudier la faisabilité de l'approche, une application sera traitée dans le paragraphe 6.4 à l'aide des techniques intervalles. Pour l'extension au cas de données multiples, de lecture plus difficile, nous renvoyons le lecteur à l'annexe (section 8.3).

6.1 Capteurs

Un *capteur* est un organe de prélèvement d'information composé de deux entités :

- Un transducteur élabore, à partir de la grandeur physique x d'intérêt, une grandeur physique de nature différente (très souvent électrique) y . La relation qui lie les valeurs de x et de y est appelée *fonction de transfert* du capteur.
- Un organe de traitement fournit ensuite à partir de y une mesure \hat{x} de x . Dans le cas statique, la relation qui permet d'obtenir \hat{x} à partir de y est appelée *courbe d'étalonnage*.

Idéalement, la fonction de transfert et la courbe d'étalonnage sont inverses l'une de l'autre, ce qui est très fréquemment supposé implicitement en pratique. Seule la connaissance de la fonction de transfert est alors nécessaire. Elle est fournie par le constructeur sous la forme d'un graphe ou d'une relation fonctionnelle. L'exemple suivant présente ces deux entités sur trois types de capteurs de température, classés selon leur complexité technologique.

Exemple 6.1.1 1. *Le principe du thermomètre à dilatation (figure 6-1) repose sur la variation du volume V de liquide dans le réservoir en fonction de la température ambiante T à estimer. À l'aide du modèle affine $V(T) = V_0(1 + aT)$ de la fonction de transfert, la courbe d'étalonnage est construite sous forme d'une échelle accolée au thermomètre de manière à pouvoir lire directement la valeur de T .*

2. *Avec un thermomètre à tension de vapeur saturante, la mesure d'une température T se fait par l'intermédiaire de la mesure d'une pression de vapeur saturante P_s . La fonction de transfert est donc une relation non-linéaire $P_s = f(T)$, illustrée par l'abaque de la figure 6-2. La courbe d'étalonnage est la relation $T = g(P_s)$ obtenue par lecture directe sur la même abaque.*

3. *La fonction de transfert d'un thermomètre à résistance est obtenue en modélisant la loi de variation de la résistance électrique d'un conducteur métallique en fonction de la température à l'aide d'un modèle polynômial :*

$$R(T) = f(T) = R_0(1 + aT + bT^2 + cT^3). \quad (6.1)$$

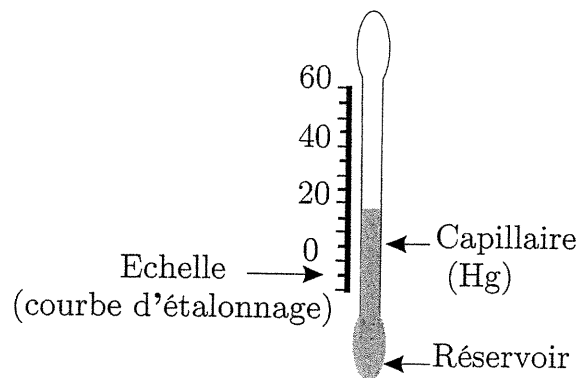


Figure 6-1: Thermomètre à dilatation

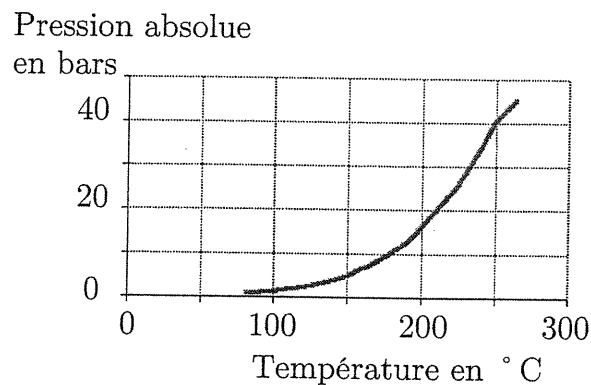


Figure 6-2: Fonction de transfert $P = f(T)$ d'un thermomètre à vapeur saturante, pour de l'eau.

Ce type de modèle, également non-linéaire, est choisi univoque dans la zone d'utilisation du thermomètre. Il reste donc à inverser ce modèle pour obtenir la courbe d'étalonnage.

Ces trois capteurs physiques ont des fonctions de transfert linéaire ou non-linéaire. En particulier, nous disposons d'un modèle des fonctions de transfert du premier et du dernier capteur. Ils se prêtent donc naturellement à la réalisation d'un capteur *logiciel*, c'est-à-dire d'un capteur intégrant un organe de traitement numérique, qui fournit la courbe d'étalonnage. Alors que dans le cas du thermomètre à dilatation, l'inversion de la fonction de transfert est aisée, l'obtention d'une courbe d'étalonnage requiert un calculateur numérique dans le cas du thermomètre à résistance. Pour le thermomètre à dilatation, il est nécessaire d'établir un modèle de f et g .

L'exemple qui précède montre bien qu'il peut exister plusieurs techniques pour mesurer une même grandeur physique x . Leurs comportements peuvent cependant être très différents suivant la valeur prise par x . Afin de pouvoir comparer ces capteurs entre eux, les normes de la métrologie ont introduit un ensemble de caractéristiques d'un capteur, que nous rappelons dans le paragraphe suivant.

6.1.1 Caractéristiques métrologiques d'un capteur

L'ensemble des définitions suivantes s'appuie sur la norme NFX07-001 pour caractériser les capteurs et les appareils de mesure [fNX86].

Définition 6.1.1 (Mesurage) Ensemble d'opérations ayant pour but de déterminer une valeur d'une grandeur.

Définition 6.1.2 (Mesurande) Grandeur particulière soumise à mesurage.

Définition 6.1.3 (Erreur de mesure) Ecart entre la valeur \hat{x} de la mesure et la valeur vraie du mesurande x^* .

Remarque 6.1.1 Les principaux types d'erreur sont résumés sur la figure 6-3 :

- a) erreur de zéro ou d'offset,
- b) erreur d'échelle,
- c) erreur de linéarité,
- d) erreur due au phénomène d'hystérésis,
- e) erreur de mobilité.

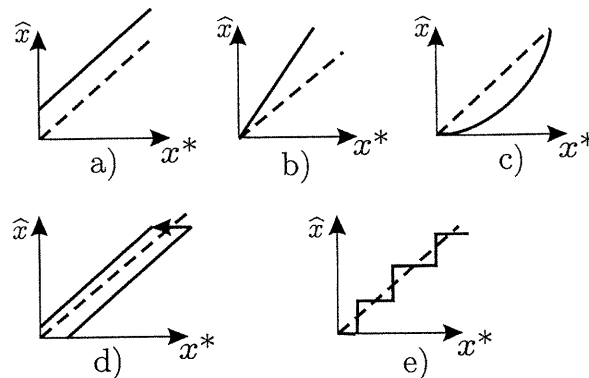


Figure 6-3: Les différents types d'erreurs de mesure. La caractéristique $\hat{x} = \hat{x}(x)$ du capteur logiciel est représentée en trait continu, la caractéristique idéale est en pointillés.

Définition 6.1.4 (Incertitude de mesure) Borne supérieure δx de la valeur absolue de l'erreur de mesure, elle caractérise l'étroitesse de l'accord entre le résultat d'un mesurage et la valeur vraie du mesurande. Le résultat d'une mesure x doit être fourni avec l'incertitude de mesure $\delta \hat{x}$ associée, de sorte que l'on puisse écrire :

$$\hat{x} - \delta x \leq x^* \leq \hat{x} + \delta x. \quad (6.2)$$

Définition 6.1.5 (Etendue de mesure) Ensemble des valeurs du mesurande x pour lesquelles les indications du capteur obtenues à l'intérieur du domaine nominal d'emploi en une seule mesure ne doivent pas être entachées d'une erreur supérieure à l'erreur de mesure maximale tolérée (voir la figure 6-4).

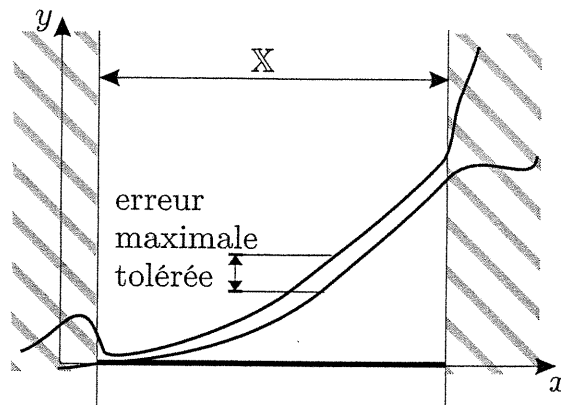


Figure 6-4: Etendue de mesure

6.1.2 Evaluation : approche stochastique

Lors de l'évaluation d'un capteur, de nombreuses caractéristiques sont testées. Parmi elles figurent la sensibilité et le temps de réponse... Ici, seules les notions de justesse et d'erreur de fidélité seront mentionnées. Ces grandeurs, qui permettent de comparer les qualités respectives de différents appareils de mesure, sont définies dans le cadre d'une modélisation stochastique de x , y et \hat{x} .

6.1.2.1 Défaut de justesse j

Un appareil de mesure est d'autant plus juste que la valeur moyenne des mesures répétées dans les mêmes conditions expérimentales est proche de la valeur vraie x^* . Le défaut de justesse j est défini par

$$j = E(\hat{x}) - x^*.$$

Plus la valeur absolue de j est petite, plus l'appareil est juste.

6.1.2.2 Fidélité et erreur de fidélité Δx

La *fidélité* est définie comme la qualité d'un appareil de mesure dont la variabilité des erreurs est faible. Dans un cadre probabiliste, l'erreur de fidélité Δx est souvent assimilée à l'écart-type de la mesure z :

$$\Delta x = \sqrt{E((\hat{x} - E(\hat{x}))^2)} = \sqrt{E((\hat{x} - x^* - j)^2)}.$$

Remarque 6.1.2 Nous dirons qu'un appareil est précis s'il est à la fois juste et fidèle. La figure 6-5 illustre le comportement de trois capteurs. Le capteur A est moins fidèle que les capteurs B et C, et le capteur B est moins juste que les capteurs A et C. Le capteur C est le plus précis.

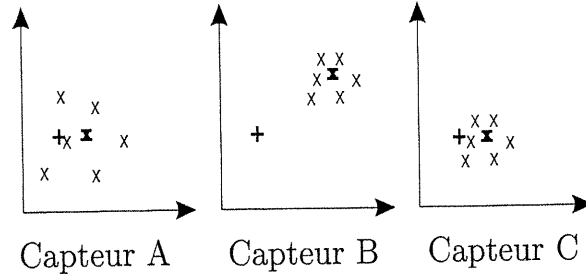


Figure 6-5: Mesures obtenues dans les mêmes conditions expérimentales avec trois capteurs de caractéristiques différentes (la vraie valeur x^* du mesurande est représentée par (+) et la moyenne $E(z)$ des mesures par une croix (x) noire).

6.1.3 Evaluation : approche intervalles

Dans le contexte de l'évaluation *a priori* des caractéristiques d'un capteur, les définitions faisant intervenir des propriétés statistiques de x , y et \hat{x} doivent être adaptées. Ainsi, il est nécessaire d'introduire de nouvelles définitions de l'incertitude de mesure, du défaut de justesse et de l'erreur de fidélité, ne faisant intervenir que les connaissances disponibles *a priori* sur le modèle et sur le caractère borné des erreurs de mesure.

Nous supposons que la vraie valeur du mesurande est x^* . Dans notre contexte, un résultat de mesure sera un intervalle $[\hat{x}] = [\underline{\hat{x}}, \overline{\hat{x}}]$, de centre

$$\hat{x}_0 = \frac{1}{2} (\overline{\hat{x}} + \underline{\hat{x}}),$$

et de rayon

$$r([\hat{x}]) = \frac{1}{2} (\overline{\hat{x}} - \underline{\hat{x}}).$$

6.1.3.1 Incertitude de mesure $\delta x(x^*)$

L'incertitude de mesure $\delta x(x^*)$ est alors l'erreur de mesure maximale commise, donnée par

$$\delta x(x^*) = \max(|\underline{\hat{x}} - x^*|, |\overline{\hat{x}} - x^*|). \quad (6.3)$$

Remarque 6.1.3 $\delta x(x^*)$ correspond donc à la magnitude de l'intervalle $[\hat{x}] - x^*$, où la magnitude de $[x]$ est définie par :

$$\text{mag}([x]) \triangleq \max(|\underline{x}|, |\overline{x}|).$$

6.1.3.2 Défaut de justesse $j(x^*)$

Nous choisissons de définir le défaut de justesse comme

$$j(x^*) = \hat{x}_0 - x^*. \quad (6.4)$$

6.1.3.3 Erreur de fidélité

Dans notre contexte, nous posons que l'erreur de fidélité d'un capteur quelconque de grandeur de sortie \hat{x} est :

$$\begin{aligned}\Delta x(x^*) &= \max \left(|\hat{x} - x^* - j(x^*)|, |\bar{\hat{x}} - x^* - j(x^*)| \right) \\ &= \max \left(|\hat{x} - \hat{x}_0|, |\bar{\hat{x}} - \hat{x}_0| \right) = |\hat{x} - \hat{x}_0| = |\bar{\hat{x}} - \hat{x}_0|\end{aligned}$$

d'où

$$\Delta x(x^*) = r([\hat{x}]). \quad (6.5)$$

Nous dirons alors qu'un appareil est *précis* pour la mesure x^* lorsque l'incertitude de mesure associée est faible, c'est-à-dire lorsqu'il est à la fois juste et fidèle. L'ensemble de ces grandeurs est illustré sur la figure 6-6.

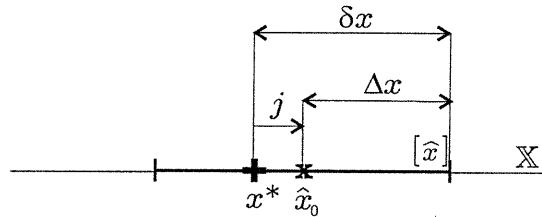


Figure 6-6: Défaut de justesse j , erreur de fidélité Δx et incertitude de mesure δx d'un capteur logiciel, pour la valeur vraie du mesurande x^*

L'exemple suivant explicite le calcul des caractéristiques définies précédemment pour un capteur logiciel très simple.

Exemple 6.1.2 Soit un capteur logiciel d'étendue de mesure $X = [0.1, 10]$ pour lequel nous disposons de modèles

- de la fonction de transfert

$$y = f(x) = \frac{1}{x},$$

- de la courbe d'étalonnage

$$\hat{x} = g(y) = \frac{1}{y}.$$

Supposons que f et g décrivent parfaitement le dispositif de mesure, et que le capteur physique est juste. Si la vraie valeur à mesurer est $x^* = 0.5$, le capteur physique délivre donc en sortie la donnée expérimentale $[y]$ de centre $\hat{y} = f(x^*) = 2$, et dont le rayon correspond à l'incertitude de mesure du capteur, notée δy .

Supposons que l'incertitude de mesure δy vaut 1. La donnée recueillie est alors une donnée intervalle $[y] = \hat{y} + \delta y[-1, 1] = [1, 3]$. La mesure obtenue $[\hat{x}]$ vaut alors

$$[\hat{x}] = g([y]) = \left[\frac{1}{3}, 1 \right],$$

d'incertitude de mesure $\delta x(x^*) = \max(|\frac{1}{3} - \frac{1}{2}|, |1 - \frac{1}{2}|) = 0.5$, et de centre $\hat{x}_0 = \frac{2}{3} \neq x^*$. D'après (6.4), le défaut de justesse du capteur logiciel vaut donc $j(x^*) = \frac{2}{3} - \frac{1}{2} = \frac{1}{6}$. Bien que le capteur physique soit juste, la présence d'une incertitude de mesure introduit un défaut de justesse dans le capteur logiciel. D'après (6.5) l'erreur de fidélité Δx du capteur logiciel vaut $r([\hat{x}]) = \frac{1}{2}(1 - \frac{1}{3}) = 0.34$. Ces grandeurs sont représentées sur la figure 6-7.

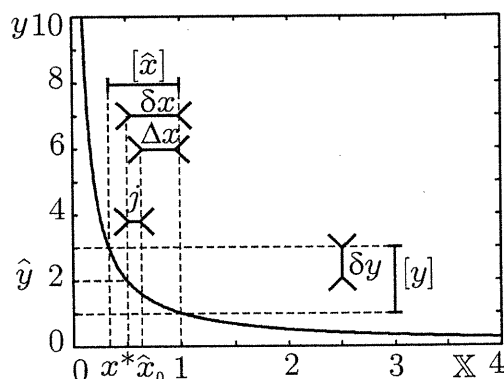


Figure 6-7: Calcul de l'incertitude de mesure, de la justesse et de l'erreur de fidélité pour le capteur de l'exemple 6.1.2

Remarque 6.1.4 L'exemple précédent souligne que l'image par une fonction non-linéaire d'un intervalle ne conserve ni le centre, ni le rayon. Ainsi, le calcul classique effectué à partir de la différentielle logarithmique est erroné, puisqu'il met en œuvre une linéarisation, ce qui suppose que la fonction f est lentement variable au point x^* . En l'occurrence, on obtiendrait pour l'exemple 6.1.2 une incertitude de mesure $\delta^l x = \frac{1}{2} \hat{x} \frac{\delta y}{y} = \frac{1}{2} \frac{\delta y}{y^2} = 0.25$.

