

ORSAY

n° d'ordre : 3074



UNIVERSITÉ DE PARIS-SUD

CENTRE D'ORSAY

THÈSE

présentée pour obtenir

Le GRADE de DOCTEUR EN SCIENCES
DE L'UNIVERSITÉ PARIS XI ORSAY

par

Luc JAULIN

**SOLUTION GLOBALE ET GARANTIE
DE PROBLÈMES ENSEMBLISTES ;**

**APPLICATION À L'ESTIMATION NON LINÉAIRE
ET À LA COMMANDE ROBUSTE**

Soutenue le **10 Février 1994** devant la Commission d'examen :

MM	BERTRAND Pierre	<i>Président</i>
	MEIZEL Dominique	<i>Rapporteur</i>
	MOORE Ramon	<i>Rapporteur</i>
	RICHALET Jacques	<i>Examineur</i>
	VIGNES Jean	<i>Examineur</i>
	WALTER Eric	<i>Examineur</i>

NOTATION

Les crochets [] ne sont jamais utilisés dans le sens de parenthèses, mais sont réservés pour représenter les encadrements binaires (intervalles, pavés et intercompacts). Les symboles – et +, utilisés comme exposant d'une quantité, signifient la valeur minimale ou maximale au sens classique, au sens vectoriel ou au sens de l'inclusion que peut prendre cette quantité sur l'encadrement binaire considéré.

Les vecteurs \mathbf{v} ou les fonctions vectorielles \mathbf{f} seront imprimés en caractères gras. Les équations vectorielles et les inégalités sont à comprendre composante par composante. Les fonctions réelles usuelles telles que \exp , \sin ... lorsqu'elles ont pour argument un vecteur deviennent des fonctions vectorielles et doivent, elles aussi, être comprises composante par composante. Elles seront imprimées en caractères gras. Par exemple, si $\mathbf{u} = (0, \pi/2, \pi)^T$ et \mathbf{v} est le vecteur de dimension 3 satisfaisant $v_i = \sin(u_i)$, $1 \leq i \leq 3$, alors, $\mathbf{v} = \mathbf{sin}(\mathbf{u}) = (0, 1, 0)^T$.

Les ensembles seront écrits en caractères double. Pour les intervalles et les pavés, nous utiliserons des minuscules.

Pour l'écriture des nombre décimaux, nous séparerons la partie décimale de la partie entière par un point, conformément à la notation anglo-saxonne. La virgule sera utilisée comme séparateur dans les intervalles, les vecteurs-ligne et les opérandes à plusieurs arguments.

$\text{clo}(A)$: adhérence de A (anglais : <i>closure of A</i>).
$C(\mathbb{R}^n)$: espace de tous les compacts de \mathbb{R}^n .
$C_f(\mathbb{R}^n)$: espace de tous les compacts pleins de \mathbb{R}^n .
∂A	: frontière de A .
$\mathbf{e}(\mathbf{p})$: erreur entre le vecteur des sorties modèles et le vecteur des données.
\mathbb{E}	: ensemble des erreurs acceptables.
ε_r	: précision requise pour les sous-pavages à obtenir.
$h_\infty(A, B)$: distance de Hausdorff de A à B .
$\bar{h}_\infty(A, B)$: distance de Hausdorff complémentaire de A à B .
$h_\infty^0(A, B)$: proximité de B à A .

$\text{int}(\mathbf{a})$: partie entière du réel a .
$\text{int}(\mathbf{A})$: intérieur de l'ensemble \mathbf{A} .
$\mathbb{I}\mathbb{R}$: ensemble de tous les intervalles de \mathbb{R} .
$\mathbb{I}\mathbb{R}^n$: ensemble de tous les pavés de \mathbb{R}^n .
$K(\mathbb{R}^n)$: ensemble de tous les sous-pavages de \mathbb{R}^n .
$L_\infty(\mathbf{a}, \mathbf{b})$: distance entre \mathbf{a} et \mathbf{b} , induite par la norme L_∞ .
$L_\infty(\mathbf{A}, \mathbf{B})$: séparation entre \mathbf{A} et \mathbf{B} .
$M(\cdot)$: structure paramétrique du modèle du système à étudier.
$M(\mathbf{p})$: modèle de structure $M(\cdot)$ associé au vecteur de paramètres \mathbf{p} .
$m_\infty(\mathbf{A}, \mathbf{B})$: distance de Hausdorff généralisée de \mathbf{A} à \mathbf{B} .
\circ	: opérateur de composition.
\mathbf{p}	: vecteur des paramètres du modèle.
\mathbf{p}^*	: vraie valeur des paramètres.
\mathbb{P}	: ensemble des paramètres acceptables <i>a priori</i> .
\mathbb{R}	: ensemble des réels.
\mathbb{R}^-	: ensemble des réels négatifs ou nuls.
\mathbb{S}	: ensemble solution des problèmes d'estimation de paramètres.
\mathbb{U}	: sphère unité dans (\mathbb{R}^n, L_∞) .
$V(\mathbf{a})$: voisinage de \mathbf{a} dans l'espace topologique auquel il appartient.
$\text{vol}(\mathbf{A})$: volume du compact \mathbf{A} .
$w(\mathbf{x})$: longueur du pavé \mathbf{x} .
\mathbb{X}	: ensemble solution du problème ensembliste considéré.
\mathbf{y}	: vecteur de toutes les mesures effectuées sur le système.
$\mathbf{y}_m(\mathbf{p})$: vecteur des sorties du modèle associé au vecteur de paramètre \mathbf{p} .
\mathbb{Y}	: ensemble à inverser.
\mathbf{x}	: intervalle ou pavé contenant la variable réelle x ou vectorielle \mathbf{x} .
$\bar{\mathbf{x}}$: milieu de l'intervalle \mathbf{x} .
\mathbf{x}^-	: borne inférieure de l'intervalle \mathbf{x} .
\mathbf{x}^+	: borne supérieure de l'intervalle \mathbf{x} .
$\bar{\mathbf{x}}$: centre du pavé \mathbf{x} .
\mathbf{x}^-	: vecteur coin inférieur du pavé \mathbf{x} .
\mathbf{x}^+	: vecteur coin supérieur du pavé \mathbf{x} .
$[\mathbf{A}]$: pavé enveloppe du compact \mathbf{A} .
\mathbf{f}	: fonction d'inclusion de \mathbf{f} .
\mathbf{f}^*	: fonction d'inclusion minimale de \mathbf{f} .
\mathbf{f}^{-1}	: fonction réciproque de \mathbf{f} , définie par $\mathbf{f}^{-1}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{f}(\mathbf{x}) \in \mathbf{A}\}$.

\times	: produit cartésien de deux ensembles.
$\#A$: cardinal de l'ensemble fini A .
\bar{A}	: complémentaire de l'ensemble A .
$\{\mathbb{K}\}$: sous-ensemble de \mathbb{R}^n formé de l'union des pavés du sous-pavage \mathbb{K} .
$\Delta\mathbb{X}$: sous-pavage indéterminé, ou couche d'incertitude de l'encadrement de \mathbb{X} .
\mathbb{X}^-	: sous-pavage contenu dans l'ensemble \mathbb{X} .
\mathbb{X}^+	: sous-pavage contenant l'ensemble \mathbb{X} .
$[\mathbb{X}^-, \mathbb{X}^+]$: intercompact enfermant \mathbb{X} .
$\overset{\subset}{\rightarrow}$: h_∞ -convergence par l'intérieur.
$\overset{\supset}{\rightarrow}$: h_∞ -convergence par l'extérieur.
$:=$: affectation d'une valeur à une variable.
\triangleq	: est égal par définition.
\diamond	: fin d'un exemple, d'une définition, d'une remarque...

CHAPITRE 1

INTRODUCTION

1.1. Problèmes ponctuels et ensemblistes

1.1.1. Les problèmes ponctuels

Les problèmes posés en mathématiques ou dans les sciences pour l'ingénieur sont souvent formulés de telle manière qu'ils demandent une solution ponctuelle (par exemple un nombre ou un vecteur réel). De nombreuses méthodes ont alors été développées pour résoudre ces types de problèmes, que nous appellerons *problèmes ponctuels*. En raison des incertitudes avec lesquelles sont connues les données intervenant dans l'énoncé du problème, des erreurs numériques introduites par le calculateur lors de la résolution ..., ces méthodes ne permettent généralement pas de bien caractériser l'incertitude avec laquelle la solution est obtenue, et nous ne savons pas quelle validité accorder au résultat. Parmi les méthodes pour résoudre les problèmes ponctuels nous parlerons de cinq types de méthodes :

- Les *méthodes exactes* permettent d'obtenir la ou les solutions (numériques ou formelles) d'une façon exacte. Ces méthodes s'appliquent à un nombre de problèmes très restreints. Un exemple d'un tel problème est de trouver l'ensemble des réels solutions de l'équation $x^2 - 1 = 0$. Les solutions obtenues -1 et 1 peuvent être obtenues avec exactitude.
- Les *méthodes garanties* sont telles que la distance, ou un majorant de cette distance, entre le résultat obtenu et la solution du problème est connue. L'incertitude avec laquelle est trouvée la solution est donc caractérisée de façon certaine.
- Les *méthodes stochastiques* balayent aléatoirement l'ensemble des solutions *a priori* possibles, afin de trouver un résultat (qui dépend du hasard) qui vérifie les contraintes et qui soit le plus acceptable au sens d'un critère de satisfaction à définir. Ces méthodes ne convergent qu'en probabilité, et ne sont que très rarement garanties (incertitude totale quant à la précision avec laquelle la solution est obtenue).
- Les *méthodes déterministes* aboutissent toujours au même résultat lorsque nous

relançons plusieurs fois l'algorithme (contrairement aux méthodes stochastiques). Le hasard n'influence donc pas le résultat.

- Les *méthodes locales* fournissent un résultat tel que tous les vecteurs appartenant à un proche voisinage du résultat dans l'espace des solutions sont "moins" satisfaisant que le résultat lui-même. Ceci ne veut pas dire, et loin s'en faut, que le résultat est la solution, mais juste que le résultat pourrait être la solution cherchée. Les méthodes locales aboutissent à des résultats qui dépendent des conditions initiales, souvent arbitraires, imposées à l'algorithme. Ces résultats peuvent donc être très éloignés, sans que nous le sachions, de la solution. Ces méthodes ne sont donc pas garanties et c'est pourtant elles qui sont de loin les plus utilisées pour les problèmes pour lesquels on ne dispose pas de méthodes exactes.
- Les *méthodes globales* convergent à coup sûr (lorsqu'elles sont déterministes) ou presque sûrement (lorsqu'elles sont stochastiques) vers la ou les solutions du problème, et ceci quelles que soient les conditions initiales imposées à l'algorithme. Les méthodes déterministes globales classiques s'appliquent à d'un nombre très restreint de problèmes comme par exemple pour la résolution d'un système d'équations linéaires ou la minimisation d'une fonction convexe.

Les méthodes disponibles pour résoudre les problèmes ponctuels ne permettent pas (sauf pour une classe très réduite de problèmes) de trouver de façon garantie un résultat global aussi proche que désiré de la solution du problème. Une reformulation ensembliste permettra, comme nous le verrons, de concevoir de nouvelles méthodes de résolution, adaptées à nos besoins de garantie et de globalité, qui ne pouvaient se concevoir avec une approche ponctuelle.

1.1.2. Les problèmes ensemblistes

Dans une reformulation ensembliste d'un problème, nous cherchons non pas une solution ponctuelle, mais un *ensemble solution* pouvant contenir une infinité de vecteurs. Un problème qui admet pour solution un ensemble sera appelé *problème ensembliste*. Parmi les problèmes ensemblistes nous distinguerons deux cas.

- Le premier est celui où le problème ensembliste se trouve associé à un problème ponctuel. L'approche ensembliste a alors uniquement pour but de pallier les

incertitudes de certaines données intervenant dans le problème ponctuel et les erreurs de calcul numérique introduites par le calculateur en les prenant en compte lors du traitement. Nous pouvons ainsi espérer obtenir un résultat garanti. Le problème ensembliste est donc formulé de telle façon que l'ensemble solution contienne, à coup sûr (ou avec une probabilité spécifiée), la ou les solutions exactes du problème ponctuel auquel il est associé. La méthode, pour qu'elle soit garantie, devra générer un ensemble résultat contenant l'ensemble solution et donc la ou les solutions du problème ponctuel.

- Le second cas est celui où le problème ensembliste n'est associé à aucun problème ponctuel. C'est le cas lorsque nous cherchons à caractériser un ensemble dans le but de calculer certaines de ses caractéristiques comme son volume ou ses propriétés de connexité.

Les problèmes ensemblistes sont toujours bien posés dans le sens où la solution existe toujours et est toujours unique (l'ensemble vide peut être solution du problème). Les approches utilisées pour résoudre les problèmes ensemblistes peuvent être regroupés en trois classes : *analytique*, *stochastique* et *garantie*.

- Les méthodes analytiques cherchent à caractériser mathématiquement l'ensemble solution d'une façon exacte en l'exprimant analytiquement, par exemple à l'aide d'inégalités. Ces recherches aboutissent fréquemment à des impasses car, même si la solution est obtenue, sa formulation est souvent tellement complexe qu'elle en devient inexploitable. On se limite alors à trouver des propriétés de l'ensemble solution (comme par exemple sa connexité, ou son volume, ...), sans résoudre complètement le problème posé.
- Dans l'approche stochastique, les problèmes ensemblistes sont résolus par balayage sélectif d'un espace dans lequel l'ensemble solution est supposé inclus. Les algorithmes fondés sur cette approche sont très simples à concevoir et s'appliquent à beaucoup de problèmes, mais le résultat obtenu ne présente aucune garantie. L'ensemble solution est alors approché par un nuage de points qui se trouve à l'intérieur de celui-ci. Ce nuage de points peut donner une très mauvaise idée, sans que nous le sachions, de l'ensemble solution.
- La troisième approche, que nous avons appelée "garantie", consiste à trouver pour résultat un *ensemble exploitable* qui contienne l'ensemble solution. Ce que nous entendons par ensemble exploitable, est un ensemble qu'il est possible de représenter

sur ordinateur avec une occupation mémoire raisonnable et dont on peut facilement calculer certaines caractéristiques importantes (comme le volume ou la connexité), vérifier qu'un élément est à l'intérieur, ... Des exemples types d'ensembles exploitables sont les ellipsoïdes ou les unions de pavés. Nous sommes ainsi amenés à trouver des algorithmes qui nous donnent à chaque itération des ensembles exploitables convergeant (dans un sens à définir) vers l'ensemble solution par l'intérieur ou par l'extérieur. Par cette même approche, il est possible de traiter des problèmes en cascade, l'énoncé de chaque problème dépendant de l'ensemble solution du problème amont. La philosophie adoptée (contraire à celle du calcul formel) consiste à aborder d'une manière numérique des problèmes qui se traitent habituellement d'une manière analytique.

C'est cette troisième approche qui sera adoptée dans ce mémoire, car dans les problèmes d'estimation et de commande, le caractère garanti des solutions est, comme nous le verrons, extrêmement important, comme nous ne manquerons pas de l'argumenter par la suite. Les ensembles exploitables que nous manipulerons seront formés d'unions de pavés et seront appelés *sous-pavages*. Leur rôle principal sera d'approximer l'ensemble solution. L'outil mathématique de base permettant la manipulation des sous-pavages est l'*analyse par intervalles*. Cet outil généralise aux intervalles et aux pavés (aussi appelés *intervalles vectoriels*) tous les calculs existant sur les réels et les vecteurs.

Les problèmes d'*inversion ensembliste* forment une grande classe des problèmes ensemblistes. Dans ce type de problèmes, l'ensemble solution est défini comme l'image réciproque par une fonction donnée d'un ensemble connu. La structure simple de ces problèmes va nous permettre de proposer un algorithme unique permettant d'approximer convenablement leur ensemble solution, et ceci quel que soit le problème d'inversion ensembliste considéré. Les propriétés de convergence de cet algorithme seront étudiées soigneusement, après une étude approfondie de la structure topologique de l'ensemble des compacts. Nous nous intéresserons à titre d'application à la résolution de quelques problèmes ensemblistes qui nous viennent de l'estimation et de la commande non linéaire. Certains des problèmes considérés sont directement des problèmes d'inversion ensembliste. D'autres demanderont la conception d'algorithmes appropriés, en général très voisins de l'algorithme d'inversion ensembliste de base.

1.2. Modélisation, estimation et commande

Introduisons le vocabulaire de base de la théorie des systèmes que nous utiliserons pour traiter des problèmes d'estimation et de commande.

Un *système* ou *processus* est un concept abstrait qui désigne l'objet que nous cherchons à *analyser* (par exemple pour comprendre son fonctionnement et connaître certaines de ses caractéristiques), ou à *commander* (pour lui donner un comportement satisfaisant). Nous agissons sur le système par ses *entrées* et nous prenons connaissance de son comportement par ses *sorties*.

Dans de nombreuses situations, nous ne connaissons, *a priori*, que très peu de choses sur ce système. Le rôle de la *modélisation* consiste alors à chercher un ou plusieurs *modèles* chargés d'imiter au mieux le comportement du système. Les modèles sont en fait des entités imaginaires et abstraites que nous pouvons comprendre, analyser et dont nous pouvons prédire le comportement. Chacun des modèles considérés doit être *homogène* au système, c'est-à-dire comporter le même nombre d'entrées et de sorties, avoir le même espace des temps, etc. C'est à travers ces modèles que nous connaissons le système et l'analyse et la commande seront effectuées à partir du ou des modèles sélectionnés après l'étape de la modélisation. La modélisation d'un système ou processus peut se décomposer en trois étapes

- *Le choix de la structure* : A partir d'informations *a priori* et d'intuition sur les propriétés du système, il s'agit de proposer un ensemble de modèles parmi lesquels nous sélectionnerons le ou les modèles finaux. Souvent, cet ensemble de modèles, sera paramétré par un *vecteur de paramètre* $\mathbf{p} \in \mathbb{P} \subset \mathbb{R}^{n_p}$ et sera appelé *structure de modèle* et noté $M(\cdot)$. L'ensemble \mathbb{P} est appelé *ensemble admissible a priori pour les paramètres*. A chaque vecteur de paramètres \mathbf{p} est donc associé un modèle $M(\mathbf{p})$ de la structure. Si, quelle que soit l'entrée, la dépendance entre les sorties du modèle $\mathbf{y}_m(\mathbf{p}, t)$ et \mathbf{p} est linéaire, le modèle sera dit *linéaire par rapport aux paramètres*.
- *Le prélèvement des données* : Après avoir fixé les conditions expérimentales, par exemple à l'aide de techniques de *planification d'expériences*, les expériences sont effectuées sur le système et les *données expérimentales* sont alors prélevées puis rangées dans le vecteur $\mathbf{y} \in \mathbb{R}^{n_y}$. Le vecteur \mathbf{y} peut correspondre à n_y mesures scalaires faites à des temps donnés sur un système monovarié (une entrée et une sortie), mais des systèmes multivariés ou des processus statiques peuvent aussi être

pris en compte. Pour les conditions expérimentales utilisées, chaque modèle $M(\mathbf{p})$ génère un vecteur de sortie $\mathbf{y}_m(\mathbf{p})$ homogène au vecteur des données \mathbf{y} . C'est-à-dire que les deux vecteurs $\mathbf{y}_m(\mathbf{p})$ et \mathbf{y} ont la même dimension, et chacune des composantes a la même signification : si la $i^{\text{ème}}$ composante de \mathbf{y} a été prélevée au temps t sur la $k^{\text{ème}}$ sortie, la $i^{\text{ème}}$ composante de $\mathbf{y}_m(\mathbf{p})$ correspond à ce que génère le modèle sur la $k^{\text{ème}}$ sortie au temps t . La dépendance entre \mathbf{y}_m et les conditions expérimentales n'apparaîtra pas explicitement puisque les conditions expérimentales seront supposées ici fixées une fois pour toutes. L'erreur entre les données \mathbf{y} et la sortie modèle $\mathbf{y}_m(\mathbf{p})$ est définie par

$$\mathbf{e}(\mathbf{p}) = \mathbf{y} - \mathbf{y}_m(\mathbf{p}). \quad (1.1)$$

- *L'estimation* : Elle consiste à retenir un ou plusieurs modèles $M(\mathbf{p})$ générant des sorties modèles $\mathbf{y}_m(\mathbf{p})$ ressemblant suffisamment aux données expérimentales, c'est-à-dire tels que l'erreur $\mathbf{e}(\mathbf{p})$ soit suffisamment petite dans un sens à définir. Lorsque un seul modèle est cherché, cas de *l'estimation ponctuelle*, on cherche souvent le modèle générant des sorties ressemblant le plus possible (au sens d'un critère à définir) aux données expérimentales. *L'estimation ensembliste* consiste au contraire à caractériser l'ensemble \mathbb{S} des paramètres (ou des modèles de la structure) acceptables. Cette approche est intéressante dans le sens où elle donne une caractérisation de la validité de l'estimation et permet de prendre en compte avec rigueur les incertitudes sur les données. En prenant un vecteur $\hat{\mathbf{p}}$ au coeur de cet ensemble, nous obtenons une estimée ponctuelle du vecteur des paramètres et l'incertitude sur cette estimation est complètement caractérisée par l'ensemble \mathbb{S} . Dans cette thèse, nous allons faire uniquement de *l'estimation non linéaire* dans le sens où \mathbf{e} est une fonction non linéaire de \mathbf{p} .

Il est souvent pratique de raisonner comme si le système étudié était un élément $M(\mathbf{p}^*)$ de la structure choisie. Le vecteur inconnu \mathbf{p}^* est alors appelé la *vraie valeur* des paramètres. Sous cette hypothèse, nous définissons le *bruit de mesure* par $\mathbf{b} = \mathbf{y} - \mathbf{y}_m(\mathbf{p}^*)$.

L'approche habituellement suivie par les automaticiens consiste à sélectionner après identification un seul modèle de la structure choisie et à calculer la commande qui sera appliquée au système à partir de ce modèle. Il s'agit donc, là aussi, d'une approche ponctuelle. L'approche que nous considérerons dans cette thèse est complètement ensembliste : après estimation, un ensemble de modèles seront sélectionnés et une commande (ou un ensemble de commandes) garantissant la stabilité de tous les modèles retenus sera recherchée. Nous espérons ainsi, en appliquant cette même commande au

système, obtenir de sa part ce même comportement. Cette approche ensembliste possède d'évidentes propriétés de robustesse et c'est pourquoi nous parlerons de *commande robuste*. Par exemple, si nous faisons l'hypothèse que notre système est un élément $M(\mathbf{p}^*)$ de notre structure, et si les hypothèses faites sur les caractéristiques du bruit de mesure \mathbf{b} sont exactes, la ou les commandes obtenues stabiliseront à coup sûr le système $M(\mathbf{p}^*)$, ce qui n'est pas le cas dans l'approche ponctuelle, où aucune garantie ne peut être espérée.

1.3. Plan de la thèse

Dans cette thèse, nous présenterons les outils de l'approche ensembliste qui nous permettront ensuite de développer des algorithmes efficaces capables de donner des solutions nouvelles à certains problèmes ensemblistes et ponctuels d'estimation et de commande.

Dans le *chapitre 2*, nous décrirons quelques problèmes ensemblistes d'estimation et de commande en rappelant les notions qui leur sont rattachées. Un certain nombre d'entre eux sont des problèmes d'inversion ensembliste, où l'ensemble solution est défini comme l'inverse par une fonction vectorielle d'un ensemble simple (hypercube, ellipsoïde, ...). C'est le cas par exemple des problèmes d'estimation non linéaire à erreur bornée pour lesquels l'ensemble solution est appelé *ensemble de vraisemblance*. Dans un contexte probabiliste, nous présenterons le problème du calcul de *régions de confiance* d'une loi de probabilité pouvant correspondre par exemple à la loi *a posteriori* du vecteur de paramètres à estimer. En ce qui concerne la commande robuste, nous nous intéresserons au problème de la caractérisation de l'*ensemble de stabilité* d'une structure de modèles $M(\cdot)$ linéaires entrées-sorties mais non linéaires paramètres-sorties, l'ensemble de stabilité étant constitué des vecteurs \mathbf{p} tels que le modèle $M(\mathbf{p})$ soit asymptotiquement stable. Enfin, nous présenterons le problème de la détermination de l'ensemble des régulateurs d'une structure donnée qui stabilisent un ensemble de modèles d'une même structure (ou de structures différentes).

Les notions d'analyse par intervalles, indispensables au développement des algorithmes d'approximation ensemblistes seront présentées au *chapitre 3*. Les concepts de fonction et de test d'inclusion, supports de tout calcul sur les intervalles, seront présentés. Des exemples seront pris parmi les fonctions vectorielles intervenant dans les énoncés des problèmes que nous nous serons proposés de résoudre dans le chapitre 2.

Dans le *chapitre 4*, nous définirons les sous-pavages qui seront des ensembles

représentables en machine formés d'unions de pavés. Ces sous-pavages seront chargés d'approximer l'ensemble solution à déterminer. Nous présenterons différentes métriques dans l'ensemble des compacts afin de déterminer la topologie utilisée pour l'analyse des propriétés de convergence des algorithmes présentés. Nous définirons ensuite les intercompacts qui sont des ensembles de compacts représentables en machine formant un voisinage de diamètre aussi petit que désiré du compact considéré.

Nous donnerons, dans le *chapitre 5*, un algorithme général d'inversion ensembliste que nous appliquerons à la résolution de quelques problèmes tests d'estimation à erreur bornée, d'estimation de conditions initiales et d'analyse d'ensemble de stabilité proposés dans le chapitre 2. L'algorithme repose principalement sur les concepts d'analyse par intervalles, et d'encadrement d'un compact par deux sous-pavages (ce qui équivaut à l'appartenance à un intercompact). Les propriétés de convergence et de complexité seront établies.

Dans le *chapitre 6*, nous verrons comment les méthodes d'inversion ensembliste développées dans le chapitre 5 s'adaptent pour permettre la résolution de problèmes ensemblistes plus complexes. Il s'agit par exemple du problème de la détermination de la région de confiance d'une loi de probabilité quelconque ou du problème du calcul de l'ensemble des régulateurs d'une structure de commande qui stabilisent à coup sûr un modèle dont les paramètres sont inconnus mais bornés. Des problèmes tests proposés dans le chapitre 2 seront résolus afin de valider les algorithmes présentés.

Le *chapitre 7* sera consacré au problème de l'optimisation d'un critère linéaire ou non linéaire sur l'ensemble solution du problème ensembliste considéré. L'intérêt de cette approche est d'obtenir rapidement un renseignement garanti sur l'ensemble inconnu sans avoir à caractériser complètement l'ensemble auparavant, d'où un gain en temps de calcul considérable. L'approche utilisée restera semblable à celles utilisées pour la résolution des problèmes ensemblistes. L'algorithme d'optimisation fondé sur les tests d'inclusion et les sous-pavages, sera validé par quelques exemples tests et comparé à l'approche signomiale, qui était la seule méthode proposée dans le domaine de l'estimation à erreurs bornées pour l'optimisation garantie sur l'ensemble solution.

Dans le *chapitre 8*, nous allons résoudre des problèmes d'optimisation sans contrainte en partitionnant l'espace de recherche en un ensemble fini de pavés sur lesquels le critère à optimiser ne puisse admettre plus d'une solution locale. En combinant ce principe de partitionnement de l'espace de recherche, à base d'analyse par intervalles, avec des méthodes d'optimisation locale, nous proposerons un algorithme d'optimisation globale et garantie qui ne présente pas les inconvénients des deux approches : la lenteur de

convergence pour une très bonne précision que possèdent les algorithmes classiques d'optimisation à base d'analyse par intervalles, et le manque de garantie du résultat des méthodes locales. Un tel algorithme est possible grâce à l'utilisation du critère de Routh qui servira à tester la convexité du critère à optimiser sur le pavé considéré.

Une disquette contenant des programmes exécutables sur compatible IBM avec écran EGA-VGA correspondant à tous les exemples illustratifs est jointe à ce document.

CHAPITRE 2

LES PROBLÈMES ENSEMBLISTES

EN ESTIMATION ET EN COMMANDE

2.1. Introduction

Les problèmes ensemblistes admettent pour solution non pas un nombre ou un vecteur, comme c'est le cas dans la plupart des problèmes, dit ponctuels, abordés dans les sciences de l'ingénieur, mais un ensemble solution qu'il s'agit alors de caractériser. Une formulation ensembliste des problèmes ponctuels de la physique permet d'approcher la solution par un ensemble la contenant à coup sûr, avec une incertitude caractérisée de façon rigoureuse, et ceci contrairement aux approches ponctuelles classiques. Parmi tous les problèmes ensemblistes, une classe importante est celle des problèmes d'inversion ensembliste qui, par leur structure simple, nous permettront d'aboutir à un algorithme unique permettant de les résoudre tous. Certains problèmes ensemblistes ne peuvent cependant pas être mis sous la forme d'un problème d'inversion ensembliste, mais nous tenterons de les résoudre avec une approche similaire. Dans ce chapitre, nous définirons ce que nous entendons par problème d'inversion ensembliste. Après un rappel succinct sur l'estimation ponctuelle classique, nous proposerons différentes formulations ensemblistes de l'estimation, que ce soit dans un cadre déterministe (estimation à erreurs bornées) ou stochastique (estimation de régions de confiance). Dans le premier cas, nous avons à faire à un problème d'inversion ensembliste alors que le second nécessite une approche spécifique. Nous proposerons, toujours dans ce chapitre, une nouvelle approche ensembliste pour l'étude de la stabilité des systèmes, et le calcul de commandes robustes. Pour chacun des problèmes posés, nous illustrerons par des applications concrètes ce qu'apporte une formulation ensembliste par rapport aux approches ponctuelles classiques. Ces applications nous serviront par la suite à tester l'efficacité des algorithmes qui seront développés.

2.2. Notation vectorielle

La notation que nous utiliserons pour les équations et inéquations vectorielles sera inspirée

de celle utilisée par le logiciel de programmation mathématique MATLAB. L'intérêt de cette notation est de simplifier l'écriture en faisant abstraction des indices des composantes vectorielles et d'obtenir ainsi des expressions mathématiques plus concises et plus claires. Dans ce système de notation, les vecteurs \mathbf{v} et les fonctions \mathbf{f} à valeurs vectorielles sont imprimés en caractères gras. Les équations et les inégalités vectorielles sont à comprendre composante par composante. Par exemple si les vecteurs \mathbf{u} et \mathbf{v} sont définis par

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \text{ et } \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, \quad (2.1)$$

alors, pour $\otimes \in \{<, >, \leq, \geq, =\}$ nous avons

$$\mathbf{u} \otimes \mathbf{v} \Leftrightarrow \begin{cases} u_1 \otimes v_1 \\ u_2 \otimes v_2 \\ u_3 \otimes v_3 \end{cases}. \quad (2.2)$$

Pour montrer le danger engendré par la manipulation de tels opérateurs, notons par exemple que la contraposée de la proposition $\mathbf{u} \leq \mathbf{v}$ n'est pas $\mathbf{u} > \mathbf{v}$ car les deux propositions peuvent être simultanément fausses. Cette contraposée sera donc notée $\mathbf{u} \not\leq \mathbf{v}$. La relation \leq n'est donc qu'une relation d'ordre partielle sur l'ensemble des vecteurs. Les fonctions réelles usuelles telles que \sin , \exp , ..., lorsqu'elles ont pour arguments des vecteurs deviennent des fonctions vectorielles et seront écrites en caractères gras. Leur évaluation se fait composante par composante. Par exemple, nous avons

$$\mathbf{\sin}(\mathbf{u}) = \mathbf{\sin} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} \sin(u_1) \\ \sin(u_2) \\ \sin(u_3) \end{pmatrix}. \quad (2.3)$$

Le vecteur argument \mathbf{u} et le vecteur image $\mathbf{\sin}(\mathbf{u})$ ont donc nécessairement la même dimension. Dans la même logique, nous aurons

$$\mathbf{u} = \mathbf{v} + \mathbf{1} \Leftrightarrow \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \Leftrightarrow \begin{cases} u_1 = v_1 + 1 \\ u_2 = v_2 + 1 \\ u_3 = v_3 + 1. \end{cases} \quad (2.4)$$

2.3. Définition d'un problème d'inversion ensembliste

Les problèmes d'inversion ensembliste, que nous allons maintenant définir, possèdent une structure simple, et de ce fait, il sera possible de trouver un algorithme unique, présenté au chapitre 5, capable de les résoudre tous.

Problème d'inversion ensembliste : Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$ une fonction continue et différentiable et \mathbb{Y} un sous-ensemble fermé de \mathbb{R}^p . Résoudre le *problème d'inversion ensembliste* consiste à caractériser l'ensemble \mathbb{X} *image réciproque* par f de \mathbb{Y} , défini par

$$\mathbb{X} = f^{-1}(\mathbb{Y}) \triangleq \{\mathbf{x} \mid f(\mathbf{x}) \in \mathbb{Y}\}. \quad (2.5)$$

L'ensemble \mathbb{X} est appelé *ensemble solution* du problème d'inversion ensembliste. La fonction ensembliste f^{-1} est appelée *fonction image réciproque* de la fonction vectorielle f .

L'*image directe* $f(\mathbb{X})$ de \mathbb{X} par la fonction f est définie par

$$f(\mathbb{X}) \triangleq \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbb{X}\}. \quad (2.6)$$

L'ensemble $f(\mathbb{R}^n)$ forme une variété différentielle (c'est-à-dire une surface de \mathbb{R}^p) appelée *variété image*. Dans le cas où $p > n$, cette variété est presque sûrement de dimension n , alors que si $p \leq n$, elle est presque sûrement de dimension p . D'après les propriétés élémentaires de la théorie des ensembles, nous avons toujours

$$f(\mathbb{X}) \subset \mathbb{Y}. \quad (2.7)$$

Remarque : Le *problème de l'image directe* consiste à caractériser l'ensemble image $f(\mathbb{X})$ d'un ensemble donné \mathbb{X} simple (pavé, ellipsoïde...). Ce type de problèmes, qui semble a priori très difficile à résoudre de façon garantie, peut se poser dans le domaine de l'automatique, par exemple pour la simulation d'un modèle ou la caractérisation du domaine atteignable d'un système. Il est toutefois possible d'obtenir un résultat satisfaisant en utilisant l'arithmétique stochastique (Vignes, 1993).

Dans de très nombreux problèmes pratiques, l'ensemble \mathbb{Y} peut se définir par un ensemble d'inégalités

$$\mathbb{Y} = \{\mathbf{y} \mid \mathbf{g}(\mathbf{y}) \leq \mathbf{0}\}. \quad (2.8)$$

Nous avons alors les équivalences

$$\mathbf{x} \in \mathbb{X} \Leftrightarrow \mathbf{f}(\mathbf{x}) \in \mathbb{Y} \Leftrightarrow \mathbf{g} \circ \mathbf{f}(\mathbf{x}) \leq \mathbf{0}. \quad (2.9)$$

En posant $\mathbf{h} = \mathbf{g} \circ \mathbf{f}(\mathbf{x})$, l'ensemble \mathbb{X} peut être décrit par l'ensemble d'inégalités

$$\mathbb{X} = \{ \mathbf{x} \mid \mathbf{h}(\mathbf{x}) \leq \mathbf{0} \}. \quad (2.10)$$

Si les inéquations définissant l'ensemble solution \mathbb{X} sont connues, nous ne connaissons pas pour autant les caractéristiques élémentaires de ce dernier (sa forme, son volume, s'il est vide, etc.). En effet, si les inégalités définissant \mathbb{Y} sont généralement très simples, celles définissant \mathbb{X} sont en général très complexes. Résoudre un problème d'inversion ensembliste revient donc très souvent à résoudre le *problème de la caractérisation d'un ensemble défini par des inégalités*. De même, il est aisé de comprendre que tout problème de caractérisation d'ensemble défini par des inégalités peut se mettre sous la forme d'un problème d'inversion ensembliste, ces deux problèmes sont donc équivalents. Il arrive que la fonction \mathbf{h} soit linéaire. C'est le cas par exemple si \mathbf{g} et \mathbf{f} sont toutes les deux linéaires, mais cela peut arriver (exceptionnellement) aussi lorsque la non linéarité de \mathbf{f} compense celle de \mathbf{g} . Dans les deux cas, nous dirons que le problème d'inversion ensembliste est linéaire. Dans ce cas, si \mathbb{Y} est un polyèdre, l'ensemble \mathbb{X} est aussi un polyèdre et toutes ses caractéristiques telles que son volume, le plus petit cube qui le contient, etc., peuvent être calculées analytiquement. Dans le cas où \mathbf{h} est non linéaire, il est parfois possible par un simple changement de variable non linéaire sur \mathbf{x} de linéariser le problème, et de simplifier ainsi sa structure. Mais dans la plupart des cas on ne peut pas faire disparaître la non-linéarité de \mathbf{h} et l'approche choisie se doit d'être non-linéaire. Les techniques à base d'analyse par intervalles que nous allons développer par la suite permettront d'approcher l'ensemble \mathbb{X} par des ensembles plus simples formés par des unions de pavés et ceci même lorsque le problème est non linéaire. Nous pourrons ainsi en déduire des propriétés sur l'ensemble solution \mathbb{X} comme par exemple savoir si \mathbb{X} est vide ou connexe, calculer son volume, trouver le plus petit pavé qui le contient, dessiner une projection de \mathbb{X} sur un plan, évaluer sa forme, etc. Donnons maintenant les bases indispensables de l'estimation ponctuelle classique avant de formuler les problèmes d'estimation ensembliste. Nous proposerons ensuite une approche ensembliste des problèmes de stabilité et de commande robuste.

2.4. Estimation ponctuelle

Supposons que nous disposions d'une structure de modèles $M(\cdot)$ pour le système auquel nous nous intéressons. A chaque vecteur de paramètres \mathbf{p} , nous pouvons donc associer un modèle particulier $M(\mathbf{p})$ de cette structure. A partir de ce modèle, et connaissant les conditions expérimentales appliquées au système qui ont permis de générer le vecteur des données \mathbf{y} , nous pouvons calculer, sous ces mêmes conditions expérimentales, la sortie modèle $\mathbf{y}_m(\mathbf{p})$ homogène aux données expérimentales \mathbf{y} . Seule la fonction $\mathbf{y}_m(\mathbf{p})$ et le vecteur de mesure \mathbf{y} seront désormais utiles pour l'estimation du vecteur de paramètre \mathbf{p} , et nous pouvons à présent faire abstraction de toute la complexité du modèle. En général, la dimension de $\mathbf{y}_m(\mathbf{p})$ et donc de \mathbf{y} est supérieure à celle de \mathbf{p} . Dans le cas de l'estimation ponctuelle, le problème posé par l'estimation est de trouver un \mathbf{p} tel que $\mathbf{y}_m(\mathbf{p}) \cong \mathbf{y}$, et nous comprenons pourquoi ce type de problèmes est souvent appelé *problème inverse* car notre idéal serait de trouver l'image inverse de \mathbf{y} par la fonction \mathbf{y}_m . Dans les approches utilisées pour résoudre les problèmes inverses, nous pouvons distinguer principalement les méthodes d'estimation projectives et les méthodes probabilistes qui rajoutent des hypothèses sur les propriétés statistiques de l'erreur $\mathbf{e}(\mathbf{p}) \triangleq \mathbf{y} - \mathbf{y}_m(\mathbf{p})$.

Dans l'estimation projective, le vecteur \mathbf{p} qui minimise la norme du vecteur erreur $\mathbf{e}(\mathbf{p})$ est recherché. Géométriquement, cela revient à projeter orthogonalement le vecteur \mathbf{y} sur la variété image de la fonction \mathbf{y}_m et à prendre le \mathbf{p} correspondant à l'antécédent par \mathbf{y}_m du projeté. Lorsque cette norme est euclidienne, cette estimation est appelée *estimation au sens des moindres carrés*.

Les méthodes probabilistes, plus sophistiquées, consistent à choisir plus ou moins arbitrairement une loi de probabilité *a priori* pour le vecteur de paramètre \mathbf{p} à estimer, une loi de probabilité *a priori* pour l'erreur $\mathbf{e}(\mathbf{p})$, afin de s'en servir pour l'estimation ponctuelle de \mathbf{p} . Cette approche renferme par exemple l'estimation bayésienne qui nous génère une loi *a posteriori* sur \mathbf{p} connaissant une loi *a priori* sur \mathbf{p} . Elle contient également l'estimation au sens du maximum de vraisemblance, à condition de prendre pour les paramètres une loi *a priori* uniforme sur l'ensemble \mathbb{P} admissible *a priori*.

L'intérêt d'avoir une estimée ponctuelle $\hat{\mathbf{p}}$ pour le vecteur de paramètre \mathbf{p} , est d'avoir un modèle unique $M(\hat{\mathbf{p}})$ ce qui simplifie de beaucoup l'exploitation ultérieure. Nous pouvons alors raisonner comme si le système à modéliser était complètement connu, même si nous savons que nous ne disposons que d'une approximation de ce système. Mais il arrive fréquemment que le modèle $M(\hat{\mathbf{p}})$ ne nous satisfasse pas car il a un comportement incohérent

avec d'autres mesures faites sur le système, ou avec l'intuition que nous en avons. Dans ce cas là, faut-il recommencer une estimation avec un autre critère ? faire d'autres hypothèses sur le bruit ? ou tout simplement rejeter la structure de modèle choisie ? Une approche par estimation ponctuelle ne permet pas de répondre à de nombreuses questions sur la qualité de l'estimation car elle ne permet pas de caractériser rigoureusement l'incertitude faite sur l'estimation, sauf dans certains cas particuliers comme par exemple dans la situation où le modèle est linéaire par rapport aux paramètres et où le bruit est gaussien et additif sur la sortie.

2.5. Estimation ensembliste à erreurs bornées

2.5.1. Introduction à l'estimation à erreurs bornées

L'estimation à erreurs bornées (voir par exemple (Walter, 1990) et les nombreux articles sur le sujet dans (Bányász et Keviczky, 1991)), s'intéresse à la caractérisation de l'ensemble \mathbb{S} des vecteurs de paramètres $\mathbf{p} \in \mathbb{P}$ tel que le vecteur des sorties du modèle $\mathbf{y}_m(\mathbf{p})$ soit compatible, dans un sens à définir, avec le vecteur des mesures \mathbf{y} . Dans ce but, définissons, après collecte des données, l'ensemble $\mathbb{Y} \subset \mathbb{R}^{n_y}$ des vecteurs des sorties du modèle acceptables. Ainsi, si le vecteur de paramètres \mathbf{p} est tel que $\mathbf{y}_m(\mathbf{p})$ soit acceptable, alors, \mathbf{p} sera un *vecteur de paramètres acceptable*. L'ensemble \mathbb{S} des vecteurs de paramètres acceptables est appelé *ensemble de vraisemblance* et peut être défini par

$$\mathbb{S} \triangleq \{\mathbf{p} \in \mathbb{P} \mid \mathbf{y}_m(\mathbf{p}) \in \mathbb{Y}\}. \quad (2.11)$$

L'approche habituellement suivie en estimation à erreurs bornées est de définir *a priori*, c'est-à-dire avant même avoir prélevé les mesures, l'ensemble des *erreurs acceptables* \mathbb{E} . Ainsi, \mathbf{p} sera un vecteur de paramètres acceptable si l'erreur $\mathbf{e}(\mathbf{p}) = \mathbf{y} - \mathbf{y}_m(\mathbf{p})$ entre les données \mathbf{y} et le vecteur des sorties du modèle $\mathbf{y}_m(\mathbf{p})$ appartient à $\mathbb{E} \subset \mathbb{R}^{n_y}$. Dans ce cas, l'ensemble \mathbb{Y} est donné par

$$\mathbb{Y} = \mathbf{y} - \mathbb{E}. \quad (2.12)$$

L'ensemble de vraisemblance est maintenant défini par

$$\mathbb{S} = \{\mathbf{p} \in \mathbb{P} \mid \mathbf{e}(\mathbf{p}) \in \mathbb{E}\}. \quad (2.13)$$

Pour tout $\mathbf{p} \in \mathbb{S}$, il existe donc un vecteur erreur acceptable $\mathbf{e} \in \mathbb{E}$ tel que

$$\mathbf{y} = \mathbf{y}_m(\mathbf{p}) + \mathbf{e}. \quad (2.14)$$

Bien que l'estimation à erreurs bornées puisse être définie en termes purement déterministes, elle peut aussi recevoir une interprétation probabiliste. Si \mathbb{P} et \mathbb{E} sont respectivement les supports des lois des densités de probabilité *a priori* pour les vecteurs aléatoires \mathbf{p} et $\mathbf{e}(\mathbf{p})$ alors, en utilisant la règle de Bayes, \mathbb{S} devient le support de la loi de probabilité *a posteriori* du vecteur aléatoire \mathbf{p} . Si $\mathbb{P} = \mathbb{R}^n$ alors \mathbb{S} est l'ensemble des vecteurs \mathbf{p} tel que la vraisemblance des données est différente de zéro.

Dans la suite, nous faisons l'hypothèse que \mathbb{P} , \mathbb{E} , et \mathbb{Y} peuvent être définis par des ensembles finis d'inégalités. Le problème d'estimation peut toujours être reformulé de façon à insérer les inégalités définissant \mathbb{P} parmi celles définissant \mathbb{E} ou \mathbb{Y} . Ainsi, \mathbb{S} peut simplement se définir en termes ensemblistes par

$$\mathbb{S} = \mathbf{y}_m^{-1}(\mathbf{y} - \mathbb{E}) = \mathbf{y}_m^{-1}(\mathbb{Y}) = \mathbf{e}^{-1}(\mathbb{E}), \quad (2.15)$$

où \mathbf{e}^{-1} et \mathbf{y}_m^{-1} sont les fonctions images réciproques (dans le sens de la théorie des ensembles) des fonctions \mathbf{e} et \mathbf{y}_m . Le problème à résoudre apparaît donc comme un problème d'inversion ensembliste. L'ensemble de vraisemblance \mathbb{S} peut aussi se définir, en termes d'inégalités en procédant comme pour (2.10) mais en tenant compte des inégalités définissant \mathbb{P} , par

$$\mathbb{S} = \{\mathbf{p} \mid \mathbf{h}(\mathbf{p}) \leq \mathbf{0}\}, \quad (2.16)$$

Le lecteur intéressé par une bibliographie sur l'estimation à erreurs bornées peut se référer aux articles généraux (Kurzhanski et Vályi, 1991; Milanese et Vicino, 1991a; Norton, 1987a; Walter et Piet-Lahanier, 1990). Les méthodes utilisées pour caractériser \mathbb{S} dépendent du type de la fonction $\mathbf{h}(\mathbf{p})$.

- Quand $\mathbf{h}(\mathbf{p})$ est affine (ou linéaire) en \mathbf{p} , et \mathbb{Y} est un polyèdre, le problème de l'estimation garantie peut être considéré comme résolu. Des méthodes ont été développées pour caractériser \mathbb{S} exactement et récursivement sur la dimension du vecteur des données \mathbf{y} (Broman et Shensa, 1990; Mo et Norton, 1990; Walter et Piet-Lahanier, 1989). Des algorithmes rapides calculant des ellipsoïdes qui contiennent \mathbb{S} à coup sûr peuvent être trouvés par exemple dans (Belforte, Bona et Cerone, 1990; Fogel et Huang, 1982; Kurzhanski et Vályi, 1991; Pronzato,

Walter et Piet-Lahanier, 1989).

- Dans le cas où $\mathbf{h}(\mathbf{p})$ est non-linéaire en \mathbf{p} , la résolution du problème de l'estimation à erreurs bornées est bien moins avancée. Les techniques à base d'exploration aléatoire (Piet-Lahanier et Walter, 1990) ne fournissent pas un résultat garanti. Pour une certaine classe de modèles, il a été montré (voir par exemple Clément et Gentil, 1990 ; Norton, 1987b) comment calculer une union de polyèdres convexes contenant \mathbb{S} à coup sûr. Ces polyèdres convexes peuvent être caractérisés exactement ou enfermés dans une union d'ellipsoïdes ou de pavés.

Quelle que soit la fonction \mathbf{h} , le calcul du plus petit pavé contenant \mathbb{S} peut être effectué par la résolution de $2 n_p$ problèmes de programmation mathématique. Chacun de ces problèmes d'optimisation correspond à la maximisation ou la minimisation d'une composante de \mathbf{p} sous les $\dim(\mathbf{h})$ contraintes inégalités définissant \mathbb{S} . Quand \mathbf{h} est affine en \mathbf{p} , la résolution de ces problèmes d'optimisation peut être faite par n'importe quelle méthode de programmation linéaire telle que la méthode du simplexe de Dantzig ou l'algorithme de Karmarkar. Dans le cas général, des méthodes d'optimisation globale doivent être utilisées. Parmi toutes les méthodes d'optimisation globale disponibles (voir par exemple (Dixon et Szego, 1975, 1978 ; Mockus, 1989 ; Zhigljavsky, 1991)), seules les méthodes déterministes (Horst et Tuy, 1990 ; Ratschek et Rokne, 1988) peuvent être utilisées car les méthodes stochastiques convergent seulement en probabilité, de sorte qu'aucune garantie sur le résultat ne peut être obtenue. Pour un grand nombre de problèmes d'intérêt pratique (tels que l'estimation de paramètres d'un modèle ARMA ou d'un modèle d'état linéaire à temps discret), la programmation signomiale peut être utilisée (Milanese and Vicino, 1991b).

Toutes les méthodes disponibles qui permettent une caractérisation garantie de l'ensemble de vraisemblance \mathbb{S} dans le cas non-linéaire se limitent donc à calculer un ensemble de forme simple (pavé polyèdre ou ellipsoïde) contenant \mathbb{S} de façon garantie. Nous proposerons dans la suite une approche nouvelle, fondée sur l'analyse par intervalles et sur le concept d'inversion ensembliste, qui permet d'obtenir une description plus détaillée de \mathbb{S} . Cette approche est similaire à celle développée récemment et indépendamment par Moore (1992).

Nous allons maintenant présenter quatre problèmes tests d'estimation à erreurs bornées qui seront utilisés à travers ce mémoire pour illustrer les différentes notions présentées. Le premier considère l'approximation d'une fonction sur un intervalle fini par une fonction appartenant à une classe fixée. Pour ce problème, une formulation identique à celle utilisée pour l'estimation d'ensemble de vraisemblance peut être utilisée. A travers cet exemple, nous montrerons que les techniques permettant de résoudre les problèmes d'inversion

ensembliste peuvent s'appliquer même si la dimension de l'espace image est infinie. Le deuxième problème test que nous allons présenter est de Milanese et Vicino (1991b), sur lequel, nous avons gelé deux paramètres afin de pouvoir donner une interprétation graphique de la méthode utilisée. Le troisième problème test reprend intégralement le problème étudié par Milanese et Vicino (1991b), qui ont estimé le plus petit pavé contenant l'ensemble de vraisemblance correspondant \mathbb{S} à l'aide d'une technique de programmation signomiale développée par Falk (1973). Nous comparerons alors les résultats obtenus par la méthode signomiale avec ceux obtenus par l'approche utilisant l'analyse par intervalles. Le quatrième problème test consiste en un problème d'estimation à erreurs bornées des conditions initiales d'un modèle d'état non linéaire, discret et à état non mesuré. Cet exemple illustrera le domaine d'application très vaste sur lequel s'applique notre méthode.

2.5.2. Problème des exponentielles approximantes

Il s'agit de trouver l'ensemble \mathbb{S} de toutes les fonctions exponentielles de la forme

$$y_m(\mathbf{p}, t) = p_1 \exp(p_2 t). \quad (2.17)$$

approximant suffisamment bien la parabole d'équation $y(t) = t^2 + 2t + 1$ sur l'intervalle $[0, 1]$ dans le sens où

$$|e(\mathbf{p}, t)| = |y(t) - y_m(\mathbf{p}, t)| \leq 1, \forall t \in [0, 1]. \quad (2.18)$$

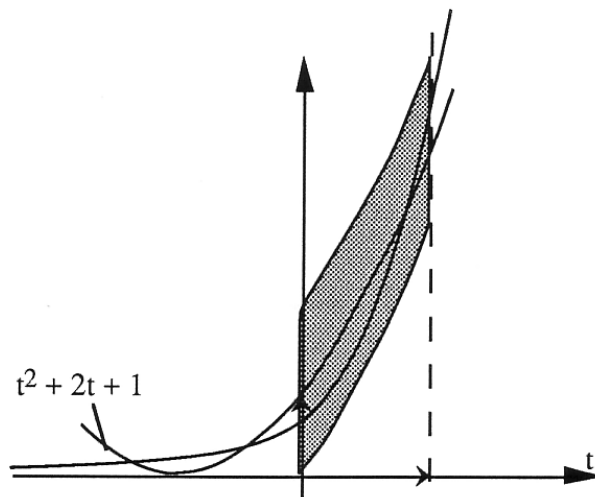


Fig. 2.1 Exponentielle acceptable comprise entre les deux paraboles sur $[0, 1]$.

La figure 2.1 représente une exponentielle acceptable. Ainsi, le vecteur \mathbf{p} est acceptable si

$$\forall t \in [0, 1], -1 \leq e(\mathbf{p}, t) \leq 1 \Leftrightarrow \begin{cases} \min_{0 \leq t \leq 1} y(t) - y_m(\mathbf{p}, t) \geq -1 \\ \max_{0 \leq t \leq 1} y(t) - y_m(\mathbf{p}, t) \leq 1. \end{cases} \quad (2.19)$$

En posant

$$e(\mathbf{p}) = \begin{pmatrix} \min_{0 \leq t \leq 1} \{t^2 + 2t + 1 - p_1 \exp(p_2 t)\} \\ \max_{0 \leq t \leq 1} \{t^2 + 2t + 1 - p_1 \exp(p_2 t)\} \end{pmatrix}, \quad (2.20)$$

l'ensemble \mathbb{S} des vecteurs \mathbf{p} acceptables est donné par

$$\mathbb{S} = \mathbf{e}^{-1}([-1, 1]) \quad (2.21)$$

où $\mathbf{1}$ représente le vecteur de dimension 2 dont les deux composantes sont égales à 1. Nous limiterons nos recherches au pavé $\mathbb{P} = [0, 5] \times [0, 5]$.

2.5.3. Problème de Milanese et Vicino simplifié

Considérons le problème d'estimation à deux paramètres inspiré de Milanese et Vicino (1991b) dans lequel le vecteur des données

$$\mathbf{y} = (7.39, 4.09, 1.74, 0.097, -2.55, -2.69, -2.07, -1.44, -0.98, -0.66)^T \quad (2.22)$$

correspond à dix mesures scalaires prélevées aux instants

$$\mathbf{t} = (0.75, 1.5, 2.25, 3, 6, 9, 13, 17, 21, 25)^T \quad (2.23)$$

sur un système sans entrées avec une sortie scalaire et évoluant dans le temps. La structure de modèle $M(\cdot)$ choisie est telle qu'au temps t , la sortie scalaire générée par le modèle $M(\mathbf{p})$ associé au paramètre \mathbf{p} inconnu est donnée par

$$y_m(\mathbf{p}, t) = 20 \exp(-p_1 t) - 8 \exp(-p_2 t). \quad (2.24)$$

Le vecteur des sorties $\mathbf{y}_m(\mathbf{p})$ du modèle $M(\mathbf{p})$ associé aux temps des mesures \mathbf{t} est donc

$$\mathbf{y}_m(\mathbf{p}) = 20 \exp(-p_1 t) - 8 \exp(-p_2 t), \quad (2.25)$$

où la $i^{\text{ème}}$ composante du vecteur $\mathbf{y}_m(\mathbf{p})$ est calculée pour la $i^{\text{ème}}$ composante de \mathbf{t} .
L'ensemble des vecteurs \mathbf{p} acceptables *a priori* est le pavé

$$\mathbb{P} = [0, 1] \times [0, 1]. \quad (2.26)$$

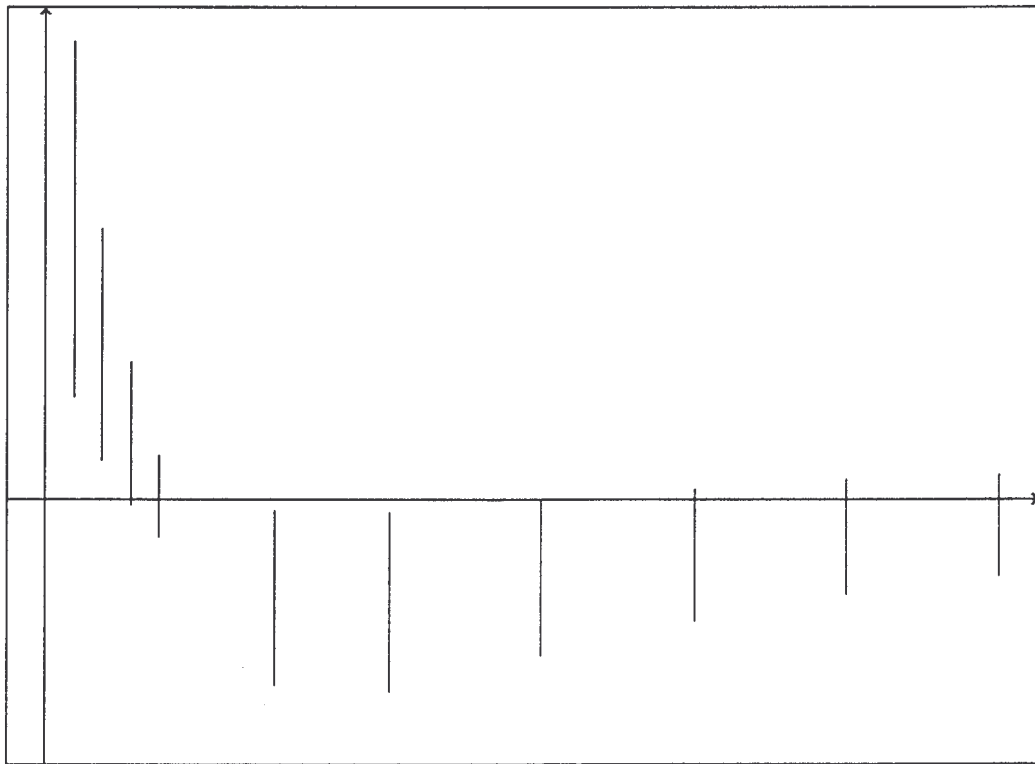


Fig. 2.2 Données et barres d'erreurs dans le problème de Milanese et Vicino simplifié dans le cadre $[-1, 26] \times [-7, 13]$ de l'espace $(t, y(t))$.

Pour être acceptable *a posteriori*, le vecteur \mathbf{p} doit être tel que l'erreur $\mathbf{e}(\mathbf{p}) = \mathbf{y} - \mathbf{y}_m(\mathbf{p})$ satisfasse

$$\mathbf{e}(\mathbf{p}) \in \mathbb{E} = \{\mathbf{e} \mid -\mathbf{e}_{\max} \leq \mathbf{e} \leq \mathbf{e}_{\max}\}, \quad (2.27)$$

avec

$$\mathbf{e}_{\max} = 0.5 |\mathbf{y}| + \mathbf{1}, \quad (2.28)$$

où $|\mathbf{y}|$ est vecteur de dimension n_y qui a pour composante la valeur absolue des composantes du vecteur \mathbf{y} et $\mathbf{1}$ représente le vecteur de dimension n_y qui a toutes ses composantes égales à 1. Les bornes de l'erreur ont été choisies dix fois plus grandes que dans (Milanese et

Vicino, 1991b) parce que l'ensemble solution \mathbb{S} obtenu avec l'erreur originale était trop plat pour donner une illustration claire de la méthode d'inversion ensembliste exposée par la suite. La figure 2.2 représente les données. Les barres d'erreurs indiquent l'incertitude associée à chaque mesure. L'ensemble solution \mathbb{S} à déterminer est l'ensemble des valeurs de \mathbf{p} tels que la sortie $y_m(\mathbf{p}, t)$ du modèle $M(\mathbf{p})$ passe à travers chacune des barres. Le problème de la caractérisation de l'ensemble $\mathbb{S} = \mathbf{e}^{-1}(\mathbb{E})$ des vecteurs de paramètres acceptables sera résolue au chapitre 5.

2.5.4. Problème de Milanese et Vicino

Considérons maintenant le problème proposé par Milanese et Vicino (1991b) sans le simplifier. Supposons à nouveau qu'aux temps (2.23), les données (2.22) aient été enregistrées sur le système à modéliser qui ne comporte qu'une sortie et aucune entrée. La structure de modèle $M(\cdot)$ choisie est telle que la sortie du modèle $M(\mathbf{p})$ associée au paramètre \mathbf{p} s'exprime par $y_m(\mathbf{p}, t) = p_1 \exp(-p_2 t) + p_3 \exp(-p_4 t)$. Le vecteur de toutes les sorties du modèle $M(\mathbf{p})$ correspondant aux temps de mesure \mathbf{t} est donc

$$\mathbf{y}_m(\mathbf{p}) = p_1 \exp(-p_2 \mathbf{t}) + p_3 \exp(-p_4 \mathbf{t}). \quad (2.29)$$

Comme dans Milanese et Vicinino (1991b), l'ensemble des erreurs acceptables \mathbb{E} est donné par

$$\mathbb{E} = \{ \mathbf{e} \mid -\mathbf{e}_{\max} \leq \mathbf{e} \leq \mathbf{e}_{\max} \}, \quad (2.30)$$

avec

$$\mathbf{e}_{\max} = 0.05 |\mathbf{y}| + 0.1 * \mathbf{1}. \quad (2.31)$$

L'ensemble des paramètres acceptables *a priori* est le pavé

$$\mathbb{P} = [2, 60] \times [0, 1] \times [-30, -1] \times [0, 0.5]. \quad (2.32)$$

La figure 2.3 représente les données. Les barres erreurs représentent l'incertitude associée à chacune d'entre elles. L'ensemble solution $\mathbb{S} = \mathbf{e}^{-1}(\mathbb{E})$ à déterminer est l'ensemble des valeurs de \mathbf{p} telles que la sortie $y_m(\mathbf{p}, t)$ du modèle $M(\mathbf{p})$ passe à travers chacune de ces barres.

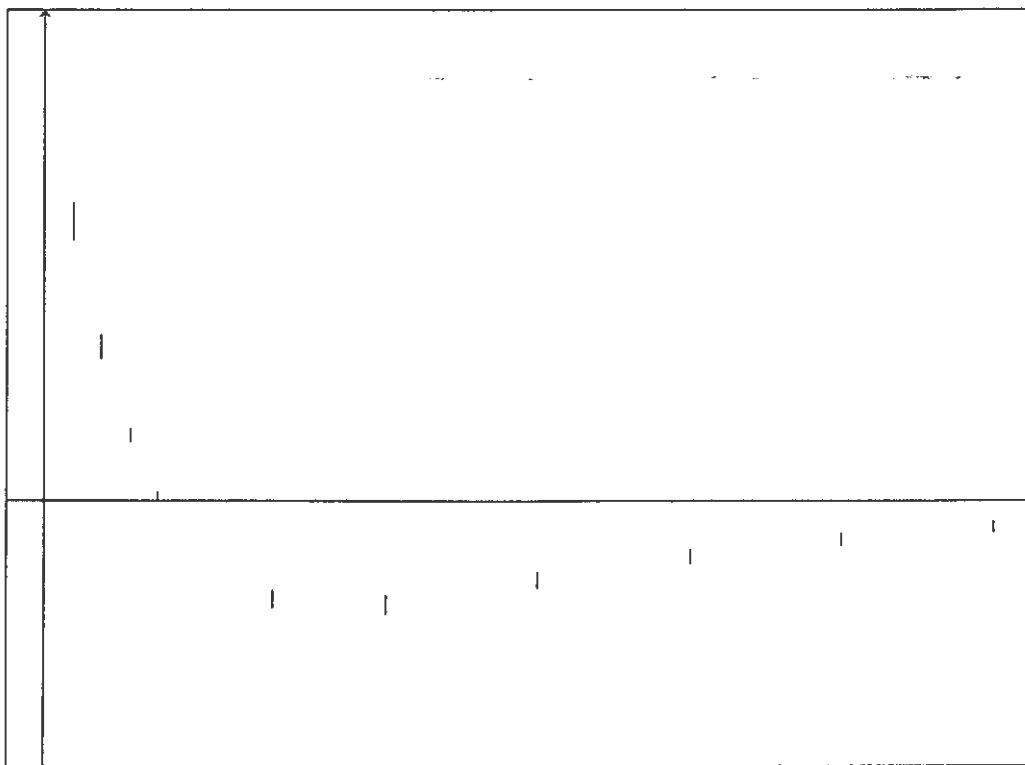


Fig. 2.3 Données et barres d'erreurs du problème d'estimation à erreurs bornées de Milanese et Vicino dans le cadre $[-1, 26] \times [-7, 13]$ de l'espace $(t, y(t))$.