Dans les prochains paragraphes, nous verrons que, suivant les hypothèses, seules certaines des caractéristiques présentées pourront être déterminées *a priori*. Au préalable, nous précisons le cadre de l'étude.

6.2 Modélisation d'un capteur logiciel

Dans toute la suite, et sur toute l'étendue de mesure, la relation entre le mesurande x et la sortie $y(x)$ du capteur physique intégré dans le capteur logiciel étudié sera décomposée en

$$y(x) = f(x, \mathbf{e}) + b(x), \quad (6.6)$$

où $f(\cdot, \cdot)$ est une fonction continue connue, $\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]$ décrit l'erreur de modélisation de la fonction de transfert et l'erreur systématique $m(e_y)$ du capteur physique (défaut de justesse), avec $\underline{\mathbf{e}}$ et $\bar{\mathbf{e}}$ supposés connus. La composante centrée de l'erreur de mesure e_y entachant le signal de sortie du capteur physique est représentée par $b(x) \in [-\Delta y(x), \Delta y(x)]$, où $\Delta y(x)$ correspond à l'erreur de fidélité du capteur physique pour la valeur x du mesurande, qui sera suivant les cas considérée comme connue ou à déterminer.

En outre, nous supposons disposer d'une courbe d'étalonnage décrite par une fonction continue

$$\hat{x} = g(y), \quad (6.7)$$

surjective sur toute l'étendue de mesure \mathbb{X} . Les fonctions f et g sont non-linéaires dans le cas général.

6.3 Détermination *a priori* des caractéristiques d'un capteur

Deux types de problèmes peuvent être considérés.

Le problème *direct* consiste à déterminer *a priori* certaines caractéristiques d'un capteur logiciel connaissant les caractéristiques du capteur physique intégré (en particulier $\Delta y(x)$).

Le problème *inverse* consiste quant à lui à déterminer les performances qu'un capteur physique devra atteindre afin que le capteur logiciel qui l'intègre satisfasse certaines contraintes de performances établies dans un cahier des charges (par exemple sur $\delta x(x)$).

6.3.1 Problème direct

D'après (6.6), si une mesure était effectuée, l'ensemble des valeurs possibles en sortie du capteur physique est

$$[y](x^*) = f(x^*, [\underline{e}, \bar{e}]) + \Delta y(x^*) [-1, 1],$$

où $\Delta y(x^*)$ est supposée connue dans ce contexte. L'ensemble des valeurs que peut alors prendre la sortie du capteur est

$$\begin{aligned} [\hat{x}](x^*) &= g([y](x^*)) \\ &= g(f(x^*, [\underline{e}, \bar{e}]) + \Delta y(x^*) [-1, 1]). \end{aligned}$$

Posons

$$\begin{aligned} h^d : \mathbb{X} \times [-1, 1] \times [\underline{e}, \bar{e}] &\longmapsto \mathbb{R} \\ (x^*, \varepsilon, \mathbf{e}) &\longmapsto g(f(x^*, \mathbf{e}) + \varepsilon \cdot \Delta y(x^*)), \end{aligned}$$

nous avons alors $[\hat{x}](x^*) = h^d(x^*, [-1, 1], [\underline{e}, \bar{e}])$.

Deux situations peuvent alors être considérées.

6.3.1.1 Dans la première situation, on suppose que $[\underline{e}, \bar{e}] = 0$

Dans ces conditions, le capteur physique est juste. Il est possible de calculer, pour tout $x^* \in \mathbb{X}$:

- le défaut de justesse du capteur logiciel

$$\begin{aligned} j(x^*) &= \hat{x}_0(x^*) - x^* \\ &= \frac{1}{2} \left(\min_{\varepsilon \in [-1, 1]} h^d(x^*, \varepsilon, \mathbf{0}) + \max_{\varepsilon \in [-1, 1]} h^d(x^*, \varepsilon, \mathbf{0}) \right) - x^*, \\ &= \frac{1}{2} \left(\min_{\varepsilon \in [-1, 1]} h^d(x^*, \varepsilon, \mathbf{0}) - x^* + \max_{\varepsilon \in [-1, 1]} h^d(x^*, \varepsilon, \mathbf{0}) - x^* \right), \end{aligned} \tag{6.8}$$

- l'erreur de fidélité du capteur logiciel

$$\begin{aligned} \Delta x(x^*) &= r([\hat{x}](x^*)) \\ &= \frac{1}{2} \left(\max_{\varepsilon \in [-1, 1]} h^d(x^*, \varepsilon, \mathbf{0}) - \min_{\varepsilon \in [-1, 1]} h^d(x^*, \varepsilon, \mathbf{0}) \right), \end{aligned} \tag{6.9}$$

- l'incertitude du capteur logiciel

$$\begin{aligned}
\delta x(x^*) &= \max(|\widehat{x}(x^*) - x^*|, |\overline{\widehat{x}}(x^*) - x^*|) \\
&= \max\left(\left|\min_{\varepsilon \in [-1,1]} h^d(x^*, \varepsilon, \mathbf{0}) - x^*\right|, \left|\max_{\varepsilon \in [-1,1]} h^d(x^*, \varepsilon, \mathbf{0}) - x^*\right|\right) \\
&= \max_{\varepsilon \in [-1,1]} |h^d(x^*, \varepsilon, \mathbf{0}) - x^*|.
\end{aligned} \tag{6.10}$$

Cependant, sauf lors de la procédure d'étalonnage d'un capteur, la valeur de x^* n'est pas connue. Nous nous intéressons donc aux caractéristiques correspondant au pire cas.

La valeur maximale \bar{j} de $|j(x^*)|$ sur l'étendue de mesure est

$$\bar{j} = \max\left(-\min_{x^* \in \mathbb{X}} j(x^*), \max_{x^* \in \mathbb{X}} j(x^*)\right). \tag{6.11}$$

La valeur maximale de l'erreur de fidélité sur l'étendue de mesure est

$$\overline{\Delta x} = \max_{x^* \in \mathbb{X}} \Delta \widehat{x}(x^*). \tag{6.12}$$

La valeur maximale $\overline{\delta x}$ de l'incertitude de mesure est

$$\begin{aligned}
\overline{\delta x} &= \max_{x^* \in \mathbb{X}} \delta x(x^*) \\
&= \max_{x^* \in \mathbb{X}} \max\left(\left|\min_{\varepsilon \in [-1,1]} h^d(x^*, \varepsilon, \mathbf{0}) - x^*\right|, \left|\max_{\varepsilon \in [-1,1]} h^d(x^*, \varepsilon, \mathbf{0}) - x^*\right|\right) \\
&= \max_{x^* \in \mathbb{X}} \max_{\varepsilon \in [-1,1]} |h^d(x^*, \varepsilon, \mathbf{0}) - x^*| \\
&= \max\left(\max_{x^* \in \mathbb{X}} \left|\min_{\varepsilon \in [-1,1]} h^d(x^*, \varepsilon, \mathbf{0}) - x^*\right|, \max_{x^* \in \mathbb{X}} \left|\max_{\varepsilon \in [-1,1]} h^d(x^*, \varepsilon, \mathbf{0}) - x^*\right|\right).
\end{aligned} \tag{6.13}$$

6.3.1.2 Dans la seconde situation, on suppose que $[\underline{e}, \bar{e}] \neq \mathbf{0}$

Dans ces conditions pour tout $x^* \in \mathbb{X}$, il n'est possible de fournir qu'un encadrement du défaut de justesse :

$$j(x^*) \in \left[\min_{\mathbf{e} \in [\underline{e}, \bar{e}]} j(x^*, \mathbf{e}), \max_{\mathbf{e} \in [\underline{e}, \bar{e}]} j(x^*, \mathbf{e}) \right], \tag{6.14}$$

où

$$j(x^*, \mathbf{e}) = \frac{1}{2} \left(\min_{\varepsilon \in [-1,1]} h^d(x^*, \varepsilon, \mathbf{e}) + \max_{\varepsilon \in [-1,1]} h^d(x^*, \varepsilon, \mathbf{e}) \right) - x^*. \tag{6.15}$$

De même pour l'incertitude de mesure $\delta x(x^*)$:

$$\delta x(x^*) \in \left[\min_{\mathbf{e} \in [\underline{e}, \bar{e}]} \delta x(x^*, \mathbf{e}), \max_{\mathbf{e} \in [\underline{e}, \bar{e}]} \delta x(x^*, \mathbf{e}) \right],$$

où

$$\begin{aligned} \delta x(x^*, \mathbf{e}) &= \max \left(\left| \min_{\varepsilon \in [-1, 1]} h^d(x^*, \varepsilon, \mathbf{e}) - x^* \right|, \left| \max_{\varepsilon \in [-1, 1]} h^d(x^*, \varepsilon, \mathbf{e}) - x^* \right| \right) \\ &= \max_{\varepsilon \in [-1, 1]} |h^d(x^*, \varepsilon, \mathbf{e}) - x^*|. \end{aligned} \quad (6.16)$$

Par contre, pour toute valeur de $x^* \in \mathbb{X}$, nous pouvons calculer l'erreur de fidélité du capteur

$$\begin{aligned} \Delta x(x^*) &= r([\widehat{x}](x^*)) \\ &= \frac{1}{2} \left(\max_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}], \varepsilon \in [-1, 1]} h^d(x^*, \varepsilon, \mathbf{e}) - \min_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}], \varepsilon \in [-1, 1]} h^d(x^*, \varepsilon, \mathbf{e}) \right), \end{aligned}$$

La valeur maximale $\overline{\Delta x}$ de l'erreur de fidélité sur l'étendue de mesure est simplement donnée par

$$\overline{\Delta x} = \max_{x^* \in \mathbb{X}} \Delta x(x^*). \quad (6.17)$$

Les deux résultats (6.13) et (6.17) font intervenir des problèmes de type MinMax, pour lesquels nous disposons d'un algorithme de résolution garanti (voir le paragraphe 6.4).

Le problème inverse consistant à déterminer *a priori* l'erreur de fidélité du capteur physique maximale acceptable $\overline{\Delta y}$ sera traité au paragraphe suivant. Comme il est plus complexe que le problème direct, nous allons formuler tout d'abord plusieurs hypothèses simplificatrices que nous relâcherons progressivement pour parvenir à résoudre le problème dans le cas général.

6.3.2 Problème inverse

6.3.2.1 On suppose x^* connu

Cette situation intervient par exemple lors de l'étalonnage d'un capteur : le cahier des charges fournit x^* (par exemple à l'aide d'un étalon de référence) et impose l'incertitude de mesure δx .

On cherche donc la plus grande valeur de $\Delta y = \Delta y(x^*)$ telle que

$$\forall y \in [y](x^*) = f(x^*, [\underline{\mathbf{e}}, \bar{\mathbf{e}}]) + \Delta y(x^*)[-1, 1], \quad (6.18)$$

on ait

$$\widehat{x} = g(y) \in [x^* - \delta x, x^* + \delta x],$$

ce qui est équivalent à

$$g([y](x^*)) \subset x^* + [-\delta x, \delta x]. \quad (6.19)$$

On suppose $[\underline{\mathbf{e}}, \bar{\mathbf{e}}] = \mathbf{0}$ Le capteur physique est donc juste, aussi

$$\Delta y(x^*) = \delta y(x^*)$$

et

$$[y](x^*) = f(x^*, \mathbf{0}) + \Delta y(x^*)[-1, 1]; \quad (6.20)$$

On recherche donc la plus grande valeur de $\Delta y(x^*)$ telle que

$$g(f(x^*, \mathbf{0}) + \Delta y(x^*)[-1, 1]) \subset x^* + [-\delta x, \delta x]. \quad (6.21)$$

Comme g est surjective, (6.21) est équivalente à

$$\Delta y(x^*)[-1, 1] \subset g^{-1}(x^* + [-\delta x, \delta x]) - f(x^*, \mathbf{0}). \quad (6.22)$$

On pose

$$h^i : \mathbb{X} \times [-\delta x, \delta x] \times [\underline{e}, \bar{e}] \longrightarrow \mathbb{R} \\ (x^*, \varepsilon, \mathbf{e}) \longmapsto g^{-1}(x^* + \varepsilon) - f(x^*, \mathbf{e}).$$

Les fonctions f et g étant continues, h^i est également continue et la condition (6.22) devient alors :

$$\begin{cases} -\Delta y(x^*) \geq \min_{\varepsilon \in [-\delta x, \delta x]} h^i(x^*, \varepsilon, \mathbf{0}) \\ \Delta y(x^*) \leq \max_{\varepsilon \in [-\delta x, \delta x]} h^i(x^*, \varepsilon, \mathbf{0}) \end{cases}$$

d'où

$$\Delta y(x^*) = \min \left(-\min_{\varepsilon \in [-\delta x, \delta x]} h^i(x^*, \varepsilon, \mathbf{0}), \max_{\varepsilon \in [-\delta x, \delta x]} h^i(x^*, \varepsilon, \mathbf{0}) \right). \quad (6.23)$$

Ce résultat est illustré sur la figure 6-8.

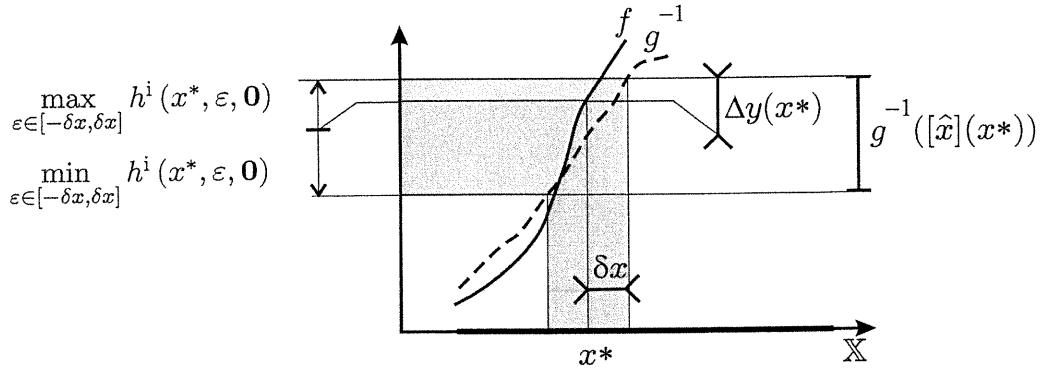


Figure 6-8: Calcul de $\Delta y(x^*)$ dans le cas où $\underline{e} = \bar{e} = 0$

Comme $\Delta y(x^*)$ doit être positif, si

$$\min_{\varepsilon \in [-\delta x, \delta x]} h^i(x^*, \varepsilon, \mathbf{0}) > 0$$

ou

$$\max_{\varepsilon \in [-\delta x, \delta x]} h^i(x^*, \varepsilon, \mathbf{0}) < 0$$

c'est-à-dire si

$$0 \notin h^i(x^*, [-\delta x, \delta x], 0)$$

ou de manière équivalente, si

$$f(x^*, 0) \notin g^{-1}(x^* + [-\delta x, \delta x]),$$

il n'existe pas de $\Delta y(x^*)$ qui satisfasse le cahier des charges.

Remarque 6.3.1 Dans ce cas, les modèles de f et g sont incompatibles avec la valeur de δx imposée. Il faut alors soit construire un modèle de courbe d'étalonnage plus précis pour le même capteur, soit changer de capteur physique.

On suppose maintenant que $[\underline{e}, \bar{e}] \neq 0$ L'équation (6.20) devient

$$[y](x^*) = f(x^*, [\underline{e}, \bar{e}]) + \Delta y(x^*)[-1, 1].$$

On cherche à nouveau $\Delta y(x^*)$ maximale telle que

$$g(f(x^*, [\underline{e}, \bar{e}]) + \Delta y(x^*)[-1, 1]) \subset x^* + [-\delta x, \delta x], \quad (6.24)$$

soit encore

$$\Delta y(x^*)[-1, 1] + f(x^*, [\underline{e}, \bar{e}]) \subset g^{-1}(x^* + [-\delta x, \delta x]). \quad (6.25)$$

Or

$$\Delta y(x^*)[-1, 1] + f(x^*, [-\underline{e}, \bar{e}]) = \left[-\Delta y(x^*) + \min_{\mathbf{e} \in [-\underline{e}, \bar{e}]} f(x^*, \mathbf{e}), \Delta y(x^*) + \max_{\mathbf{e} \in [-\underline{e}, \bar{e}]} f(x^*, \mathbf{e}) \right].$$

La condition (6.24) est donc équivalente à :

$$\begin{cases} -\Delta y(x^*) + \min_{\mathbf{e} \in [-\underline{e}, \bar{e}]} f(x^*, \mathbf{e}) \geq \min_{\varepsilon \in [-\delta x, \delta x]} g^{-1}(x^* + \varepsilon) \\ \Delta y(x^*) + \max_{\mathbf{e} \in [-\underline{e}, \bar{e}]} f(x^*, \mathbf{e}) \leq \max_{\varepsilon \in [-\delta x, \delta x]} g^{-1}(x^* + \varepsilon), \end{cases}$$

soit encore

$$\begin{cases} \Delta y(x^*) \leq - \left(\min_{\varepsilon \in [-\delta x, \delta x]} g^{-1}(x^* + \varepsilon) - \min_{\mathbf{e} \in [-\underline{e}, \bar{e}]} f(x^*, \mathbf{e}) \right), \\ \Delta y(x^*) \leq \max_{\varepsilon \in [-\delta x, \delta x]} g^{-1}(x^* + \varepsilon) - \max_{\mathbf{e} \in [-\underline{e}, \bar{e}]} f(x^*, \mathbf{e}), \end{cases}$$

$$\begin{cases} \Delta y(x^*) \leq - \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e} \in [-\underline{e}, \bar{e}]} h^i(x^*, \varepsilon, \mathbf{e}), \\ \Delta y(x^*) \leq \max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e} \in [-\underline{e}, \bar{e}]} h^i(x^*, \varepsilon, \mathbf{e}). \end{cases} \quad (6.26)$$

Par conséquent

$$\Delta y(x^*) = \min \left(- \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e} \in [-\underline{e}, \bar{e}]} h^i(x^*, \varepsilon, \mathbf{e}), \max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e} \in [-\underline{e}, \bar{e}]} h^i(x^*, \varepsilon, \mathbf{e}) \right) \quad (6.27)$$

Si $\min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} h^i(x^*, \varepsilon, \mathbf{e}) > 0$ ou $\max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} h^i(x^*, \varepsilon, \mathbf{e}) < 0$, il n'existe à nouveau pas de solution $\Delta y(x^*)$ satisfaisante (voir également la remarque 6.3.1).