Milanese et Vicino (1991b) se sont intéressés à trouver le plus petit pavé qui contienne \mathcal{S} . Pour cela, ils ont utilisé l'algorithme d'optimisation signomiale proposé par Falk (1973), et qui sera rappelé au chapitre sept. Les résultats qu'ils ont obtenus seront donnés à cette occasion.

2.5.5. Problème des conditions initiales

Considérons le modèle dont l'évolution dynamique est régie par l'équation d'état à temps discret

$$\begin{cases} x_1(k+1) = \cos(x_1(k) x_2(k)) \\ x_2(k+1) = 3 x_1(k) - \sin(x_2(k)) \\ y_m(k) = x_1^2(k) - x_2(k) \end{cases} \quad (2.33)$$

Les conditions initiales, notées

$$\mathbf{x}(0) = \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \mathbf{p}, \quad (2.34)$$

sont à estimer. Nous avons à notre disposition dix mesures $y(k)$ ($k = 0, \dots, 9$) qui ont été générées en simulant le modèle avec pour vraie valeur des paramètres $\mathbf{x}(0) = (2 \ 1)^T$ puis en bruitant le résultat par ajout d'une erreur aléatoire ϵ sur chaque $y(k)$, satisfaisant $-0.5 \leq \epsilon \leq 0.5$. L'ensemble des mesures bruitées ainsi obtenues est donné par le vecteur

$$\mathbf{y} = (y(0), \dots, y(9))^T = (3, -5, 0.6, 2.2, -3.8, -1.4, 0.4, -1.2, -1.8, 2.6)^T. \quad (2.35)$$

L'ensemble \mathbb{S} à caractériser est formé de tous les $\mathbf{x}(0) = \mathbf{p}$ tels que $\mathbf{e}(\mathbf{p}) \in \mathbb{E}$ avec

$$\mathbb{E} = 0.5 [-1, 1], \quad (2.36)$$

où $[-1, 1]$ représente l'hypercube parallèle aux axes centré en l'origine et de côté 2. La fonction erreur $\mathbf{e}(\mathbf{p})$ peut être donnée par une expression formelle très complexe, mais peut être décrite simplement par l'algorithme suivant, écrit en pseudo PASCAL.

```

x1(0) := p1 ; x2(0) := p2 ;
For k := 0 to 9 do
  begin
    ym(k) := x12(k) - x2(k) ;
    e(k) := y(k) - ym(k) ;
    x1(k+1) := cos(x1(k) * x2(k)) ;
    x2(k+1) := 3 x1(k) - sin(x2(k)) ;
  end ;

```

(2.37)

Le vecteur $\mathbf{e}(\mathbf{p})$ est alors donné par

$$\mathbf{e}(\mathbf{p}) = \begin{pmatrix} e(0) \\ \dots \\ e(9) \end{pmatrix}. \quad (2.38)$$

L'ensemble des conditions initiales à caractériser est donc $\mathbb{S} = \mathbf{e}^{-1}(\mathbb{E})$.

2.6. Régions de confiance Bayésiennes

2.6.1. Introduction

Soit $\pi_{\text{prior}}(\mathbf{p})$ la densité de probabilité *a priori* du vecteur des paramètres \mathbf{p} à estimer. Soit $\pi_{\mathbf{y}}(\mathbf{y} \mid \mathbf{p})$, la vraisemblance du vecteur des données \mathbf{y} lorsque le vecteur des paramètres prend la valeur \mathbf{p} . Cette vraisemblance est calculée à partir de la connaissance d'une loi de probabilité pour le bruit corrompant les données (voir par exemple Walter et Pronzato, 1994). L'expression analytique de la densité *a posteriori* pour les paramètres compte tenu des données \mathbf{y} est directement obtenue par la règle de Bayes :

$$\pi_{\text{post}}(\mathbf{p} \mid \mathbf{y}) = \frac{\pi_{\mathbf{y}}(\mathbf{y} \mid \mathbf{p}) * \pi_{\text{prior}}(\mathbf{p})}{\pi_{\mathbf{y}}(\mathbf{y})} \quad (2.39)$$

où $\pi_{\mathbf{y}}(\mathbf{y})$ est la loi marginale pour \mathbf{y} de la loi du couple (\mathbf{p}, \mathbf{y}) . Comme $\pi_{\mathbf{y}}(\mathbf{y})$ ne dépend pas de \mathbf{p} , il a le rôle d'un facteur de normalisation choisi de telle façon que $\pi_{\text{post}}(\mathbf{p} \mid \mathbf{y})$ soit une densité de probabilité. En général, du fait de la complexité de son expression formelle, cette densité n'est pas directement exploitable. L'approche usuelle (voir par exemple Norton 1986) consiste donc à déduire de la loi *a posteriori* $\pi_{\text{post}}(\mathbf{p} \mid \mathbf{y})$, une estimée ponctuelle pour le vecteur aléatoire \mathbf{p} , par exemple celle qui maximise π_{post} (estimation du maximum à posteriori). Une telle approche fait perdre une grande quantité d'information sur les paramètres par rapport à la connaissance de $\pi_{\text{post}}(\mathbf{p} \mid \mathbf{y})$. Nous ignorons en particulier avec quelle incertitude \mathbf{p} est estimé. De plus, cette estimée est trouvée suivant un critère plus ou moins arbitraire, et nous ne savons pas toujours quelle validité accorder au résultat ainsi trouvé.

Une approche ensembliste naturelle consiste à chercher l'ensemble \mathbb{S}_{α} de plus petit volume qui contient \mathbf{p} avec une probabilité d'au moins α par la loi π_{post} . L'ensemble \mathbb{S}_{α} est appelé *région de confiance* de probabilité α . Il contient quant à lui des informations intéressantes sur l'incertitude avec laquelle \mathbf{p} est connu. Nous nous placerons dans la situation générique où \mathbb{S}_{α} est unique. Un cas dégénéré se produit par exemple si π_{post} est une loi uniforme et α quelconque. Dans la suite, π_{post} sera noté tout simplement π . Si $\Pi(\mathbb{A})$ est la probabilité de l'ensemble \mathbb{A} par la loi π , \mathbb{S}_{α} peut aussi être défini par

$$\begin{cases} \mathbb{S}_{\alpha} = \pi^{-1}([s_{\alpha}, +\infty[), \\ \Pi(\mathbb{S}_{\alpha}) = \alpha. \end{cases} \quad (2.40)$$

Il s'agit là d'un système d'équation ensembliste très couplé puisque la première équation

dépend d'un seuil s_α fixé par la deuxième équation qui dépend elle-même de l'ensemble \mathbb{S}_α qui doit satisfaire la première équation. La figure 2.4 illustre cette définition et montre que seuls les paramètres de densité de probabilité maximum appartiennent à \mathbb{S}_α .

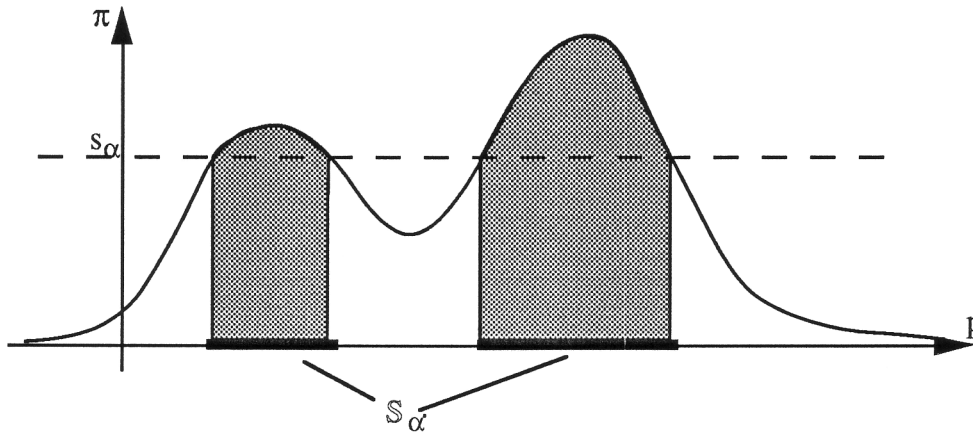


Fig. 2.4 Illustration de la définition d'une région de confiance.

Le problème de la caractérisation de \mathbb{S}_α , bien que proche d'un problème d'inversion ensembliste, n'en est pas vraiment un car l'ensemble à inverser est inconnu. Il dépend d'un seuil inconnu s_α qu'il faut calculer pour que la probabilité *a posteriori* de l'ensemble \mathbb{S}_α soit α . Dans le cas où π est gaussienne, les régions de confiance sont des ellipsoïdes dont nous savons déterminer l'équation. Dans le cas général, les méthodes classiques ne permettent pas d'exprimer sous une forme exploitable la région de confiance de probabilité α associée à une loi de probabilité. Les outils basés sur l'analyse par intervalles proposés au chapitre 3 nous permettront de développer un algorithme qui générera un ensemble exploitable approchant convenablement l'ensemble solution \mathbb{S}_α . L'exemple suivant illustrera le fonctionnement et l'efficacité de la méthode.

2.6.2. Problème de la région de confiance triangulaire

Supposons qu'après application de la règle de Bayes l'expression analytique de la densité *a posteriori* pour un vecteur de paramètres \mathbf{p} de dimension 2 obtenue soit donnée par

$$\pi(\mathbf{p}) = \begin{cases} \exp(-p_1) \exp(-p_2) & \text{si } \mathbf{p} \geq \mathbf{0} \\ 0 & \text{sinon} \end{cases} \quad (2.41)$$

Nous souhaitons déterminer l'ensemble $\mathbb{S}_{0.95}$ de surface minimale tel que $\Pi(\mathbb{S}_{0.95}) = 0.95$.

Ce problème présente le double avantage d'être à deux dimensions, ce qui permettra de tracer $\mathbb{S}_{0,95}$, et d'être soluble analytiquement, d'où une vérification possible des résultats fournis par l'algorithme. Les isodensités sur $(\mathbb{R}^+)^2$ de seuil $0 < s < 1$ vérifient

$$\exp(-p_1) * \exp(-p_2) = s \Leftrightarrow p_1 + p_2 = -\ln(s) \quad (2.42)$$

Les isodensités sont donc des triangles isocèles, rectangles en l'origine. Une intégration formelle triviale montre que la probabilité de ce triangle est donnée par

$$\Pi(\pi^{-1}([s, +\infty[)) = 1 - s (1 - \ln(s)). \quad (2.43)$$

Ainsi, le seuil s_α associé à l'ensemble solution \mathbb{S}_α satisfait

$$1 - s_\alpha (1 - \ln(s_\alpha)) = \alpha. \quad (2.44)$$

Pour $\alpha = 0.95$, il vient

$$s_{0,95} (1 - \ln(s_{0,95})) = 0.05 \quad (2.45)$$

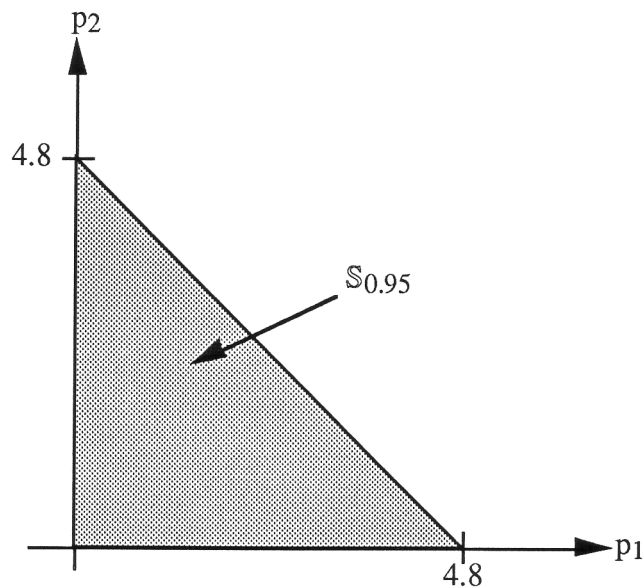


Fig. 2.5 Région de confiance $\mathbb{S}_{0,95}$ de la loi de probabilité (détermination analytique).

La solution trouvée par dichotomie est approximativement

$$s_{0,95} \cong 0.0087. \quad (2.46)$$

Les deux cotés égaux du triangle isocèle ont donc pour longueur $-\ln(0.0087) \cong 4.8$. La figure 2.5 représente l'ensemble $\mathbb{S}_{0.95}$ qui vient d'être calculé analytiquement.

2.7. Ensemble de stabilité

2.7.1. Introduction

Considérons pour structure de modèle $M(\cdot)$ du système à décrire, une structure telle que pour tout \mathbf{p} , le modèle $M(\mathbf{p})$ admette une représentation d'état linéaire d'ordre fini n , invariante dans le temps et telle que le paramètre \mathbf{p} qui est partiellement inconnu influence non-linéairement la sortie du modèle. Ce modèle peut être décrit par une représentation d'état de dimension finie ou par une matrice de transfert. Il est bien connu que les propriétés de stabilité de $M(\mathbf{p})$ sont complètement déterminées par celles de son polynôme caractéristique.

$$P(s) = \sum_{k=0}^n a_k s^k. \quad (2.47)$$

Si le modèle est à temps continu la stabilité asymptotique est assurée si et seulement si toutes les racines de ce polynôme caractéristique ont leur partie réelle strictement négative. Dans ce cas, le polynôme est dit *stable* ou *Hurwitz*. Pour un modèle à temps discret, la stabilité asymptotique est assurée si et seulement si toutes les racines de son polynôme caractéristique sont à l'intérieur du cercle unité du plan complexe. Le polynôme est alors dit *stable* ou *Schur*.

En supposant que le coefficient a_n ne puisse s'annuler, il est toujours possible de rendre le polynôme unitaire en divisant tous ses coefficients par a_n : les propriétés de stabilité du polynôme n'en seront pas affectées. Dans ce cas, le vecteur

$$\mathbf{a} = (a_0, \dots, a_{n-1})^T \quad (2.48)$$

sera appelé *vecteur des coefficients du polynôme unitaire*. Dans beaucoup de problèmes d'ingénierie, le polynôme caractéristique dépend des paramètres incertains du modèle du processus, et il est très utile de caractériser le *domaine de stabilité*, c'est-à-dire l'ensemble \mathbb{S} de toutes les valeurs des paramètres incertains \mathbf{p} qui correspondent à un modèle $M(\mathbf{p})$ asymptotiquement stable (voir par exemple (Siljak, 1989) pour un article de synthèse). Notre étude se limitera aux modèles à temps continu bien que la transposition aux modèles

linéaires à temps discret soit triviale. L'approche que nous proposerons produit des résultats garantis contrairement aux méthodes basées sur l'exploration systématique d'une grille ou sur une exploration aléatoire de l'espace des paramètres. Soit $P_p(s)$ le polynôme caractéristique du modèle $M(p)$ considéré. Il peut être écrit sous la forme

$$P_p(s) = \sum_{k=0}^n a_k(p) s^k, \quad (2.49)$$

où le polynôme P_p est supposé mis sous forme unitaire. Dans notre nouvelle approche, les coefficients $a_k(p)$ du polynôme peuvent être n'importe quelle fonction du vecteur des paramètres incertains p calculable à partir de p par un algorithme sans connexions logiques (comme if, while, ...). La classe des polynômes ainsi traitée est donc bien plus vaste que celle pour laquelle le célèbre théorème de Kharitonov (1978) et toutes ses extensions peuvent s'appliquer. Dans ce formalisme, il est également possible de traiter le modèle d'un système en boucle fermée, où p pourra contenir à la fois les paramètres incertains du modèle en boucle ouverte et les paramètres du système de commande. Une bonne connaissance de \mathcal{S} nous renseignera sur le choix des paramètres du système de commande dans le but d'assurer une robustesse satisfaisante vis-à-vis des incertitudes sur les paramètres du modèle en boucle ouverte. Essayons maintenant de redéfinir l'ensemble solution

$$\mathcal{S} \triangleq \{p \mid P_p \text{ Hurwitz}\} \quad (2.50)$$

en termes d'inversion ensembliste grâce à une utilisation formelle du critère de Routh dont nous allons rappeler le principe.

Critère de Routh

Le critère de Routh, abondamment utilisé pour l'analyse de stabilité des systèmes linéaires (voir par exemple DiStefano, Stubberud et Williams (1990), Faedo (1953), Leipholz (1987), Walter et Jaulin (1994)), est une méthode pour déterminer les signes des parties réelles des racines d'un polynôme à coefficients réels

$$P(s) = a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + a_1 s + a_0. \quad (2.51)$$

On construit tout d'abord la *table de Routh*, définie comme suit

$$\begin{array}{cccccc}
 a_n & a_{n-2} & a_{n-4} & a_{n-6} & \dots & \\
 a_{n-1} & a_{n-3} & a_{n-5} & a_{n-7} & \dots & \\
 b_1 & b_2 & b_3 & b_4 & \dots & \\
 c_1 & c_2 & c_3 & c_4 & \dots & \\
 \dots & \dots & \dots & \dots & \dots &
 \end{array} \tag{2.52}$$

où

$$\begin{aligned}
 b_1 &= (a_{n-1} a_{n-2} - a_n a_{n-3}) / a_{n-1} \\
 b_2 &= (a_{n-1} a_{n-4} - a_n a_{n-5}) / a_{n-1}, \dots \\
 c_1 &= (b_1 a_{n-3} - a_{n-1} b_2) / b_1 \\
 c_2 &= (b_1 a_{n-5} - a_{n-1} b_3) / b_1, \dots
 \end{aligned} \tag{2.53}$$

La table se continue ainsi horizontalement et verticalement jusqu'à ce que des zéros soient obtenus. Le nombre des racines à parties réelles positives est alors donné par le nombre de changement dans la première colonne de cette table.

L'utilisation du critère de Routh pour étudier la stabilité d'un polynôme à coefficients incertains n'est pas nouvelle, puisque Faedo (1953) a développé une version primitive de l'analyse par intervalles pour en déduire des conditions suffisantes de stabilité. A partir de l'expression formelle de $P_{\mathbf{p}}(s)$, nous pouvons construire une table de Routh formelle, où les coefficients de la table dépendent uniquement des composantes de \mathbf{p} . L'application du critère de Routh aboutit à n inégalités portant sur les composantes de \mathbf{p} qui constituent, pour un \mathbf{p} générique, un ensemble de conditions nécessaires et suffisantes de stabilité asymptotique du modèle $M(\mathbf{p})$. Ces inégalités sont de la forme

$$f_i(\mathbf{p}) > 0, i = 1, \dots, n. \tag{2.54}$$

Posons $\mathbb{Y} =]0, \infty[)^n$ et soit \mathbf{f} la fonction vectorielle qui a pour fonctions coordonnées les f_i . L'ensemble de stabilité $\mathbb{S} = \mathbf{f}^{-1}(\mathbb{Y})$ est donc la solution d'un problème d'inversion ensembliste. Les deux exemples suivants serviront de support pour tester l'algorithme d'inversion ensembliste développé ultérieurement. Le premier est un exemple sans grand intérêt pratique, mais illustrant le champ d'application de la méthode. L'ensemble solution que nous allons découvrir possède une forme étrange qui fait de cet exemple une curiosité mathématique. Le second exemple montre l'intérêt pratique de l'approche ensembliste de l'étude de stabilité.

2.7.2. Problème des palmes

Le nom de ce problème ensembliste vient de la forme de l'ensemble solution dont les composantes ressemblent à des feuilles de palmier. Dans ce problème nous cherchons à caractériser le domaine de stabilité du polynôme

$$P_p(s) = s^3 + (p_1 p_2) s^2 + p_1^2 s + p_1 p_2. \quad (2.55)$$

Ce polynôme présente l'intérêt d'avoir une forte non-linéarité en \mathbf{p} qui permet de montrer l'immensité du champ d'action de la méthode que nous développerons par la suite. Considérons tout d'abord un polynôme $P(s)$ unitaire de degré 3 :

$$P(s) = s^3 + a_2 s^2 + a_1 s + a_0. \quad (2.56)$$

La table de Routh associée à $P(s)$ dans le cas générique est

1	a_1	0
a_2	a_0	0
$\frac{a_1 a_2 - a_0}{a_2}$	0	
a_0		

(2.57)

D'après le critère de Routh, une condition nécessaire et suffisante de stabilité asymptotique est que tous les éléments de la première colonne de la table soient de même signe. Comme le premier élément de la première colonne est positif, il faut que les 3 derniers éléments de la première colonne soient strictement positifs, ce qui se traduit par les inégalités

$$\left\{ \begin{array}{l} a_0 > 0 \\ a_2 > 0 \\ a_1 a_2 - a_0 > 0. \end{array} \right. \quad (2.58)$$

Ainsi, les conditions nécessaires et suffisantes sur \mathbf{p} pour que le polynôme $P_p(s)$ soit asymptotiquement stable sont

$$\mathbf{f}(\mathbf{p}) = \begin{pmatrix} p_1 p_2 \\ \sin(p_1 p_2) \\ p_1^2 \sin(p_1 p_2) - p_1 p_2 \end{pmatrix} > \mathbf{0}. \quad (2.59)$$

L'ensemble solution peut donc être défini en termes d'inversion ensembliste par

$$\mathbb{S} = \mathbf{f}^{-1}([0, \infty]^3). \quad (2.60)$$

2.7.3. Problème du col de cygne

Le nom donné à ce problème vient lui aussi de la forme l'ensemble solution qui peut faire penser à un col de cygne. Considérons un modèle d'un système décrit par la fonction de transfert

$$\frac{B(s)}{A(s)} = \frac{k \omega_0^2}{(1 + T s)(s^2 + 2 z \omega_0 s + \omega_0^2)}. \quad (2.61)$$

Les valeurs nominales pour les paramètres sont fixées à $k = 1$, $T = -1$, $z = 1$ et $\omega_0 = 1$. Les racines du polynôme caractéristique $A(s)$ pour les valeurs nominales des paramètres sont 1, -1 et -1 . Le modèle nominal est donc instable en boucle ouverte. Supposons maintenant que nous voulions commander le modèle nominal avec une commande PID insérée dans la chaîne directe et une contre réaction unitaire (voir figure 2.6). La fonction de transfert du PID a pour structure

$$\frac{R(s)}{S(s)} = \frac{c_1 + c_2 s + c_3 s^2}{s}. \quad (2.62)$$

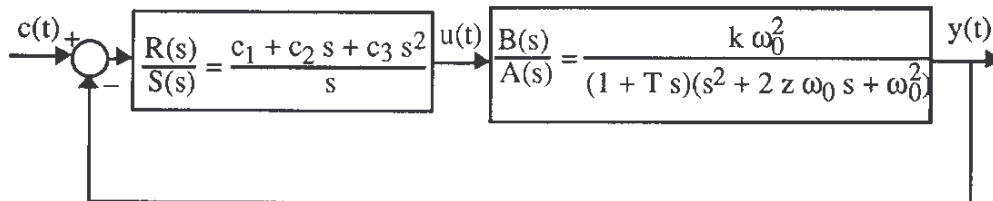


Fig. 2.6 Modèle régulé par un PID.

Le polynôme caractéristique unitaire du modèle en boucle fermée est alors :

$$P(s) = A(s).S(s) + B(s).R(s) =$$

$$s^4 + [2z\omega_0 + T^{-1}] s^3 + \left[\frac{2z\omega_0}{T} + \omega_0^2 \left(1 + \frac{kc_3}{T} \right) \right] s^2 + \frac{\omega_0^2 \cdot (1 + c_2k)}{T} s + \frac{\omega_0^2 kc_1}{T} \quad (2.63)$$

c'est-à-dire pour les valeurs nominales des paramètres

$$P_n(s) = s^4 + s^3 - (1 + c_3) s^2 - (1 + c_2) s - c_1. \quad (2.64)$$

En appliquant à ce polynôme le critère de Routh, il est facile de vérifier que le modèle nominal est stabilisé pour $c_1 = -1$, $c_2 = -2$ et $c_3 = -6$. Le problème que nous allons considérer est celui de la robustesse de la commande associée au réglage du PID vis-à-vis d'une incertitude sur la valeur de la constante de temps T (supposée différente de zéro) et sur le coefficient d'amortissement z . Dans ce but, nous voulons caractériser l'ensemble \mathbb{S} des couples (z, T) tels que le modèle en boucle fermée reste stable en laissant les paramètres ω_0 et k à leur valeur nominale. Ceci revient à considérer l'ensemble des couples (z, T) tels que le polynôme caractéristique

$$P_p(s) = s^4 + (2z + T^{-1}) s^3 + \left(\frac{2z - 6}{T} + 1 \right) s^2 - \frac{1}{T} s - \frac{1}{T} \quad (2.65)$$

soit Hurwitz. Rappelons que la table de Routh associée au polynôme

$$P(s) = s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0 \quad (2.66)$$

est donnée par

1	a ₂	a ₀
a ₃	a ₁	0
$\frac{a_2 a_3 - a_1}{a_3}$	a ₀	0
$a_1 - \frac{a_0 a_3^2}{a_2 a_3 - a_1}$	0	0
a ₀		

(2.67)

D'après le critère de Routh, les conditions nécessaires et suffisantes de stabilité asymptotique sont

$$\left\{ \begin{array}{l} a_0 > 0 \\ a_3 > 0 \\ a_2 a_3 - a_1 > 0 \\ a_1 a_2 a_3 - a_1^2 - a_0 a_3^2 > 0. \end{array} \right. \quad (2.68)$$

L'ensemble de stabilité peut donc s'écrire sous la forme $\mathbb{S} = \mathbf{f}^{-1}(\mathbb{Y})$ avec

$$\mathbf{f}(\mathbf{p}) = \mathbf{f}(T, z) = \begin{pmatrix} -T^{-1} \\ 2z + T^{-1} \\ T^{-2} (2z - 6 + T) (2zT + 1) + T^{-1} \\ -T^{-3} (2z - 6 + T) (2zT + 1) - T^{-2} + T^{-3} (2zT + 1)^2 \end{pmatrix} \quad (2.69)$$

et

$$\mathbb{Y} = (]0, \infty[)^4. \quad (2.70)$$

Remarques

(i) Pour les problèmes à deux paramètres, il est possible de tracer toutes les courbes de l'espace des paramètres le long desquelles s'annule un élément de la première colonne de la table de Routh. Ces courbes partitionnent le domaine d'intérêt en régions associées à un nombre donné de racines instables. La détermination graphique de l'ensemble de stabilité est alors immédiate. Des résultats similaires peuvent être obtenus par les techniques de D-décomposition (Leipholz, 1987). Toutefois, ces méthodes font intervenir des équations implicites difficiles à manipuler. De plus, elles ne s'étendent pas facilement aux problèmes faisant intervenir plus de deux paramètres, contrairement à la méthode que nous allons proposer par la suite. En effet, la partition de l'espace donnée par des inéquations non linéaires ne permet que difficilement d'en déduire les caractéristiques élémentaires de l'ensemble solution comme de savoir s'il est vide ou non.

(ii) La complexité de l'expression formelle associée à \mathbf{f} augmente rapidement avec le degré n du polynôme $P_p(s)$. L'utilisation du calcul formel sur ordinateur (voir par exemple (Davenport, Siret, Tournier, 1987)) peut être nécessaire pour traiter des problèmes plus complexes.

(iii) La méthode se transpose directement au cas discret en remplaçant le critère de Routh par celui de Schur-Cohn.

(iv) Dans le cas continu, en effectuant le changement de variable $s' = s + \lambda$, où λ est un réel positif connu, il est possible de caractériser toutes les valeurs des paramètres tels que le degré de stabilité soit plus grand que λ , c'est-à-dire que toutes les réponses impulsionnelles du modèle convergent vers zéro plus rapidement que $\exp(-\lambda t)$.

2.8. Ensemble stabilisant

2.8.1. Introduction

Intéressons nous maintenant au problème de la commande robuste d'un système. Supposons que le modèle $M(\mathbf{p})$ choisi pour décrire ce système soit un modèle linéaire entrées-sorties et invariant dans le temps. Ce modèle peut être décrit par des équations différentielles si le modèle est à temps continu, ou par des équations de récurrence si le modèle est à temps discret. La structure de modèle $M(\cdot)$ considérée peut être non-linéaire en \mathbf{p} . Nous ferons l'hypothèse simplificatrice que le système est exactement équivalent à un modèle particulier $M(\mathbf{p}^*)$ de la structure $M(\cdot)$. La valeur exacte du paramètre \mathbf{p}^* est inconnue, mais nous supposerons qu'elle appartient à coup sûr à un ensemble $\mathbb{S}_{\mathbf{p}}$, qui a pu par exemple être obtenu par une technique d'estimation garantie à erreurs bornées. Nous voulons maintenant stabiliser le système avec un régulateur $R(\mathbf{c})$ appartenant à une structure fixée $R(\cdot)$ dépendant non-linéairement du vecteur des paramètres du régulateur \mathbf{c} (voir figure 2.7).

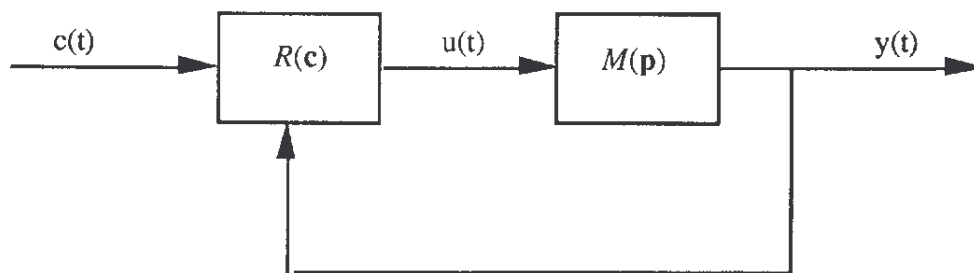


Fig. 2.7 Modèle incertain à commander par un régulateur de structure $R(\cdot)$.

Nous voulons caractériser l'ensemble stabilisant $\mathbb{S}_{\mathbf{c}}$ de toutes les valeurs de \mathbf{c} qui stabilisent tous les modèles $M(\mathbf{p})$ pour $\mathbf{p} \in \mathbb{S}_{\mathbf{p}}$. Nous aurons ainsi l'ensemble des commandes de la structure choisie qui stabilisent à coup sûr le système $M(\mathbf{p}^*)$. Soit $P_{\mathbf{c},\mathbf{p}}(s)$ le polynôme caractéristique du modèle $M(\mathbf{p})$ commandé par le régulateur $R(\mathbf{c})$. La stabilité du modèle commandé est équivalente à la stabilité de son polynôme caractéristique. D'après le critère de

Routh dans le cas où le modèle est à temps continu, ou le critère de Schur-Cohn dans le cas discret, des conditions nécessaires et suffisantes pour la stabilité de $P_{c,p}$ et donc du modèle contrôlé associé pour n'importe quelles valeurs des paramètres c et p peuvent être obtenues sous forme d'inégalités

$$f(c,p) > 0 ; i = 1, \dots, n. \quad (2.71)$$

L'ensemble stabilisateur peut donc être défini par

$$\mathbb{S}_c \triangleq \{c \mid \forall p \in \mathbb{S}_p, f(c,p) > 0\}. \quad (2.72)$$

Le problème du calcul de l'ensemble stabilisant \mathbb{S}_c ne peut pas se mettre sous la forme d'un problème d'inversion ensembliste, pourtant, le même type d'approche sera utilisé pour sa caractérisation.