La figure 6-9 illustre le calcul de $\Delta y(x^*)$ tel qu'il est décrit par (6.27).

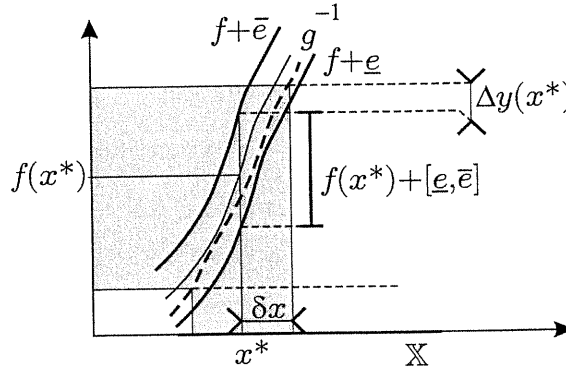


Figure 6-9: Calcul de $\Delta y(x^*)$ dans le cas où $[\underline{\mathbf{e}}, \bar{\mathbf{e}}] \neq 0$

6.3.2.2 On suppose x^* inconnu

Cette situation est la plus fréquemment rencontrée en pratique, puisque la seule information disponible avant la mesure est le fait que $x^* \in \mathbb{X}$. Nous allons nous intéresser à la caractérisation de la plus grande valeur de l'erreur de fidélité $\overline{\Delta y}$ du capteur physique telle que l'incertitude de mesure de x^* reste inférieure à une valeur spécifiée δx par le cahier des charges, et cela, quel que soit x^* appartenant à l'étendue de mesure \mathbb{X} .

La valeur $\overline{\Delta y}$ recherchée est la borne supérieure de l'ensemble des valeurs Δy satisfaisant

$$\forall x^* \in \mathbb{X}, g([y](x^*)) \subset x^* + [-\delta x, \delta x],$$

avec

$$[y](x^*) = f(x^*, [\underline{\mathbf{e}}, \bar{\mathbf{e}}]) + \Delta y [-1, 1].$$

$\overline{\Delta y}$ est donc la plus grande valeur de Δy satisfaisant

$$\forall x^* \in \mathbb{X}, g(f(x^*, [\underline{\mathbf{e}}, \bar{\mathbf{e}}]) + \Delta y [-1, 1]) \subset x^* + [-\delta x, \delta x]. \quad (6.28)$$

Cette contrainte est équivalente à

$$\forall x^* \in \mathbb{X}, f(x^*, [\underline{\mathbf{e}}, \bar{\mathbf{e}}]) + \Delta y [-1, 1] \subset g^{-1}(x^* + [-\delta x, \delta x]),$$

qui à son tour se décompose en deux sous-contraintes

$$\forall x^* \in \mathbb{X}, \begin{cases} \min_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} f(x^*, \mathbf{e}) - \Delta y \geq \min_{\varepsilon \in [-\delta x, \delta x]} g^{-1}(x^* + \varepsilon) \\ \max_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} f(x^*, \mathbf{e}) + \Delta y \leq \max_{\varepsilon \in [-\delta x, \delta x]} g^{-1}(x^* + \varepsilon), \end{cases}$$

c'est-à-dire

$$\forall x^* \in \mathbb{X}, \begin{cases} \Delta y \leq - \left(\min_{\varepsilon \in [-\delta x, \delta x]} g^{-1}(x^* + \varepsilon) - \min_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} f(x^*, \mathbf{e}) \right) \\ \Delta y \leq \max_{\varepsilon \in [-\delta x, \delta x]} g^{-1}(x^* + \varepsilon) - \max_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} f(x^*, \mathbf{e}), \end{cases}$$

ou encore

$$\forall x^* \in \mathbb{X}, \begin{cases} \Delta y \leq - \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} h^i(x^*, \varepsilon, \mathbf{e}) \\ \Delta y \leq \max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} h^i(x^*, \varepsilon, \mathbf{e}). \end{cases}$$

Par conséquent, $\overline{\Delta y}$ vaut

$$\begin{aligned} \overline{\Delta y} &= \min_{x^* \in \mathbb{X}} \min \left(- \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} h^i(x^*, \varepsilon, \mathbf{e}), \max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} h^i(x^*, \varepsilon, \mathbf{e}) \right) \\ &= \min \left(\min_{x^* \in \mathbb{X}} \left(- \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} h^i(x^*, \varepsilon, \mathbf{e}) \right), \min_{x^* \in \mathbb{X}} \left(\max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} h^i(x^*, \varepsilon, \mathbf{e}) \right) \right) \\ &= \min \left(- \max_{x^* \in \mathbb{X}} \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} h^i(x^*, \varepsilon, \mathbf{e}), \min_{x^* \in \mathbb{X}} \max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e} \in [\underline{\mathbf{e}}, \bar{\mathbf{e}}]} h^i(x^*, \varepsilon, \mathbf{e}) \right). \quad (6.29) \end{aligned}$$

A nouveau, si la valeur $\overline{\Delta y}$ obtenue est négative, le cahier des charges ne peut pas être satisfait.

Nous avons montré que les problèmes direct et inverse se formulent comme des problèmes "MinMax" continus. Dans la partie suivante, nous montrons qu'il est possible de résoudre ces problèmes de façon garantie, et nous les appliquons à un exemple d'illustration.

6.4 Application : étude d'un thermomètre

Le problème d'optimisation MinMax a fait l'objet de plusieurs travaux dans la littérature de l'analyse par intervalles [ZNE90], [Jau01]. L'algorithme d'optimisation utilisé dans la suite de ce document est l'algorithme de partitionnement MINIMAX, pour lequel nous n'avons pas apporté d'améliorations notables. Le lecteur désireux de connaître le principe détaillé de cet algorithme pourra se reporter à [JKDW01], page 126. MINIMAX calcule de façon garantie un encadrement de la solution c_n de la série d'optimisation suivante :

$$\begin{aligned} c_1(p_2, \dots, p_n) &= \min_{p_1 \in [p_1]} c_0(p_1, \dots, p_n) \\ c_2(p_3, \dots, p_n) &= \max_{p_2 \in [p_2]} c_1(p_2, \dots, p_n) \\ &\dots \\ c_i(p_{i+1}, \dots, p_n) &= (-1)^i \max_{p_i \in [p_i]} c_{i-1}(p_i, \dots, p_n) \\ &\dots \\ c_n &= (-1)^n \max_{p_n \in [p_n]} c_{n-1}(p_n), \end{aligned}$$

où $c_0(p_1, \dots, p_n)$ est une fonction connue pour laquelle on dispose d'une fonction d'inclusion convergente $[c_0](\cdot, \dots, \cdot)$. Si i est pair, $(-1)^i = 1$ et

$$c_i(p_{i+1}, \dots, p_n) = \max_{p_i \in [p_i]} c_{i-1}(p_i, \dots, p_n).$$

Sinon, la notation $c_i(p_{i+1}, \dots, p_n) = (-1)^i \max_{p_i \in [p_i]} c_{i-1}(p_i, \dots, p_n)$ se lit

$$c_i(p_{i+1}, \dots, p_n) = \min_{p_i \in [p_i]} c_{i-1}(p_i, \dots, p_n).$$

Le pessimisme dû à la fonction d'inclusion est contrôlé dans l'algorithme MINIMAX par le réglage d'un paramètre ε , qui fixe la taille maximale des pavés bissectés. Bien entendu, la diminution de ε engendre une augmentation du temps de calcul.

Nous abordons dans la suite l'étude d'un exemple académique : l'évaluation des performances d'une chaîne de mesure utilisant un thermomètre à vapeur saturante. Il s'agit d'un travail préliminaire de faisabilité de l'approche.

6.4.1 Contexte

Nous considérons l'exemple d'une chaîne de mesure indirecte de température utilisant un thermomètre à vapeur saturante, tel que celui décrit dans l'exemple 1. Ce thermomètre exploite la relation entre la pression de vapeur saturante P_s d'un corps (nous choisirons ici le cas de l'eau), et la température T de ce corps. Le relevé des couples (T, P_s) fournit une courbe expérimentale appelée courbe de vaporisation. Cette courbe est purement expérimentale, et il n'existe pas de modèle à base de connaissance capable de décrire cette courbe sur toute la plage de température que nous allons envisager. Par contre, plusieurs modèles ont été proposés selon la plage de température considérée. Sur la plage de température $T \in [90, 300]$ °C le plus souvent le modèle de Duperray est utilisé :

$$P_s(T) = \left(\frac{T}{100} \right)^4, \quad (6.30)$$

où P_s est exprimée en atm, et T est la température de l'eau exprimée en °C. L'écart relatif entre la sortie de ce modèle et la valeur lue sur l'abaque peut varier de 0.12% à 7.7%.

Le modèle de Rankine, défini par

$$\ln P_s(T) = 13.7 - \frac{5120}{T + 273.16}, \quad (6.31)$$

où à nouveau, P_s est exprimée en atm et T en °C, est utilisé sur la plage $[5, 140]$ °C. L'écart relatif varie alors de 0.39% à 4.1%.

Il existe donc une plage commune d'utilisation de ces deux modèles, qui vaut $T \in [90, 140]$. Afin de souligner la différence de comportement entre ces deux modèles, nous élargissons cette plage à la zone d'étude \mathbb{T}

$$\mathbb{T} = [90, 170].$$

Nous considérons dans la suite deux thermomètres à vapeur saturante \mathcal{T}_1 et \mathcal{T}_2 d'étendue de mesure \mathbb{T} .

La fonction de transfert de \mathcal{T}_1 est simulée à partir du modèle de Rankine

$$P_s = f_1(T, e_1) = \exp \left(13.7 - \frac{5120}{T + 273.16} \right) (1 + e_1), \quad (6.32)$$

où $e_1 \in [-0.077, 0.077]$, tandis que la fonction de transfert de \mathcal{T}_2 est simulée à partir du modèle

de Duperray

$$P_s = f_2(T, e_2) = \left(\frac{T}{100}\right)^4 (1 + e_2), \quad (6.33)$$

où $e_2 \in [-0.041, 0.041]$.

Dans une première étude, nous nous placerons dans un cadre idéalisé, où nous supposons que le comportement des deux thermomètres est parfaitement décrit par ces deux modèles en posant $e_1 = e_2 = 0$. Nous dirons donc que ces capteurs sont justes. Les fonctions de transfert correspondantes de ces deux thermomètres sur leur étendue de mesure sont représentées à la figure 6-10. Ces deux thermomètres semblent similaires, et l'erreur maximale entre les pressions de vapeur saturante est obtenue pour $T = 170^\circ\text{C}$ et vaut 0.2 atm .

Bien entendu, en réalité, ces fonctions de transfert ne sont pas connues. Il est alors nécessaire d'établir une courbe d'étalonnage g à partir de l'information disponible. Nous choisissons, pour ces deux thermomètres, la fonction g comme l'inverse exacte de la fonction de transfert $f_1(\cdot, 0)$:

$$\hat{T} = g(P_s) = \frac{5120}{13.7 - \ln P_s} - 273.16$$

de sorte que, dans le cadre idéalisé:

- $f_1(\cdot, 0) = g^{-1}(\cdot)$ pour le thermomètre \mathcal{T}_1 ,
- $f_2(\cdot, 0) \neq g^{-1}(\cdot)$: le choix de la courbe d'étalonnage a introduit une erreur de modèle pour le thermomètre \mathcal{T}_2 .

Nous allons étudier les caractéristiques de ces deux thermomètres en deux temps. Nous abordons premièrement le problème direct, en déterminant l'erreur de mesure de température commise avec ces deux thermomètres selon l'incertitude sur le relevé de la pression de vapeur saturante P_s . Nous étudierons ensuite le problème inverse, qui consiste à évaluer l'erreur de fidélité maximale $\overline{\Delta P_s}$ tolérable en sortie de ces deux thermomètres afin d'obtenir l'incertitude de mesure δT requise par le cahier des charges.

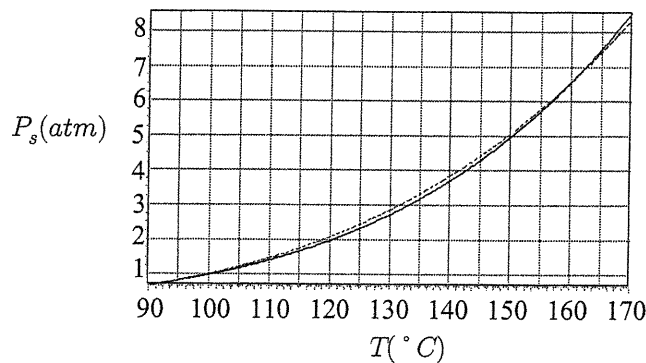


Figure 6-10: Modèles de Duperray (en trait plein) et de Rankine (en pointillés) de la loi $P_s(T)$ de vaporisation de l'eau : l'écart maximal entre les sorties de ces deux modèles vaut 0.2 atm et est obtenue pour $T = 170^\circ\text{C}$

6.4.2 Problème direct

Dans cette partie, nous allons caractériser les performances de la chaîne de mesure utilisant l'un ou l'autre de ces deux thermomètres.

6.4.2.1 Cadre idéalisé

Nous supposons dans cette partie que les comportements respectifs des deux thermomètres \mathcal{T}_1 et \mathcal{T}_2 sont exactement décrits par les fonctions de transfert $f_1(\cdot, 0)$ et $f_2(\cdot, 0)$. La seule différence entre les capteurs logiciels résultants réside donc uniquement dans l'erreur introduite lors du choix de la courbe d'étalonnage de \mathcal{T}_2 .

Distorsion due au choix de g Afin d'étudier cette différence, nous posons dans un premier temps $\Delta P_s = 0$. A l'aide de l'algorithme MINIMAX, nous calculons alors un encadrement garanti du défaut de justesse (voir (6.11)), de l'erreur de fidélité (voir (6.12)) et de l'incertitude de mesure (voir (6.13)) sur toute l'étendue de mesure \mathbb{T} .

Pour le thermomètre \mathcal{T}_1 , l'algorithme MINIMAX assure en 1.4s (pour $\varepsilon = 0.01$) que le défaut de justesse \bar{j} , l'erreur de fidélité et l'incertitude de mesure vérifient

$$\begin{aligned}\bar{j} &\in [-9.7, 9.7] \cdot 10^{-3}, \\ \Delta\bar{T} &\in [-4.8, 4.8] \cdot 10^{-3}, \\ \delta\bar{T} &\in [0, 9.7] \cdot 10^{-3}.\end{aligned}$$

En effet, pour ce thermomètre $g = f_1^{-1}$ et le capteur physique est juste. Pour toute valeur T^* du mesurande dans \mathbb{T} , \bar{j} , l'erreur de fidélité $\Delta\bar{T}$ et l'incertitude de mesure $\delta\bar{T}$ sont donc nulles.

Dans le cas du thermomètre \mathcal{T}_2 , MINIMAX fournit en 30ms (pour une profondeur de bissection de l'algorithme $\varepsilon = 0.01$) les encadrements garantis du défaut de justesse (6.11), de l'erreur de fidélité (6.12) et de l'incertitude de mesure δT (6.13) :

$$\begin{aligned}\bar{j} &\in [1.59, 1.61], \\ \Delta\bar{T} &\in [-4.10^{-3}, 4.10^{-3}], \\ \delta\bar{T} &\in [1.59, 1.61].\end{aligned}$$

Ceci permet en particulier d'établir qu'il existe un défaut de justesse \bar{j} et une incertitude de mesure strictement positifs. Pour étudier cela, nous avons tracé l'évolution de la valeur absolue du défaut de justesse $j(T^*)$ pour un ensemble discret de valeurs de T^* dans l'étendue de mesure à l'aide de l'algorithme MINIMAX, pour le problème décrit par l'équation (6.8). L'erreur introduite dans le capteur logiciel utilisant le thermomètre \mathcal{T}_2 introduit un défaut de justesse $j(T)$ non-nul. En $T = 125^\circ C$ (voir la figure 6-11), la valeur absolue de $j(T)$ est maximale et atteint 1.6, qui appartient bien à l'intervalle $[1.59, 1.61]$ qui a été fourni par l'algorithme dans le cas où T^* est inconnu. Remarquons que ces valeurs ne correspondent pas à la température $T = 170^\circ C$ pour laquelle l'écart entre les deux modèles est maximale (voir la figure 6-10). Cette différence est due à la distorsion introduite par la courbure de la fonction g . En $T = 95^\circ C$ et $T = 160^\circ C$, le défaut de justesse est nul car les deux modèles f_2 et g^{-1} coïncident parfaitement.