2.8.2. Problème du lobe stabilisant

Considérons, à titre d'application, le problème de la détermination d'un ensemble de correcteurs stabilisant pour un ensemble de modèles $M(p)$ décrits par la fonction de transfert

$$\frac{B(p, s)}{A(p, s)} = \frac{k \omega_0^2}{(1 + T s)(s^2 + 2 z \omega_0 s + \omega_0^2)}, \quad (2.73)$$

et dont le vecteur des paramètres $p = [z, T, \omega_0, k]^T$ satisfait

$$z \in [0.95, 1.05], T \in [-1.05, -0.95], \omega_0 \in [0.95, 1.05], k \in [0.95, 1.05]. \quad (2.74)$$

Cela peut correspondre à la situation où le fabricant garantit que le processus délivré a un comportement identique au modèle $M(p)$, où le vecteur de paramètres p ne peut s'écarter de sa valeur nominale $p_n = (z_n, T_n, \omega_{0n}, k_n)^T = (1, -1, 1, 1)^T$ de plus de 5%. Ainsi, l'ensemble \mathbb{S}_p des paramètres incertains est

$$\mathbb{S}_p = [0.95, 1.05] \times [-1.05, -0.95] \times [0.95, 1.05] \times [0.95, 1.05] \quad (2.75)$$

Proposons nous de stabiliser l'ensemble des modèles correspondants par un régulateur ayant la structure d'un PID classique

$$\frac{R(s)}{S(s)} = \frac{c_1 + c_2 s + c_3 s^2}{s} \quad (2.76)$$

inséré dans la chaîne directe et une contre-réaction unitaire (voir la figure 2.6). Une commande garantie peut consister par exemple à trouver un vecteur $\mathbf{c} = [c_1, c_2, c_3]^T$ qui garantisse la stabilité de tous les modèles $M(\mathbf{p})$ pour $\mathbf{p} \in \mathbb{S}_p$. Afin de permettre une représentation graphique en deux dimension, nous fixerons $c_3 = -6$, car nous avons vu au paragraphe 2.7.3 que $R([-1, -2, -6])$ stabilise le modèle nominal $M(\mathbf{p}_n)$. Nous nous intéresserons alors à l'ensemble \mathbb{S}_c de toutes les valeurs des paramètres c_1 et c_2 du régulateur qui sont tels que le modèle en boucle fermée $M(\mathbf{c}, \mathbf{p})$ de polynôme caractéristique $P_{\mathbf{p},\mathbf{c}}$ soit stable pour tout \mathbf{p} de \mathbb{S}_p . L'ensemble \mathbb{S}_c , que nous allons dessiner après la résolution du problème, présente la forme d'un lobe, ce qui explique le nom donné à ce problème. Notons que la méthode que nous allons proposer peut très bien s'appliquer dans le cas où c_3 ne serait pas fixé. Rappelons que l'expression du polynôme caractéristique $P_{\mathbf{p},\mathbf{c}}$ du modèle en boucle fermée donnée en (2.63) est

$$P_{\mathbf{p},\mathbf{c}}(s) = s^4 + [2 z \omega_0 + T^{-1}] s^3 + [2 z \omega_0 T^{-1} + \omega_0^2 (1 - 6 k T^{-1})] s^2 + \omega_0^2 (1 + c_2 k) T^{-1} s + \omega_0^2 k c_1 T^{-1}. \quad (2.77)$$

En procédant comme dans les problèmes des ensembles stabilisants, une expression formelle de $\mathbf{f}(\mathbf{c}, \mathbf{p})$ est immédiatement obtenue à l'aide de la table de Routh. Pour ne pas avoir une expression formelle trop longue, exprimons $\mathbf{f}(\mathbf{c}, \mathbf{p})$ comme composée de deux fonctions

$$\mathbf{f}(\mathbf{c}, \mathbf{p}) = \mathbf{g} \circ \mathbf{a}(\mathbf{c}, \mathbf{p}) \quad (2.78)$$

où \mathbf{a} correspond à la fonction vectorielle qui associe à chaque couple de paramètres (\mathbf{c}, \mathbf{p}) , le vecteur $\mathbf{a}(\mathbf{c}, \mathbf{p})$ des coefficients du polynôme $P_{\mathbf{p},\mathbf{c}}(s)$, et \mathbf{g} est la fonction qui associe au vecteur des coefficients du polynôme unitaire le vecteur formé des expressions formelles apparaissant dans les inégalités (2.68) découlant de la table de Routh (2.67).

$$\mathbf{a}(\mathbf{p}, \mathbf{c}) = \begin{pmatrix} a_0(\mathbf{p}, \mathbf{c}) \\ a_1(\mathbf{p}, \mathbf{c}) \\ a_2(\mathbf{p}, \mathbf{c}) \\ a_3(\mathbf{p}, \mathbf{c}) \end{pmatrix} = \begin{pmatrix} \omega_0^2 k c_1 T^{-1} \\ \omega_0^2 (1 + c_2 k) T^{-1} \\ 2 z \omega_0 T^{-1} + \omega_0^2 (1 - 6 k T^{-1}) \\ 2 z \omega_0 + T^{-1} \end{pmatrix}, \quad (2.79)$$

$$\mathbf{g}(\mathbf{a}) = \begin{pmatrix} a_0 \\ a_3 \\ a_2 a_3 - a_1 \\ a_1 a_2 a_3 - a_1^2 - a_0 a_3^2 \end{pmatrix}. \quad (2.80)$$

Il nous faut donc trouver l'ensemble \mathbb{S}_c défini par

$$\mathbb{S}_c = \{\mathbf{c} \mid \forall \mathbf{p} \in \mathbb{S}_p, \mathbf{f}(\mathbf{c}, \mathbf{p}) > \mathbf{0}\}. \quad (2.81)$$

2.9. Conclusion

Dans ce chapitre, nous avons présenté ce que pouvait apporter une formulation ensembliste dans les domaines de l'estimation et de la commande. Nous n'avons pas parlé des méthodes à mettre en œuvre pour résoudre ces problèmes ensemblistes et qui forment le sujet essentiel des prochains chapitres.

Une formulation ensembliste des problèmes permet de caractériser de façon rigoureuse les incertitudes des résultats causées par des incertitudes sur les données. L'ensemble solution contient toute l'information qui était contenue dans le problème de départ et qui était susceptible de nous intéresser, et ceci contrairement aux approches ponctuelles. Dans le cas idéal où la structure du modèle $M(\cdot)$ contient le système étudié $M(\mathbf{p}^*)$, et sous la condition que nos hypothèses sur la bornitude du bruit de mesure soient correctes, la solution du problème d'estimation ensembliste est un ensemble qui contient à coup sûr la vraie valeur du paramètre \mathbf{p}^* . Si la commande est calculée par une approche ensembliste garantie, nous sommes certains que la commande obtenue stabilisera notre système.

Les problèmes ensemblistes semblent en première approche très complexes à résoudre. Pourtant, grâce aux outils nés de l'analyse par intervalles et développés dans les prochains chapitres, nous arriverons à résoudre un grand nombre d'entre eux. Les outils utilisés serviront même à résoudre des problèmes ponctuels que nous n'arrivions pas à résoudre de façon globale et garantie avec les approches classiques. Il s'agit de problèmes d'optimisation globale, garantie et non-linéaire, dont nous nous servirons pour calculer certaines caractéristiques numériques de l'ensemble solution dans le but d'obtenir rapidement une information précise et garantie sur cet ensemble sans avoir à le caractériser complètement.

CHAPITRE 3

CALCUL ET ANALYSE

PAR INTERVALLES

3.1. Introduction

Lors des calculs numériques sur ordinateur, il est souvent impossible de caractériser la validité des résultats généralement faussés par la propagation d'erreurs numériques. C'est l'une des raisons principales qui a motivé la manipulation d'intervalles à la place de réels. Tout nombre incertain est alors représenté par un intervalle le contenant de façon garantie. L'ensemble des techniques permettant le calcul sur les intervalles forme une branche relativement récente des mathématiques appliquées. C'est le *calcul par intervalles*, apparu il y a une trentaine d'années et qui a connu un développement rapide (voir par exemple (Moore, 1979 ; Neumaier, 1990 ; Ratschek et Rokne, 1988 ; Jahn, 1991)). Cette approche permet une évaluation rigoureuse et garantie du résultat d'un calcul numérique sur machine.

Le calcul par intervalles manipule les intervalles comme un nouveau type de nombres. Un intervalle est représenté par une paire ordonnée de nombres réels associés à ses extrémités, de la même façon qu'un nombre rationnel a/b est représenté par une paire de deux nombres entiers relatifs et qu'un nombre complexe $x + i.y$ est représenté par une paire de deux nombres réels.

Un intervalle a une double nature, à la fois de nombre et d'ensemble contenant une infinité de nombres réels. Un grand nombre d'algorithmes utilisant le calcul par intervalles exploite cette dualité et combinent les opérations sur les ensembles, telles que l'union et l'intersection, avec les opérations arithmétiques sur les réels qui s'étendent très facilement aux intervalles.

Certains algorithmes sur les intervalles sont des extensions naturelles d'algorithmes réels (c'est-à-dire ponctuels) déjà existants. Dans ce cas, un intervalle représente la quantité réelle incertaine qui lui est associée dans l'algorithme réel et qui est supposée lui appartenir à coup sûr. Quand ses deux extrémités sont identiques, l'intervalle, dit *dégénéré*, peut s'identifier à la quantité réelle correspondante qui est donc connue avec certitude. Si, lors d'un calcul, les

données incertaines en entrée du calcul appartiennent à coup sûr à des intervalles connus, le résultat du calcul par intervalles sera un intervalle contenant la solution à coup sûr.

D'autres algorithmes sur les intervalles ont une optique totalement différente et ne correspondent à aucun algorithme réel. L'approche qu'ils utilisent, qui sera appelée *analyse par intervalles*, se sert d'intervalles pour partitionner la droite réelle et en déduire une solution garantie au problème posé. Le résultat obtenu diffère donc essentiellement de celui des méthodes classiques, effectuant une exploration aléatoire de l'espace de recherche, qui peuvent trouver une solution acceptable au problème posé mais sans aucune garantie.

Un autre aspect intéressant et utile des algorithmes obtenus par analyse par intervalles est leur convergence finie. Si un algorithme génère une séquence d'intervalles emboîtés les uns dans les autres (dans le sens où le $n^{\text{ème}}$ intervalle est inclus dans le $n-1^{\text{ème}}$), comme le calcul est fait en précision finie, il est possible de montrer qu'au bout d'un nombre fini d'étapes la séquence ne change plus. L'algorithme s'arrête dès que le résultat de deux itérations successives sont strictement égaux.

Il est possible de microprogrammer les opérations sur les intervalles, voire de les faire exécuter par un processeur spécialisé câblé esclave du CPU, de façon à réduire la durée d'exécution. Dans le cas d'algorithmes sur les intervalles correspondant à des algorithmes réels, le temps d'exécution CPU pourra alors devenir strictement identique à celui requis par l'algorithme réel correspondant.

Enfin, nous verrons que le calcul par intervalles peut aussi être vue comme un langage simple adapté à la manipulation d'inégalités.

3.2. Intervalles réels et arithmétique sur les intervalles

3.2.1. Intervalles

Un intervalle x de \mathbb{R} ou *intervalle réel* est un sous-ensemble connexe fermé et borné de \mathbb{R}

$$x = [x^-, x^+] \triangleq \{x \mid x^- \leq x \leq x^+\}. \quad (3.1)$$

Dans la littérature, on parle parfois d'intervalles ouverts, mais ici tous les intervalles considérés seront fermés. Une définition d'intervalles non bornés peut être utilisée dans

certaines applications, (Ratschek et Rokne, 1988, page 29), mais nous nous limiterons dans ce mémoire à la définition ci-dessus. La *longueur* d'un intervalle est définie par

$$w(\mathbf{x}) \triangleq x^+ - x^- \quad (3.2)$$

L'intervalle \mathbf{x} permet une représentation en machine d'un réel x connu avec plus ou moins d'incertitude. Il suffit, pour cela, que nous soyons certains que $x \in \mathbf{x}$. La longueur $w(\mathbf{x})$ de \mathbf{x} caractérise alors l'incertitude avec laquelle x est connue. Un intervalle \mathbf{x} permet aussi une représentation en machine d'un réel x quelconque. Par exemple, si $x = 1/3$, et s'il n'est possible de représenter en machine que deux digits décimaux (en virgule fixe), l'intervalle $\mathbf{x} = [0.33, 0.34]$ représentera en machine le nombre x . Le calcul sur les intervalles permet de généraliser aux intervalles tous les calculs que nous savions faire sur les réels, afin d'évaluer de façon rigoureuse l'incertitude avec laquelle les quantités manipulées sont connues.

3.2.2. Arithmétique sur les intervalles.

L'arithmétique classique est fondée sur la science des nombres rationnels. Les seules opérations autorisées sont l'addition, la soustraction, la multiplication et la division. L'*arithmétique sur les intervalles* (Moore, 1979) généralise ces quatre opérations aux intervalles et en étudie les propriétés. Les intervalles pourront ainsi être manipulés comme des nombres. Rappelons tout d'abord les deux principes du calcul ensembliste largement utilisés en mathématiques. Si A et B sont deux sous-ensembles de \mathbb{R}^n

$$A \ \& \ B \triangleq \{a \ \& \ b \mid a \in A \text{ et } b \in B\}, \quad (3.3)$$

$$f(A) \triangleq \{f(a) \mid a \in A\}, \quad (3.4)$$

où $\&$ est une loi binaire sur \mathbb{R}^n , et f une fonction ayant \mathbb{R}^n pour ensemble de départ. Les opérations sur les intervalles s'en déduisent immédiatement

$$\begin{aligned} \mathbf{x} + \mathbf{y} &= \{x + y \mid x \in \mathbf{x} \text{ et } y \in \mathbf{y}\} = [x^- + y^-, x^+ + y^+], \\ -\mathbf{x} &= \{-x \mid x \in \mathbf{x}\} = [-x^+, -x^-], \\ \mathbf{x} - \mathbf{y} &= \{x - y \mid x \in \mathbf{x} \text{ et } y \in \mathbf{y}\} = [x^- - y^+, x^+ - y^-], \\ 1 / \mathbf{x} &= \{1 / x \mid x \in \mathbf{x}\} = [1/x^+, 1/x^-] \text{ (si } 0 \notin \mathbf{x}\text{)}, \\ \mathbf{x} * \mathbf{y} &= \{x * y \mid x \in \mathbf{x} \text{ et } y \in \mathbf{y}\}, \\ &= [\min(x^- * y^-, x^- * y^+, x^+ * y^-, x^+ * y^+), \max(x^- * y^-, x^- * y^+, x^+ * y^-, x^+ * y^+)], \\ \mathbf{x}^2 &= \{x^2 \mid x \in \mathbf{x}\}. \end{aligned} \quad (3.5)$$

Notons que $x^2 \neq x * x$ car par exemple si $x = [-1, 1]$, $x^2 = [0, 1]$ alors que $x * x = [-1, 1]$. Si $0 \in x$, l'ensemble $\{1/x \mid x \in x\}$ n'est pas borné et ne peut donc pas être représenté par un intervalle tel que nous l'avons défini.

3.2.3. Arithmétique sur les intervalles représentables en machine

L'arithmétique sur les intervalles représentables en machine peut être considérée comme une approximation sur calculateur numérique, de l'arithmétique sur les intervalles. Soient x et y deux réels inconnus sur lesquels la seule information disponible est qu'ils appartiennent respectivement aux intervalles x et y . Ces deux intervalles peuvent ne pas être représentables en machine de façon exacte, comme c'est le cas pour $[0, 1/3]$. Remplaçons alors x et y par les plus petits intervalles représentables en machine x_M et y_M tels que $x \subset x_M$ et $y \subset y_M$. Intéressons nous maintenant à la quantité $x \& y$, où $\&$ est un opérateur quelconque. Nous avons $x \& y \in x \& y \subset x_M \& y_M$. Mais $x_M \& y_M$ peut ne pas être représentable en machine et il conviendra alors de le remplacer par l'intervalle représentable en machine $(x_M \& y_M)_M$. Ce principe forme la base de l'arithmétique sur les intervalles représentables en machine qui garantit que la solution inconnue du calcul numérique faite en précision finie appartient à l'intervalle représentable en machine calculé. Les erreurs d'approximation numérique sont ainsi évaluées de façon garantie (quoique souvent pessimiste). Le résultat obtenu sera un intervalle qui contiendra à coup sûr la solution exacte. Sans les précautions prises par l'arithmétique sur les intervalles représentables en machine, le calculateur aurait tronqué arbitrairement les extrémités des intervalles manipulés et le résultat obtenu n'aurait plus été garanti. L'arithmétique sur les intervalles représentables en machine a donné naissance à des extensions commercialisées de FORTRAN et PASCAL (IBM, 1986; Kulisch, 1987). L'exemple numérique suivant (tiré de Moore, 1981) illustre la différence entre l'arithmétique sur les intervalles et l'arithmétique sur les intervalles représentable en machine.

Exemple 3.1 : Si $x = [-0.613 \cdot 10^{-2}, -0.61 \cdot 10^{-2}]$ et $y = [1, 3]$. Calculons $z = x * (1 + 1/y)$. En utilisant l'arithmétique par intervalles exacte, on obtient $z = x * (1 + 1/[1, 3]) = x * [4/3, 2] = [-0.01226, -0.0083333...]$. Cet intervalle, ainsi que les intervalles intermédiaires ne sont pas représentables en machine. En refaisant le même raisonnement avec l'arithmétique par intervalles représentables en machine avec une précision finie de trois chiffres, il vient $1/y \subset [0.333 \cdot 10^0, 0.100 \cdot 10^1]$ donc $1 + 1/y \subset [1.333 \cdot 10^0, 0.200 \cdot 10^1]$ et enfin $z \subset [-1.123 \cdot 10^{-1}, -0.811 \cdot 10^{-2}]$. Si $x \in x$ et $y \in y$ alors le réel $z = x * (1 + 1/y)$ est donc compris

de façon certaine dans l'intervalle $[-1.123.10^{-1}, -0.811.10^{-2}]$. \diamond

Remarque 3.1 : Par la suite nous ferons abstraction de tous ces problèmes numériques et nous utiliserons simplement l'arithmétique sur les intervalles, mais en toute rigueur, il conviendrait d'utiliser l'arithmétique sur les intervalles représentables en machine. \diamond

3.2.4. Propriétés algébriques de l'arithmétique sur les intervalles

Les propriétés de l'arithmétique sur les intervalles sont des conséquences directes des définitions des opérations qu'elle utilise. Par exemple, il est facile de montrer que la multiplication et l'addition sont toutes les deux associatives et commutatives. Par contre, la distributivité de l'addition par rapport à la multiplication n'est pas toujours vérifiée. Nous avons toutefois la propriété dite de sous-distributivité

$$\mathbf{x} * (\mathbf{y} + \mathbf{z}) \subset \mathbf{x} * \mathbf{y} + \mathbf{x} * \mathbf{z} . \quad (3.6)$$

Si $\&$ est un opérateur sur les réels tel que $+$, $-$, $*$, $/$, nous avons alors la propriété dite *d'inclusion monotone* :

$$\mathbf{x}_1 \subset \mathbf{x}_2 \text{ et } \mathbf{y}_1 \subset \mathbf{y}_2 \Rightarrow \mathbf{x}_1 \& \mathbf{y}_1 \subset \mathbf{x}_2 \& \mathbf{y}_2 . \quad (3.7)$$

Ces quelques propriétés nous suffiront pour les manipulations sur les intervalles que nous aurons à effectuer. Pour plus d'information, consulter (Moore 1979).

3.3. Fonctions d'inclusion

3.3.1. Définition, inclusion monotone, propriété de convergence

Soit $f : \mathbb{R} \rightarrow \mathbb{R}$, une fonction réelle, la fonction $f : \mathbb{IR} \rightarrow \mathbb{IR}$ sera appelée *fonction d'inclusion* de f si et seulement si elle vérifie

$$f(\mathbf{x}) \subset f(\mathbf{x}) \quad (3.8)$$

pour tout intervalle \mathbf{x} . Souvent, l'image de l'intervalle \mathbf{x} par la fonction f est inaccessible à

l'utilisateur. En effet, trouver dans le cas général l'image d'un intervalle \mathbf{x} par une fonction f nécessite plusieurs calculs d'extrema que nous ne savons résoudre que dans certains cas particuliers. En revanche, pour la plupart des fonctions réelles f , il est très facile de trouver une fonction d'inclusion, comme nous le verrons par la suite. Ainsi, si nous savons avec certitude que le réel x appartient à l'intervalle \mathbf{x} , nous aurons la garantie que l'image $f(x)$ appartient à l'intervalle $f(\mathbf{x})$. Pour une même fonction f , il existe bien sûr une infinité de fonctions d'inclusion. La fonction d'inclusion \mathbf{f} de f sera dite *monotone* si elle satisfait pour tout couple d'intervalles \mathbf{x} et \mathbf{y}

$$\mathbf{x} \subset \mathbf{y} \Rightarrow \mathbf{f}(\mathbf{x}) \subset \mathbf{f}(\mathbf{y}). \quad (3.9)$$

Toutes les fonctions d'inclusion manipulées par la suite seront de ce type. Une fonction d'inclusion \mathbf{f} de f sera dite *convergente* si elle satisfait, pour toute suite d'intervalles \mathbf{x}

$$w(\mathbf{x}) \rightarrow 0 \Rightarrow w(\mathbf{f}(\mathbf{x})) \rightarrow 0, \quad (3.10)$$

ce qui revient à dire que quand \mathbf{x} tend vers un réel x , son intervalle image par \mathbf{f} tend vers le réel $f(x)$. De telles fonctions d'inclusion se trouvent aisément pour la plupart des fonctions f utilisées en physique, pourvu que f soit continue. Cette propriété permettra d'obtenir des algorithmes convergents. Dans la suite, une fonction d'inclusion sera par défaut monotone et convergente. Toutes ces notions se généralisent immédiatement aux intervalles représentables en machine, qu'il faudrait utiliser en toute rigueur pour avoir un résultat garanti.

3.3.2. Fonction d'inclusion minimale

Pour une même fonction f , il existe une infinité de fonctions d'inclusion. Mais, parmi toutes celles-ci, il en existe une qui est minimale au sens de l'inclusion. Nous l'appellerons la *fonction d'inclusion minimale*, notée \mathbf{f}^* . Cette fonction d'inclusion est nécessairement monotone, et convergente si f est continue.

Exemple 3.2 : Notons \exp la fonction exponentielle. Comme cette fonction est croissante, la fonction d'inclusion minimale est définie par

$$\exp^*(\mathbf{x}) = \exp^*([x^-, x^+]) = [\exp(x^-), \exp(x^+)] = \exp(\mathbf{x}). \quad (3.11)$$

L'intervalle image de \mathbf{x} par la fonction d'inclusion \exp^* est donc identique à l'image par la fonction \exp de \mathbf{x} . Du fait de la continuité, la propriété de convergence de la fonction d'inclusion \exp^* est satisfaite. \diamond

La même approche peut être utilisée pour toute fonction f continue et monotone. L'intervalle image $f(\mathbf{x})$ a pour bornes les images des extrémités x^- et x^+ de l'intervalle \mathbf{x} . Notons que si f n'est pas monotone sur \mathbb{R} elle peut l'être sur l'intervalle \mathbf{x} considéré. C'est le cas par exemple si $0 \notin f(\mathbf{x})$, où f' est une fonction d'inclusion de la dérivée f' de f .

Exemple 3.3 : Soit \sin la fonction sinus. Elle n'est pas monotone, et la technique précédente ne peut plus être appliquée pour calculer la fonction d'inclusion minimale associée \sin^* . A l'aide du graphe de la fonction sinus, il est aisé de comprendre que l'image $\sin^*(\mathbf{x})$ de l'intervalle \mathbf{x} est calculée comme suit :

$$\begin{aligned} \text{si } \exists k \in \mathbb{Z} \mid 2k\pi - \pi/2 \in \mathbf{x} & \text{ alors } \sin^-(\mathbf{x}) = -1 \\ & \text{sinon } \sin^-(\mathbf{x}) = \min(\sin x^-, \sin x^+), \\ \text{si } \exists k \in \mathbb{Z} \mid 2k\pi + \pi/2 \in \mathbf{x} & \text{ alors } \sin^+(\mathbf{x}) = 1 \\ & \text{sinon } \sin^+(\mathbf{x}) = \max(\sin x^-, \sin x^+), \\ \sin^*(\mathbf{x}) & = [\sin^-(\mathbf{x}), \sin^+(\mathbf{x})]. \end{aligned} \tag{3.12}$$

De même, il est possible pour toutes les *fonctions élémentaires* comme \cos , \log , \arcsin , \arccos , \tan ... de calculer les fonctions d'inclusion minimales. Ces fonctions sont par ailleurs préprogrammées dans les langages sur les intervalles cités ci-dessus. Comme nous avons $\exp^*(\mathbf{x}) = \exp(\mathbf{x})$, $\sin^*(\mathbf{x}) = \sin(\mathbf{x})$, etc, les fonctions d'inclusion minimales des fonctions élémentaires seront désormais notées en caractère normal.

3.3.3. Fonction d'extension naturelle

Les situations où la fonction d'inclusion minimale f^* de f peut être calculée sont en fait très rares. Quand f résulte de compositions d'opérateur ou de fonctions élémentaires réelles, nous n'arrivons généralement pas à la calculer. Dans ce cas, une fonction d'inclusion, appelée *fonction d'extension naturelle*, peut être obtenue en recopiant l'expression formelle de f tout en remplaçant son argument x par l'intervalle \mathbf{x} et les fonctions élémentaires par les fonctions d'inclusion minimales associées.

Bien que Moore (1979) ait montré que $w(f(\mathbf{x})) - w(f(x)) = O(w(\mathbf{x}))$, la fonction d'extension

naturelle est en générale loin d'être minimale. Elle peut toutefois être sensiblement améliorée en remaniant l'expression formelle de f ou en utilisant le fait que l'intersection de fonctions d'inclusion est une fonction d'inclusion. Avant de calculer la fonction d'extension naturelle, il faudra donc se poser la question de comment organiser l'expression formelle de f pour obtenir une fonction d'inclusion aussi fine que possible.

Exemple 3.4 : Si $f_1(x) = x - x^2$, $f_2(x) = x(1 - x)$ et $f_3(x) = -(x - 1/2)^2 + 1/4$, nous avons $f_1 = f_2 = f_3$, bien que les expressions formelles de ces trois fonctions soient différentes. Les fonctions d'extension naturelles associées sont respectivement $f_1(\mathbf{x}) = \mathbf{x} - \mathbf{x}^2$, $f_2(\mathbf{x}) = \mathbf{x} * (1 - \mathbf{x})$ et $f_3(\mathbf{x}) = (\mathbf{x} - 1/2)^2 - 1/4$, qui conduisent à des résultats différents. Si par exemple $\mathbf{x} = [0, 1]$, alors,

$$\begin{aligned} f_1(\mathbf{x}) &= [0, 1] - [0, 1]^2 = [0, 1] - [0, 1] = [-1, 1], \\ f_2(\mathbf{x}) &= [0, 1] * (1 - [0, 1]) = [0, 1] * [0, 1] = [0, 1], \\ f_3(\mathbf{x}) &= -([0, 1] - 1/2)^2 + 1/4 = -[-1/2, 1/2]^2 - 1/4 = [1/4, 0] - 1/4 = [0, 1/4]. \quad \diamond \end{aligned}$$

Il est possible de montrer que f_3 est minimale car, dans son expression formelle, la variable x n'apparaît qu'une seule fois. Tel n'est pas le cas pour f_1 et f_2 car on effectue des opérations sur des intervalles associés aux mêmes quantités réelles en faisant comme si elles étaient indépendantes. Ce phénomène est connu sous le nom de *problème de dépendance*. Dans le cas général, pour avoir une fonction d'inclusion relativement précise, nous avons tout intérêt à réduire le nombre d'occurrences de la variable x dans l'expression de f . Parmi les nombreux artifices possibles, on peut factoriser f ou, lorsque f est polynomiale, l'exprimer sous la forme du reste quadratique. Par exemple, pour la fonction

$$f(x) = x^8 - 2x^7 + x^6 - x^5 - 4x^4 - 4x^3 + 4x^2 - 6x + 5, \quad (3.13)$$

le nombre d'occurrence de x peut passer de 8 à 5 en écrivant f sous la forme :

$$\begin{aligned} f(x) &= (x^8 - 2x^7 + x^6) + (-x^5 - 4x^4 - 4x^3) + (4x^2 - 6x + 5) \\ &= x^6(x^2 - 2x + 1) - x^3(x^2 + 4x + 4) + (4x^2 - 6x + 5) \\ &= x^6(x - 1)^2 - x^3(x + 2)^2 + (2x - 3)^2. \end{aligned} \quad (3.14)$$

3.3.4. Fonction d'inclusion par développement de Taylor

Si f est n fois différentiable, \bar{x} le centre de \mathbf{x} et $f^{(n)}$ une fonction d'inclusion de la dérivée

$n^{\text{ème}}$ de f alors, la *fonction d'inclusion centrée* est donnée par

$$f(\mathbf{x}) \subset f(\bar{\mathbf{x}}) + f'(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}}) + \frac{f''(\bar{\mathbf{x}})}{2!} (\mathbf{x} - \bar{\mathbf{x}})^2 + \dots + \frac{f^{(n)}(\bar{\mathbf{x}})}{n!} (\mathbf{x} - \bar{\mathbf{x}})^n \quad (3.15)$$

Dans le cas ou $n = 1$, nous avons donc

$$f(\mathbf{x}) = f(\bar{\mathbf{x}}) + f'(\bar{\mathbf{x}}) (\mathbf{x} - \bar{\mathbf{x}}) \quad (3.16)$$

Cette fonction d'inclusion est quasiment minimale lorsque $w(\mathbf{x})$ tend vers 0. Hansen a prouvé que $w(f(\mathbf{x})) - w(f(\bar{\mathbf{x}})) = O(w^2(\mathbf{x}))$. Pour s'en convaincre, il suffit d'imaginer un agrandissement sur la fonction autour de \mathbf{x} pour \mathbf{x} infiniment petit. Comme $f'(\bar{\mathbf{x}})$ est infiniment petit, $w(f(\mathbf{x}))/w(f(\bar{\mathbf{x}}))$ tend vers 1, ce qui n'est pas le cas pour une fonction d'extension naturelle lorsqu'intervient un problème de dépendance. Cette forme est donc à recommander pour les algorithmes qui font intervenir des pavés de plus en plus petits. Notons que cette fonction d'inclusion existe aussi lorsque f est une fonction vectorielle.

3.4. Cas multidimensionnel

3.4.1. Pavés

Dans le cas scalaire, les intervalles permettaient de remplacer les valeurs ponctuelles de nombres réels par des sous-ensembles de \mathbb{R} . La structure simple des intervalles a permis le développement d'un calcul sur les intervalles généralisant le calcul sur les réels, autorisant ainsi une analyse globale d'un ensemble infini de réels avec un nombre fini d'opérations. De la même façon, le calcul vectoriel peut être généralisé en remplaçant les valeurs ponctuelles des vecteurs de \mathbb{R}^n par des sous-ensembles de \mathbb{R}^n de structure suffisamment simple pour permettre la généralisation du calcul vectoriel. Dans la littérature, les structures d'ensemble qui ont été considérées sont les pavés (Moore, 1979), les ellipsoïdes et les polytopes (voir Moore, 1979, p 17). Parmi ces trois types d'ensemble, nous avons choisi d'utiliser les pavés en raison de leur facilité d'utilisation et des propriétés qui en découlent.

Un *pavé*, ou *intervalle vectoriel*, \mathbf{x} de \mathbb{R}^n est formé par le produit cartésien de n intervalles réels. La notion d'intervalle vectoriel s'étend sans difficulté aux matrices. Nous noterons \mathbf{A} l'intervalle matriciel associé à la matrice \mathbf{A} . Les pavés seront notés indifféremment :

$$\mathbf{x} = [x_1^-, x_1^+] \times \dots \times [x_n^-, x_n^+] = \mathbf{x}_1 \times \dots \times \mathbf{x}_n = [\mathbf{x}^-, \mathbf{x}^+] = \begin{pmatrix} [x_1^-, x_1^+] \\ \dots \\ [x_n^-, x_n^+] \end{pmatrix}, \quad (3.17)$$

où les vecteurs \mathbf{x}^- et \mathbf{x}^+ ont pour coordonnées respectives $(x_1^-, x_2^-, \dots, x_n^-)$ et $(x_1^+, x_2^+, \dots, x_n^+)$. Les intervalles scalaires $\mathbf{x}_i = [x_i^-, x_i^+]$ sont appelés *composantes* du pavé \mathbf{x} . L'ensemble de tous les pavés de \mathbb{R}^n sera noté \mathbb{IR}^n . Les vecteurs \mathbf{x} sont des pavés *dégénérés* avec $\mathbf{x}^- = \mathbf{x}^+ = \mathbf{x}$. La *longueur* $w(\mathbf{x})$ du pavé $\mathbf{x} \in \mathbb{IR}^n$ est donnée par

$$w(\mathbf{x}) \triangleq \max_i \{x_i^+ - x_i^-\}. \quad (3.18)$$

Un *plan principal* d'un pavé \mathbf{x} est un plan de symétrie de ce pavé qui est normal à un axe i correspondant à un côté de longueur maximale, c'est-à-dire $i \in \{j \mid w(\mathbf{x}) = w(\mathbf{x}_j)\}$. Bien sûr, un tel plan n'est pas nécessairement unique. Pourtant, dans le but d'alléger la rédaction, nous ferons comme s'il l'était. La figure 3.1 représente un pavé de dimension 2, avec son plan principal qui, dans ce cas particulier, est une droite.