Par ailleurs, puisque ΔP_s est nul, l'erreur de fidélité $\Delta\bar{T}$ calculée à l'aide de (6.9) est nulle. L'incertitude de mesure $\delta\bar{T}$, calculée par (6.10) est alors égale à \bar{j} . La Figure 6-12 illustre à nouveau l'encadrement $[\delta T(T^*)]$ très fin fourni par MINIMAX à l'aide de l'équation (6.13) pour un ensemble discret de valeurs T^* . La valeur maximale est $1.6^\circ C$ pour $T = 125^\circ C$.

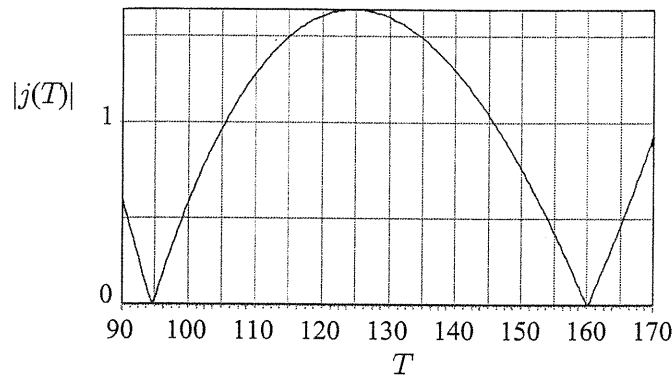


Figure 6-11: Valeur absolue du défaut de justesse $j(T)$ du capteur utilisant le thermomètre \mathcal{T}_2 en l'absence d'incertitude de mesure δP_s

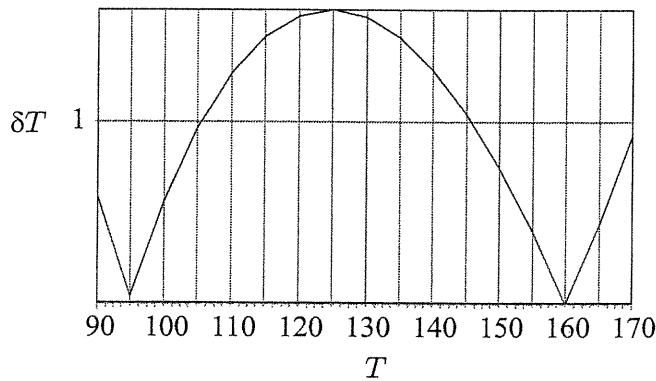


Figure 6-12: Incertitude de mesure δT associée au thermomètre \mathcal{T}_2 pour $\delta P_s = 0$

Jusqu'à présent, nous avons négligé l'erreur de fidélité ΔP_s en sortie des capteurs physiques. Dans le paragraphe suivant nous étudions l'influence de la valeur de δP_s sur les résultats obtenus en sortie des deux capteurs logiciels.

Influence de δP_s Supposons maintenant que l'erreur de fidélité ΔP_s n'est plus nulle. Les évolutions de \bar{j} , $\delta\bar{T}$ et $\Delta\bar{T}$ en fonction de la valeur de ΔP_s pour le thermomètre \mathcal{T}_1 sont présentées à la figure 6-13, pour une valeur de ε choisie égale à 0.01. Le diamètre des intervalles obtenus par MINIMAX est maintenant suffisamment élevé pour distinguer les deux courbes représentant l'évolution des deux bornes des intervalles.

Puisque $f = g^{-1}$, seule ΔP_s introduit un défaut de justesse et une erreur de fidélité. Pour $\Delta P_s = 0$, \bar{j} et $\delta\bar{T}$ sont nuls. Lorsque ΔP_s augmente, la valeur \bar{j} du défaut de justesse croît, et il en est de même pour l'erreur de fidélité et l'incertitude de mesure (voir la figure 6-10).

La même étude a été réalisée pour le thermomètre \mathcal{T}_2 (voir la figure 6-14). L'erreur due au fait que $f_2 \neq g^{-1}$ se traduit par un comportement différent de \bar{j} et de $\delta\bar{T}$ pour les faibles valeurs de ΔP_s . Lorsque ΔP_s augmente, cette erreur devient négligeable face à l'incertitude de mesure, et le comportement asymptotique de \bar{j} , $\Delta\bar{T}$ et $\delta\bar{T}$ s'apparente alors à celui obtenu pour le thermomètre \mathcal{T}_1 .

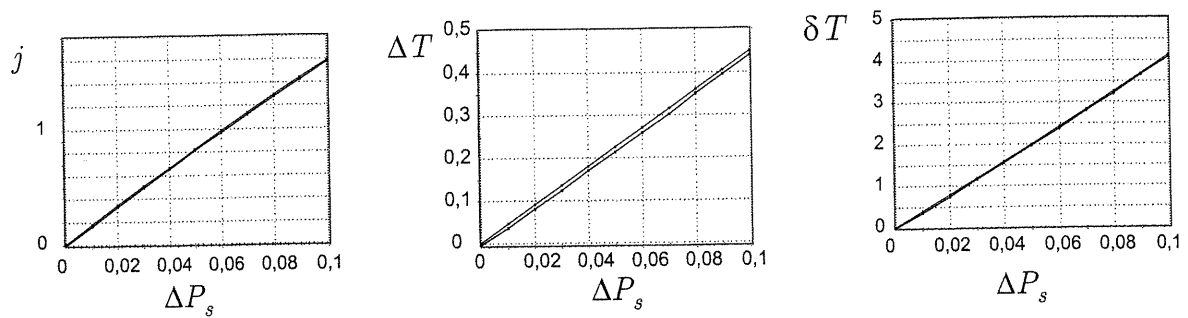


Figure 6-13: De gauche à droite : étude du défaut de justesse, de l'erreur de fidélité et de l'incertitude de mesure en fonction de ΔP_s lorsque $f = g^{-1}$

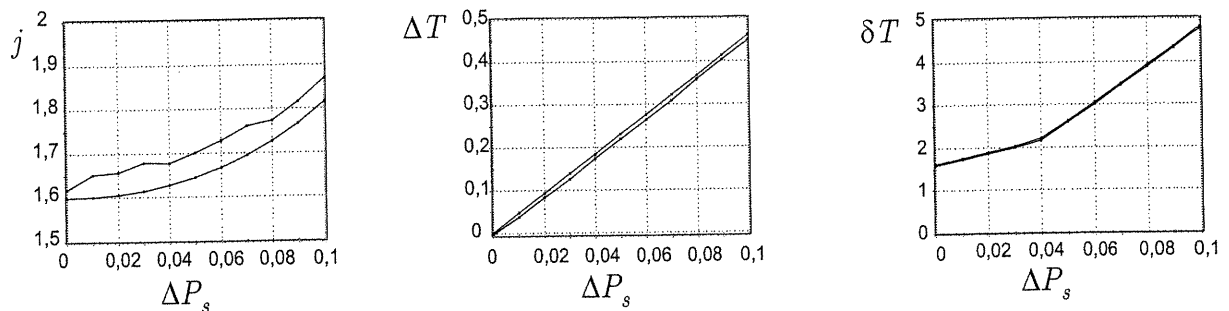


Figure 6-14: De gauche à droite : évolution de \bar{j} , $\bar{\delta T}$ et $\bar{\Delta T}$ en fonction de la valeur de ΔP_s pour le thermomètre T_2

6.4.2.2 Cas $e \neq 0$

Sortons maintenant du cadre idéalisé. Le comportement des deux capteurs suit la loi de vaporisation, que l'on peut décrire à l'aide des fonctions de transfert f_1 et f_2 à condition de tenir compte de l'erreur de modèle commise. Cette erreur est une grandeur incertaine pour laquelle nous disposons de bornes expérimentales (voir (6.32) et (6.33)).

Restreignons-nous ici à l'étude de l'erreur de fidélité. Les évolutions de $\bar{\Delta T}$ en fonction de ΔP_s sont représentées sur la figure 6-15 pour les deux thermomètres. Pour une valeur de ΔP_s donnée, l'incertitude de mesure est plus faible si on utilise le thermomètre T_1 , puisque dans ce cas $f_1 = g^{-1}$.

6.4.3 Problème inverse

Il s'agit désormais de déterminer l'incertitude de mesure $\overline{\Delta P_s}$ maximale tolérable pour pouvoir espérer mesurer T avec une incertitude de mesure δT fixée. Nous allons étudier, pour chaque thermomètre, l'évolution de l'incertitude de mesure $\overline{\Delta P_s}$ requise en fonction de la valeur δT imposée dans le cahier des charges. Bien entendu, cette valeur doit être positive. Aussi nous repérerons à chaque fois la valeur minimale de δT pour laquelle il existe une valeur $\overline{\Delta P_s}$ positive. Nous noterons δT^o cette grandeur, qui correspond donc à l'incertitude de température minimale atteignable. Nous concluons donc que :

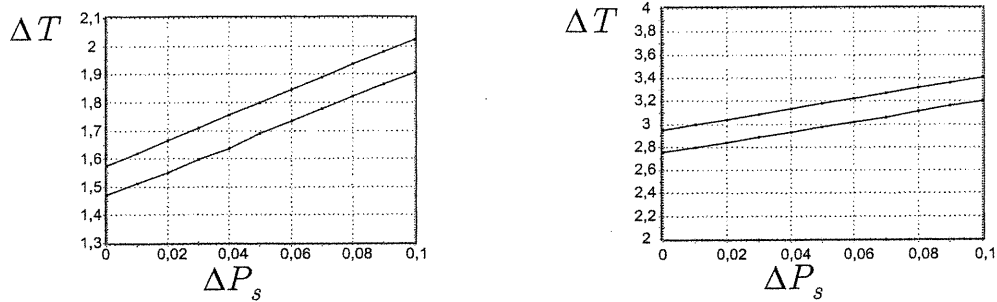


Figure 6-15: Evolution de δT en fonction de δP_s en tenant compte de l'erreur de modèle, pour les thermomètres \mathcal{T}_1 (à gauche) et \mathcal{T}_2 (à droite)

- si l'incertitude de mesure δT imposée est inférieure à δT° , alors le thermomètre utilisé ne peut satisfaire le cahier des charges,
- dans le cas contraire, le thermomètre utilisé satisfait cette contrainte *a priori*.

6.4.3.1 Thermomètre \mathcal{T}_1

Considérons le thermomètre \mathcal{T}_1 , et supposons que $e_1 = 0$. Dans ce cas, l'incertitude de mesure ΔP_s est égale à l'erreur de fidélité δP_s du thermomètre \mathcal{T}_1 . Le gabarit obtenu à l'aide de l'algorithme MINMAX et de l'équation (6.23), pour différentes valeurs de δT est présenté à la figure 6-16. Il permet de vérifier que ΔP_s augmente avec δT : plus l'incertitude de mesure tolérée sur T est importante, plus la contrainte de performance imposée au thermomètre peut être relâchée. Une simple lecture permet de déterminer la condition pour laquelle il est possible de satisfaire le cahier des charges. Par exemple, pour obtenir $\delta T = 0.5$, il est impératif que $\overline{\Delta P_s}$ soit inférieure à 0.0123. Enfin, la valeur minimale δT° de l'incertitude de mesure δT qui peut être satisfaite appartient à l'intervalle $[0, 0.0014]^\circ C$. Elle est obtenue en relevant sur la courbe les valeurs $\underline{\delta T}^\circ$ et $\overline{\delta T}^\circ$ telles que

$$\begin{aligned} \overline{\Delta P_s}(\overline{\delta T}^\circ) &= 0, \\ \underline{\Delta P_s}(\underline{\delta T}^\circ) &= 0. \end{aligned}$$

Supposons maintenant qu'il existe une erreur $e_1 \in [-0.041, 0.041]$. Dans ce cas, les valeurs de δP_s et de ΔP_s ne sont plus égales. Nous déterminons l'erreur de fidélité $\overline{\Delta P_s}$ de \mathcal{T}_1 à l'aide de l'équation (6.27), pour différentes valeurs de δT . Le gabarit obtenu est représenté à la figure 6-17. Il n'est désormais plus possible d'assurer l'obtention d'une mesure de T avec une incertitude de mesure inférieure à $\underline{\delta T}^\circ$, où

$$\delta T^\circ \in [1.4, 1.6]^\circ C,$$

6.4.3.2 Thermomètre \mathcal{T}_2

La même étude a pu être réalisée pour le thermomètre \mathcal{T}_2 . Lorsque $e_2 = 0$, à nouveau $\delta P_s = \Delta P_s$. L'algorithme MINIMAX nous permet de tracer l'évolution de δP_s en fonction de δT , représentée

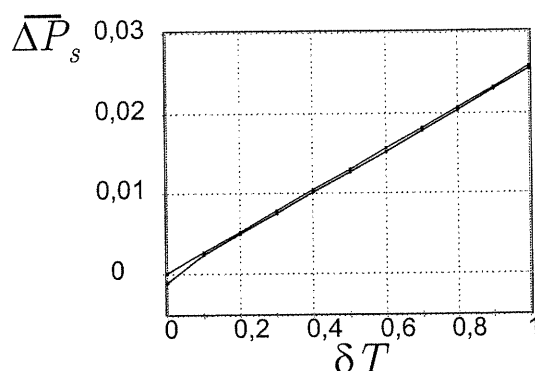


Figure 6-16: Evolution de δP_s du thermomètre \mathcal{T}_1 en fonction de l'incertitude δT requise

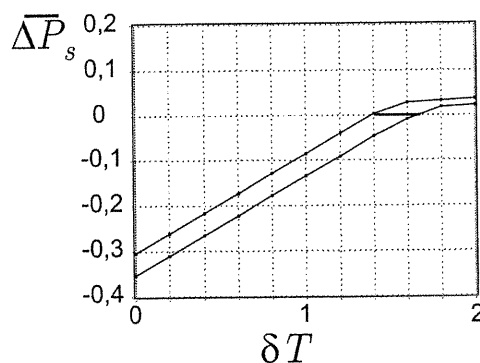


Figure 6-17: Evolution de ΔP_s du thermomètre \mathcal{T}_1 en fonction de la valeur de δT imposée pour $e_1 = 0.077$

sur la figure 6-18. Comme dans ce cas, le seuil δT° atteint la valeur

$$\delta T^\circ \in [1.55, 1.67]^\circ C,$$

il n'est pas possible d'obtenir avec ce capteur une incertitude de mesure inférieure à $1.55^\circ C$.

Supposons alors l'existence d'une erreur e de 7% sur le modèle de Duperray. L'allure de l'évolution de $\overline{\Delta P}_s$ en fonction de la valeur de δT imposée est présentée à la figure 6-19. On obtient désormais

$$\delta T^\circ \in [3.6, 4.2]^\circ C.$$

Nous avons considéré jusqu'à présent le cas d'un capteur unique. Le cas plus réaliste où l'on dispose de plusieurs données requiert une écriture relativement lourde, aussi est-il présenté dans l'annexe. Il s'agit de premiers pas vers l'élaboration d'un critère d'optimisation d'essais, dans un souci de planification d'expériences.

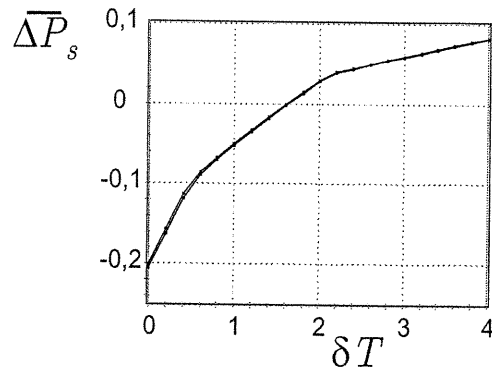


Figure 6-18: Evolution de $\delta P_s = \Delta P_s$ en fonction de la valeur de δT imposée

6.5 Conclusions

Dans ce chapitre, nous avons envisagé l'étude des performances du capteur physique utilisé dans la chaîne de mesure indirecte. Nous avons distingué deux problèmes. Dans le problème direct, nous avons caractérisé les performances escomptées de cette chaîne en fonction des caractéristiques du capteur physique. Nous avons en particulier étudié l'incertitude de mesure et le défaut de justesse de la chaîne. Dans le problème inverse, nous supposons que le cahier des charges impose l'incertitude de mesure en sortie de la chaîne. L'objectif est alors d'étudier si le capteur physique nous permettra d'obtenir cette incertitude. Nous avons alors montré qu'il était possible de déterminer la relation liant l'incertitude de mesure en sortie de la chaîne à l'incertitude de mesure du capteur physique.