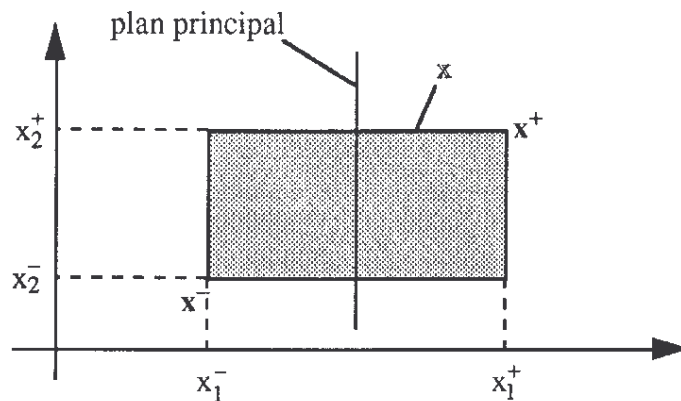


Fig. 3.1 Pavé \mathbf{x} avec son plan principal.

Le *pavé enveloppe* $[A]$ d'un ensemble borné A de \mathbb{R}^n est le plus petit pavé (au sens de l'inclusion) de \mathbb{IR}^n qui contient A .

$$[A] \triangleq \bigcap \{ \mathbf{x} \in \mathbb{IR}^n \mid A \subset \mathbf{x} \}. \quad (3.19)$$

3.4.2. Calcul vectoriel sur les intervalles

Dans les espaces vectoriels réels, les deux seules opérations autorisées sont l'addition de deux vecteurs et la multiplication d'un vecteur par une constante réelle, aussi appelée multiplication externe. Ces deux opérations vectorielles peuvent se généraliser aux pavés et permettront ainsi de manipuler les pavés au même titre que des vecteurs. Il est possible de définir l'addition de deux pavés comme suit :

$$\mathbf{x} + \mathbf{y} \triangleq \{\mathbf{x} + \mathbf{y} \mid \mathbf{x} \in \mathbf{x} \text{ et } \mathbf{y} \in \mathbf{y}\} = [\mathbf{x}^- + \mathbf{y}^-, \mathbf{x}^+ + \mathbf{y}^+], \quad (3.20)$$

ce qui garantit que si $\mathbf{x} \in \mathbf{x}$ et $\mathbf{y} \in \mathbf{y}$ alors $\mathbf{x} + \mathbf{y} \in \mathbf{x} + \mathbf{y}$. D'une façon similaire, la multiplication externe peut être définie par

$$\lambda \cdot \mathbf{x} \triangleq \{\lambda \cdot \mathbf{x} \mid \mathbf{x} \in \mathbf{x}\} = \begin{cases} [\lambda \mathbf{x}^-, \lambda \mathbf{x}^+] & \text{si } \lambda \geq 0 \\ [\lambda \mathbf{x}^+, \lambda \mathbf{x}^-] & \text{si } \lambda \leq 0 \end{cases} \quad (3.21)$$

3.4.3. Fonctions d'inclusion vectorielles

3.4.3.1. Définitions

Soit une fonction vectorielle $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$, la fonction ensembliste $f : \mathbb{IR}^n \rightarrow \mathbb{IR}^p$ sera appelée *fonction d'inclusion* de f si et seulement si pour tout pavé \mathbf{x} elle vérifie $f(\mathbf{x}) \subset f(\mathbf{x})$. Comme c'est le cas à une dimension, l'intersection de deux fonctions d'inclusion est bien évidemment une fonction d'inclusion ; la fonction d'inclusion f sera dite *monotone* si, pour tout couple de pavés \mathbf{x} et \mathbf{y} , on a

$$\mathbf{x} \subset \mathbf{y} \Rightarrow f(\mathbf{x}) \subset f(\mathbf{y}) ; \quad (3.22)$$

elle sera *convergente* si, pour toute suite de pavés \mathbf{x}_k , on a

$$w(\mathbf{x}_k) \rightarrow 0 \Rightarrow w(f(\mathbf{x}_k)) \rightarrow 0. \quad (3.23)$$

Il n'existe donc pas de fonction d'inclusion convergente pour f si f n'est pas continue. Dans la suite, les fonctions d'inclusion considérées seront par défaut monotones et convergentes.

3.4.3.2. Fonctions d'inclusion minimale

Parmi toutes les fonctions d'inclusion f de la fonction vectorielle $f: \mathbb{R}^n \rightarrow \mathbb{R}^p$, il en existe une seule qui soit minimale au sens de l'inclusion. C'est la *fonction d'inclusion minimale*, notée f^* et définie par

$$f^*: \mathbb{R}^n \rightarrow \mathbb{IR}^p; \mathbf{x} \rightarrow [\{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{x}\}]. \quad (3.24)$$

Le pavé $f^*(\mathbf{x})$ est donc le plus petit pavé de \mathbb{IR}^p qui contient $f(\mathbf{x})$, c'est-à-dire le pavé enveloppe de l'ensemble $f(\mathbf{x})$. La fonction d'inclusion minimale est nécessairement monotone, et si f est continue, elle est convergente. La figure 3.2 illustre la définition de la fonction d'inclusion minimale et celle de la fonction d'inclusion dans le cas général. Nous savons parfois calculer la fonction d'inclusion minimale, comme le montrent les deux exemples suivants.

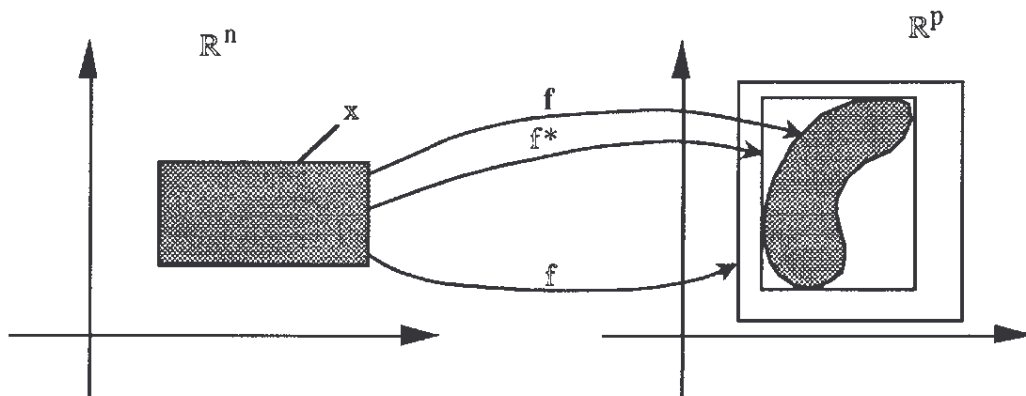


Fig. 3.2 Image directe f , fonction d'inclusion minimale f^* et fonction d'inclusion f non minimale.

Exemple 3.5 : Somme de deux exponentielles.

Considérons la fonction $f: \mathbb{R}^4 \rightarrow \mathbb{R}; \mathbf{x} \rightarrow x_1 \exp(x_2) + x_3 \exp(x_4)$. D'après la monotonie de la fonction exponentielle, nous avons $\exp(x_i) = \exp(x_i) = [\exp(x_i^-), \exp(x_i^+)]$. Comme les variables x_i n'apparaissent chacune qu'une fois dans l'expression formelle de f , la fonction d'inclusion minimale de f est donc $f^*: \mathbb{IR}^4 \rightarrow \mathbb{IR}; \mathbf{x} \rightarrow x_1 * \exp(x_2) + x_3 * \exp(x_4)$. Plaçons nous maintenant dans le cas particulier où $x_1 > 0$ et $x_3 > 0$, alors $f([\mathbf{x}^-, \mathbf{x}^+]) = [x_1^- \exp(x_2^-) + x_3^- \exp(x_4^-), x_1^+ \exp(x_2^+) + x_3^+ \exp(x_4^+)]$. Il est en effet évident d'après les propriétés de l'arithmétique sur les intervalles que f^* est donc une fonction d'inclusion. De

plus, pour toute fonction d'inclusion f de f , nous avons par définition $f(\mathbf{x}) \subset f(\mathbf{x}) \Rightarrow \{f(\mathbf{x}^-), f(\mathbf{x}^+)\} \subset f(\mathbf{x}) \Rightarrow f^*([\mathbf{x}^-, \mathbf{x}^+]) = [f(\mathbf{x}^-), f(\mathbf{x}^+)] \subset f(\mathbf{x})$. La fonction d'inclusion f^* est donc minimale. \diamond

Exemple 3.6 : Sinusoïde.

Considérons la fonction $f: \mathbb{R}^2 \rightarrow \mathbb{R}; \mathbf{x} \rightarrow x_1 \sin x_2$. En constatant que les deux paramètres x_1 et x_2 apparaissent indépendamment dans l'expression de f , nous trouvons immédiatement que la fonction d'inclusion minimale de f est $f^*: \mathbb{R}^2 \rightarrow \mathbb{R}; \mathbf{x}_1 * \sin(\mathbf{x}_2)$. \diamond

Dans ces deux exemples, la fonction f prend ses valeurs dans \mathbb{R} . Dans le cas où f prend ses valeurs dans \mathbb{R}^p , la fonction d'inclusion minimale f de f a pour fonctions coordonnées les fonctions d'inclusion minimale f_i des fonctions coordonnées f_i de f .

Exemple 3.7 : Problème des exponentielles approximantes.

Calculons la fonction d'inclusion minimale de la fonction $e(\mathbf{p})$ définie par (2.20). Comme $p_1 \geq 0$, la fonction d'inclusion minimale pour $e(\mathbf{p})$ est donnée par

$$e(\mathbf{p}) = \left(\begin{array}{l} [\min_{0 \leq t \leq 1} \{t^2 + 2t + 1 - p_1^+ \exp(p_2^+ t)\}, \min_{0 \leq t \leq 1} \{t^2 + 2t + 1 - p_1^- \exp(p_2^- t)\}] \\ [\max_{0 \leq t \leq 1} \{t^2 + 2t + 1 - p_1^+ \exp(p_2^+ t)\}, \max_{0 \leq t \leq 1} \{t^2 + 2t + 1 - p_1^- \exp(p_2^- t)\}] \end{array} \right) \quad (3.25)$$

Une évaluation de $e(\mathbf{p})$ demande donc la résolution de quatre problèmes d'optimisation monodimensionnelle simples. Chacun de ces problèmes peut être résolu en utilisant n'importe quelle méthode d'optimisation locale monodimensionnelle deux fois (une fois initialisée à $t = 0$, et une autre fois à $t = 1$). Nous atteindrons ainsi toujours l'extremum global sur $[0, 1]$, comme on peut s'en apercevoir en dérivant la fonction à minimiser. \diamond

Exemple 3.8 : Problème de Milanese et Vicino.

Comme $p_1 \geq 0$, $p_3 \leq 0$ et $t > 0$, la fonction d'inclusion de $y_m(\mathbf{p})$ définie en (2.29) est

$$y_m(\mathbf{p}) = [p_1^- \exp(-p_2^+ t) + p_3^- \exp(-p_4^- t), p_1^+ \exp(-p_2^- t) + p_3^+ \exp(-p_4^+ t)]. \quad (3.26)$$

Exemple 3.9 : Problème de la région de confiance triangulaire.

La loi de probabilité $\pi : \mathbb{R}^2 \rightarrow \mathbb{R}$ introduite en (2.41) et définie par

$$\pi(\mathbf{p}) = \begin{cases} \exp(-p_1) * \exp(-p_2) & \text{si } \mathbf{p} \geq \mathbf{0} \\ 0 & \text{sinon} \end{cases} \quad (3.27)$$

admet pour fonction d'inclusion minimale $\pi^*(\mathbf{p}) = [\pi^-(\mathbf{p}), \pi^+(\mathbf{p})]$, avec

$$\begin{aligned} \pi^-(\mathbf{p}) &= \pi(p_1^+, p_2^+) \text{ si } p_1^- \text{ ou } p_2^- \geq 0 \text{ et } 0 \text{ sinon,} \\ \pi^+(\mathbf{p}) &= \pi(\sup(0, p_1^-), \sup(0, p_2^-)) \text{ si } p_1^+ \text{ ou } p_2^+ \geq 0 \text{ et } 0 \text{ sinon.} \end{aligned} \quad (3.28)$$

3.4.3.3. Fonctions d'extension naturelles

Quand la fonction f utilisée résulte de compositions d'opérations ou de fonctions élémentaires réelles, qu'elle soit décrite par une formule explicite ou par un algorithme ne contenant pas de branchements conditionnels comme *if* ou *while*, il est possible de construire une fonction d'inclusion, appelée *fonction d'extension naturelle*, en recopiant l'expression formelle de f (ou l'algorithme décrivant f), en remplaçant les coordonnées x_i de l'argument \mathbf{x} par les composantes x_i du pavé \mathbf{x} , tous les opérateurs arithmétiques réels par les opérateurs arithmétiques sur les intervalles correspondants et les fonctions élémentaires par les fonctions d'inclusion minimales associées. Si dans l'expression formelle de chacune des fonctions coordonnées, chaque composante x_i de \mathbf{x} n'apparaît qu'une seule fois, la fonction d'extension naturelle est égale à la fonction d'inclusion minimale.

Donnons maintenant, à titre d'exemple, quelques fonctions d'extension naturelle de fonctions vectorielles intervenant dans la définition des problèmes ensemblistes proposés au chapitre 2.

Problème des conditions initiales : Un algorithme par intervalles exprimant la fonction d'extension naturelle $\mathbf{e}(\mathbf{p})$ de la fonction $\mathbf{e}(\mathbf{p})$, donné en (2.37) et (2.38) sous forme d'algorithme, est donc

```

x1(0) := p1 ; x2(0) := p2 ;
For k = 0 to 9 do
  begin
    ym(k) := x1^2(k) - x2(k) ;
    e(k) := y(k) - ym(k) ;
    x1(k+1) := cos(x1(k) * x2(k)) ;
    x2(k+1) := 3 x1(k) - sin(x2(k)) ;
  end ;

```

(3.29)

L'intervalle $e(k)$ correspond à la $(k + 1)$ ^{ième} composante de la fonction $e(p)$.

Problème des palmes : Soit f la fonction utilisée dans le problème de stabilité des palmes. Elle est donnée en (2.59) par

$$f : \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \rightarrow \begin{pmatrix} p_1 p_2 \\ \sin (p_1 p_2) \\ p_1^2 \sin (p_1 p_2) - p_1 p_2 \end{pmatrix}. \quad (3.30)$$

La fonction d'extension naturelle de f est :

$$f : p \rightarrow \begin{pmatrix} p_1 * p_2 \\ \sin (p_1 * p_2) \\ p_1^2 \sin (p_1 * p_2) - p_1 * p_2 \end{pmatrix}. \quad (3.31)$$

Pour mieux comprendre comment utiliser une telle expression, considérons le cas particulier où $p = [-1, 1] \times [0, 2]$. Dans ce cas

$$f(p) = \begin{pmatrix} [-1, 1] * [0, 2] \\ \sin ([-1, 1] * [0, 2]) \\ [-1, 1]^2 \sin ([-1, 1] * [0, 2]) - [-1, 1] * [0, 2] \end{pmatrix} \quad (3.32)$$

$$= \begin{pmatrix} [-2, 2] \\ \sin ([-2, 2]) \\ [0, 1] \sin ([-2, 2]) - [-2, 2] \end{pmatrix} = \begin{pmatrix} [-2, 2] \\ [-1, 1] \\ [-1, 1] - [-2, 2] \end{pmatrix} = \begin{pmatrix} [-2, 2] \\ [-1, 1] \\ [-3, 3] \end{pmatrix}. \quad (3.33)$$

L'ensemble image $f(p)$ est donc inclus dans le pavé $f(p)$. Le calcul du pavé-enveloppe de

$f(\mathfrak{p})$ demanderait quatre optimisations non-linéaires sur le pavé \mathfrak{p} . La non-minimalité de la fonction d'extension naturelle de f provient du fait que dans l'expression de f_3 , les variables p_1 et p_2 apparaissent deux fois. Par contre, les deux premières fonctions coordonnées de f sont minimales.

Problème du col de cygne : Rappelons que dans ce problème d'analyse de stabilité défini dans le chapitre 2, la fonction f à inverser introduite en (2.69) est donnée par

$$f(\mathfrak{p}) = \begin{pmatrix} -T^{-1} \\ 2z + T^{-1} \\ T^{-2} (2z - 6 + T) (2zT + 1) + T^{-1} \\ -T^{-3} (2z - 6 + T) (2zT + 1) - T^{-2} + T^{-3} (2zT + 1)^2 \end{pmatrix}, \quad (3.34)$$

où $\mathfrak{p} = (T, z)^T$. La fonction d'extension naturelle associée à cette expression formelle de f est

$$f(\mathfrak{p}) = \begin{pmatrix} -T^{-1} \\ 2z + T^{-1} \\ T^{-2} (2z - 6 + T) (2zT + 1) + T^{-1} \\ -T^{-3} (2z - 6 + T) (2zT + 1) - T^{-2} + T^{-3} (2zT + 1)^2 \end{pmatrix}. \quad (3.35)$$

Cette fonction d'inclusion est très loin d'être minimale du fait de la complexité de l'expression formelle de f .

Notons que, en calculant le vecteur signe du pavé $f(\mathfrak{p})$, nous obtenons des conditions suffisantes de stabilité et d'instabilité pour tout un pavé de paramètres intervenant non-linéairement dans la détermination des coefficients du polynôme caractéristique. Ces conditions deviennent *presque* nécessaires et suffisantes lorsque la longueur du pavé \mathfrak{p} tend vers zéro. Les seules conditions de ce type qui existaient, à notre connaissance, dans la littérature étaient pour une classe très limitée de dépendances paramètres-coefficients du polynômes caractéristiques.

3.4.3.4. Test d'inclusion

Un test sur un élément \mathbf{x} de \mathbb{R}^n est une fonction binaire $t : \mathbb{R}^n \rightarrow \{\text{vrai, faux}\}$. Nous dirons que $t : \mathbb{R}^n \rightarrow \{\text{vrai, faux, indéterminé}\}$ est un *test d'inclusion* de t si les deux implications suivantes sont satisfaites

$$\forall \mathbf{x}, t(\mathbf{x}) = \text{vrai} \Rightarrow t(\mathbf{x}) = \text{vrai}, \quad (3.36)$$

$$\forall \mathbf{x}, t(\mathbf{x}) = \text{faux} \Rightarrow t(\mathbf{x}) = \text{faux}. \quad (3.37)$$

Ce test d'inclusion sera dit *convergent* si pour presque tout $\mathbf{x} \in \mathbb{R}^n$, $\exists \varepsilon > 0$ tel que le pavé $\mathbf{x}_\varepsilon = \mathbf{x} + \varepsilon [-\mathbf{1}, \mathbf{1}]$, où $\mathbf{1}$ est le vecteur de \mathbb{R}^n dont tous les éléments sont égaux à 1, soit tel que $t(\mathbf{x}_\varepsilon) \neq \text{indéterminé}$.

Soit A un compact de \mathbb{R}^n et soit $t_A : \mathbb{R}^n \rightarrow \{\text{vrai, faux}\}$ le test défini comme suit :

$$\begin{aligned} t_A(\mathbf{x}) = \text{vrai} &\Leftrightarrow \mathbf{x} \in A, \\ t_A(\mathbf{x}) = \text{faux} &\Leftrightarrow \mathbf{x} \notin A. \end{aligned} \quad (3.38)$$

Un test d'inclusion t_A de t_A est appelé *test d'inclusion pour l'ensemble A*. Ce test d'inclusion servira par exemple lorsque l'ensemble à analyser A n'est pas défini par un ensemble d'inégalités, mais par un test d'appartenance. Un test d'inclusion convergent pour cet ensemble permettra alors de caractériser l'ensemble A de façon garantie.

3.5. Analyse par intervalles

Dans le sens courant, l'*analyse* d'un problème consiste à discerner les éléments de sa solution par une série de raisonnements successifs cherchant à le décomposer en un ensemble de problèmes élémentaires. En mathématique, l'analyse est souvent assimilée au calcul infinitésimal dans le sens où de nombreux problèmes complexes deviennent très simples si on les décompose en une infinité de problèmes infinitésimaux. De nombreux problèmes issus de la physique ont pour solution un point ou un sous-ensemble de \mathbb{R}^n . Recouvrir le domaine de recherche *a priori* par un ensemble fini de pavés, afin d'encadrer la solution du problème restreint à chacun des pavés puis d'en déduire la solution globale du problème initial, permet de résoudre de nombreux problèmes difficilement solubles avec les méthodes classiques. Ce principe d'analyse fondé sur l'arithmétique sur les intervalles et sur

le concept de fonction d'inclusion est appelé *analyse par intervalles*. Dans les méthodes et algorithmes qui en découlent, les pavés manipulés deviennent généralement de plus en plus petits et l'on se trouve ainsi confronté à des problèmes de convergence et de calcul de limites caractéristiques des problèmes infinitésimaux. Nous allons maintenant décrire deux algorithmes permettant de trouver des solutions aux problèmes de résolution d'égalités linéaires perturbées et d'équations non-linéaires. Ces algorithmes sont des outils fréquemment utilisés par les algorithmes d'analyse par intervalles. Nous illustrerons enfin cette philosophie en présentant le premier algorithme d'optimisation globale et garantie apparu dans la littérature qui utilise l'analyse par intervalles.

3.5.1. Résolution des équations linéaires perturbées par Gauss-Seidel

Une équation linéaire d'ordre fini n s'écrit sous la forme

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad (3.39)$$

Dans de nombreuses applications, les éléments de la matrice \mathbf{A} et du vecteur \mathbf{b} sont connus avec une certaine précision et peuvent être bornés par une matrice intervalle \mathbf{A} et un vecteur intervalle \mathbf{b} . L'équation linéaire perturbée s'écrit alors

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}. \quad (3.40)$$

Elle admet pour solution l'ensemble

$$\mathbb{X} = \{ \mathbf{x} \mid \mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{A} \in \mathbf{A} \text{ et } \mathbf{b} \in \mathbf{b} \} \quad (3.41)$$

La forme de \mathbb{X} est souvent très compliquée (\mathbb{X} peut être non convexe) et nous allons nous limiter à trouver un pavé \mathbf{x} contenant \mathbb{X} de façon certaine. Nous supposons que tous les éléments a_{ii} de la diagonale de \mathbf{A} ne contiennent pas zéro. Si ce n'est pas le cas, on peut reconditionner le système, par exemple en multipliant (3.40) de chaque côté par l'inverse de la matrice centre de \mathbf{A} , si celle-ci est inversible. La *méthode de Gauss-Seidel*, dont nous allons donner le principe, semble être la plus adaptée à la résolution de ce problème (Hansen, 1992). Supposons que $\mathbb{X} \subset \mathbf{x}$, où \mathbf{x} est un pavé connu en début d'itération. La $i^{\text{ème}}$ équation du système s'écrit

$$a_{i1} \cdot x_1 + \dots + a_{in} \cdot x_n = b_i \quad (3.42)$$

comme $0 \notin a_{ii}$, posons

$$y_i = \frac{a_{i1} \cdot x_1 + \dots + a_{i,i-1} \cdot x_{i-1} + a_{i,i+1} \cdot x_{i+1} + \dots + a_{in} \cdot x_n}{a_{ii}}, \quad (3.43)$$

alors l'encadrement proposé par le pavé

$$x' = (x_1, \dots, x_i \cap y_i, \dots, x_n) \quad (3.44)$$

est plus précis que celui fourni par x . Cette itération doit se faire pour $i = 1, \dots, n$ et ceci plusieurs fois. Pour les conditions de convergence et les améliorations possibles, voir Hansen (1992).

3.5.2. Méthode de Newton par intervalles

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, chercher un zéro de f revient à résoudre l'équation $f(x) = 0$. La *méthode de Newton par intervalles* est un algorithme $n(f, x)$ qui génère un pavé contenant à coup sûr tous les zéros de f sur le pavé x . L'avantage essentiel de la méthode est que si le pavé initial x est suffisamment petit, et si x n'admet qu'une seule solution alors la méthode va converger très rapidement.

Si $x_0 \in x$ alors, d'après le théorème de la valeur moyenne, pour tout $x \in x$, il existe $x_c \in x$ tel que

$$f(x) = f(x_0) + J(x_c)(x - x_0), \quad (3.45)$$

où $J(x_c)$ désigne la matrice Jacobienne de f au point x_c . Si $f(x_0) = 0$ et $x = \bar{x} = \text{milieu}(x)$, alors il existe $x_c \in x$ tel que

$$f(\bar{x}) = J(x_c)(\bar{x} - x_0) \Leftrightarrow J(x_c) \cdot x_0 = J(x_c) \cdot \bar{x} - f(\bar{x}). \quad (3.46)$$

La solution x_0 satisfait donc l'équation linéaire perturbée

$$J(x) \cdot x_0 = J(x) \cdot \bar{x} - f(\bar{x}), \quad (3.47)$$

soluble par la méthode de Gauss-Seidel. Nous noterons x_0 l'encadrement ainsi obtenu. Remarquons qu'il existe des méthodes de dérivation automatique (Rall, 1981) pour obtenir J

lorsque f est donnée par un algorithme. Ces méthodes génèrent un algorithme associé à la jacobienne J à partir du laquelle il est possible de trouver un algorithme associé à la fonction d'inclusion J . L'algorithme de Newton par intervalles, donné ci-dessous, génère un encadrement pour la solution de l'équation $f(x) = 0$ à partir d'un pavé de recherche initial x . Il se présente sous la forme suivante

- Pas 1 Résoudre, par exemple par la méthode de Gauss Seidel, l'équation linéaire perturbée (3.47) et appeler x_0 l'encadrement obtenu pour la solution.
- Pas 2 $x := x \cap x_0$. Si le critère d'arrêt n'est pas satisfait alors aller au pas 1.

Une fois cette procédure terminée, x contient à coup sûr tous les zéros de f contenus dans le pavé initial. Si x suffisamment petit et si x ne contient qu'une seule solution, alors, la convergence est de type quadratique dans le sens où

$$w(x(k+1)) = \alpha \cdot w^2(x(k)) \quad (3.48)$$

où α est fixé par les données du problème. La figure 3.3 illustre cette rapidité de convergence dans le cas réel pour x très petit et une fonction presque affine sur x . Si nous faisons agir une itération de l'algorithme de Newton par intervalles sur le pavé x , nous obtenons un pavé contenant l'intersection de la zone hachurée avec le pavé x .

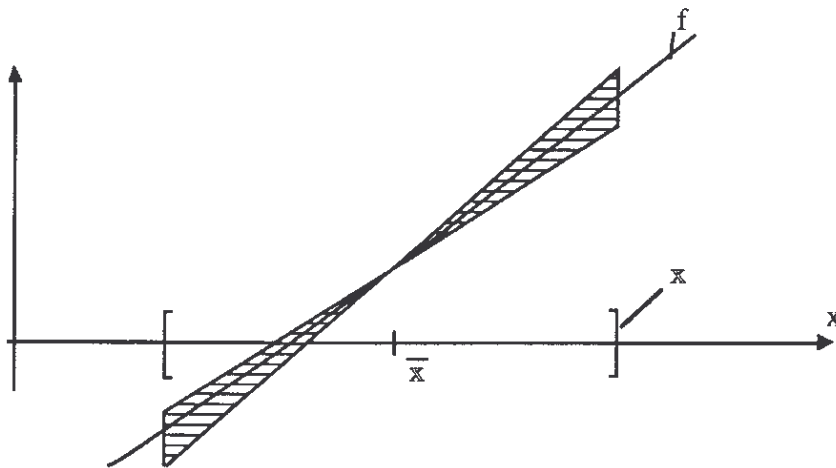


Fig. 3.3 Illustration de la rapidité de convergence de la méthode de Newton par intervalles.

Si l'algorithme aboutit à un pavé x vide, aucune solution n'existe sur le pavé initial. Si au

pas 2, on obtient $\mathbf{x}_0 \subset \text{int}(\mathbf{x})$ alors, f possède une solution unique sur le pavé initial. Cette méthode est loin d'être satisfaisante car elle peut rester bloquée à un pavé (comme par exemple lorsque \mathbf{x} renferme 2 zéros). Cependant, elle n'oublie jamais de solutions, et lorsqu'elle converge vers un zéro de f , elle le fait avec la rapidité d'une optimisation locale de type Newton. Des algorithmes plus globaux, comme par exemple celui donné au paragraphe suivant, seront alors chargés d'isoler un zéro dans un pavé assez petit pour que la méthode de Newton par intervalles puisse donner en un temps très court un zéro en toute garantie.

3.5.3. Algorithme d'optimisation globale de Moore-Skelboe

A l'origine, le calcul sur les intervalles a été développé pour effectuer du calcul numérique en précision finie tout en obtenant un résultat garanti. Un grand bond a été fait lorsque Skelboe (1974) a combiné les idées de Moore sur le calcul sur les intervalles et les fonctions d'inclusion avec les idées d'optimisation de type *branch and bound* pour en déduire un algorithme, dit de Moore-Skelboe, déterministe d'optimisation globale sans contrainte. Cet algorithme vise à minimiser la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ sur un pavé de recherche initial $\mathbf{x}(0)$ (éventuellement très grand). Nous présentons cet algorithme non pas parce que nous l'avons utilisé pour résoudre nos problèmes ensemblistes (pour lesquels il serait peu efficace), mais tout d'abord pour son intérêt historique et aussi parce qu'il nous a servi de support pour tous les algorithmes que nous avons pu développer par la suite.

Le principe de cet algorithme est de partitionner le pavé de recherche $\mathbf{x}(0)$ en pavés de longueur variable. A chaque itération, la recherche est effectuée sur le pavé \mathbf{x} sur lequel f a la plus petite borne inférieure, ce qui augmente les chances d'y trouver l'optimum f^* . Un minorant y^- de f sur un pavé \mathbf{x} est déterminé à l'aide d'une fonction d'inclusion f de f . Les pavés \mathbf{x} susceptibles de contenir l'optimum, ainsi que le minorant y^- de f calculé sur chacun d'eux forment un ensemble de couples (\mathbf{x}, y^-) qui seront rangés dans la liste L .

Algorithme de Moore-Skelboe

- Pas 1 $\mathbf{x} := \mathbf{x}(0)$.
- Pas 2 $\mathbf{y} := f(\mathbf{x})$
- Pas 3 Poser $L := ((\mathbf{x}, \mathbf{y}^-))$.
- Pas 4 Bisséquer \mathbf{x} suivant son axe principal en deux pavés \mathbf{x}_1 et \mathbf{x}_2 .
- Pas 5 $\mathbf{y}_1 := f(\mathbf{x}_1)$; $\mathbf{y}_2 := f(\mathbf{x}_2)$.
- Pas 6 Enlever $(\mathbf{x}, \mathbf{y}^-)$ de la liste L .
- Pas 7 Insérer les paires $(\mathbf{x}_1, \mathbf{y}_1^-)$ et $(\mathbf{x}_2, \mathbf{y}_2^-)$ dans L de telle façon que toutes les paires soient triées dans l'ordre croissant de leur second membre.
- Pas 8 Appeler la première paire $(\mathbf{x}, \mathbf{y}^-)$.
- Pas 9 Aller au pas 4.

Les conditions de convergence de cet algorithme ont été étudiées dans Ratschek et Rokne (1988). Lorsque ces conditions sont vérifiées, la variable \mathbf{y}^- tend par valeurs inférieures vers le minimum global f^* . Par contre, la longueur de la liste L tend vers l'infini, ce qui rend très vite laborieuse la manipulation d'une telle liste qui doit rester triée en permanence. Notons enfin que si à l'optimum global, le gradient de f s'annule, c'est-à-dire $\mathbf{f}'(\mathbf{x}^*) = \mathbf{0}$, l'utilisation de méthode de Newton par intervalles peut améliorer de façon spectaculaire la convergence.

3.6. Conclusion

Les pavés ont une structure simple qui permet de les manipuler comme des vecteurs à l'aide du calcul sur les intervalles qui vient d'être décrit. La nature ensembliste des pavés autorise un calcul garanti sur une infinité non-dénombrable de vecteurs avec un nombre fini d'opérations. Nous avons présenté le calcul sur les intervalles (qu'il soit mono ou multi-dimensionnel), ainsi que les propriétés qui en découlent. Les fonctions d'inclusion permettent de généraliser les fonctions vectorielles et sont telles que l'image d'un pavé est aussi un pavé. Nous avons donné des méthodes pour calculer ces fonctions d'inclusion, et présenté l'intérêt de ce concept pour une analyse garantie des résultats du calcul numérique. Nous avons donné le principe de l'analyse par intervalles qui consiste à partitionner l'espace de recherche *a priori* du problème considéré en un ensemble fini de pavés pour ensuite étudier leurs caractéristiques numériques à l'aide du calcul sur les intervalles. Par élimination des pavés ne pouvant contenir la solution, nous pouvons approcher la solution cherchée de

façon garantie. Nous avons présenté la résolution d'équations linéaires perturbées par Gauss-Seidel ainsi que la résolution de systèmes d'équations non linéaires par la méthode de Newton par intervalles. Ces outils s'avèrent indispensables pour l'élaboration d'algorithmes efficaces fondés sur l'analyse par intervalles. L'algorithme d'optimisation globale de Moore-Skelboe nous a permis de montrer comment l'analyse par intervalles s'appliquait de façon garantie à la résolution de problèmes globaux.

Nous allons, dans les prochains chapitres, utiliser l'analyse par intervalles pour résoudre les problèmes ensemblistes d'estimation et de commande proposés au le chapitre 2. Afin de faciliter la compréhension et l'analyse de la convergence des algorithmes que nous serons amenés à développer, nous allons présenter au chapitre suivant le concept de sous-pavage utilisé pour approximer et encadrer les ensembles compacts de \mathbb{R}^n . Les notions de topologie sur l'ensemble des compacts, indispensables pour les définitions de convergence des algorithmes, seront également introduites.

CHAPITRE 4

ENCADREMENT ENSEMBLISTE

ET SOUS-PAVAGES

4.1. Introduction

Un problème ensembliste a pour solution un ensemble \mathbb{X} qui peut généralement être défini de façon exacte, par exemple par un ensemble d'inégalités non linéaires. Cependant, la représentation exacte de cet ensemble ainsi obtenue est en général inexploitable. En effet, si \mathbb{X} est donné par un ensemble d'inéquations, comment savoir si \mathbb{X} est vide ou connexe ? Quel est son volume, sa forme ? Nous pouvons alors considérer que l'ensemble \mathbb{X} , bien que défini de façon exacte, reste inconnu puisque nous ne savons pas répondre aux questions les plus élémentaires à son sujet. Une autre approche consiste à approximer l'ensemble solution \mathbb{X} par un ensemble exploitable, par exemple, une union d'ellipsoïdes, une union de pavés, ou un polyèdre. Nous pourrions alors approcher certaines caractéristiques de \mathbb{X} en calculant la caractéristique correspondante de l'ensemble exploitable l'approximant. Les ensembles approximants que nous allons considérer sont formés par des unions de pavés, que nous appellerons *sous-pavages*. L'obtention de ces sous-pavages se fera bien sûr à l'aide de l'analyse par intervalles. Nous commencerons par définir certaines distances sur les compacts afin de disposer d'une topologie sur les ensembles et de définir plus rigoureusement ce que nous entendons par approximation ensembliste. Lorsque l'on veut faire du calcul garanti, on ne se satisfait pas d'une approximation, mais on cherche plutôt à encadrer les variables manipulées, qu'elles soient ponctuelles ou ensemblistes, et à effectuer les calculs sur ces encadrements. Pour le calcul réel, nous utilisons des intervalles pour encadrer les réels, et manipulons ces intervalles pour faire du calcul garanti. Nous verrons qu'il est possible, de façon similaire, d'encadrer les sous-ensembles de \mathbb{R}^n par deux sous-pavages avec autant de précision que désiré. Cet encadrement définira un voisinage de l'ensemble considéré dans l'espace des compacts, que nous appellerons *intercompact*. Le calcul numérique ensembliste pourra donc se faire par calcul sur des sous-pavages ou sur les intercompacts qui permettent un encadrement des ensembles manipulés. Le calcul par intervalles sera l'outil utilisé pour trouver de tels sous-pavages encadrant un ensemble défini analytiquement par le problème dont il est solution. En exploitant les notions de

topologie développées en début de chapitre, nous étudierons, par une approche géométrique, les discontinuités pouvant survenir dans les problèmes d'inversion ensembliste.

4.2. Distances sur l'ensemble des compacts

Diverses distances entre les compacts seront utiles pour l'étude de la convergence des algorithmes ensemblistes présentés aux chapitres 5 et 6.

4.2.1. Distance de Hausdorff

Munissons l'ensemble \mathbb{R}^n de la distance L_∞ induite par la norme classique L_∞ définie par

$$L_\infty(\mathbf{x}, \mathbf{y}) \triangleq \max_{i \in \{1, \dots, n\}} (|y_i - x_i|). \quad (4.1)$$

Définition 4.1 : La *distance séparatrice* de deux ensembles A et B est définie par

$$L_\infty(A, B) \triangleq \inf_{\mathbf{a} \in A, \mathbf{b} \in B} L_\infty(\mathbf{a}, \mathbf{b}), \quad (4.2)$$

Dans ce sens, la distance séparatrice entre la France et l'Angleterre est d'environ trente kilomètres. Notons que L_∞ n'est pas une distance sur l'ensemble $C(\mathbb{R}^n)$ de tous les compacts de \mathbb{R}^n , puisque la distance séparatrice entre deux ensembles d'intersection non vide est nulle.

Définition 4.2 : La *proximité* de A à B est définie par

$$h_\infty^0(A, B) \triangleq \inf \{r \in \mathbb{R}^+ \mid A \subset B + rU\}, \quad (4.3)$$

où U est la sphère unité de (\mathbb{R}^n, L_∞) . Notons que $h_\infty^0(.,.)$ n'est pas un opérateur symétrique, et que la proximité peut être infinie si A n'est pas borné. La notion de proximité se comprend de façon intuitive si nous considérons par exemple le cas d'une maison qui borde une rivière. Nous dirons que la proximité de la maison à la rivière est faible (c'est-à-dire que la maison est proche de la rivière) alors que la proximité de la

rivière à la maison est très grande : ce n'est pas parce que nous nous trouvons proches de la rivière que nous sommes proches de la maison. L'opérateur de proximité possède les propriétés suivantes

$$h_{\infty}^0(A, B) = \sup_{a \in A} h_{\infty}^0(a, B) \text{ (ici, } a \text{ est vu comme un singleton),} \quad (4.4)$$

$$A \subset B \Rightarrow h_{\infty}^0(A, B) = 0, \quad (4.5)$$

$$h_{\infty}^0(A, \partial B) = \max\{h_{\infty}^0(A, B), h_{\infty}^0(A, \bar{B})\}, \quad (4.6)$$

$$A \subset B \Rightarrow h_{\infty}^0(A, C) \leq h_{\infty}^0(B, C), \quad (4.7)$$

où \bar{B} est le complément de B dans \mathbb{R}^n et ∂B est la frontière de l'ensemble B .

Démonstration : Bornons nous à prouver (4.6), les autres propriétés étant triviales.

D'après (4.4), $h_{\infty}^0(A, \partial B) = \sup_{a \in A} h_{\infty}^0(a, \partial B)$. Or $h_{\infty}^0(a, \partial B) = \max\{h_{\infty}^0(a, B), h_{\infty}^0(a, \bar{B})\}$. Ainsi, nous avons $h_{\infty}^0(A, \partial B) = \sup_{a \in A} \max\{h_{\infty}^0(a, B), h_{\infty}^0(a, \bar{B})\} = \max(\sup_{a \in A} h_{\infty}^0(a, B), \sup_{a \in A} h_{\infty}^0(a, \bar{B})) \stackrel{(4.4)}{=} \max\{h_{\infty}^0(A, B), h_{\infty}^0(A, \bar{B})\}$ \diamond

Définition 4.3 : La *distance de Hausdorff* (voir par exemple (Berger, 1979)) entre deux sous-ensembles A et B de \mathbb{R}^n est donnée par $h_{\infty}(A, B) \stackrel{\Delta}{=} \max\{h_{\infty}^0(A, B), h_{\infty}^0(B, A)\}$. \diamond

Dans la littérature, il est possible de trouver la définition équivalente

$$h_{\infty}(A, B) \stackrel{\Delta}{=} \inf\{r \mid A \subset B + rU \text{ et } B \subset A + rU\}. \quad (4.8)$$

La démonstration de l'équivalence entre ces deux définitions est triviale. L'opérateur h_{∞} est une distance dans l'ensemble des compacts $C(\mathbb{R}^n)$, c'est-à-dire que pour tout compact A, B et C appartenant à $C(\mathbb{R}^n)$, nous avons les propriétés des distances

- (i) séparabilité : $h_{\infty}(A, B) = 0 \Rightarrow A = B$,
- (ii) symétrie : $h_{\infty}(A, B) = h_{\infty}(B, A)$,
- (iii) inégalité triangulaire : $h_{\infty}(A, C) \leq h_{\infty}(A, B) + h_{\infty}(B, C)$.

La distance de Hausdorff possède entre autres les propriétés suivantes :

$$\mathbb{B} \subset \mathbb{A} \Rightarrow h_{\infty}^0(\mathbb{A}, \mathbb{B}) = h_{\infty}(\mathbb{A}, \mathbb{B}), \quad (4.9)$$

$$h_{\infty}^0(\mathbb{A}, \mathbb{B}) = h_{\infty}(\mathbb{B}, \mathbb{A} \cup \mathbb{B}), \quad (4.10)$$

$$h_{\infty}^0(\mathbb{A}, \partial\mathbb{B}) = \max\{h_{\infty}(\mathbb{B}, \mathbb{A} \cup \mathbb{B}), h_{\infty}(\bar{\mathbb{B}}, \mathbb{A} \cup \bar{\mathbb{B}})\}, \quad (4.11)$$

$$h_{\infty}(\mathbb{A}, \mathbb{B}) \geq h_{\infty}(\mathbb{A} \cup \mathbb{C}, \mathbb{B} \cup \mathbb{C}). \quad (4.12)$$

Les démonstrations de (4.9), (4.10) et (4.12) sont triviales, (4.11) découle directement de (4.6) et (4.10).

Remarque 4.1 : Soit $\mathbb{A}_1 = \mathbb{A} \cup \{\mathbf{a}\}$, où \mathbf{a} est un point très éloigné de \mathbb{A} . La distance $h_{\infty}(\mathbb{A}_1, \mathbb{A})$ est très grande. Soit maintenant \mathbb{A}_2 , le compact obtenu en déformant légèrement les frontières de l'ensemble \mathbb{A} et en creusant un nombre fini de trous minuscules dans \mathbb{A} . La distance $h_{\infty}(\mathbb{A}_2, \mathbb{A})$ est alors très petite. Ceci illustre le manque de finesse de la topologie induite par la distance de Hausdorff, puisque des ensembles ayant des caractéristiques très différentes peuvent se trouver très proches au sens de cette distance. Une topologie plus fine sera nécessaire pour définir les conditions de convergence des algorithmes d'inversion ensembliste proposés au chapitre 5. Ceci justifie l'introduction de nouvelles distances entre compacts.

4.2.2. Distance de Hausdorff complémentaire

Définition 4.4 : La distance de Hausdorff complémentaire entre deux sous-ensembles \mathbb{A} et \mathbb{B} de \mathbb{R}^n est définie par $\bar{h}_{\infty}(\mathbb{A}, \mathbb{B}) \stackrel{\Delta}{=} h_{\infty}(\bar{\mathbb{A}}, \bar{\mathbb{B}})$, où $\bar{\mathbb{A}}$ et $\bar{\mathbb{B}}$ désignent respectivement les complémentaires dans \mathbb{R}^n de \mathbb{A} et \mathbb{B} . \diamond

Proposition 4.1 : L'opérateur \bar{h}_{∞} est une semi-distance dans $C(\mathbb{R}^n)$. \diamond

Démonstration : (i) L'opérateur \bar{h}_{∞} n'est pas une distance car si \mathbb{A} et \mathbb{B} sont des singletons alors $\bar{h}_{\infty}(\mathbb{A}, \mathbb{B}) = 0$. La propriété de séparabilité d'une distance n'est donc pas vérifiée. (ii) La symétrie de \bar{h}_{∞} résulte de la définition 4.4. Vérifions maintenant l'inégalité triangulaire (iii). Si $\text{clo}(\mathbb{A})$ désigne la clôture ou l'adhérence de \mathbb{A} , alors, pour tout \mathbb{A}, \mathbb{B} et \mathbb{C} appartenant à $C(\mathbb{R}^n)$, $\bar{h}_{\infty}(\mathbb{A}, \mathbb{B}) = h_{\infty}(\bar{\mathbb{A}}, \bar{\mathbb{B}}) = h_{\infty}(\text{clo}(\bar{\mathbb{A}}), \text{clo}(\bar{\mathbb{B}})) = h_{\infty}(\text{clo}(\bar{\mathbb{A}}) \cap \mathbb{Z}, \text{clo}(\bar{\mathbb{B}}) \cap \mathbb{Z})$ où \mathbb{Z} est une sphère compacte infiniment grande. L'inégalité triangulaire

satisfaite par l'opérateur h_∞ lorsqu'il opère sur des compacts nous donne $h_\infty(\text{clo}(\bar{A}) \cap Z, \text{clo}(\bar{B}) \cap Z) \leq h_\infty(\text{clo}(\bar{A}) \cap Z, \text{clo}(\bar{C}) \cap Z) + h_\infty(\text{clo}(\bar{C}) \cap Z, \text{clo}(\bar{B}) \cap Z) = h_\infty(\bar{A}, \bar{C}) + h_\infty(\bar{C}, \bar{B}) = \bar{h}_\infty(A, C) + \bar{h}_\infty(C, B)$. \diamond

Comme h_∞ , la distance de Hausdorff complémentaire \bar{h}_∞ ne caractérise pas finement la différence entre deux compacts comme l'illustre l'exemple suivant. Soit A_1 le compact obtenu en perçant un trou infiniment petit dans le compact A , alors $\bar{h}_\infty(A, A_1)$ n'est pas infiniment petit. Par contre, si A_2 est le compact obtenu ajoutant à A un nombre fini de points loin de celui-ci, alors $\bar{h}_\infty(A, A_2)$ sera nulle. Des compacts de caractéristiques différentes peuvent donc se trouver à une distance \bar{h}_∞ nulle l'un de l'autre.

4.2.3. Distance de Hausdorff généralisée

En combinant h_∞ et \bar{h}_∞ , il est possible d'obtenir une nouvelle distance qui évite les défauts de chacune d'elles.

Définition 4.5 : La *distance de Hausdorff généralisée* entre deux sous-ensembles A et B de \mathbb{R}^n est définie par $m_\infty(A, B) \triangleq \max(h_\infty(A, B), \bar{h}_\infty(A, B))$ \diamond

Proposition 4.2 : L'opérateur m_∞ est une distance sur $C(\mathbb{R}^n)$. \diamond

Démonstration : (i) Si $m_\infty(A, B) = 0$ alors $h_\infty(A, B) = 0$ donc $A = B$. (ii) m_∞ est évidemment symétrique. (iii) Pour tout A, B et C appartenant à $C(\mathbb{R}^n)$, l'inégalité triangulaire est satisfaite par h_∞ et \bar{h}_∞ , ce qui implique que $m_\infty(A, B) = \max(h_\infty(A, B), \bar{h}_\infty(A, B)) \leq \max(h_\infty(A, C) + h_\infty(C, B), \bar{h}_\infty(A, C) + \bar{h}_\infty(C, B)) \leq \max(h_\infty(A, C), \bar{h}_\infty(A, C)) + \max(h_\infty(C, B), \bar{h}_\infty(C, B)) = m_\infty(A, C) + m_\infty(C, B)$. Ainsi, m_∞ vérifie l'inégalité triangulaire. \diamond

4.3. Sous-pavages

Les algorithmes d'exploration ont pour but de couvrir l'espace de recherche. Dans le contexte de l'analyse par intervalles, ce recouvrement se fait par un ensemble de pavés, qui correspond aux notions suivantes.

4.3.1. Définitions

Définition 4.6 : Un *sous-pavage* \mathbb{K} de \mathbb{R}^n est un ensemble (fini ou non) de pavés de \mathbb{R}^n de volume non nul qui ne se recouvrent pas, ce qui signifie que leur intersection deux à deux est vide sauf éventuellement sur leurs frontières. \diamond

L'ensemble des sous-pavages de \mathbb{R}^n sera noté $K(\mathbb{R}^n)$

Définition 4.7 : Si A est le sous-ensemble de \mathbb{R}^n formé par l'union de tous les pavés de \mathbb{K} , alors \mathbb{K} est appelé *pavage* de A . \diamond

Quand aucune ambiguïté ne peut apparaître, l'ensemble $\{\mathbb{K}\}$ formé par l'union de tous les pavés de \mathbb{K} sera aussi noté \mathbb{K} . $\{\mathbb{K}\}$ est un sous-ensemble de \mathbb{R}^n constitué de points, alors que \mathbb{K} est un ensemble de pavés.

Définition 4.8: L'*ensemble d'accumulation* d'un sous-pavage \mathbb{K} est la limite du sous-ensemble de \mathbb{R}^n formée par l'union de tous les pavés de \mathbb{K} qui ont une longueur inférieure à ε , quand ε tend vers zéro. \diamond

Puisque les sous-pavages ne contiennent pas de pavé de longueur nulle, l'ensemble d'accumulation d'un sous-pavage fini est nécessairement vide.

Définition 4.9 : Une suite de sous-pavages $(\mathbb{K}(k))_{k \in \mathbb{N}}$ s'*accumule* sur un sous-ensemble A de \mathbb{R}^n si A est l'ensemble d'accumulation du sous-pavage $\mathbb{K}(\infty)$. \diamond

Remarquons que même lorsque tous les sous-pavages de la suite sont finis la suite peut converger vers un sous-pavage $\mathbb{K}(\infty)$ de cardinal infini.

Définition 4.10 : Soit n_b le nombre minimum de pavés de \mathbb{R}^n de longueur plus petite que ε nécessaire pour recouvrir $A \subset \mathbb{R}^n$. La *dimension fractale* de A est (Cherbit, 1987)

$$m \stackrel{\Delta}{=} \lim_{\varepsilon \rightarrow 0^+} \sup \frac{\ln(n_b)}{\ln\left(\frac{1}{\varepsilon}\right)}. \quad \diamond$$

Nous supposons les variétés A considérées suffisamment régulières pour que leur dimension fractale soit égale à leur dimension classique, et que la limite supérieure soit égale à la limite simple.

Contre-exemple : Un exemple classique de variété dont la dimension classique diffère de la dimension fractale est la sous-variété de \mathbb{R}^2 définie par l'équation polaire

$$\rho = \log(\theta). \quad (4.13)$$

En traçant son graphe, nous constatons que sa dimension fractale est deux alors que sa dimension classique de la variété vaut un. Ceci vient du fait que la spirale ainsi formée se resserre tellement lorsque ρ tend vers l'infini que la variété tend à recouvrir presque complètement le plan. Ce cas peut être considéré comme un cas dégénéré. \diamond

L'introduction de la notion de dimension fractale d'une variété se comprendra aisément plus tard lorsqu'une curieuse similitude entre le calcul de la complexité de l'algorithme d'inversion ensembliste et la définition de la dimension fractale de l'ensemble solution pourra être observée.

4.3.2. Propriétés du cardinal d'un sous-pavage

Dans ce paragraphe, nous allons définir les notions de base utilisées pour le calcul de la complexité des algorithmes ensemblistes. Nous allons pour cela donner un encadrement du cardinal d'un sous-pavage \mathbb{K} recouvrant une sous-variété \mathbb{A} donnée, pour une dimension fractale de \mathbb{A} et une précision de recouvrement données. Définissons tout d'abord la relation d'ordre partielle \triangleleft sur l'ensemble des suites réelles positives :

$$u \triangleleft v \Leftrightarrow \lim_{n \rightarrow \infty} \frac{\ln(u_n)}{\ln(v_n)} \leq 1. \quad (4.14)$$

Cette relation d'ordre induit une relation d'équivalence \sim définie par

$$u \sim v \Leftrightarrow u \triangleleft v \text{ et } v \triangleleft u. \quad (4.15)$$

Exemple 4.1 : si $u_n = n^2$ et $v_n = 2n^2$, les deux suites sont équivalentes. Si maintenant $v_n = n^3$, les deux suites ne sont plus équivalentes, nous avons seulement $u \triangleleft v$. Cette équivalence peut donc être vue comme une équivalence de degrés polynomiaux puisque dans le cas des suites polynomiales, seul le degré des polynômes est pris en compte. \diamond

Remarque 4.2 : L'introduction des relations \triangleleft et \sim s'explique par le fait que si ϵ est une suite positive convergant vers zéro et si n_b est défini comme dans la définition 4.10,

alors $n_b \sim \varepsilon^{-m}$. Cela nous donne un ordre d'idée de la complexité de l'approximation d'une variété par un sous-pavage. L'intuition derrière cette équivalence peut être comprise en considérant un pavé de \mathbb{R}^m recouvert par des hypercubes de côté ε . Quand ε est divisé par deux, le nombre n_b d'hypercubes de côté ε nécessaires au recouvrement du pavé est multiplié par 2^m . \diamond

Théorème 4.1 : Soit ε_d une suite de réels positifs convergeant vers zéro. Soit \mathbb{K} un sous-pavage recouvrant une variété compacte $A \subset \mathbb{K} \subset \mathbb{R}^n$, tel que $h_\infty(\mathbb{K}, A) \leq \varepsilon_d$, et qu'aucun pavé de \mathbb{K} n'ait une longueur plus petite que ε_d . Soit m la dimension de A supposée strictement inférieure à n . Alors $(\varepsilon_d)^{-m} \triangleleft \#\mathbb{K} \triangleleft (\varepsilon_d)^{-n}$ pour presque tous les ensembles A . \diamond

Démonstration : Comme $m < n$, presque partout sur l'ensemble A les pavés de \mathbb{K} ont une longueur du même ordre de grandeur que ε_d . D'après la remarque 4.2, le nombre minimum de pavés avec une longueur plus petite que ε_d nécessaire pour recouvrir A vérifie $n_b \sim (\varepsilon_d)^{-m}$. Nous avons ainsi l'inégalité $(\varepsilon_d)^{-m} \triangleleft \#\mathbb{K}$. Puisqu'aucun pavé de \mathbb{K} n'a une longueur plus petite que ε_d , $\#\mathbb{K}$ est inférieur au nombre d'hypercubes de côté ε_d nécessaire pour recouvrir le pavé enveloppe $[A]$ de A , qui vaut $\text{vol}([A]) * (\varepsilon_d)^{-n}$. Nous avons donc $\#\mathbb{K} \triangleleft (\varepsilon_d)^{-n}$. \diamond

Le théorème 4.1 sera très utile pour l'analyse de la complexité de l'algorithme d'inversion ensembliste proposé au chapitre 5.

Remarque 4.3 : Dans le théorème 4.1, "pour presque tout A " a pour rôle d'exclure les situations dégénérées comme celles où A est une union finie de pavés, car alors $\#\mathbb{K}$ reste fini. \diamond

4.4. Encadrement d'un compact entre deux sous-pavages

Les sous-pavages ont pour but principal d'approximer et plus particulièrement d'encadrer les ensembles que nous considérerons. Soit $(C(\mathbb{R}^n), \subset, m_\infty)$ l'ensemble des compacts de \mathbb{R}^n muni de la relation d'ordre partielle \subset et de la distance de Hausdorff généralisée m_∞ . L'ensemble des sous-pavages finis est dense par l'extérieur dans $(C(\mathbb{R}^n), \subset, m_\infty)$, c'est-à-dire que

$$\forall A \in C(\mathbb{R}^n), \forall \varepsilon > 0, \exists \mathbb{K}^+ \in K(\mathbb{R}^n) \mid A \subset \mathbb{K}^+ \text{ et } m_\infty(\mathbb{K}^+, A) < \varepsilon. \quad (4.16)$$

Toutefois, cette propriété ne s'applique pas aux sous-pavages intérieurs. Considérons par exemple un segment du plan \mathbb{R}^2 . Il peut être approximé d'aussi près que désiré par un sous-pavage extérieur, mais pas par un sous-pavage intérieur. Il est en effet impossible de trouver un pavé de volume non nul inclus dans ce segment. Pour éviter ce type de problème, nous restreindrons notre étude à la (grande) classe des compacts pleins définis comme suit.

Définition 4.11 : Le compact A est *plein* si $\text{clo}(\text{int}(A)) = A$. ◁

L'ensemble de tous les compacts pleins de \mathbb{R}^n sera noté $C_f(\mathbb{R}^n)$. L'indice f provient de la traduction anglaise *full* du mot *plein*.

Exemple 4.2 : Une variété de \mathbb{R}^n de dimension $m < n$ n'est pas un compact plein alors qu'une sphère pleine est un compact plein. ◁

Proposition 4.3 : \bar{h}_∞ est une distance dans $C_f(\mathbb{R}^n)$. ◁

Démonstration : (i) Pour tout A et B appartenant à $C_f(\mathbb{R}^n)$, nous avons $\bar{h}_\infty(A, B) = h_\infty(\bar{A}, \bar{B}) = h_\infty(\text{clo}(\bar{A}), \text{clo}(\bar{B})) = h_\infty(\text{clo}(\bar{A}) \cap Z, \text{clo}(\bar{B}) \cap Z)$, où Z est une sphère compacte suffisamment grande pour contenir A et B . Ainsi, $\bar{h}_\infty(A, B) = 0 \Rightarrow h_\infty(\text{clo}(\bar{A}) \cap Z, \text{clo}(\bar{B}) \cap Z) = 0$, ce qui implique, comme h_∞ est une distance dans $C(\mathbb{R}^n)$ qui vérifie donc la propriété de séparabilité, $\text{clo}(\bar{A}) \cap Z = \text{clo}(\bar{B}) \cap Z \Leftrightarrow \text{clo}(\bar{A}) = \text{clo}(\bar{B}) \Leftrightarrow \text{int}(A) = \text{int}(B) \Rightarrow \text{clo}(\text{int}(A)) = \text{clo}(\text{int}(B))$, d'où $A = B$ d'après la définition 4.11. Les deux autres propriétés (ii) et (iii) d'une distance résultent de la proposition 4.1. ◊

Il découle immédiatement de la proposition 4.3 que m_∞ est une distance dans $C_f(\mathbb{R}^n)$. Tous les sous-pavages finis de \mathbb{R}^n appartiennent à $C_f(\mathbb{R}^n)$. L'ensemble des sous-pavages est dense par l'intérieur et par l'extérieur dans $(C_f(\mathbb{R}^n), \subset, m_\infty)$. Ainsi,

$$\forall A \in C_f(\mathbb{R}^n), \exists \mathbb{K}^-(k) \mid \begin{cases} \mathbb{K}^-(k) \subset A \\ m_\infty(\mathbb{K}^-(k), A) \rightarrow 0 \end{cases} \quad (4.17)$$

$$\forall A \in C_f(\mathbb{R}^n), \exists \mathbb{K}^+(k) \mid \begin{cases} A \subset \mathbb{K}^+(k) \\ m_\infty(\mathbb{K}^+(k), A) \rightarrow 0 \end{cases} \quad (4.18)$$

Pour tout $\mathbb{X} \in C_f(\mathbb{R}^n)$, il est donc possible de trouver deux sous-pavages finis \mathbb{X}^- et \mathbb{X}^+ tels que $\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+$. Dans les sous-pavages que nous manipulerons, nous aurons toujours $\mathbb{X}^- \subset \mathbb{X}^+$ au sens où chaque pavé de \mathbb{X}^- appartient aussi à \mathbb{X}^+ . La quantité $\Delta\mathbb{X} = \mathbb{X}^+ - \mathbb{X}^-$ est donc un sous-pavage, que nous appellerons *couche d'incertitude*, car $\Delta\mathbb{X}$ forme l'ensemble des vecteurs tels que nous ne savons pas s'ils sont à l'intérieur ou à l'extérieur de \mathbb{X} . La figure 4.1 illustre cette notion d'encadrement d'un compact par deux sous-pavages et de couche d'incertitude de cet encadrement.

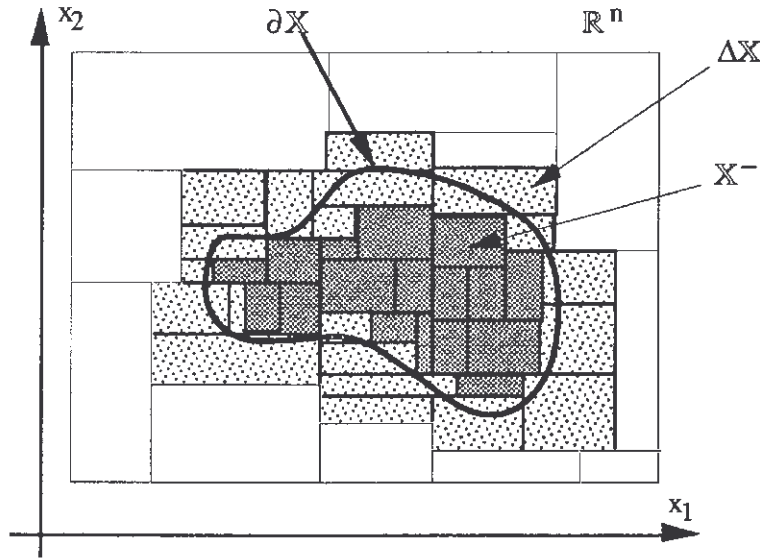


Fig. 4.1 Encadrement d'un compact par deux sous-pavages.

Le sous-ensemble $V(\mathbb{X})$ de $C(\mathbb{R}^n)$ constitué de tous les compacts \mathbb{X}' tels que $\mathbb{X}^- \subset \mathbb{X}' \subset \mathbb{X}^+$ possède un diamètre $m_\infty(\mathbb{X}^-, \mathbb{X}^+)$ aussi petit que voulu. $V(\mathbb{X})$ forme donc un voisinage de \mathbb{X} , qui sera dénommé *intercompact* dans le prochain paragraphe. Nous pouvons dire que \mathbb{X}^- et \mathbb{X}^+ enferment ou encadrent le compact plein \mathbb{X} . Dans la suite, nous chercherons à encadrer l'ensemble solution du problème ensembliste à résoudre par deux sous-pavages. Cela ne sera donc possible que si l'ensemble solution est un compact plein, ce qui se trouve en général être le cas.

Encadrement d'une caractéristique : Soit Z une fonction croissante de $(C(\mathbb{R}^n), \subset)$ dans un espace métrique partiellement ordonné $(Z, <)$. Par exemple, $Z(\mathbb{X})$ peut correspondre au volume de \mathbb{X} , à son pavé enveloppe $[\mathbb{X}]$, ou à n'importe quelle fonction de \mathbb{X} résultant de la maximisation d'un critère convexe sur \mathbb{X} . Toutes ces caractéristiques sont très faciles à calculer pour les sous-pavages. Si $\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+$, alors $Z(\mathbb{X}^-) < Z(\mathbb{X}) < Z(\mathbb{X}^+)$. Si Z est une fonction continue autour de \mathbb{X} , alors $Z(\mathbb{X}^-) \xrightarrow{\leftarrow} Z(\mathbb{X})$ et $Z(\mathbb{X}^+) \xrightarrow{\rightarrow}$

$Z(\mathbb{X})$ lorsque $m_\infty(\mathbb{X}^-, \mathbb{X}^+) \rightarrow 0$.

◊

Il va ainsi être possible d'encadrer, sous certaines conditions, l'ensemble solution \mathbb{X} du problème ensembliste entre deux sous-pavages enfermant un voisinage de \mathbb{X} dans $(C(\mathbb{R}^n), m_\infty)$ avec un diamètre aussi petit que désiré.