Comme ces études sont effectuées a priori, aucune donnée n'est disponible. Dans ce contexte où nous souhaitons obtenir des conclusions valides quelle que puisse être la valeur prise par le vecteur de paramètres, nous avons posé ces deux problèmes sous la forme de critères "MinMax". A partir de l'algorithme MINIMAX garanti, qui s'appuie sur l'analyse par intervalles, nous avons étudié ces deux problèmes pour un thermomètre à pression de vapeur saturante. L'extension aux cas de données multiples a également été envisagée. La formulation du problème en présence d'un unique capteur en fonctionnement discret est une des approches possibles pour définir un

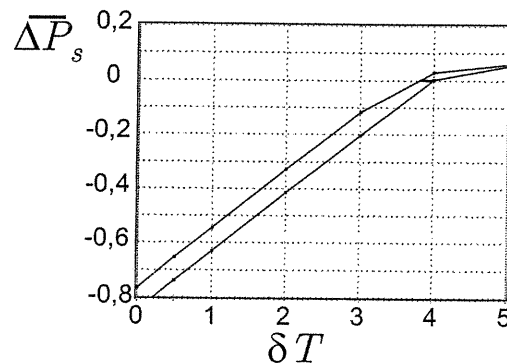


Figure 6-19: Evolution de ΔP_s en fonction de la valeur de δT imposée en présence d'une erreur e_2

critère de choix du protocole expérimental, selon les caractéristiques du capteur physique utilisé.

Chapitre 7

Conclusions et perspectives

Initialement développée pour le contrôle des erreurs d'arrondi dans les calculateurs numériques, l'analyse par intervalles tient aujourd'hui un rôle important dans de nombreuses applications (voir [KK96] et le site web [WEBf]). Outre la démonstration assistée par ordinateur de conjectures [FS96] qui accroît régulièrement sa renommée, cette approche numérique permet de résoudre de façon garantie de nombreux problèmes non-linéaires fréquents dans les domaines des sciences physiques et des sciences de l'ingénieur. Ainsi, l'identification de paramètres de modèles non-linéaires dans un contexte à erreurs bornées a pris naturellement une place prépondérante.

Mais l'identification proprement dite ne constitue qu'une étape dans le processus complexe de construction de modèles ou d'estimation de grandeurs physiques. Nous avons alors souhaité montrer, au travers de cette thèse, comment les approches ensemblistes garanties peuvent être utilisées tout au long de cette démarche. Pour cela, nous avons été amenés, dans un premier temps, à montrer que de nombreux problèmes rencontrés au cours de la construction de modèles peuvent se formuler comme des problèmes ensemblistes. Dans un deuxième temps, nous avons développé des algorithmes spécifiques adaptés à la résolution de ces problèmes.

7.1 Les problèmes ensemblistes dans la construction de modèles

Deux types d'étapes jalonnent la construction d'un modèle.

Les *étapes a priori* sont antérieures au recueil des données. Y figurent, entre autres, la caractérisation de la structure de modèle $\mathcal{M}(\cdot)$ (chapitre 5) et la sélection d'un capteur (chapitre 6). Dans ces étapes, il s'agit de préparer au mieux l'estimation, alors qu'aucune valeur du vecteur de paramètres \mathbf{x} à estimer n'est encore disponible. Afin de déterminer si la structure de modèle ou le capteur choisi satisferont les attentes de l'expérimentateur et les contraintes du cahier des charges une fois les données recueillies, ils sont soumis à des tests. Il sera alors possible de prendre une décision si ces tests s'avèrent concluants pour toutes les valeurs de \mathbf{x} dans un espace de recherche – éventuellement égal à \mathbb{R}^m – fourni par l'expérimentateur.

Dans les étapes dites *a posteriori*, les données expérimentales sont disponibles. Il s'agit alors d'estimer \mathbf{x} à partir de ces données, et d'évaluer l'incertitude associée à l'estimation (chapitre 4). Pour cela, deux erreurs doivent être prises en compte : l'erreur de mesure, qui, souvent décrite par une variable aléatoire, peut également être modélisée par un intervalle, à condition de connaître sa plage de variation, et l'erreur de modèle, naturellement beaucoup plus difficile à quantifier. Lorsqu'on dispose de bornes sur ces deux erreurs, l'estimée de \mathbf{x} regroupe l'ensemble des valeurs compatibles avec la structure de modèle, les données recueillies, et ces bornes. C'est le contexte à erreurs bornées, auquel nous nous sommes restreints dans ce document.

Les problèmes posés dans ces étapes sont de nature différente, mais font tous intervenir des

ensembles : ce sont donc des problèmes ensemblistes, que les approches ensemblistes que nous avons décrites visent à résoudre de manière garantie.

7.1.1 Avant le recueil des données

Dans les étapes *a priori*, il s'agit d'effectuer un test sur un ensemble de valeurs possibles dans un espace de recherche.

Dans la caractérisation d'une structure de modèle, on cherche à déterminer le sous-ensemble des valeurs de \mathbf{x} qui ne satisfont pas une propriété donnée. Si ce sous-ensemble est vide, alors le test est concluant sur tout l'espace de recherche. L'approche classique consiste à décrire ce sous-ensemble à l'aide des méthodes de calcul formel. Mais cette description, qui peut parfois être trop lourde pour être utilisable en pratique, peut également se révéler impossible si ce sous-ensemble n'est pas de mesure nulle. Au contraire, l'approche numérique garantie que nous avons proposée dans ce document fournit une description *garantie* de ce sous-ensemble. Nous avons pour cela formulé de nouveaux tests numériques et garantis d'identifiabilité et de discernabilité (chapitre 5).

La sélection d'un capteur est une étape d'évaluation *a priori* du dispositif expérimental mis en œuvre, qui peut aider l'expérimentateur à estimer, avant toute saisie de mesures, la qualité du résultat qu'il peut escompter. Peu d'approches systématiques sont disponibles. L'approche probabiliste consiste à soumettre le capteur à des séries de tests correspondant à différentes valeurs de \mathbf{x} . Cette description nécessairement non exhaustive de l'espace de recherche ne permet pas non plus d'assurer la garantie du résultat une fois l'expérience réalisée. Nous avons proposé là aussi d'utiliser les outils numériques garantis pour évaluer certaines performances de ce capteur, en supposant disponible un modèle de sa fonction de transfert. Nous avons montré comment il était possible de caractériser *a priori* les performances du capteur physique utilisé selon l'incertitude de mesure associée à l'estimée du vecteur de paramètres. Après avoir étudié l'influence des caractéristiques du capteur sur le résultat de l'estimation, nous avons proposé un test simple afin de déterminer si le capteur sélectionné peut permettre d'atteindre l'incertitude de mesure requise par le cahier des charges.

7.1.2 Après le recueil des données

Dans les étapes *a posteriori*, il s'agit de décrire l'ensemble de toutes les valeurs compatibles avec l'expérience. Lorsque le vecteur de paramètres regroupe toutes les grandeurs incertaines présentes dans la chaîne de mesure, l'estimation peut s'écrire sous la forme d'un problème d'inversion ensembliste. Dans le cas contraire, l'estimation revient à projeter l'ensemble de toutes les grandeurs incertaines compatibles avec l'expérience sur le sous-espace du vecteur de paramètres. Nous avons proposé pour cela un algorithme de projection (chapitre 3), et deux applications concrètes ont été envisagées : l'estimation de paramètres cinétiques en électrochimie et des paramètres thermo-physiques d'un échantillon de PVC à partir d'un dispositif de mesure nouveau (chapitre 4).

7.2 Nouveaux algorithmes ensemblistes

Sous le terme d'approches ensemblistes, nous avons regroupé les méthodes qui permettent de décrire un ensemble. Parmi celles-ci, l'approche numérique garantie, que nous avons suivie tout au long de ce document, possède deux avantages notables. Elle permet d'une part de coder et de manipuler simplement un ensemble, et d'autre part de tirer des conclusions fiables, en calculant des approximations intérieure et extérieure de cet ensemble exactement représentables par un

calculateur numérique. Comme ces approximations sont obtenues par un partitionnement de l'espace de recherche en pavés, ceci assure que la description obtenue est exhaustive, et permet par ailleurs l'utilisation des règles de calcul garanti de l'analyse par intervalles. A cause de ce partitionnement, la complexité des algorithmes ensemblistes utilisés est exponentielle en le nombre des paramètres. Comme les contracteurs permettent de réduire la taille d'un pavé sans le bissecter, ils peuvent être mis à profit dans ces algorithmes pour obtenir des encadrements garantis d'ensembles de dimension plus élevée que ceux étudiés jusqu'à alors. En particulier, nous avons souligné l'intérêt d'un contracteur utilisant la propagation de contraintes pour réduire des pavés de taille importante (chapitre 3), et éliminer rapidement des pavés qui ne sont pas admissibles.

7.3 Perspectives

Seules quelques étapes de la construction de modèles ont été étudiées dans ce document. En particulier, nous avons omis l'étape de planification d'expériences, qui constitue un axe de recherche particulièrement intéressant à envisager. Par ailleurs, nous n'avons considéré ici que l'estimation de paramètres physiques, dans des objectifs de mesure. Les algorithmes développés peuvent être aménagés de sorte à envisager l'estimation de paramètres dans des objectifs de commande.

7.3.1 Planification d'expériences

Nous avons abordé au chapitre 6 l'étude de la sélection d'un capteur. Or cette étape est fortement liée à la planification d'expériences, puisque la détermination d'un plan d'expériences optimal pourra entraîner la modification ou le déplacement d'un capteur. Aussi avons-nous ébauché dans l'annexe le traitement de plusieurs données, selon qu'elles sont obtenues à partir d'un même capteur ou à partir de plusieurs capteurs. Il est alors envisagé de montrer que la détermination du protocole expérimental, qui, associé à un capteur, permet d'obtenir la meilleure incertitude d'estimation revient à résoudre un problème MinMax. Cette étude peut être mise en parallèle avec les travaux initiés dans [PW96], qui ont montré la possibilité d'étudier la planification d'expériences dans un contexte à erreurs bornées.

7.3.2 Commande

L'algorithme de projection que nous avons proposé permet de caractériser un ensemble solution d'un système de contraintes liées par un quantificateur existentiel. Il se prête donc particulièrement bien à l'estimation de grandeurs incertaines en présence de facteurs expérimentaux inconnus. Nous avons montré au chapitre 3 que ce même algorithme peut être exploité afin de manipuler des contraintes utilisant un quantificateur universel. Il peut donc être utilisé dans des objectifs de commande. Il est ainsi possible de déterminer l'ensemble des valeurs d'un correcteur PID qui stabilisent un système linéaire non-linéairement paramétré, les paramètres étant incertains mais bornés. Cette idée a été illustrée dans l'article [JBW02] récemment accepté.

Chapitre 8

Annexe

8.1 Table des symboles

8.1.1 Notations mathématiques ponctuelles

x	p.10	scalaire ponctuel
\mathbf{x}	p.7	vecteur colonne ponctuel
\mathbf{x}^T	p.43	vecteur transposé de \mathbf{x}
\mathbf{A}	p.47	matrice ponctuelle
z_r	p.116	partie réelle du complexe ponctuel z
z_i	p.116	partie imaginaire du complexe ponctuel z
s	p.110	variable de Laplace
ω	p.115	pulsation

8.1.2 Notations ensemblistes

\mathbb{X}	p.10	ensemble
$\partial\mathbb{X}$	p.85	frontière de \mathbb{X}
$[\mathbb{X}]$	p.21	pavé enveloppe de \mathbb{X}
$\text{Vol}(\mathbb{X})$	p.12	volume de l'ensemble \mathbb{X}
\emptyset	p.12	ensemble vide
\mathbb{R}	p.10	corps des réels
\mathbb{IR}	p.22	ensemble des intervalles
\mathbb{B}	p.31	ensemble des booléens
\mathbb{IB}	p.31	ensemble des intervalles booléens
\mathcal{R}	p.13	ensemble des réels exactement représentables en machine
$\downarrow x$	p.184	plus grand nombre flottant inférieur au réel x
$\uparrow x$	p.184	plus petit nombre flottant supérieur au réel x

8.1.3 Grandeurs relatives à l'estimation

\hat{x}	p.7	estimée de la grandeur incertaine x
x^*	p.100	vraie valeur de la grandeur incertaine x
j	p.159	défaut de justesse
Δx	p.159	erreur de fidélité associée la mesure de x
δx	p.158	incertitude de mesure de x
\mathbb{X}	p.8	domaine de recherche <i>a priori</i> de x (étendue de mesure)

8.1.4 Paramètres relatifs aux algorithmes

es	p.27	arguments entrée-sortie de l'algorithme
e	p.27	argument d'entrée
s	p.54	sortie de l'algorithme
t_c	p.33	temps de calcul (seconde: s)
n_b	p.35	nombre de bisections effectuées
ε	p.33	profondeur de bisection (taille maximale des pavés bisectés)
δ	p.34	précision de l'encadrement
$\underline{\mathbb{X}}$	p.12	approximation intérieure de \mathbb{X}
$\overline{\mathbb{X}}$	p.12	approximation extérieure de \mathbb{X}
\mathcal{L}	p.36	liste

8.2 Mise en œuvre informatique

Les algorithmes présentés dans le corps du document ont été implémentés en C++ à l'aide de la bibliothèque développée par M. Kieffer [Kie99], inspirée des langages XSC [HHKR95a]. Bien entendu, de nombreux autres logiciels de calcul par intervalles (citons par exemple Profil/Bias [Knü94] et Intlab [Rum01]) sont disponibles. La page Web [WEBf] présente une liste des nombreux logiciels destinés aux intervalles qui ont été développés depuis les années 1960. D'autres logiciels sont dédiés à la programmation par contraintes à l'aide de langages déclaratifs (PrologIV [Webd]). Malgré sa moins grande maniabilité, nous avons utilisé uniquement une classe C++ pour la décomposition des contraintes dans cette thèse. Soulignons cependant l'arrivée récente du logiciel RealPaver, gratuit et disponible à l'adresse [Webk], qui utilise la propagation de contraintes et la box-consistance.

Pour une plus grande lisibilité, les problèmes relatifs à la mise en œuvre de l'analyse par intervalles ont été volontairement occultés dans le corps du document. De nombreuses introductions à la mise en œuvre du calcul par intervalles sont disponibles ([Wal98], [KM81] et [Kul96]). Nous avons donc choisi dans cette partie de rappeler succinctement au lecteur quelques points importants de cette implémentation, en particulier pour la manipulation des contraintes.

8.2.1 Arrondi sur les réels

Comme nous l'avons montré dans l'introduction, un nombre x en virgule flottante est représenté à l'aide d'une base, d'un exposant et d'une mantisse finis. A cause de la précision limitée de cette représentation, seul le sous-ensemble \mathcal{R} du corps des réels peut être exactement représenté, et tout réel qui n'appartient pas à \mathcal{R} est inévitablement approximé. Afin de contrôler cette approximation de manière à perdre le moins d'information possible, la norme IEEE 754 introduit différentes *fonctions d'arrondi* : arrondi au plus près, arrondi vers le haut (\uparrow) et arrondi vers le bas (\downarrow) (voir la figure 8-1). Bien entendu, ces trois fonctions laissent l'ensemble \mathcal{R} invariant.

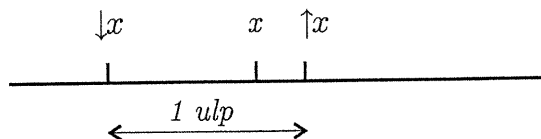


Figure 8-1: Arrondis vers le haut et vers le bas du réel x : la distance séparant x^+ de x^- vaut 1 *ulp* et dépend de l'exposant de x

Remarque 8.2.1 Soit $\bar{\aleph}$ le plus grand nombre représentable en machine dans le format sélectionné. Par exemple, $\bar{\aleph} = 1.80 \times 10^{308}$ pour une mantisse de longueur $p = 53$ et un exposant inférieur ou égal à 1024. Alors

$$\uparrow \bar{\aleph} = +\infty.$$

Puisque trois directions d'arrondi sont possibles pour la représentation d'un réel x , le résultat d'une opération simple faisant intervenir x peut donc prendre différentes valeurs suivant la fonction d'arrondi utilisée. L'arithmétique sur \mathcal{R} est alors définie pour chaque opération et chaque fonction d'arrondi. Pour assurer le maximum de précision de la représentation du résultat d'une opération \circ en virgule flottante et précision finie, celle-ci doit correspondre à la valeur arrondie de la même opération réalisée sur les réels.

Remarque 8.2.2 Le type d'arrondi est contrôlé par le processeur conformément à la norme IEEE 754. En C++, il suffit d'insérer une simple instruction : par exemple, l'arrondi vers le haut se fait par la commande :

```
_control87(RC_UP, MCW_RC);
```

Seul le type d'arrondi est contrôlé : il est en effet difficile d'estimer l'erreur d'arrondi associée, qui dépend de la valeur du réel que l'on cherche à représenter. Pour pouvoir assurer la garantie du résultat, il n'est donc pas possible de se satisfaire de ce type de représentation. Dans tous les algorithmes présentés dans ce document, les nombres réels x seront représentés à l'aide d'intervalles du type $\lfloor x, \uparrow x$ si $x \notin \mathcal{R}$ et d'intervalle dégénéré $[x, x]$ si $x \in \mathcal{R}$, en respectant les règles exposées dans le paragraphe suivant. Les deux types d'arrondi \downarrow et \uparrow sont donc les deux modes que nous utiliserons par la suite.

8.2.2 Arrondi sur les intervalles

Comme nous avons défini les intervalles à partir des réels, il s'agit maintenant de représenter ces intervalles.

Soit \mathcal{IR} l'ensemble des intervalles représentables en machine, c'est-à-dire

$$\mathcal{IR} = \{[x] \in \mathbb{IR} \mid \underline{x} \in \mathcal{R}, \bar{x} \in \mathcal{R}\}.$$

Rappelons que nous ne considérons que des intervalles fermés, mais d'autres types d'intervalles peuvent être envisagés [AH83].