4.5. Les intercompacts.

Le calcul sur les intervalles permet de faire du calcul sur les réels en encadrant le réel incertain x entre deux réels x^- et x^+ définissant ainsi un ensemble $[x^-, x^+]$ qui contient x de façon garantie. De la même façon, le calcul ensembliste numérique garanti est concevable si les ensembles \mathbb{X} manipulés sont encadrés par deux sous-pavages \mathbb{X}^- et \mathbb{X}^+ définissant ainsi un *intercompact* noté $[\mathbb{X}^-, \mathbb{X}^+]$ qui contient \mathbb{X} à coup sûr. Rappelons qu'un élément de l'intercompact $[\mathbb{X}^-, \mathbb{X}^+]$ est un compact de \mathbb{R}^n alors qu'un élément de $[x^-, x^+]$ est un réel. Un *intercompact* englobe l'ensemble des compacts compris (au sens de l'inclusion) entre la borne inférieure \mathbb{X}^- et la borne supérieure \mathbb{X}^+ , de même qu'un *intervalle* englobe l'ensemble des valeurs comprises entre ses deux bornes. Le calcul sur les intercompacts serait donc au calcul ensembliste ce que le calcul sur les intervalles est au calcul réel. Le calcul intercompact repose sur le calcul par sous-pavages, par exemple

$$[\mathbb{X}^-, \mathbb{X}^+] + [\mathbb{Y}^-, \mathbb{Y}^+] \triangleq [\mathbb{X}^- + \mathbb{Y}^-, \mathbb{X}^+ + \mathbb{Y}^+], \quad (4.19)$$

où l'addition de deux sous-pavages est définie comme suit,

$$\mathbb{K}_1 + \mathbb{K}_2 \triangleq \{ \mathbf{k}_1 + \mathbf{k}_2 \mid \mathbf{k}_1 \in \mathbb{K}_1 \text{ et } \mathbf{k}_2 \in \mathbb{K}_2 \}. \quad (4.20)$$

Cherchons maintenant à effectuer un encadrement ensembliste garanti de l'ensemble $\mathbb{Y} = \mathbf{f}(\mathbb{X})$ où \mathbb{X} appartient à coup sûr à l'intercompact $[\mathbb{X}^-, \mathbb{X}^+]$. Si \mathbf{f}^+ est une fonction d'inclusion de \mathbf{f} , en posant

$$\mathbf{f}^+(\mathbb{K}) \triangleq \{ \mathbf{f}(\mathbf{k}) \mid \mathbf{k} \in \mathbb{K} \}, \quad (4.21)$$

nous avons

$$\mathbb{Y} = \mathbf{f}(\mathbb{X}) \subset \mathbf{f}(\mathbb{X}^+) \subset \mathbf{f}^+(\mathbb{X}^+). \quad (4.22)$$

f^+ peut être appelée *fonction d'inclusion extérieure*, car $f(\mathfrak{x}) \subset f^+(\mathfrak{x})$. Pour avoir une borne intérieure pour \mathfrak{Y} , il nous faudrait une *fonction d'inclusion intérieure* f^- , c'est-à-dire telle que pour tout pavé \mathfrak{x} , nous ayons $f^-(\mathfrak{x}) \subset f(\mathfrak{x})$. Ainsi, nous aurions

$$f^-(\mathfrak{X}^-) \subset f(\mathfrak{X}^-) \subset f(\mathfrak{X}) = \mathfrak{Y} \quad (4.23)$$

Par exemple, dans le cas monodimensionnel, si f est une fonction continue de x , une fonction d'inclusion intérieure est donnée par $f^-([x^-, x^+]) = [\min(f(x^-), f(x^+)), \max(f(x^-), f(x^+))]$. Dans le cas multidimensionnel, la détermination de fonctions d'inclusions intérieures reste ouvert.

On peut étendre de cette façon tous les calculs existant sur les pavés aux sous-pavages et aux intercompacts. Pour cela, il peut être intéressant de définir d'autres types d'opérateur sur les sous-pavages tels que

- la *réduction*, qui consiste, à partir d'un sous-pavage, à construire un sous-pavage équivalent avec un nombre inférieur de pavés, qui seront bien sûr plus gros ;
- l'*inversion inclusive*, qui consiste à calculer un intercompact contenant l'image réciproque par une fonction f d'un ensemble \mathfrak{Y} connu par un intercompact qui le renferme.

Les outils nécessaires à cette inversion seront développés au chapitre 5. Le nombre $m_\infty(\mathfrak{X}^-, \mathfrak{X}^+)$ est appelé *longueur* ou *diamètre* de l'intercompact $[\mathfrak{X}^-, \mathfrak{X}^+]$. Le sous-ensemble $\Delta\mathfrak{X}$ de \mathbb{R}^n défini par $\Delta\mathfrak{X} = \{\mathfrak{X}^+\} - \{\mathfrak{X}^-\} = \{\mathfrak{X}^+\} \cap \{\overline{\mathfrak{X}^-}\}$ est appelé *couche d'incertitude de l'encadrement*. L'ensemble $\Delta\mathfrak{X}$ forme une couche enfermant la frontière $\partial\mathfrak{X}$ de \mathfrak{X} , et l'encadrement de \mathfrak{X} par l'intercompact $[\mathfrak{X}^-, \mathfrak{X}^+]$ ne permet pas de dire si un élément de $\Delta\mathfrak{X}$ appartient ou non à \mathfrak{X} . Le volume de la couche d'incertitude de l'intercompact semble être une notion intéressante pour caractériser la grossièreté de l'encadrement de \mathfrak{X} .

Il serait intéressant de concevoir un langage de programmation ensembliste garantie qui permette la déclaration et la gestion des intercompacts. Nous n'utiliserons cependant pas dans nos algorithmes le calcul intercompact car celui-ci nécessiterait un approfondissement conséquent.

4.6. Topologie du problème d'inversion ensembliste

Dans ce paragraphe, nous allons nous intéresser à la continuité de la fonction f^{-1} introduite dans le problème d'inversion ensembliste d'équation $\mathbb{X} = f^{-1}(\mathbb{Y})$. Comme nous utilisons plusieurs distances dans l'ensemble des compacts, il convient d'introduire des notations permettant de distinguer le type de continuité considéré pour une fonction ensembliste donnée.

Notation : Si h et g sont respectivement deux distances (ou semi-distances) dans les espaces de compacts $C(\mathbb{R}^n)$ et $C(\mathbb{R}^p)$ alors, dire que la fonction ensembliste $f : C(\mathbb{R}^n) \rightarrow C(\mathbb{R}^p)$ est $(h-g)$ -continue, signifie que $\forall A, B \in C(\mathbb{R}^n), h(A, B) \rightarrow 0 \Rightarrow g(f(A), f(B)) \rightarrow 0$. Quand la distance h considérée dans $C(\mathbb{R}^n)$ est la même que celle considérée dans $C(\mathbb{R}^p)$, nous dirons que f est h -continue. Si une fonction ensembliste est à la fois h_1-g_1 -continue et h_2-g_2 -continue, nous dirons que f est (h_1-g_1) et (h_2-g_2) -continue. \diamond

Supposons que $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$ soit continue et localement injective. Il est aisé de comprendre que la fonction ensembliste associée $f : C(\mathbb{R}^n) \rightarrow C(\mathbb{R}^p)$ est h_∞, \bar{h}_∞ et m_∞ -continue autour de n'importe quel compact de $C(\mathbb{R}^n)$. Notons que la fonction inverse $f^{-1} : \mathbb{R}^p \rightarrow \mathbb{R}^n$ est en général non définie (même lorsque $n = p$), mais la fonction ensembliste réciproque $f^{-1} : C(\mathbb{R}^p) \rightarrow C(\mathbb{R}^n)$ est toujours définie. Par contre, elle n'est jamais partout continue. Supposons maintenant que $\mathbb{Y} \in C_f(\mathbb{R}^p)$, et que $\mathbb{X} = f^{-1}(\mathbb{Y}) \in C(\mathbb{R}^n)$, et tentons d'étudier le type de discontinuité de f^{-1} pouvant apparaître autour de \mathbb{Y} . Pour cela, nous utiliserons la définition suivante.

Définition 4.12 : Le vecteur $x_i \in \mathbb{R}^n$ est *instable* si $x_i \in f^{-1}(\partial\mathbb{Y})$. \diamond

Lorsque le vecteur x_i est instable, il existe un compact $\mathbb{Y}^- \subset \text{int}(\mathbb{Y})$ m_∞ -proche de \mathbb{Y} tel que $x_i \notin f^{-1}(\mathbb{Y}^-)$, alors que $x_i \in f^{-1}(\mathbb{Y})$. Les comportements qualitatifs possibles de la variété image $f(\mathbb{R}^n)$ relativement à l'ensemble \mathbb{Y} peuvent être classifiés selon le type de discontinuité rencontrée. Dans ce but, nous allons considérer un compact $\mathbb{Y}^- \subset \text{int}(\mathbb{Y})$ tel que \mathbb{Y}^- soit m_∞ -proche de \mathbb{Y} (\mathbb{Y}^- existe puisque \mathbb{Y} est plein). Nous allons alors tenter de classifier les différents types de discontinuité de la fonction ensembliste f^{-1} qu'il est possible de rencontrer. La figure 4.2 illustre, dans le cas monodimensionnel ($n = 1$), trois types de discontinuité, où les variétés images sont repérées par (a),(b),(c), et le type de continuité (repérée par (d)) qu'il est possible de rencontrer.

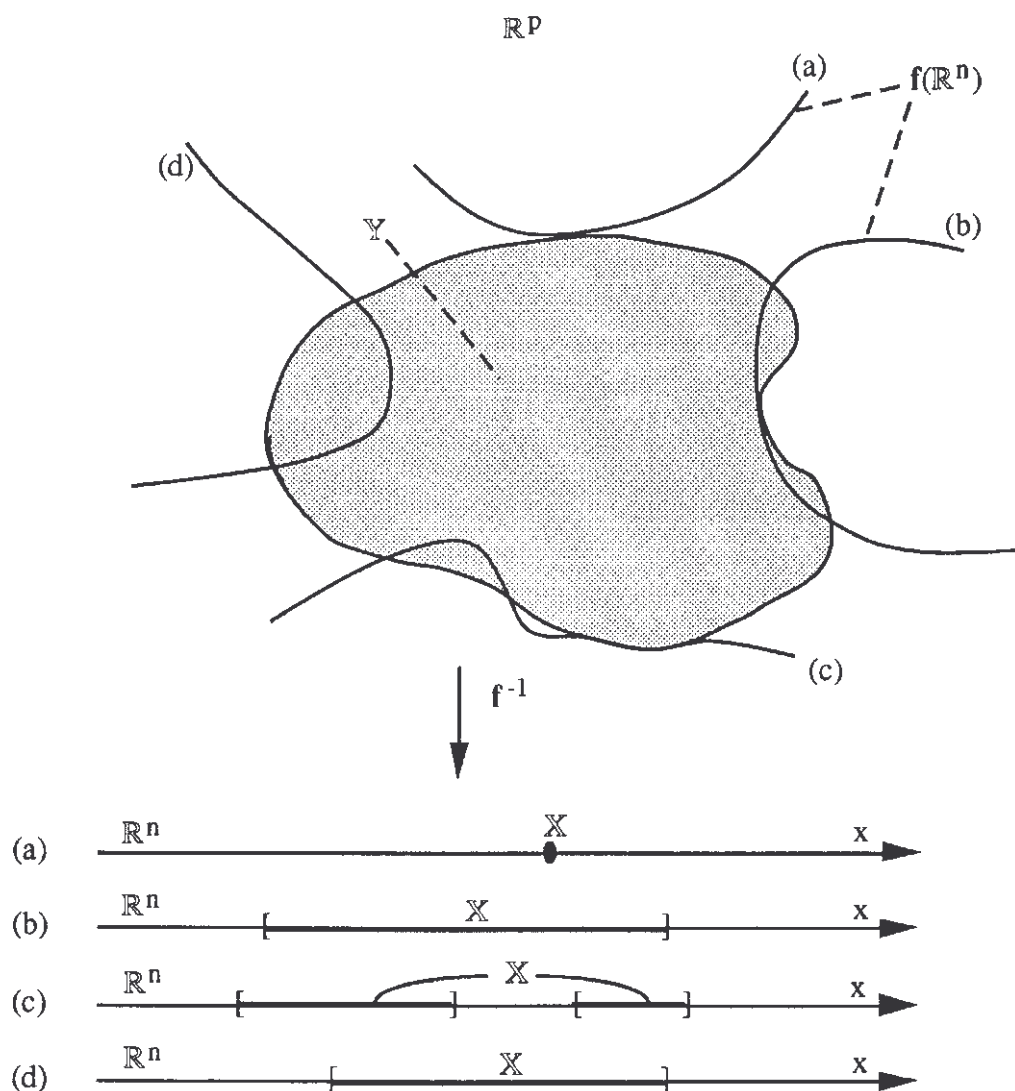


Fig. 4.2 Quatre situations pouvant apparaître dans le problème d'inversion ensembliste.

a) **Discontinuité extérieure**

Définition 4.13 : f^{-1} est *extérieurement discontinue* en Y s'il existe un vecteur instable x_i et un voisinage $V(x_i)$ de x_i tel que $f(V(x_i)) \cap \text{int}(Y) = \emptyset$. \diamond

Évidemment, $f^{-1}(Y^-)$ n'est pas h_∞ -proche de $f^{-1}(Y)$, et donc f^{-1} est $(m_\infty-h_\infty)$ -discontinue autour de Y . Cette discontinuité est qualifiée d'extérieure car la variété $f(\mathbb{R}^n)$ vient tangenter la frontière ∂Y de Y par l'extérieur. C'est le cas sur la figure 4.2 (a).

b) Discontinuité intérieure

Définition 4.14 : f^{-1} est *intérieurement discontinue* en Y s'il existe un voisinage $V(x_i)$ de x_i tel que $f(V(x_i)) \subset Y$. \diamond

L'ensemble $f^{-1}(Y^-)$ n'est pas \bar{h}_∞ -proche de $f^{-1}(Y)$ et donc f^{-1} est $(m_\infty\text{-}\bar{h}_\infty)$ -discontinue autour de Y . Nous dirons que f^{-1} possède une discontinuité intérieure en Y car la variété $f^{-1}(\mathbb{R}^n)$ vient tangenter la frontière par l'intérieur. C'est le cas par exemple sur la figure 4.2 (b).

(c) Discontinuité complète

Définition 4.15 : f^{-1} est *complètement discontinue* en Y s'il existe un voisinage $V(x_i)$ de x_i tel que $f(V(x_i)) \subset \partial Y$.

Nous avons en x_i la superposition de discontinuités intérieure et extérieure autour de Y . L'ensemble $f^{-1}(Y^-)$ n'est ni h_∞ ni \bar{h}_∞ -proche de $f^{-1}(Y)$. Nous dirons que f^{-1} possède une discontinuité complète en Y car f^{-1} est $(m_\infty\text{-}h_\infty)$ et $(m_\infty\text{-}\bar{h}_\infty)$ -discontinue autour de Y . Nous avons un ensemble $f^{-1}(\partial Y)$ de vecteurs instables de mesure non nulle. C'est le cas par exemple sur la figure 4.2 (c).

(d) Transversalité

Définition 4.16 : f^{-1} est *transverse* en x_i s'il existe un voisinage $V(x_i)$ de x_i tel que $f(V(x_i))$ et ∂Y soient des sous-variétés transverses autour du point $f(x_i)$. \diamond

Rappelons que deux sous-variétés sont transverses au point d'intersection x_i si (éventuellement après avoir fait agir préalablement un homéomorphisme sur l'espace image) les deux espaces tangents sont supplémentaires. Cette condition de transversalité est satisfaite pour tous les vecteurs instables x_i si et seulement si f^{-1} est $m_\infty\text{-}m_\infty$ -continue autour de Y . Cette situation est générique, et il ne s'agit pas d'une situation de discontinuité. C'est le cas par exemple sur la figure 4.2 (d).

Dans tous les cas étudiés ci-dessus, un agrandissement infinitésimal de \mathbb{Y} n'engendre jamais une modification considérable de l'ensemble $\mathbb{X} = f^{-1}(\mathbb{Y})$ au sens de la distance m_∞ . Par contre, une réduction infinitésimale de \mathbb{Y} peut engendrer, selon le type de la discontinuité de f^{-1} , une modification non négligeable de l'ensemble \mathbb{X} , toujours au sens de m_∞ . Dans le cas d'une discontinuité complète, un sous-ensemble de volume non nul \mathbb{X} va disparaître. Dans le cas d'une discontinuité extérieure qui n'est pas complète, \mathbb{X} sera amputé d'un ensemble, formant une composante non-connexe de plus faible dimension. Dans le cas d'une discontinuité intérieure qui ne soit pas complète, un trou ou une faille de dimension inférieure à n va se creuser dans \mathbb{X} .

4.7. Conclusion

Les sous-pavages sont des sous-ensembles compacts de \mathbb{R}^n de structure très simple (puisque'ils sont formés par des unions de pavés), qui ont pour but premier d'approximer simplement les sous-ensembles de \mathbb{R}^n que nous serons amenés à considérer. Ils présentent le double avantage d'être facilement représentables en machine et de pouvoir supporter les calculs ensemblistes grâce au calcul par intervalles. Certaines caractéristiques de l'ensemble solution \mathbb{X} peuvent alors être estimées de façon garantie par calcul de ces caractéristiques pour un sous-pavage \mathbb{K} approximant \mathbb{X} . Avec deux sous-pavages, il est possible d'enfermer un sous-ensemble de \mathbb{R}^n dans un sous-ensemble, appelé intercompact, de l'espace des compacts. Ces intercompacts sont représentables en machine et capables de décrire sous une forme exploitable et garantie les ensembles manipulés. Dans la suite, les algorithmes développés n'utiliseront cependant pas le calcul intercompact car ils sont suffisamment simples pour être compris aisément sans cet outil, mais le calcul intercompact semble très adapté au développement et à la conception d'algorithmes ensemblistes plus complexes. Dans ce chapitre, nous avons présenté les différentes distances dans l'espace des compacts qui seront utilisées par la suite pour démontrer la convergence des algorithmes ensemblistes. Nous avons aussi donné un ordre de grandeur du nombre de pavés nécessaires à ces sous-pavages pour recouvrir un ensemble considéré avec une certaine précision. Ceci permettra l'analyse de complexité des algorithmes proposés par la suite. Une vision géométrique des problèmes de discontinuité intrinsèques au problème d'inversion ensembliste a été proposée. Toutes les notions présentées dans ce chapitre s'avéreront indispensables pour attaquer la résolution du problème général d'inversion ensembliste qui fera l'objet du prochain chapitre.

CHAPITRE 5

ALGORITHME

D'INVERSION ENSEMBLISTE

5.1. Introduction

L'approche que nous avons choisie pour caractériser l'ensemble solution \mathbb{X} du problème d'inversion ensembliste $\mathbb{X} = f^{-1}(\mathbb{Y})$ consiste à encadrer \mathbb{X} par deux sous-pavages \mathbb{X}^- et \mathbb{X}^+ tels que $\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+$, formant ainsi un intercompact $[\mathbb{X}^-, \mathbb{X}^+]$ enfermant \mathbb{X} . Nous supposons que la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$ est continue, qu'elle admet une fonction d'inclusion convergente connue f et que l'ensemble \mathbb{Y} est un compact plein. Les notions d'inversion ensembliste et d'encadrement de l'ensemble solution par deux sous-pavages sont illustrées par la figure 5.1. Notons que dans tous les cas, nous avons $f(\mathbb{X}) = f \circ f^{-1}(\mathbb{Y}) = \mathbb{Y} \cap f(\mathbb{R}^n) \subset \mathbb{Y}$, avec $f(\mathbb{X}) = \mathbb{Y}$ seulement si f est surjective, ce qui n'est jamais le cas dans les applications considérées dans cette thèse.

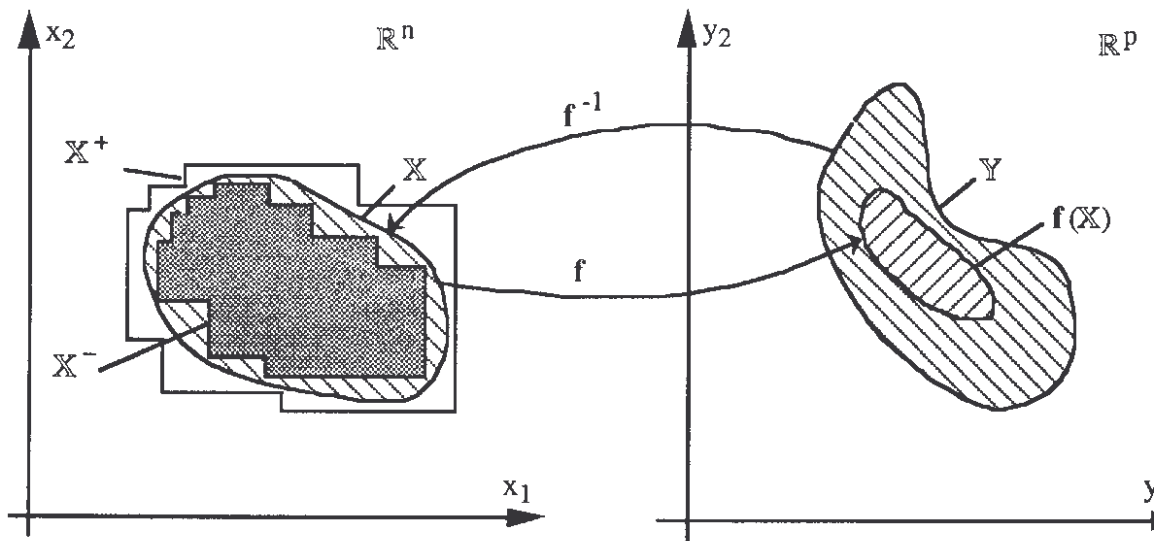


Fig. 5.1 Encadrement de l'ensemble solution du problème d'inversion ensembliste par deux sous-pavages.

Nous avons défini au chapitre 4 une topologie pour l'ensemble $C_f(\mathbb{R}^n)$ des compacts pleins grâce à l'introduction d'une distance m_∞ sur $C_f(\mathbb{R}^n)$. L'ensemble des sous-pavages est dense par l'extérieur et par l'intérieur dans $(C_f(\mathbb{R}^n), \subset, m_\infty)$. Ainsi, si nous supposons que $\mathbb{X} \in C_f(\mathbb{R}^n)$, alors, il existe deux sous-pavages \mathbb{X}^- et \mathbb{X}^+ définissant un intercompact $[\mathbb{X}^-, \mathbb{X}^+]$ qui forme un voisinage de \mathbb{X} de diamètre $m_\infty(\mathbb{X}^-, \mathbb{X}^+)$ aussi petit que désiré. Si Z est une caractéristique croissante de \mathbb{X} à valeurs dans l'ensemble ordonné $(Z, <)$, un encadrement de la caractéristique $Z(\mathbb{X})$ de l'ensemble solution est donné par $Z(\mathbb{X}^-) < Z(\mathbb{X}) < Z(\mathbb{X}^+)$. Si de plus Z est continue autour de \mathbb{X} cet encadrement peut être obtenu avec une précision aussi fine que désiré.

Nous allons maintenant développer un algorithme pour la recherche effective de l'intercompact $[\mathbb{X}^-, \mathbb{X}^+]$, de diamètre aussi petit que désiré, qui enferme la solution $\mathbb{X} = f^{-1}(Y)$ du problème d'inversion ensembliste, sous l'hypothèse que f^{-1} soit h_∞ et \bar{h}_∞ -continue en Y et que \mathbb{X} soit un compact plein. Cet algorithme peut s'adapter de façon à fournir l'encadrement de n'importe quelle caractéristique $Z(\mathbb{X})$ continue et monotone avec autant de précision que désiré.

5.2. Algorithme d'inversion ensembliste

Nous supposons, par souci de simplicité, que \mathbb{X} est borné et inclus dans le pavé connu *a priori* $\mathbf{x}(0)$, qui sera utilisé comme domaine de recherche initial. L'extension au cas non-borné est immédiate grâce à l'utilisation de pavés non-bornés (ou intervalles vectoriels généralisés) (voir par exemple Hansen (1992), page 10).

5.2.1. Tests d'acceptabilité et d'inacceptabilité d'un pavé.

L'algorithme SIVIA (*Set Inverter Via Interval Analysis*) (Jaulin, Walter 1993b), que nous allons présenter, s'applique à n'importe quelle fonction f pour laquelle une fonction d'inclusion f peut être calculée. Nous dirons qu'un pavé \mathbf{x} est *acceptable* si $\mathbf{x} \subset \mathbb{X}$ et *inacceptable* si $\mathbf{x} \cap \mathbb{X} = \emptyset$. Sinon, \mathbf{x} est *ambigu*. L'analyse par intervalles nous donne deux conditions, illustrées par la figure 5.2, pour décider de l'acceptabilité de \mathbf{x} .

$$\text{Si } f(\mathbf{x}) \subset Y \quad \Rightarrow \quad \mathbf{x} \subset \mathbb{X} \quad (\text{le pavé } \mathbf{x} \text{ est acceptable}). \quad (5.1)$$

$$\text{Si } f(\mathbf{x}) \cap Y = \emptyset \quad \Rightarrow \quad \mathbf{x} \cap \mathbb{X} = \emptyset \quad (\text{le pavé } \mathbf{x} \text{ est inacceptable}). \quad (5.2)$$

Dans tous les autres cas, le pavé \mathbf{x} sera dit *indéterminé*. Notons que tout pavé indéterminé n'est pas nécessairement ambigu, mais peut être acceptable ou inacceptable. Par contre, tout pavé ambigu est nécessairement indéterminé. La figure 5.2 montre comment un pavé inacceptable peut être indéterminé et donc pourquoi ces deux conditions ne sont que suffisantes.

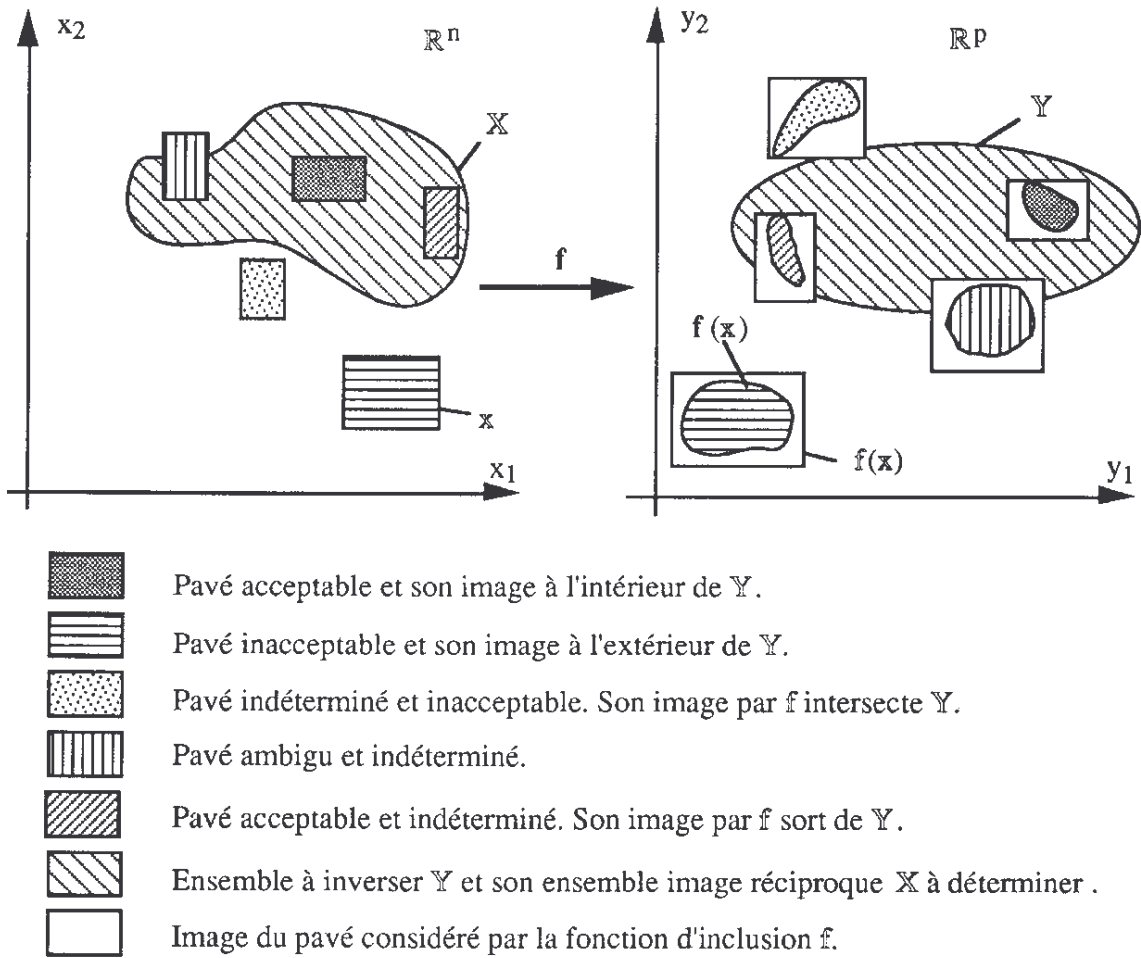


Fig. 5.2 Conditions suffisantes d'acceptabilité et d'inacceptabilité d'un pavé.

5.2.2. Principe de l'algorithme d'inversion ensembliste

Le principe de l'encadrement de l'ensemble solution \mathbb{X} repose sur trois étapes :

- (i) Définir le pavé $\mathbf{x}(0)$ des vecteurs acceptables *a priori* sur lequel la recherche sera effectuée.
- (ii) Proposer un pavage \mathbb{K} pour $\mathbf{x}(0)$.
- (iii) Calculer $f(\mathbf{x})$ pour chacun des pavés de \mathbb{K} . Trois situations peuvent apparaître :
 - 1) $f(\mathbf{x}) \subset \mathbb{Y} \Rightarrow \mathbf{x} \subset \mathbb{X}$, (\mathbf{x} est acceptable).
 - (L'implication devient une équivalence si f est minimale et \mathbb{Y} est un pavé).
 - 2) $f(\mathbf{x}) \cap \mathbb{Y} = \emptyset \Rightarrow \mathbf{x} \cap \mathbb{X} = \emptyset$, (\mathbf{x} est inacceptable).
 - 3) Sinon, \mathbf{x} est indéterminé, (il peut être ambigu).

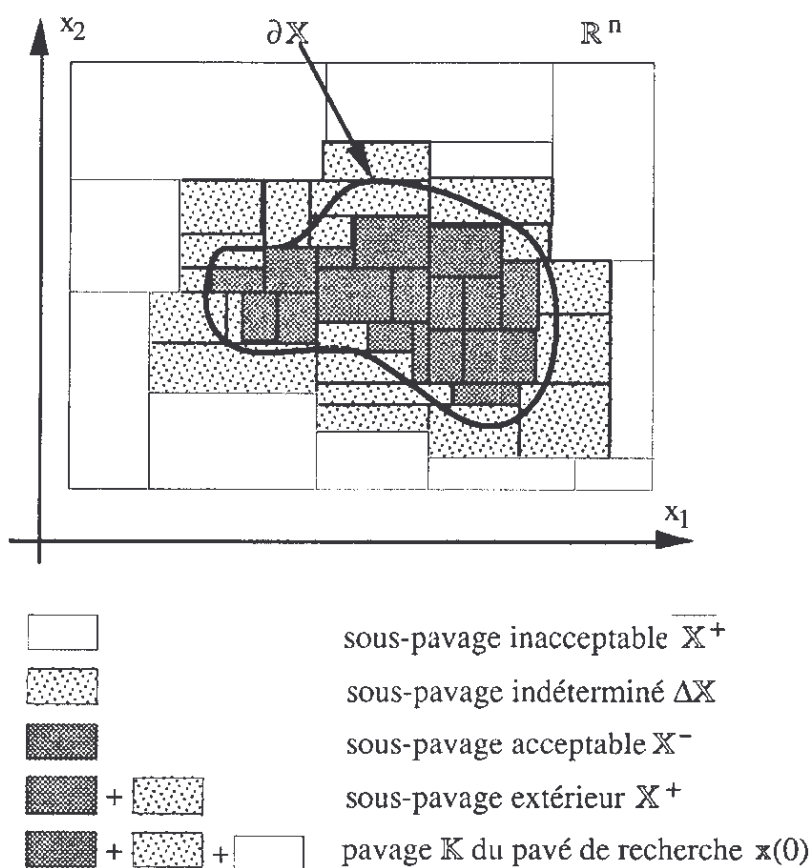


Fig. 5.3 Encadrement du compact solution par deux sous-pavages.

Le pavage \mathbb{K} se trouve maintenant partitionné en trois sous-pavages \mathbb{X}^- , $\Delta\mathbb{X}$ et $\overline{\mathbb{X}^+}$ correspondant respectivement aux pavés acceptables, indéterminés et inacceptables, avec $\mathbb{X}^+ \triangleq \mathbb{X}^- \cup \Delta\mathbb{X}$. Ces sous-pavages vérifient

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+ \quad (5.3)$$

$$\partial\mathbb{X} \subset \Delta\mathbb{X} \quad (5.4)$$

$$\text{vol}(\mathbb{X}^-) \leq \text{vol}(\mathbb{X}) \leq \text{vol}(\mathbb{X}^+) \quad (5.5)$$

$$[\mathbb{X}^-] \subset [\mathbb{X}] \subset [\mathbb{X}^+] \quad (5.6)$$

La figure 5.3 illustre le pavage du pavé de recherche ainsi que l'encadrement d'un compact par deux sous-pavages.