L'intervalle vide est particulièrement nécessaire dans les contracteurs. En effet, il apparaît lors de l'intersection de deux ensembles disjoints, or lors de la contraction d'un intervalle inconsistant. De même, les notions d'infini nous seront très utiles. Nous définissons alors l'ensemble $\overline{\mathcal{IR}}$ comme

$$\overline{\mathcal{IR}} = \mathcal{IR} \cup \{[x, +\infty], x \in \mathcal{R}\} \cup \{[-\infty, x], x \in \mathcal{R}\} \cup [-\infty, +\infty] \cup [\text{NaN}, \text{NaN}],$$

où NaN représente l'élément Not a Number de la norme IEEE 754¹. Il nous sera très utile pour coder l'ensemble vide \emptyset , puisqu'il est absorbant pour toute opération arithmétique effectuée sur un ordinateur, de la même manière que \emptyset pour les opérations arithmétiques ensemblistes.

La fonction d'arrondi \bigcirc pour les intervalles vérifie les propriétés :

$$\begin{aligned} [x] \subset [y] &\Rightarrow \bigcirc [x] \subset \bigcirc [y], \\ \bigcirc [x] = [x] &\iff [x] \in \overline{\mathcal{IR}}. \end{aligned}$$

¹Il existe différents NaN [JKDW01]. Pour plus de clarté, nous ne ferons cependant pas la distinction ici.

La représentation de l'intervalle $[z] \in \overline{\mathcal{IR}}$ résultant d'une opération arithmétique ou d'une fonction sera obtenue avec une précision maximale si $[z]$ est égal à l'arrondi de la même opération effectuée sur les intervalles en précision infinie. La précision maximale est atteinte lorsque le diamètre de $[z]$ est de 1 ulp.

8.2.3 Garantie des calculs

La fonction d'arrondi utilisée, que nous notons \uparrow est définie de la façon suivante

$$\forall [x] \in \mathbb{IR}, \uparrow [x] = [\downarrow \underline{x}, \uparrow \bar{x}] \in \overline{\mathcal{IR}}.$$

8.2.3.1 Opérations arithmétiques

Pour toute opération arithmétique \circ , on a donc

$$\begin{aligned} \uparrow ([x] \circ [y]) &\in \overline{\mathcal{IR}}, \\ [x] \circ [y] &\subset \uparrow ([x] \circ [y]). \end{aligned}$$

De plus, la norme IEEE 754 garantit une erreur de calcul des bornes des intervalles inférieures à 1 ulp.

8.2.3.2 Fonctions

Les fonctions élémentaires sont naturellement étendues selon le principe précédent. Ainsi

$$\exp([x]) = [\downarrow \exp(\underline{x}), \uparrow \exp(\bar{x})].$$

Différents problèmes apparaissent lors de la mise en œuvre des autres fonctions.

Domaine de définition Ainsi, l'évaluation de l'image $[y] = f([x])$ d'un intervalle $[x]$ qui n'est pas inclus dans le domaine \mathbb{D} de définition de f fait l'objet d'un litige. Deux approches peuvent être envisagées. La première considère que cette image est vide, et renvoie $[\text{NaN}, \text{NaN}]$. La seconde approche décompose l'intervalle $[x]$ sous la forme $[x] = ([x] \cap \mathbb{D}) \cup ([x] \cap \neg \mathbb{D})$, de sorte que $[y] = f([x] \cap \mathbb{D})$.

Exemple 8.2.1 Dans la première approche, $\ln([-1, 1]) = [\text{NaN}, \text{NaN}]$, tandis que dans la seconde $\ln([-1, 1]) = [-\infty, 0]$.

Nous avons suivie cette seconde approche afin de ne pas éliminer sans raison apparente des variables dans les graphes des contraintes. Considérons en effet la contrainte

$$\mathcal{H} : (z = x \ln(x + y), x \in [-0.5, 0.5], y \in [-0.5, 0.5], z \in [-100, 100]).$$

La décomposition de \mathcal{H} en contraintes primitives fournit le systèmes de contraintes

$$\begin{aligned} h_1 : a_1 &= x + y, \\ h_2 : a_2 &= \ln(a_1), \\ h_3 : a_3 &= x.a_2, \end{aligned}$$

Dans l'algorithme FALL, la propagation de la contrainte h_2 fournit $[a_2] = \ln([-1, 1])$. Si la première approche est suivie sans aucune précaution, alors $[a_2] = [\text{NaN}, \text{NaN}]$ et le pavé $[x] \times [y] \times [z]$ est éliminé, bien qu'il ne soit pas inconsistant. La seconde approche, en fournissant $[a_2] = [-\infty, 0]$, évite cet écueil.

Fonctions transcendantes Les fonctions transcendantes (\cos , \sin , \tan) forment un cas spécifique, pour lesquelles quelques précautions doivent être prises. En effet, ces fonctions décomposables en séries entières infinies sont souvent codées à l'aide de séries tronquées, ce qui limite la qualité de leur mise en œuvre. Même en utilisant la fonction arrondi présentée ci-dessus, il n'est pas possible d'assurer que l'intervalle $[z]$ contenu contient le résultat exact. L'approche suivie consiste alors à élargir l'intervalle $[z]$ d'un nombre de ulps qui dépend de la qualité de la mise en œuvre.

8.2.3.3 Consistances faibles

L'extension des solveurs de CSP discrets pour la caractérisation d'ensembles continus \mathbb{X}_i suppose de tester tous les points des domaines \mathbb{X}_i . En raison de la précision finie des calculateurs, la consistance d'arc ne peut donc pas être directement mise en œuvre. Pour rendre possible ce calcul en précision finie, les informaticiens ont proposé de définir d'autres types de consistances locales, plus faibles. La première d'entre elles s'appuie sur la remarque suivante.

Remarque 8.2.3 Si $[x_i]$ est arc-consistant avec le CSP \mathcal{H} , alors les valeurs \underline{x}_i et \bar{x}_i sont consistantes avec \mathcal{H} .

L'homme a proposé de tester la contraposée de cette remarque afin de détecter l'inconsistance de $[x_i]$. La complexité de l'algorithme résultant est raisonnable, au détriment de l'optimalité du résultat, car des valeurs inconsistantes peuvent subsister. Ainsi, une contrainte \mathcal{H}_j est dite *2B-consistante* sur le domaine $[x_i]$ si et seulement si

$$\left\{ \begin{array}{l} \exists p \in [\downarrow \underline{x}_i, \uparrow \underline{x}_i[\text{ tel que } p \in \pi_i (\mathbb{S}_j \cap [\mathbf{x}]) \\ \exists q \in]\downarrow \bar{x}_i, \uparrow \bar{x}_i] \text{ tel que } q \in \pi_i (\mathbb{S}_j \cap [\mathbf{x}]) \end{array} \right\}.$$

Comme la suivante, cette approche suppose de pouvoir calculer exactement l'enveloppe des relations réelles associées aux contraintes, ce qui n'est malheureusement pas possible avec les fonctions transcendantes, dont la précision n'est pas garantie, et dépend de la mise en œuvre effectuée.

En remarquant qu'il suffit de détecter une valeur inconsistante dans un domaine pour que celui-ci ne soit pas localement consistant, la notion de consistance d'enveloppe² a alors été définie. Cette consistance est donc une approximation relativement naturelle de la consistance d'arc. Une contrainte \mathcal{H}_j est dite *consistante par enveloppe* sur le domaine $[x_i]$ si et seulement si

$$[x_i] = \uparrow [\pi_i (\mathbb{S}_j \cap [\mathbf{x}])].$$

Là encore, la mise en œuvre de ce test pour des contraintes qui ne sont pas primitives est compromise par le problème de dépendance [Gra98]. Il est nécessaire de décomposer le CSP \mathcal{H} en contraintes primitives et de vérifier la consistance sur ces nouvelles contraintes. Malheureusement $\mathcal{D}(\mathcal{H})$ peut être consistant par enveloppe même si \mathcal{H} ne l'est pas.

Enfin, la *box-consistance* a deux avantages. Elle permet d'une part de remédier aux problèmes liés à l'introduction de variables auxiliaires [BGGP99] D'autre part, elle utilise des évaluations intervalles des contraintes, qui peuvent être directement implémentées. La box-consistance de la contrainte \mathcal{H}_j est vérifiée numériquement si

$$\forall p \in \{[\downarrow \underline{x}_i, \uparrow \underline{x}_i[,]\downarrow \bar{x}_i, \uparrow \bar{x}_i]\}, ([x_1], \dots, [x_{i-1}], p, [x_{i+1}], \dots, [x_m]) \in \mathbb{S}_j.$$

² *Hull-consistency* dans la langue de Yeats

8.3 Evaluation de la chaîne de mesure : cas de données multiples

L'évaluation *a priori* du capteur logiciel lorsque une seule donnée est disponible fait l'objet du chapitre 6. Dans ce paragraphe, nous étendons ces résultats au cas de capteurs logiciels qui fournissent une mesure \hat{x} de x calculée à partir d'un ensemble de n données (y_1, \dots, y_n) . Nous distinguons pour cela deux types de capteurs logiciels :

- ceux où les n données proviennent de n capteurs physiques mis en parallèle,
- ceux où les n données proviennent d'un unique capteur physique, dans n conditions expérimentales différentes.

Dans ces deux cas, une loi de fusion de ces données dans le capteur logiciel doit être choisie. Dans un premier temps, nous supposons que ces n données proviennent de n capteurs différents.

8.3.1 n capteurs physiques

Dans ce cas, nous supposons que chaque capteur dispose de sa propre fonction de transfert f_i , de sorte que l'on puisse écrire :

$$y_i(x) = f_i(x, \mathbf{e}_i) + b_i(x),$$

où $\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]$ regroupe l'erreur de modélisation de la fonction de transfert f_i et le défaut de justesse du $i^{\text{ème}}$ capteur physique. L'erreur de mesure $b_i(x)$ entachant le signal de sortie du $i^{\text{ème}}$ capteur physique est également à support borné : $b_i(x) \in [-\Delta y_i(x), \Delta y_i(x)]$, où $\Delta y_i(x)$ représente l'erreur de fidélité du $i^{\text{ème}}$ capteur physique pour la valeur x du mesurande. Les fonctions $f_i(x)$ ainsi que les bornes $\underline{\mathbf{e}}_i$ et $\bar{\mathbf{e}}_i$ sont supposées connues pour tout i variant de 1 à n .

8.3.1.1 Problème direct

Soit g_i la courbe d'étalonnage du $i^{\text{ème}}$ capteur. Différentes lois de fusion de données peuvent être établies. Nous choisissons de considérer le cas où le capteur logiciel fournit

$$[\hat{x}](x^*) = \bigcap_{i \in \{1, \dots, n\}} [\hat{x}_i](x^*), \quad (8.1)$$

où $[\hat{x}_i](x^*)$ est la mesure obtenue à partir de la sortie du $i^{\text{ème}}$ capteur à l'aide de la courbe d'étalonnage g_i . La loi de fusion consiste donc ici à prendre l'intersection des résultats de tous les capteurs, mais d'autres lois pourraient être considérées.

En posant

$$h_i^d : \mathbb{X} \times [-1, 1] \times [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i] \longmapsto \mathbb{R} \\ (x^*, \varepsilon, \mathbf{e}_i) \longrightarrow g_i(f_i(x^*, \mathbf{e}_i) + \varepsilon \Delta y(x^*)),$$

on montre que

$$\begin{aligned}
 \widehat{x}(x^*) &= \left[\max_{i \in \{1, \dots, n\}} \min_{\varepsilon \in [-1, 1], \mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} g_i(f_i(x^*, \mathbf{e}_i) + \varepsilon \cdot \Delta y_i(x^*)), \right. \\
 &\quad \left. \min_{i \in \{1, \dots, n\}} \max_{\varepsilon \in [-1, 1], \mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} g_i(f_i(x^*, \mathbf{e}_i) + \varepsilon \cdot \Delta y_i(x^*)) \right] \\
 &= \left[\max_{i \in \{1, \dots, n\}} \min_{\varepsilon \in [-1, 1], \mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^d(x^*, \varepsilon, \mathbf{e}_i), \min_{i \in \{1, \dots, n\}} \max_{\varepsilon \in [-1, 1], \mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^d(x^*, \varepsilon, \mathbf{e}_i) \right] \quad (8.2)
 \end{aligned}$$

Cet intervalle peut éventuellement être vide, ce qui traduit le fait que les hypothèses *a priori* considérées pour chaque capteur ne sont pas compatibles entre elles.

On suppose les n capteurs justes On a donc $\underline{\mathbf{e}}_i = \bar{\mathbf{e}}_i = \mathbf{0}$, pour tout $i \in \{1, \dots, n\}$. Pour le capteur logiciel, à partir de (8.2), il est alors possible de calculer directement pour toute valeur de $x^* \in \mathbb{X}$:

- le défaut de justesse

$$\begin{aligned}
 j(x^*) &= \frac{1}{2} \left(\widehat{x}(x^*) + \overline{x}(x^*) \right) - x^* \\
 &= \frac{1}{2} \left(\max_{i \in \{1, \dots, n\}} \min_{\varepsilon \in [-1, 1]} h_i^d(x^*, \varepsilon, \mathbf{0}) + \min_{i \in \{1, \dots, n\}} \max_{\varepsilon \in [-1, 1]} h_i^d(x^*, \varepsilon, \mathbf{0}) \right) - x^*,
 \end{aligned}$$

- l'incertitude de mesure

$$\begin{aligned}
 \delta x(x^*) &= \max(|\widehat{x}(x^*) - x^*|, |\overline{x}(x^*) - x^*|) \\
 &= \max \left(\left| \max_{i \in \{1, \dots, n\}} \min_{\varepsilon \in [-1, 1]} h_i^d(x^*, \varepsilon, \mathbf{0}) - x^* \right|, \left| \min_{i \in \{1, \dots, n\}} \max_{\varepsilon \in [-1, 1]} h_i^d(x^*, \varepsilon, \mathbf{0}) - x^* \right| \right),
 \end{aligned}$$

- l'erreur de fidélité

$$\begin{aligned}
 \Delta x(x^*) &= \frac{1}{2} \left(\overline{x}(x^*) - \widehat{x}(x^*) \right) \\
 &= \frac{1}{2} \left(\min_{i \in \{1, \dots, n\}} \max_{\varepsilon \in [-1, 1]} h_i^d(x^*, \varepsilon, \mathbf{0}) - \max_{i \in \{1, \dots, n\}} \min_{\varepsilon \in [-1, 1]} h_i^d(x^*, \varepsilon, \mathbf{0}) \right).
 \end{aligned}$$

Lorsque x^* est inconnu, les valeurs maximales de ces grandeurs sont obtenues par

$$\bar{j} = \max_{x^* \in \mathbb{X}} j(x^*), \quad \overline{\delta x} = \max_{x^* \in \mathbb{X}} \delta x(x^*), \quad \overline{\Delta x} = \max_{x^* \in \mathbb{X}} \Delta x(x^*).$$

Un au moins des capteurs physiques n'est pas juste A nouveau, seul un encadrement du défaut de justesse peut être obtenu. En effet, pour une valeur donnée $x^* \in \mathbb{X}$ et pour tout $(\mathbf{e}_1, \dots, \mathbf{e}_n) \in [\underline{\mathbf{e}}_1, \bar{\mathbf{e}}_1] \times \dots \times [\underline{\mathbf{e}}_n, \bar{\mathbf{e}}_n]$, le défaut de justesse vaut

$$\begin{aligned}
 j(x^*, \mathbf{e}_1, \dots, \mathbf{e}_n) &= \frac{1}{2} \left(\min_{i \in \{1, \dots, n\}} \bigcap_{i \in \{1, \dots, n\}} h_i^d(x^*, [-1, 1], \mathbf{e}_i) + \max_{i \in \{1, \dots, n\}} \bigcap_{i \in \{1, \dots, n\}} h_i^d(x^*, [-1, 1], \mathbf{e}_i) \right) - x^* \\
 &= \frac{1}{2} \left(\max_{i \in \{1, \dots, n\}} \min_{\varepsilon \in [-1, 1]} h_i^d(x^*, \varepsilon, \mathbf{e}_i) + \min_{i \in \{1, \dots, n\}} \max_{\varepsilon \in [-1, 1]} h_i^d(x^*, \varepsilon, \mathbf{e}_i) \right) - x^*.
 \end{aligned}$$

Par conséquent, pour tout $x^* \in \mathbb{X}$

$$j(x^*) \in \left[\min_{\mathbf{e}_1 \in [\underline{\mathbf{e}}_1, \bar{\mathbf{e}}_1], \dots, \mathbf{e}_n \in [\underline{\mathbf{e}}_n, \bar{\mathbf{e}}_n]} j(x^*, \mathbf{e}_1, \dots, \mathbf{e}_n), \max_{\mathbf{e}_1 \in [\underline{\mathbf{e}}_1, \bar{\mathbf{e}}_1], \dots, \mathbf{e}_n \in [\underline{\mathbf{e}}_n, \bar{\mathbf{e}}_n]} j(x^*, \mathbf{e}_1, \dots, \mathbf{e}_n) \right].$$

De même seul un encadrement de l'incertitude de mesure peut être calculé. En effet, pour une valeur donnée $x^* \in \mathbb{X}$ et pour tout $(\mathbf{e}_1, \dots, \mathbf{e}_n) \in [\underline{\mathbf{e}}_1, \bar{\mathbf{e}}_1] \times \dots \times [\underline{\mathbf{e}}_n, \bar{\mathbf{e}}_n]$, l'incertitude de mesure vaut

$$\begin{aligned} \delta x(x^*, \mathbf{e}_1, \dots, \mathbf{e}_n) &= \max \left(\left| \min_{i \in \{1, \dots, n\}} \bigcap h_i^d(x^*, [-1, 1], \mathbf{e}_i) - x^* \right|, \right. \\ &\quad \left. \left| \max_{i \in \{1, \dots, n\}} \bigcap h_i^d(x^*, [-1, 1], \mathbf{e}_i) - x^* \right| \right) \\ &= \max \left(\left| \max_{i \in \{1, \dots, n\}} \min_{\varepsilon \in [-1, 1]} h_i^d(x^*, \varepsilon, \mathbf{e}_i) - x^* \right|, \right. \\ &\quad \left. \left| \min_{i \in \{1, \dots, n\}} \max_{\varepsilon \in [-1, 1]} h_i^d(x^*, \varepsilon, \mathbf{e}_i) - x^* \right| \right). \end{aligned}$$

Par conséquent, pour tout $x^* \in \mathbb{X}$,

$$\delta x(x^*) \in \left[\min_{\mathbf{e}_1 \in [\underline{\mathbf{e}}_1, \bar{\mathbf{e}}_1], \dots, \mathbf{e}_n \in [\underline{\mathbf{e}}_n, \bar{\mathbf{e}}_n]} \delta x(x^*, \mathbf{e}_1, \dots, \mathbf{e}_n), \max_{\mathbf{e}_1 \in [\underline{\mathbf{e}}_1, \bar{\mathbf{e}}_1], \dots, \mathbf{e}_n \in [\underline{\mathbf{e}}_n, \bar{\mathbf{e}}_n]} \delta x(x^*, \mathbf{e}_1, \dots, \mathbf{e}_n) \right].$$

Par contre, l'erreur de fidélité peut être donnée de manière exacte.

$$\Delta x(x^*) = r \left(\bigcap_{i \in \{1, \dots, n\}} [\hat{x}_i](x^*) \right).$$

En utilisant (8.2), on obtient

$$\begin{aligned} \Delta x(x^*) &= \frac{1}{2} \left(\min_{i \in \{1, \dots, n\}} \max_{\varepsilon \in [-1, 1], \mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^d(x^*, \varepsilon, \mathbf{e}_i) \right. \\ &\quad \left. - \max_{i \in \{1, \dots, n\}} \min_{\varepsilon \in [-1, 1], \mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^d(x^*, \varepsilon, \mathbf{e}_i) \right). \end{aligned}$$

La valeur maximale $\overline{\Delta x}$ de l'erreur de fidélité sur l'étendue de mesure est ensuite simplement donnée par

$$\overline{\Delta x} = \max_{x^* \in \mathbb{X}} \Delta x(x^*). \quad (8.3)$$

8.3.1.2 Problème inverse

Nous recherchons la valeur maximale $\overline{\Delta y}$ de l'erreur de fidélité pour les n capteurs telle que l'incertitude de mesure de x^* reste inférieure à δx . Ainsi, si

$$\Delta y_i \leq \overline{\Delta y}, \quad i = 1, \dots, n$$

alors le cahier des charges sera satisfait.

D'après (6.27), pour chaque capteur pris indépendamment, si x^* est connu on peut écrire

$$\Delta y_i(x^*) = \min \left(- \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^i(x^*, \varepsilon, \mathbf{e}_i), \max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^i(x^*, \varepsilon, \mathbf{e}_i) \right)$$

où

$$h_i^i : \mathbb{X} \times [-1, 1] \times [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i] \mapsto \mathbb{R} \\ (x^*, \varepsilon, \mathbf{e}_i) \longrightarrow g_i^{-1}(x^* + \varepsilon \cdot \delta x) - f_i(x^*, \mathbf{e}_i)$$

Lorsqu'on prend en compte les n données, il vient alors

$$\begin{aligned} \Delta y(x^*) &= \min_{i \in \{1, \dots, n\}} \min \left(- \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^i(x^*, \varepsilon, \mathbf{e}_i), \max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^i(x^*, \varepsilon, \mathbf{e}_i) \right) \\ &= \min \left(- \max_{i \in \{1, \dots, n\}} \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^i(x^*, \varepsilon, \mathbf{e}_i), \right. \\ &\quad \left. \min_{i \in \{1, \dots, n\}} \max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^i(x^*, \varepsilon, \mathbf{e}_i) \right). \end{aligned}$$

Si x^* est inconnu, alors

$$\begin{aligned} \Delta y &= \min_{x^* \in \mathbb{X}} \Delta y(x^*) \\ &= \min_{x^* \in \mathbb{X}} \min \left(- \max_{i \in \{1, \dots, n\}} \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^i(x^*, \varepsilon, \mathbf{e}_i), \right. \\ &\quad \left. \min_{i \in \{1, \dots, n\}} \max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^i(x^*, \varepsilon, \mathbf{e}_i) \right) \\ &= \min \left(- \max_{i \in \{1, \dots, n\}} \max_{x^* \in \mathbb{X}} \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^i(x^*, \varepsilon, \mathbf{e}_i), \right. \\ &\quad \left. \min_{i \in \{1, \dots, n\}} \min_{x^* \in \mathbb{X}} \max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h_i^i(x^*, \varepsilon, \mathbf{e}_i) \right). \end{aligned}$$

8.3.2 Cas d'un capteur unique

Dans ce contexte, nous supposons qu'on effectue une série de n mesures à partir du même capteur. Nous supposons que ces n données seront obtenues dans diverses conditions expérimentales. Soit $\xi \in \Xi \subset \mathbb{R}^n$ le vecteur des conditions expérimentales. On peut donc écrire le modèle de la fonction de transfert sous la forme

$$\mathbf{y} = \mathbf{f}(x, \mathbf{e}, \xi),$$

où

$$\begin{cases} y_1 = f(x, \mathbf{e}, \xi_1) \\ \dots \\ y_i = f(x, \mathbf{e}, \xi_i) \\ \dots \\ y_n = f(x, \mathbf{e}, \xi_n) \end{cases},$$

et il s'agit de déterminer les relations entre les différentes caractéristiques des capteurs afin de satisfaire les contraintes souhaitées. Soit

$$\begin{aligned} h^{\text{de}} : \mathbb{X} \times [-1, 1] \times [\underline{\mathbf{e}}, \bar{\mathbf{e}}] \times \Xi &\rightarrow \mathbb{R} \\ (x^*, \varepsilon, \mathbf{e}, \boldsymbol{\xi}) &\longrightarrow g(f(x^*, \mathbf{e}, \xi_i) + \varepsilon \cdot \Delta y(x^*)) \\ h^{\text{ie}} : \mathbb{X} \times [-\delta x, \delta x] \times [\underline{\mathbf{e}}, \bar{\mathbf{e}}] \times \Xi &\rightarrow \mathbb{R} \\ (x^*, \varepsilon, \mathbf{e}, \boldsymbol{\xi}) &\longmapsto g^{-1}(x^* + \varepsilon) - f(x^*, \xi_i) \end{aligned}$$

On a alors dans le cas général,

$$\Delta y = \min \left(- \max_{i \in \{1, \dots, n\}} \max_{x^* \in \mathbb{X}} \min_{\varepsilon \in [-\delta x, \delta x]} \max_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h^i(x^*, \varepsilon, \mathbf{e}_i, \boldsymbol{\xi}), \right. \\ \left. \min_{i \in \{1, \dots, n\}} \min_{x^* \in \mathbb{X}} \max_{\varepsilon \in [-\delta x, \delta x]} \min_{\mathbf{e}_i \in [\underline{\mathbf{e}}_i, \bar{\mathbf{e}}_i]} h^i(x^*, \varepsilon, \mathbf{e}_i, \boldsymbol{\xi}) \right).$$

Comme la valeur Δy dépend des n conditions expérimentales, une des perspectives envisagées à la suite de ce travail est de définir le critère $\Delta y(\boldsymbol{\xi})$ à optimiser de manière à obtenir la meilleure répartition des valeurs de ξ_i dans l'espace de recherche a priori Ξ de manière à obtenir la valeur optimale de Δy , mais ceci sort du cadre de cette étude.

Index

- évaluation, 128
- acceptable, 95
- accumulation (zone d'), 32
- annulation (effet d'), 13
- approximation extérieure, 20, 33
- approximation extérieure (d'un CSP), 43
- approximation intérieure, 33
- arbre, 56
- arrondi (fonctions d'), 184
- ascendant, 58
- bissection, 29
- box-consistance, 65
- branche, 58
- capteur, 156
- capteur logiciel, 157
- caractérisation, 7, 139
- complet (algorithme), 9
- conjonction, 67
- consistance avec ses ascendants, 58
- consistance d'arc, 41
- consistance globale, 41
- contracteur, 43
- contrainte primitive, 52
- convergence (ordre de), 25
- convergente (fonction d'inclusion), 25
- courbe d'étalonnage, 156
- décomposition d'une contrainte, 56
- discernabilité, 139
- disjonction, 67
- donnée aberrante, 134
- encadrement, 12
- ensemble complémentaire, 43
- ensembles, 14
- enveloppe, 21
- erreur de mesure, 90
- erreur de modélisation, 90
- erreur de sortie, 89
- erreurs bornées (contexte d'), 92
- estimateur ponctuel, 94
- facteur expérimental, 112
- feuille, 58
- fiabilité, 135
- fidélité, 159
- fonction de transfert d'un capteur, 156
- fonction solution associée, 54
- garantie, 9, 14
- graphe cyclique, 56
- graphe de représentation, 51
- identifiabilité, 139
- ImageSP, 35
- incertitude de mesure, 92
- inclusion (test d'), 32
- intervalle booléen, 31
- inversion ensembliste, 95
- inversion explicite (lemme d'), 45
- justesse, 159
- mesure indirecte, 7, 89
- monotone au sens de l'inclusion, 25
- monotonie, 26
- opération étendue, 17
- opération ensembliste pure, 17
- optimal (test d'inclusion), 32
- pavé enveloppe, 23
- pessimisme, 20
- phénomène d'enveloppement, 21, 23
- point fixe, 44
- point fixe (résolveur à), 46
- polytope, 86
- précision, 34
- problème de dépendance, 20
- Project, 76
- projection, 17, 41
- projection (procédure de), 52

prolongeables par continuité (fonctions), 27

réipient, 20

réipients, 12

résumé, 142

réunification, 29

racine, 58

recouvrement, 12

relation réelle, 51

Sivia, 32, 33

SiviaP, 66, 71

sous-pavages, 31

sous-résolveur, 44

sous-vecteur, 43

structurelle (propriété), 141

support, 10

virgule flottante (arithmétique à), 12

Bibliographie

- [ACS94] M. V. A. Andrade, J. L. D. Comba, and J. Stolfi. Affine arithmetic. In *Interval'94*, St Petersburg, 1994.
- [AH83] G. Alefeld and J. Herzberger. *Introduction to Interval Computation*. Academic Press, New York, 1983.
- [AM01] R. Alt and S. Markov. On the algebraic properties of stochastic arithmetic. comparison to interval arithmetic. In W. Krämer and J.W. von Gudenberg, editors, *Scientific Computing, Validated Numerics, Interval Methods*, pages 331–342, Boston, 2001. Kluwer Academic publishers.
- [A.N85] Institute of Electrical and Electronics Engineers A.N.S.I. *A standard for binary floating-point arithmetic*. ANSI/IEEE Std. 754-1985, New York, 1985.
- [BB01] K. Buehler and W. Barth. *A new intersection algorithm for parametric surfaces based on LIEs*, pages 179–190. Kluwer Academic/ Plenum Publishers, w. kraemer and j.w. von gudenberg edition, 2001.
- [BBC⁺97] J.-C. Bajard, O. Beaumont, J.-M. Chesneaux, M. Daumas, J. Erhel, D. Michelucci, J.-M. Muller, B. Philippe, N. Revol, J.-L. Roch, and J. Vignes. . *Qualité des calculs sur ordinateur*. Masson, Paris, 1997.
- [BBJ⁺01a] I. Braems, F. Berthier, L. Jaulin, M. Kieffer, and E. Walter. Estimation et traitement garanti des incertitudes de mesure: applications en électrochimie. In F. Lepoutre, D. Placko, and Y. Surrel, editors, *Instrumentation pour les mesures physiques, C2I*, volume 1, pages 55–62. Hermès Science Publications, Paris, 2001.
- [BBJ⁺01b] I. Braems, F. Berthier, L. Jaulin, M. Kieffer, and E. Walter. Guaranteed estimation of electrochemical parameters by set inversion using interval analysis. *Journal of Electroanalytical Chemistry*, 495(1): 1–9, 2001.
- [BDM⁺93] F. Berthier, J. Diard, C. Montella, L. Pronzato, and E. Walter. Sur l'Identifiabilité structurelle de la réaction de Volmer-Heyrovsky. *J. Chim. Phys.*, 90: 2069–2081, 1993.
- [BDM⁺95] F. Berthier, J.-P. Diard, C. Montella, L. Pronzato, and E. Walter. Choice of experimental method induced by structural properties of mechanisms. *J. Electroanalytical Chemistry*, 399: 1–6, 1995.
- [BDPW96] F. Berthier, J.-P. Diard, L. Pronzato, and E. Walter. Identifiability and distinguishability in electrochemistry. *Automatica*, 32: 973–984, 1996.

- [BG97] F. Benhamou and L. Granvilliers. Automatic generation of numerical redundancies for nonlinear constraint solving. *Reliable Computing*, 3(3): 335–344, 1997.
- [BG00] Frédéric Benhamou and Frédéric Goualard. Universally quantified interval constraints. In Rina Dechter, editor, *Proceedings of the Sixth International Conference on Principles and Practice of Constraint Programming (CP'2000), Lecture Notes in Computer Science vol. 1894*, pages 67–82. SpringerVerlag, 2000.
- [BGGP99] F. Benhamou, F. Goualard, L. Granvilliers, and J. F. Puget. Revising hull and box consistency. In *Proceedings of the International Conference on Logic Programming*, pages 230–244, Las Cruces, NM, 1999.
- [BGLC99] F. Benhamou, F. Goualard, E. Languenou, and M. Christie. An algorithm to compute inner approximations of relations for interval constraints. In *Conference on Perspectives of Systems Informatics (PSI'99)*, 1999.
- [BJKW01a] I. Braems, L. Jaulin, M. Kieffer, and E. Walter. Guaranteed numerical alternatives to structural identifiability testing. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 3122–3128, 2001.
- [BJKW01b] I. Braems, L. Jaulin, M. Kieffer, and E. Walter. Set computation, computation of volumes and data safety. In W. Kraemer and J. W. von Gudenberg, editors, *Scientific Computing, Validated Numerics, Interval Methods*, pages 267–280, Boston, 2001. Kluwer Academic Publishers.
- [BJKW02] I. Braems, L. Jaulin, M. Kieffer, and E. Walter. Identifiabilités, discernabilités et analyse par intervalles. In *Conférence Internationale Francophone d'Automatique*, pages 823–828, Nantes, 2002.
- [BL01] F.G. Boese and W. Luther. Enclosure of the zero set of polynomials in several complex variables. *Multidimensional systems and signal processing*, 12: 165–197, 2001.
- [BM98] M. Berz and K. Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4(4): 361–369, 1998.
- [Bou86] B. A. Boukamp. A nonlinear least squares fit procedure for analysis of immitance data of electrochemical systems. *Solid State Ionics*, 20: 31, 1986.
- [Bra99] I. Braems. *Estimation Nonlinéaire Robuste et Garantie Grâce À L'analyse Par Intervalles; Applications En Électrochimie*. D.E.A. Automatique et Traitement du Signal, Paris-Sud., 1999.
- [BSV⁺01] C. Bliiek, P. Spellucci, L.N. Vicente, A. Neumaier, L. Granvilliers, E. Monfroy, F. Benhamou, E. Huens, P. Van Hentenrynck, D. Sam-Haroud, and B. Faltings. Algorithms for solving nonlinear constrained and optimization problems: the state of the art. Technical report, A progress report of the COCONUT project, <http://www.mat.univie.ac.at/~neum/glopt/coconut/StArt.html>, 2001.
- [CET98] Michèle Neuilly CETAMA. *Modélisation et estimation des erreurs de mesure*. Lavoisier TechDoc, Paris, 1998.

- [CG96] M.J. Chapman and K.R. Godfrey. Nonlinear compartmental model indistinguishability. *Automatica*, 32(3): 419–422, 1996.
- [CGV94] M. J. Chapman, K.R. Godfrey, and S. Vajda. Indistinguishability for a class of nonlinear compartmental models. *Math. Biosciences*, 119: 77–95, 1994.
- [Che95] J.-M. Chesneaux. *L'arithmétique stochastique et le logiciel CADNA*. PhD thesis, Université Pierre et Marie Curie, Paris, 1995.
- [Chr02] M. Christie. Camera benchmark. 2002. communication privée.
- [Cle87] J. C. Cleary. Logical arithmetic. *Future Computing Systems*, 2(2): 125–149, 1987.
- [Dav87] E. Davis. Constraint propagation with interval labels. *Artificial Intelligence*, 32(3): 281–331, 1987.
- [DB99] J. R. Dygas and M.W. Breiter. Variance of errors and elimination of outliers in the least squares analysis of impedance spectra. *Electrochimica Acta*, 44: 4163–4174, 1999.
- [DGM91] J.P. Diard, B. Le Gorrec, and C. Montella. Proceeding of the electrochemical impedance forum. page 145, 1991.
- [DGM96] J.-P. Diard, B. Le Gorrec, and C. Montella. *Cinétique Électrochimique*. Hermann collection Méthodes, 1996.
- [dH95] J.M. Van den Hof. Structural identifiability from input-output observations. Technical Report BS-R9514, CWI, Amsterdam, 1995.
- [Did97] O. Didrit. *Analyse par intervalles pour l'automatique; résolution globale et garantie de problèmes non linéaires en robotique et commande robuste*. PhD dissertation, Université Paris-Sud, Orsay, France, juin 1997.
- [Dor00] P. Dorato. Quantified multivariate polynomial inequalities. *IEEE Control Systems Magazine*, pages 48–58, 2000.
- [DS92] D.Z. D'Argenio and A. Schumitzky. *ADAPT II User's Guide*. Biomedical Simulation Resource, University of Southern California, Los Angeles, 1992.
- [DVJBNP01] L. Denis-Vidal, G. Joly-Blanchard, C. Noiret, and M. Petitot. An algorithm to test identifiability of non-linear systems. *Proc. NOLCOS'01, St Petersburg*, juillet 2001.
- [DYA97] P. Dorato, Wei Yang, and C. Abdallah. Robust multi-objective feedback design by quantifier elimination. *Journal of Symbolic Computation*, 24: 153–159, 1997.
- [fNX86] Norme française NF X 07-001. *Vocabulaire international des termes fondamentaux et généraux de métrologie*. AFNOR, Octobre 1986.
- [FS96] C.L. Fefferman and L.A. Seco. Interval arithmetic in quantum mechanics. In R.B. Kearfott and V. Kreinovich, editors, *Applications in Interval Computations*, pages 145–167, Dordrecht, The Netherlands, 1996. Kluwer.
- [GCV94] K.R. Godfrey, M.J. Chapman, and S. Vajda. Identifiability and distinguishability of nonlinear pharmacokinetic models. *J. Pharmacokin. Biopharm*, 22: 229–251, 1994.

- [GNS92] G. C. Goodwin, B. Ninness, and M. E. Salgado. Quantification of uncertainty in estimation. *Proc. American Control Conferences*, 1992.
- [Gra98] L. Granvilliers. *Consistances locales et transformations symboliques de contraintes d'intervalles*. PhD dissertation, Université d'Orléans, France, 1998.
- [Han68] E. R. Hansen. On solving systems of equations using interval arithmetic. *Mathematical Computing*, 22: 374–384, 1968.
- [Han69] In E. Hansen, editor, *Topics in Interval Analysis*. Oxford University Press, London, 1969.
- [Han79] E. R. Hansen. Global optimization using interval analysis: the one-dimensionnal case. *Journal of Optimization Theory and Applications*, 29(3): 331–344, 1979.
- [Han92] E. R. Hansen. *Global Optimization using Interval Analysis*. Marcel Dekker, New York, NY, 1992.
- [HHKR95a] R. Hammer, M. Hocks, U. Kulisch, and D. Ratz. *C++ Toolbox for Verified Computing*. Springer-Verlag, Berlin Heidelberg New York, 1995.
- [HHKR95b] R. Hammer, M. Hocks, U. Kulish, and D. Ratz. *C++ Toolbox for Verified Computing*. Springer-Verlag, Berlin, Germany, 1995.
- [HMK97] P. Van Hentenryck, D. MacAllister, and D. Kapur. Solving polynomial systems using a branch and prune approach. *SIAM Journal on Numerical Analysis*, 34(2): 797–827, 1997.
- [HT90] R. Horst and H. Tuy. *Global Optimization*. Springer-Verlag, Berlin, 1990.
- [Hyv92] E. Hyvönen. Constraint reasoning based on interval arithmetic; the tolerance propagation approach. *Artificial Intelligence*, 58(1-3): 71–112, 1992.
- [IV99] Andrei A. Istratov and Oleg F. Vyvenko. Exponential analysis in physical phenomena. *American Institute of Physics*, 70(2): 1233–1257, february 1999.
- [Jau94] L. Jaulin. *Solution globale et garantie de problèmes ensemblistes; application à l'estimation non linéaire et à la commande robuste*. PhD dissertation, Université Paris-Sud, Orsay, France, 1994. Available at: <http://www.istia.univ-angers.fr/~jaulin/thesejaulin.zip>.
- [Jau00a] L. Jaulin. Interval constraint propagation with application to bounded-error estimation. *Automatica*, 36(10): 1547–1552, 2000.
- [Jau00b] L. Jaulin. *Le calcul ensembliste par analyse par intervalles*. Habilitation à diriger des recherches, Université Paris-Sud, Orsay, France, 2000. Available at: <http://www.istia.univ-angers.fr/~jaulin/hdrjaulin.zip>.
- [Jau01] L. Jaulin. Reliable minimax parameter estimation. *Reliable Computing*, 7(3): 231–246, 2001.
- [JBKW01] L. Jaulin, I. Braems, M. Kieffer, and E. Walter. Nonlinear state estimation using forward-backward propagation of intervals. In W. Kraemer and J. W. von Gudenberg, editors, *Scientific Computing, Validated Numerics, Interval Methods*, pages 191–204, Boston, 2001.

- [JBW02] L. Jaulin, I. Braems, and E. Walter. Interval methods for nonlinear identification and robust control. In *41st Conference on Decision and Control*, 2002. accepté.
- [JGW⁺97] L. Jaulin, J. L. Godet, E. Walter, A. Elliasmine, and Y. Leduff. Light scattering data analysis via set inversion. *Journal of Physics A: Mathematical and General*, pages 7733–7738, 1997.
- [JKBW01] L. Jaulin, M. Kieffer, I. Braems, and E. Walter. Guaranteed nonlinear estimation using constraint propagation on sets. *International Journal of Control*, 74(18): 1772–1782, 2001.
- [JKDW01] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, London, 2001.
- [JP90] J.A Jacquez and T. Perry. Parameter estimation - local identifiability of parameters. *Amer. J. Physiol.*, 258: E727–E736, 1990.
- [JW93a] L. Jaulin and E. Walter. Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis. *Mathematics and Computers in Simulation*, 35(2): 123–137, 1993.
- [JW93b] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4): 1053–1064, 1993.
- [JW96] L. Jaulin and E. Walter. Guaranteed tuning, with application to robust control and motion planning. *Automatica*, 32(8): 1217–1221, 1996.
- [JW99] L. Jaulin and E. Walter. Guaranteed bounded-error parameter estimation for nonlinear models with uncertain experimental factors. *Automatica*, 35(5): 849–856, 1999.
- [JWD96] L. Jaulin, E. Walter, and O. Didrit. Guaranteed robust nonlinear parameter bounding. In *Proceedings of CESA'96 IMACS Multiconference (Symposium on Modelling, Analysis and Simulation)*, pages 1156–1161, Lille, France, 1996.
- [Kea96] R. B. Kearfott. *Rigorous Global Search: Continuous Problems*. Kluwer, Dordrecht, the Netherlands, 1996.
- [Kie99] M. Kieffer. *Estimation ensembliste par analyse par intervalles, application à la localisation d'un véhicule*. PhD dissertation, Université Paris-Sud, Orsay, France, 1999.
- [Kim98] H. Kimura. Non-uniqueness, uncertainty and complexity in modeling. *Journal of Applied Computation, Control, Signals and Circuits*, 1: 455–485, 1998.
- [KJBW01] M. Kieffer, L. Jaulin, I. Braems, and E. Walter. Guaranteed set computation with subpavings. In W. Kraemer and J. W. von Gudenberg, editors, *Scientific Computing, Validated Numerics, Interval Methods*, pages 167–178, Boston, 2001.
- [KJW02] M. Kieffer, L. Jaulin, and E. Walter. Guaranteed recursive nonlinear state bounding using interval analysis. *International Journal of Adaptive Control and Signal Processing*, 6(3): 193–218, 2002.
- [KK96] R. B. Kearfott and V. Kreinovich, editors. *Applications of Interval Computations*. Kluwer, Dordrecht, the Netherlands, 1996.

- [KM81] U. Kulisch and W. L. Miranker. *Computer Arithmetic in Theory and Practice*. Academic Press, New York, NY, 1981.
- [Knü94] O. Knüppel. PROFIL/BIAS – A fast interval library. *Computing*, 53: 277–287, 1994.
- [KR50] T.C. Koopmans and O. Reiersol. The identification of structural characteristics. *Ann. Math. Stat.*, 21(2): 165–181, 1950.
- [Kul96] U. Kulish. Numerical algorithms with automatic results verification. In J. Renegar, M. Schub, and S. Smale, editors, *The Mathematics of Numerical Analysis*, volume 32 of *Lectures in Applied Mathematics*, pages 471–502. American Mathematical Society, 1996.
- [KW98] M. Kieffer and E. Walter. Interval analysis for guaranteed nonlinear parameter estimation. In A. C. Atkinson, L. Pronzato, and H. P. Wynn, editors, *MODA 5-Advances in Model-Oriented Data Analysis and Experiment Design*, pages 115–125. Physica-Verlag, Heidelberg, Germany, 1998.
- [KWBJ01] M. Kieffer, E. Walter, I. Braems, and L. Jaulin. Interval analysis for nonlinear parameter and state estimation: contributions and limitations. In *5th IFAC symposium on NonLinear and Control Systems*, Saint Petersburg, 2001.
- [Lho93] Olivier Lhomme. Consistency techniques for numeric CSPs. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 232–238, Chambéry, France, 1993.
- [Mac77] A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1): 99–118, 1977.
- [MCR02] R. H. Milocco, E. B. Castro, and S. G. Real. Parameter identification and model validation of adsorption and electron transfer processes using impedances. *Electrochimica Acta*, pages 2035–2041, 2002.
- [MNPLW96] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, editors. *Bounding Approaches to System Identification*. Plenum Press, New York, NY, 1996.
- [Moo59] R. E. Moore. Automatic error analysis in digital computation. Technical Report LMSD-48421 Lockheed Missiles and Space Co, Palo Alto, CA, 1959.
- [Moo66] R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.
- [Moo79] R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, PA, 1979.
- [Moo92] R. E. Moore. Parameter sets for bounded-error data. *Mathematics and Computers in Simulation*, 34(2): 113–119, 1992.
- [MR91] U. Montanari and F. Rossi. Constraint relaxation may be perfect. *Artificial Intelligence*, 48(2): 143–170, 1991.
- [Neu] A. Neumaier. Taylor forms - use and limits, to appear. <http://www.mat.univie.ac.at/~eum/papers.htm>.
- [Neu90] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, UK, 1990.

- [Noi00] C. Noiret. *Utilisation du calcul formel pour l'identifiabilité de modèles paramétriques et nouveaux algorithmes en estimation de paramètres*. PhD thesis, Université de Technologie de Compiègne, 2000.
- [Nor94] J. P. Norton, editor. Special Issue on Bounded-Error Estimation: Issue 1. 1994. *International Journal of Adaptive Control and Signal Processing* 8(1): 1–118.
- [Nor95] J. P. Norton, editor. Special Issue on Bounded-Error Estimation: Issue 2. 1995. *International Journal of Adaptive Control and Signal Processing* 9(1): 1–132.
- [OA96] M. E. Orazem and P. Agarwal. Application of measurement models to electrohydrodynamic impedance spectroscopy. *J. Electrochem. Soc.*, 143: 948–960, 1996.
- [Oll90] F. Ollivier. *Le problème de l'identifiabilité structurelle globale : approche théorique, méthodes effectives et bornes de complexité*. PhD thesis, Ecole Polytechnique, 1990.
- [PLW90] H. Piet-Lahanier and E. Walter. Characterization of non-connected parameter uncertainty regions. *Mathematics and Computers in Simulation*, 32: 553–560, 1990.
- [PP98] M. S. Petkovic and L. D. Petkovic. *Complex interval arithmetic and its applications*. Wiley-VCh, Berlin, 1998.
- [PT95] W.H. Press and S.A. Teukolsky. *Numerical Recipes in C. The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1995. Second edition.
- [PW96] L. Pronzato and E. Walter. Robustness to outliers of bounded-error estimators and consequences on experiment design. In M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, editors, *Bounding Approaches to System Identification*, pages 199–212, New York, 1996. Plenum.
- [RLWV85] A. Raksanyi, Y. Lecourtier, E. Walter, and E. Venot. Identifiability and distinguishability testing in computer algebra. *Math. Biosci.*, 77(1-2): 245–266, 1985.
- [RMP93] M. Robert, M. Marchandiaux, and M. Porte. *Capteurs intelligents et méthodologie d'évaluation (Série automatique)*. Hermès Science, Paris, 1993.
- [RTFJC01] N. Ramdani, M. Tayakout-Fayolle, C. Jallut, and Y. Candau. Etude de l'identifiabilité structurelle des paramètres lors de la caractérisation thermiques des matériaux. *Congrès Français de Thermique*, mai 2001.
- [Rum01] S. M. Rump. INTLAB - INTerval LABoratory. In J. Grabmeier, E. Kaltofen, and V. Weispfennig, editors, *Handbook of Computer Algebra: Foundations, Applications, Systems*. Springer-Verlag, Heidelberg, Germany, 2001.
- [Sal98] G. Salié. *J. Electroanal. Chem.*, 447: 211–234, 1998.
- [SAM85] Interface électrochimique 1286. Technical report, Schlumberger instruments, 1985.
- [Sch68] F. C. Schweppe. Recursive state estimation: unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, 13(1): 22–28, 1968.

- [Sed01] A. Sedoglavic. *Méthodes seminumériques en algèbre différentielle: applications à l'étude des propriétés structurelles de systèmes différentiels algébriques en automatique*. PhD thesis, Ecole Polytechnique, 2001.
- [TK98] E. Tang-Kwor. *Contribution au développement de méthodes périodiques de mesure de propriétés thermophysiques des matériaux opaques*. PhD thesis, Université Paris XII Val-de-Marne, 1998.
- [VA98] T. J. VanderNoot and I. Abrahams. *J. Electroanal. Chem.*, 448: 17–23, 1998.
- [VGR89] S. Vajda, K. R. Godfrey, and H. Rabitz. Similarity transformation approach to identifiability analysis of nonlinear compartmental models. *Math. Biosc.*, 93 (2): 217–248, 1989.
- [VR94] S. Vajda and H. Rabitz. Identifiability and distinguishability of general reaction systems. *J. Phys. Chem.*, 89: 5265–5271, 1994.
- [Wal75] D.L. Waltz. Generating semantic descriptions from drawings of scenes with shadows. In P. H. Winston, editor, *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, New York, NY, 1975.
- [Wal82] E. Walter. *Identifiability of state space models*. Springer-Verlag, Berlin, Germany, 1982.
- [Wal90] E. Walter, editor. Special Issue on Parameter Identification with Error Bounds. 1990. *Mathematics and Computers in Simulation* 32(5-6): 447–607.
- [Wal98] G. W. Walster. The extended real interval system. Available at: <http://www.mscs.edu/globsol/walster-papers.html>, 1998.
- [Weba] <http://dutita0.twi.tudelft.nl/users/vuik/wi211/disasters.html>.
- [WEBb] <http://elib.zib.de/pub/packages/mathprog/polyth/>.
- [WEBc] <http://math.washington.edu/~hillman/pub/volume>.
- [Webd] <http://prologianet.univ-mrs.fr/Us/prolog4.html>.
- [Webe] <http://www-sop.inria.fr/cafe/Evelyne.hubert/webdiffalg/diffalg.html>.
- [WEBf] <http://www.cs.utep.edu/interval-comp/main.html>.
- [Webg] <http://www.inria.fr/recherche/equipes/loco.fr.html>.
- [Webh] http://www.mscs.mu.edu/~globsol/javaintervalcalc/i_c.html.
- [Webi] <http://www.risc.uni-linz.ac.at/people/sratscha/>.
- [Webj] <http://www.saam.com>.
- [Webk] <http://www.sciences.univ-nantes.fr/info/perso/permanents/granvil/realpaver/>.
- [Webl] <http://www.sun.com/software/sundev/news/features/intervals.html>.
- [Webm] <http://www.uga.edu/cms/Elec.html>.

- [Whi93] L. Darrel Whitley, editor. *Foundations of Genetic Algorithms.2*. Morgan Kaufmann Publishers, San Mateo, 1993.
- [WL81] E. Walter and Y. Lecourtier. Unidentifiable compartmental models: What to do? *Math. Biosci.*, 56: 1–25, 1981.
- [WL82] E. Walter and Y. Lecourtier. Global approaches to identifiability testing for linear and nonlinear state space models. *Mathematics and Computers in Simulation*, 24: 472–482, 1982.
- [WP90] E. Walter and L. Pronzato. Qualitative and quantitative experiment design for phenomenological models, a survey. *Automatica*, 26: 195–213, 1990.
- [WP97] E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer-Verlag, London, UK, 1997.
- [Zhi91] A. A. Zhiglavsky. *Theory of Global Random Search*. Kluwer, Dordrecht, the Netherlands, 1991.
- [ZNE90] S. Zuhe, A. Neumaier, and M.C. Eiermann. Solving minimax problems by interval methods. *BIT*, 30: 742–751, 1990.