5.2.3. SIVIA

L'algorithme SIVIA met en œuvre de façon récursive le principe d'encadrement par sous-pavage que nous venons de présenter. Il utilise une structure dynamique particulière pour stocker ses résultats intermédiaires. Rappelons qu'une structure dynamique est un ensemble d'éléments liés par une ou plusieurs relations. Ces relations, qui ne sont pas nécessairement symétriques, peuvent par exemple être des relations de parenté, des relations de voisinage, ou des relations hiérarchiques. En informatique, les relations associées à un élément sont représentées par des pointeurs qui mémorisent l'emplacement des éléments avec lesquels il est en relation. Une telle représentation permet de restructurer complètement les liens relationnels de l'ensemble sans avoir à modifier l'emplacement mémoire de chacun des éléments, d'où un gain de temps considérable dans la gestion mémoire de ces éléments.

Un exemple d'une telle structure est la pile. Dans une pile, chaque élément connaît l'emplacement exacte de l'élément qui est en dessous de lui dans la pile (pensons à une pile d'assiettes). La figure 5.4 illustre la représentation informatique d'une pile. Sur cette figure, une pile de 6 éléments, numérotés de 1 à 6, est représentée sous sa forme classique (a) puis telle qu'elle est mémorisée en machine (b). L'élément 1 forme le fond de pile et l'élément 6, le sommet de pile. Sur la figure 5.4.(b), les flèches représentent des pointeurs, et la petite case à droite de chacun des éléments représente le champ de cet élément qui contient l'adresse en mémoire de l'élément se trouvant juste en dessous de lui

dans la pile. Bien sûr, l'élément 1 ne pointe sur personne puisqu'il est au fond de la pile. Le champ associé prend alors une valeur neutre. En machine, la pile toute entière est représentée par un seul pointeur (pile) pointant vers le sommet de pile. Nous remarquons qu'en machine l'aspect géométrique de la pile n'est pas du tout respecté.

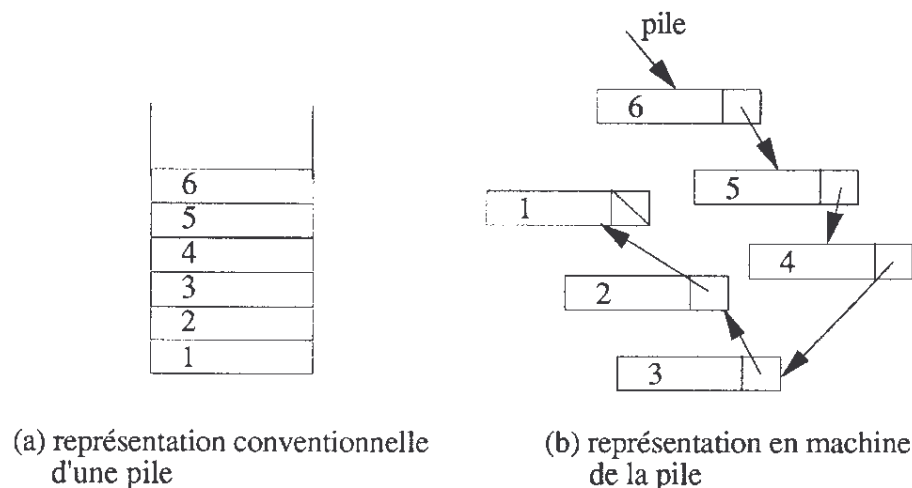


Fig. 5.4 Représentation d'une pile.

Dans une pile, seulement trois opérations sont possibles. La première consiste à *empiler*, c'est-à-dire à mettre un élément au sommet de la pile. La deuxième consiste à *dépiler*, c'est-à-dire à enlever l'élément placé au sommet de la pile. La troisième opération permet de tester si la pile est vide ou non. La structure dynamique utilisée par SIVIA est une pile de pavés. Le pavé considéré à l'itération k sera noté $\mathbf{x}(k)$, ϵ_r sera la précision requise pour les sous-pavages \mathbf{X}^- et $\Delta\mathbf{X}$. C'est-à-dire qu'une fois l'algorithme achevé, tous les pavés indéterminés auront une longueur inférieure ou égale à ϵ_r . L'algorithme SIVIA qui résoud le problème d'inversion ensembliste à l'aide de l'analyse par intervalles, peut être décrit de la façon suivante.

<u>Entrées</u>	Fonction d'inclusion	:	f .
	Ensemble à inverser	:	\mathbb{Y} .
	Pavé de recherche <i>a priori</i> contenant \mathbb{X}	:	$\mathbf{x}(0)$.
	Précision requise pour le pavage	:	ε_r .

Initialisation $k := 0$, pile := \emptyset , $\mathbb{X}^- := \emptyset$, $\Delta\mathbb{X} := \emptyset$.

Itération k

- Pas 1 Si $f(\mathbf{x}(k)) \subset \mathbb{Y}$, alors $\mathbb{X}^- := \mathbb{X}^- \cup \mathbf{x}(k)$. Aller au pas 4.
- Pas 2 Si $f(\mathbf{x}(k)) \cap \mathbb{Y} = \emptyset$, alors aller au pas 4.
- Pas 3 Si $w(\mathbf{x}(k)) \leq \varepsilon_r$, alors $\Delta\mathbb{X} := \Delta\mathbb{X} \cup \mathbf{x}(k)$,
sinon bissecter $\mathbf{x}(k)$ suivant un plan principal et empiler les deux pavés résultants.
- Pas 4 Si la pile n'est pas vide, alors dépiler dans $\mathbf{x}(k+1)$, incrémenter k de un et aller au pas 1, sinon FIN ◇

5.2.4. Propriétés de SIVIA

L'algorithme SIVIA génère deux sous-pavages \mathbb{X}^- et $\Delta\mathbb{X}$. La dépendance de ces sous-pavages en ε_r sera omise pour simplifier les notations.

5.2.4.1. Convergence

Démontrons maintenant que l'encadrement de \mathbb{X} généré par SIVIA

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+ \stackrel{\Delta}{=} \mathbb{X}^- \cup \Delta\mathbb{X}, \tag{5.7}$$

définit un intercompact $[\mathbb{X}^-, \mathbb{X}^+]$ renfermant \mathbb{X} et dont le diamètre $w([\mathbb{X}^-, \mathbb{X}^+])$ converge vers zéro lorsque ε_r tend vers zéro.

Lemme 1 : $\lim_{\varepsilon_r \rightarrow 0} h_{\infty}^0(f(\Delta\mathbb{X}), \partial\mathbb{Y}) = 0$. ◇

Ceci signifie que pour ε_r infiniment proche de 0, si $\mathbf{x} \in \Delta\mathbb{X}$, alors $f(\mathbf{x})$ est très proche de la frontière $\partial\mathbb{Y}$. Et donc, même si \mathbf{x} est inacceptable et loin de l'ensemble solution \mathbb{X} , \mathbf{x}

est presque acceptable dans le sens où une légère modification de \mathbb{Y} le rend acceptable.

Démonstration : Si $\mathbf{x} \in \Delta\mathbb{X}$, alors $w(\mathbf{x}) \leq \varepsilon_r$, la fonction d'inclusion convergente f (à fournir en entrée de l'algorithme) vérifie $w(f(\mathbf{x})) \rightarrow 0$ lorsque $\varepsilon_r \rightarrow 0$. Comme $f(\mathbf{x})$ n'est ni à l'intérieur ni à l'extérieur de \mathbb{Y} , il intersecte sa frontière $\partial\mathbb{Y}$. Par conséquent, $\forall \mathbf{x} \in \Delta\mathbb{X}$, $h_\infty^0(f(\mathbf{x}), \partial\mathbb{Y}) \rightarrow 0 \Rightarrow h_\infty^0(f(\Delta\mathbb{X}), \partial\mathbb{Y}) \rightarrow 0$. \langle

Théorème 5.1 : Si f^{-1} est h_∞ - et \bar{h}_∞ -continue autour de \mathbb{Y} alors \mathbb{X}^- , \mathbb{X}^+ et $\Delta\mathbb{X}$ satisfont

- (i) $\Delta\mathbb{X} \xrightarrow{\supset} \partial\mathbb{X}$,
- (ii) $\mathbb{X}^+ \xrightarrow{\supset} \mathbb{X}$,
- (iii) $\mathbb{X}^- \xrightarrow{\subset} \mathbb{X}$ (si \mathbb{X} est plein),

lorsque ε_r tend vers zéro, avec $\xrightarrow{\supset}$ et $\xrightarrow{\subset}$ signifiant respectivement la h_∞ -convergence par l'extérieur et par l'intérieur. \langle

Démonstration

(i) : D'après (4.11), nous avons

$$\begin{aligned} h_\infty^0(f(\Delta\mathbb{X}), \partial\mathbb{Y}) &= \max\{h_\infty(\mathbb{Y}, f(\Delta\mathbb{X}) \cup \mathbb{Y}), h_\infty(\bar{\mathbb{Y}}, f(\Delta\mathbb{X}) \cup \bar{\mathbb{Y}})\} \\ &= \max\{h_\infty(\mathbb{Y}, f(\Delta\mathbb{X}) \cup \mathbb{Y}), \bar{h}_\infty(\mathbb{Y}, \overline{f(\Delta\mathbb{X}) \cup \bar{\mathbb{Y}}})\}. \end{aligned} \quad (5.8)$$

D'après le lemme 1, il en découle

$$\begin{cases} h_\infty(\mathbb{Y}, f(\Delta\mathbb{X}) \cup \mathbb{Y}) \rightarrow 0 \\ \bar{h}_\infty(\mathbb{Y}, \overline{f(\Delta\mathbb{X}) \cup \bar{\mathbb{Y}}}) \rightarrow 0 \end{cases} \quad (5.9)$$

lorsque ε_r tend vers 0. Comme f^{-1} est h_∞ et \bar{h}_∞ continue en \mathbb{Y} , nous avons aussi

$$\begin{cases} h_\infty(f^{-1}(\mathbb{Y}), f^{-1}(f(\Delta\mathbb{X}) \cup \mathbb{Y})) \rightarrow 0 \\ \bar{h}_\infty(f^{-1}(\mathbb{Y}), f^{-1}(\overline{f(\Delta\mathbb{X}) \cup \bar{\mathbb{Y}}})) \rightarrow 0. \end{cases} \quad (5.10)$$

En utilisant la définition 4.4 et en remplaçant $f^{-1}(\mathbb{Y})$ par \mathbb{X} , nous obtenons:

$$\begin{cases} h_\infty(\mathbb{X}, f^{-1} \circ f(\Delta\mathbb{X}) \cup \mathbb{X}) \rightarrow 0 \\ h_\infty(\overline{\mathbb{X}}, f^{-1} \circ f(\Delta\mathbb{X}) \cup \overline{\mathbb{X}}) \rightarrow 0. \end{cases} \quad (5.11)$$

d'où, d'après (4.11), $h_\infty^0(f^{-1} \circ f(\Delta\mathbb{X}), \partial\mathbb{X}) \rightarrow 0$. Or $\Delta\mathbb{X} \subset f^{-1} \circ f(\Delta\mathbb{X})$, et donc d'après (4.7) nous avons $h_\infty^0(\Delta\mathbb{X}, \partial\mathbb{X}) \rightarrow 0$. En utilisant (4.9) et le fait que $\partial\mathbb{X} \subset \Delta\mathbb{X}$, nous obtenons finalement $h_\infty(\Delta\mathbb{X}, \partial\mathbb{X}) \rightarrow 0$, qui démontre (i).

(ii) : D'après (4.12), on a $h_\infty(\Delta\mathbb{X}, \partial\mathbb{X}) \geq h_\infty(\Delta\mathbb{X} \cup \mathbb{X}, \partial\mathbb{X} \cup \mathbb{X}) = h_\infty(\mathbb{X}^+, \mathbb{X}) \rightarrow 0$.

(iii) : Soit $\varepsilon > 0$ un nombre réel infiniment petit. Du fait que \mathbb{X} est plein, il existe un sous-pavage fini $\mathbb{K}_1 \subset \mathbb{X}$ tel que

$$\begin{cases} h_\infty(\mathbb{K}_1, \mathbb{X}) < 2\varepsilon \\ L_\infty(\mathbb{K}_1, \partial\mathbb{X}) > \varepsilon. \end{cases} \quad (5.12)$$

Or, pour ε_r suffisamment petit, $h_\infty(\Delta\mathbb{X}, \partial\mathbb{X}) < \varepsilon$, ainsi $\mathbb{K}_1 \cap \Delta\mathbb{X} = \emptyset$, c'est-à-dire, $\mathbb{K}_1 \subset \mathbb{X}^-$. Quand $\varepsilon \rightarrow 0$, $\mathbb{K}_1 \xrightarrow{c} \mathbb{X}$ et donc $\mathbb{X}^- \xrightarrow{c} \mathbb{X}$. \diamond

5.2.4.2. Complexité

A chaque itération k , $\mathbb{K}(k) = \mathbf{x}(k) \cup \text{pile}(k) \cup \mathbb{X}^-(k) \cup \Delta\mathbb{X}(k) \cup \overline{\mathbb{X}^+}(k)$ forme un pavage du domaine de recherche $\mathbf{x}(0)$. Lorsque ε_r tend vers zéro, le pavage $\mathbb{K}(\varepsilon_r) = \mathbb{K}(k_f(\varepsilon_r))$ s'accumule sur la frontière $\partial\mathbb{X}$, qui est génériquement de dimension $n - 1$, où $k_f(\varepsilon_r)$ est le nombre total d'itérations utilisées par l'algorithme pour une précision de ε_r . Tous les pavés de \mathbb{K} ont une longueur supérieure à ε_r , et donc, le théorème 4.1 implique que, presque sûrement, $(\varepsilon_r)^{-n+1} \triangleleft \#\mathbb{K} \triangleleft (\varepsilon_r)^{-n}$ où $\#\mathbb{K}$ désigne le cardinal du pavage \mathbb{K} . La principale limitation de SIVIA réside donc dans le temps de calcul (proportionnel à $\#\mathbb{K}$ qui augmente exponentiellement avec le nombre de paramètres). Ceci est dû à la nécessité pour le pavage de s'accumuler sur la frontière $\partial\mathbb{X}$ de l'ensemble solution \mathbb{X} . Il semble impossible de donner une limite plus précise de la complexité de l'algorithme car cette complexité dépend d'un grand nombre de facteurs tels que les caractéristiques de \mathbb{X} (forme, longueur, aire de la frontière $\partial\mathbb{X}$, ...) et la position relative de la variété image $f(\mathbb{R}^n)$ par rapport à \mathbb{Y} .

5.2.4.3. Mémoire utilisée

Quand nous nous intéressons uniquement à dessiner une projection de l'ensemble solution \mathbb{X} ou à calculer une de ses caractéristiques telle que son pavé enveloppe $[\mathbb{X}]$ ou son volume $\text{vol}(\mathbb{X})$, seule la pile prend une place significative en mémoire. Il est en effet possible d'éviter de mémoriser les sous-pavages \mathbb{X}^- et $\Delta\mathbb{X}$ en utilisant une technique récursive. Pour d'autres caractéristiques comme les caractéristiques de forme (connexité, convexité), il faut faire très attention d'éviter de mémoriser des informations inutiles en utilisant le concept de *résumé exhaustif* d'un sous-pavage (Jaulin et Walter, 1993a). Sinon, la quantité mémoire requise pour ranger les sous-pavages va augmenter linéairement avec le nombre d'itérations, ce qui peut engendrer un dépassement mémoire même pour des problèmes de dimension modeste. Notons toutefois que le pavage doit être exploré même s'il n'est pas mémorisé. Ainsi, le concept de résumé exhaustif d'un sous-pavage permet d'économiser de la mémoire mais pas du temps de calcul.

Afin de mieux comprendre comment il est possible d'éviter de mémoriser ces sous-pavages, donnons une autre forme de SIVIA dans le cas où la seule caractéristique de \mathbb{X} qui nous intéresse est son volume.

Initialisation

pile := \emptyset , vol⁻ := 0, vol⁺ := 0, \mathbf{x} := $\mathbf{x}(0)$.

Itération k

- | | | | | |
|-------|--|-------|--|---|
| Pas 1 | Si $f(\mathbf{x}) \subset \mathbb{Y}$, | alors | vol ⁻ := vol ⁻ + volume(\mathbf{x}). | |
| | | | vol ⁺ := vol ⁺ + volume(\mathbf{x}). Aller au pas 4. | |
| Pas 2 | Si $f(\mathbf{x}) \cap \mathbb{Y} = \emptyset$, | alors | aller au pas 4. | |
| Pas 3 | Si $w(\mathbf{x}) \leq \varepsilon_f$, | alors | vol ⁺ := vol ⁺ + volume(\mathbf{x}), | |
| | sinon bissequer \mathbf{x} selon le plan principal et empiler les deux pavés résultants. | | | |
| Pas 4 | Si pile $\neq \emptyset$, alors dépiler dans \mathbf{x} et aller au pas 1, sinon FIN. | | | ◇ |

Comme le volume est une caractéristique croissante, nous avons nécessairement en fin d'exécution

$$\text{vol}^- \leq \text{volume}(\mathbb{X}) \leq \text{vol}^+. \quad (5.13)$$

Notons que l'adaptation au cas où la caractéristique qui nous intéresse est le pavé enveloppe $[\mathbb{X}]$ est immédiate. Le théorème suivant met en évidence la place mémoire extraordinairement petite requise par la pile.

Théorème 5.2 : Le nombre d'éléments dans la pile vérifie :

$$\#pile < n \cdot \text{int}(\log_2(w(\mathbf{x}(0))) - \log_2(\epsilon_r) + 1) \quad \diamond$$

Démonstration : Si $\#pile \geq k.n$, alors le plus petit pavé (rangé au sommet de la pile) a pour taille $\epsilon \leq 2^{-k} \cdot w_1$, où w_1 est la taille du plus grand pavé (rangé en fond de pile), car chaque pavé dans la pile est au moins deux fois moins volumineux que celui qui le précède. Comme $w_1 \leq w(\mathbf{x}(0))$, nous avons la proposition suivante

$$\#pile \geq k.n \Rightarrow \exists \mathbf{x} \in \text{pile}, w(\mathbf{x}) \leq 2^{-k} \cdot w(\mathbf{x}(0)). \quad (5.14)$$

La contraposée de cette proposition est :

$$\forall \mathbf{x} \in \text{pile}, w(\mathbf{x}) > 2^{-k} \cdot w(\mathbf{x}(0)) \Rightarrow \#pile < k.n. \quad (5.15)$$

Or $\forall \mathbf{x} \in \text{pile}, w(\mathbf{x}) \geq \epsilon_r > 2^{-k} \cdot w(\mathbf{x}(0))$ pour tout $k > \log_2(w(\mathbf{x}(0))) - \log_2(\epsilon_r)$ et donc pour $k = \text{int}(\log_2(w(\mathbf{x}(0))) - \log_2(\epsilon_r) + 1)$. En utilisant la proposition (5.15), ceci établit le théorème 5.2. \diamond

Remarque 5.1 : Même pour n très grand, la taille maximale de la pile reste raisonnable. Par exemple, si $n = 100$, $w(\mathbf{x}(0)) = 10^4$ et $\epsilon_r = 10^{-10}$, le théorème 5.2 indique que $\#pile < 4600$. \diamond

5.2.4.4. Parallélisation de SIVIA

Les architectures parallèles pour les ordinateurs permettent de traiter, en même temps, un très grand nombre de calculs indépendants. Plus un algorithme possède d'opérations indépendantes, et plus il est parallélisable. SIVIA se parallélise très facilement, et nous en donnons ci-dessous une version parallélisée qui peut fonctionner avec r processeurs.

Entrées

Fonction d'inclusion convergente	:	f .
Ensemble à inverser	:	Y .
Pavé de recherche <i>a priori</i>	:	$x(0)$.
Précision requise pour le pavage	:	ε_r .

Initialisation

$k := 0$, pile := \emptyset , $X^- := \emptyset$, $\Delta X := \emptyset$.

Itération k

- Pas 1 Couper $x(k)$ de façon à former r pavés. Soit Z cet ensemble de pavés.
- Pas 2 Ranger dans X^- tous les pavés z de Z tels que $f(z) \subset Y$.
- Pas 3 Éliminer de Z tous les pavés z tels que $f(z) \cap Y = \emptyset$.
- Pas 4 Ranger dans ΔX tous les pavés z tels que $w(z) \leq \varepsilon_r$.
- Pas 5 Empiler tous les pavés de Z qui restent dans la pile.
- Pas 6 Si la pile n'est pas vide, alors dépiler dans $x(k+1)$,
incrémenter k et aller au pas 1.
- Pas 7 FIN ◇

5.2.4.5. Stratégie de bisection

Il arrive fréquemment que l'espace des paramètres ne soit pas isotrope, parce que les sensibilités de f relativement aux différentes composantes de x ne sont pas du même ordre de grandeur. La technique de bisection suivant l'axe principal suggérée dans la description de l'algorithme peut alors manquer d'efficacité. Le problème qui se pose est donc de choisir une stratégie de bisection qui permette une convergence aussi rapide que possible. Une première idée consiste à pondérer chacune des composantes de x intervenant lors du calcul de la taille de façon à rendre le problème isotrope. Il semble cependant difficile de suggérer une stratégie de bisection rationnelle pour le choix des poids car l'anisotropie dépend souvent de l'endroit considéré dans l'espace des paramètres. Une autre approche consiste à bisectionner suivant la direction de x qui minimise un critère dépendant de paramètres divers comme par exemple de la matrice

jacobienne $J(\mathbf{x})$. Une étude plus approfondie des stratégies de bisection des pavés dans le cadre de l'optimisation globale est proposée dans Kearfott (1987). Lorsque le problème est sévèrement anisotrope, l'amélioration peut être assez spectaculaire. \diamond

5.2.5 Algorithme d'inversion ensembliste de Moore

L'algorithme qui suit (Moore, 1992) a été développé en même temps et indépendamment de SIVIA. L'idée de base est identique, mais la principale différence réside dans la quantité de mémoire nécessaire. En effet, l'algorithme de Moore utilise une file d'attente alors que SIVIA utilise une pile. Il a ici été reformulé par nos soins de façon à utiliser le vocabulaire, les notations et les concepts développés dans ce mémoire ce qui permet de mieux faire ressortir les corrélations entre les deux algorithmes.

Entrées

Fonction d'inclusion convergente	:	f .
Ensemble à inverser	:	Y .
Pavé de recherche <i>a priori</i>	:	$\mathbf{x}(0)$.
Précision requise pour le pavage	:	ϵ_r .

Initialisation

$k := 0$, file := \emptyset , $X^- := \emptyset$.

Itération k

- Pas 1 Bisséquer $\mathbf{x}(k)$ suivant un plan principal en deux pavés \mathbf{x}_1 et \mathbf{x}_2 .
 - Pas 2 Si $f(\mathbf{x}_1) \subset Y$, alors $X^- := X^- \cup \mathbf{x}_1$. Aller au pas 4.
 - Pas 3 Si $f(\mathbf{x}_1) \cap Y \neq \emptyset$ alors mettre \mathbf{x}_1 en fin de file.
 - Pas 4 Si $f(\mathbf{x}_2) \subset Y$, alors $X^- := X^- \cup \mathbf{x}_2$. Aller au pas 6.
 - Pas 5 Si $f(\mathbf{x}_2) \cap Y \neq \emptyset$, alors mettre \mathbf{x}_2 en fin de file.
 - Pas 6 Mettre le premier pavé de la file dans $\mathbf{x}(k)$
 - Pas 7 Si $w(\mathbf{x}(k)) > \epsilon_r$ alors incrémenter k de 1 et aller au pas 1.
- FIN. \diamond

La file est un sous-pavage qui contient à tout instant la frontière ∂X . Elle est identique au sous-pavage ΔX obtenu, pour les mêmes paramètres d'entrées, par SIVIA. A chaque instant, le premier pavé de la file se trouve être le plus volumineux de la file. La quantité mémoire nécessaire pour calculer une caractéristique comme le volume est énorme

puisque l'algorithme de Moore, du fait de la structure de file utilisée, doit mémoriser un nombre minimum de pavés de l'ordre de

$$\#file \geq \text{vol}_{n-1}(\partial\mathbb{X}) * (\epsilon_r)^{-n+1}, \quad (5.16)$$

où vol_p désigne la longueur si $p = 1$, la surface si $p = 2$... La démonstration de cette borne inférieure est triviale. Si, par exemple, nous voulons utiliser cet algorithme pour calculer le volume de la sphère d'équation $x^2 + y^2 + z^2 \leq 1$, avec $\epsilon_r = 0.01$, il nous faudra mémoriser au moins 120 000 pavés, alors qu'avec SIVIA, ce nombre ne pourra dépasser 24. Par contre, lorsque nous cherchons à calculer un encadrement de \mathbb{X} par deux sous-pavages, les deux algorithmes sont strictement équivalents dans le sens où la quantité mémoire nécessaire, le temps de calcul et le résultat obtenu sont strictement identiques.

5.2.6. Discontinuité de la fonction réciproque

La qualité de l'encadrement de \mathbb{X} donné par l'algorithme peut être quantifiée par le calcul de la h_∞ -distance entre \mathbb{X}^- et \mathbb{X}^+ , qui correspond au diamètre de l'intercompact $[\mathbb{X}^-, \mathbb{X}^+]$. Il se peut, dans certains cas très rares, que ce diamètre soit très grand alors que ϵ_r est infiniment petit. Cela est alors dû à une discontinuité de la fonction ensembliste réciproque f^{-1} , situation abordée au paragraphe 4.6 et illustrée par la figure 5.5. Sur cette figure, la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$ qui est continue s'étend en une fonction ensembliste h_∞ -continue $f : C(\mathbb{R}^n) \rightarrow C(\mathbb{R}^p)$. Et pourtant f^{-1} est h_∞ -discontinue autour de \mathbb{Y} comme illustré dans la situation (b) : il est toujours possible de trouver deux compacts \mathbb{Y} et $\mathbb{Y} + d\mathbb{Y}$ tels que $h_\infty(\mathbb{Y} + d\mathbb{Y}, \mathbb{Y})$ tend vers zéro alors que $h_\infty(f^{-1}(\mathbb{Y} + d\mathbb{Y}), f^{-1}(\mathbb{Y}))$ reste éloignée de zéro. Notons que si f est continue, la zone de h_∞ -discontinuité de f^{-1} a une mesure nulle dans $C(\mathbb{R}^p)$. Nous pouvons donc dire que f^{-1} est h_∞ -continue par morceaux.

Les situations telles que celle décrite sur la figure 5.5 (b) ne contredisent pas le théorème de convergence 5.1, puisque celui-ci est valide seulement dans les zones de continuité. Rappelons qu'une description géométrique plus complète de ces problèmes de discontinuité est donnée au chapitre 4.

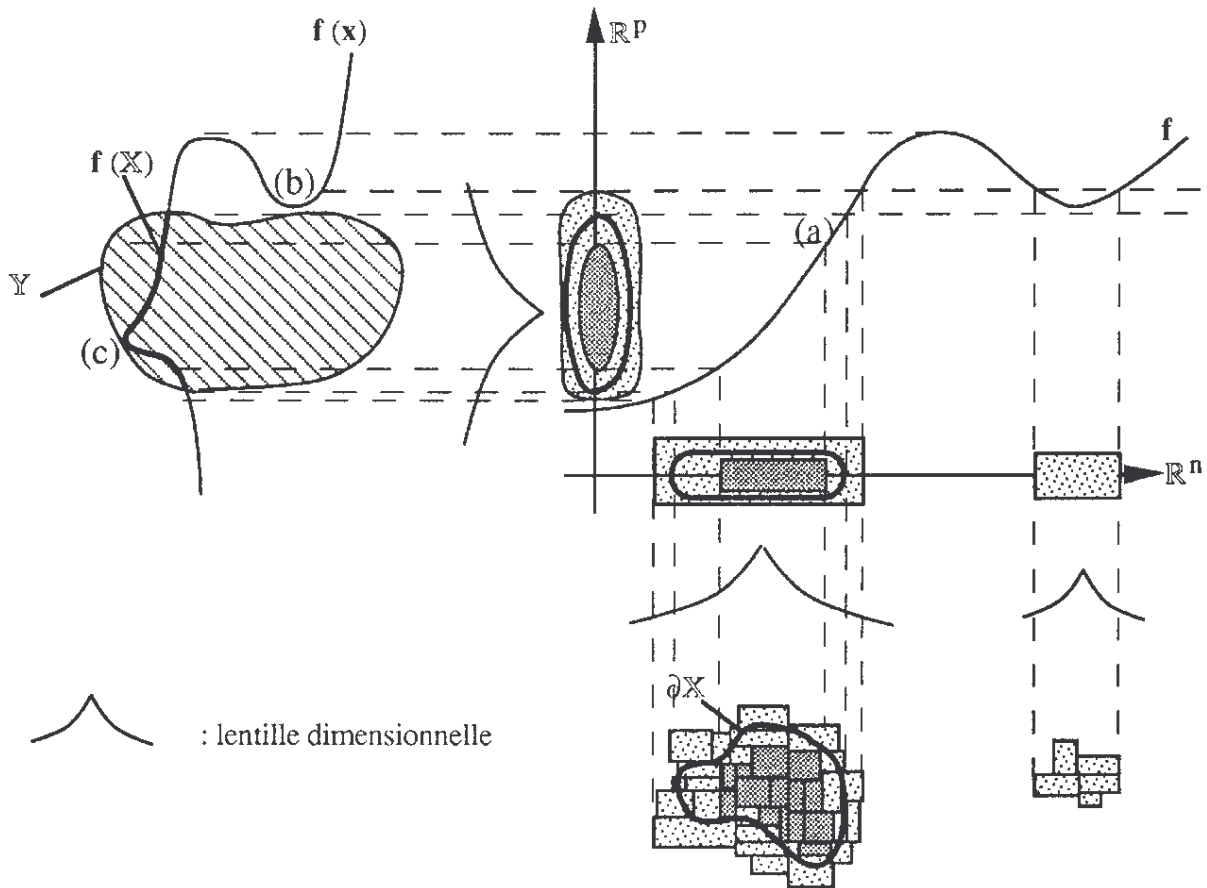


Fig. 5.5 Visualisation globale de la géométrie du problème d'inversion ensembliste.

5.2.7. Inversion intercompacte

Supposons que nous voulions calculer une approximation garantie de l'inverse $[X^-, X^+]$ par la fonction f de l'intercompact $[Y^-, Y^+]$, où les bornes de l'intercompact Y^- et Y^+ sont des sous-pavages. Les compacts $X^- = f^{-1}(Y^-)$ et $X^+ = f^{-1}(Y^+)$ forment un intercompact $[X^-, X^+] = f^{-1}([Y^-, Y^+])$ tel que $\forall Y \in [Y^-, Y^+], f^{-1}(Y) \in [X^-, X^+]$. Mais X^- et X^+ ne sont pas des sous-pavages et on ne sait représenter en machine que les intercompacts dont les bornes sont des sous-pavages. Nous allons donc nous contenter d'un intercompact $[X^{--}, X^{++}] \supset [X^-, X^+]$ et nous aurons ainsi une approximation garantie de l'intercompact solution. SIVIA permet d'encadrer $f^{-1}(Y^-)$ et $f^{-1}(Y^+)$:

$$\begin{aligned} X^{--} &\subset f^{-1}(Y^-) \subset X^{-+}, \\ X^{+-} &\subset f^{-1}(Y^+) \subset X^{++}, \end{aligned} \tag{5.17}$$

L'intercompact $[\mathbb{X}^{--}, \mathbb{X}^{++}]$ que nous permet d'obtenir SIVIA est donc une approximation garantie de l'intercompact inverse $[\mathbb{X}^-, \mathbb{X}^+] = \mathbf{f}^{-1}([\mathbb{Y}^-, \mathbb{Y}^+])$. La convergence de $[\mathbb{X}^{--}, \mathbb{X}^{++}]$ vers $[\mathbb{X}^-, \mathbb{X}^+]$ lorsque la précision ε_r demandée lors de l'utilisation de SIVIA tend vers zéro est presque toujours obtenue. Les conditions théoriques de convergence sont immédiatement obtenues à partir des conditions de convergence de SIVIA, et données par le théorème 5.1.

5.2.8. Reformulation du problème d'inversion

Le choix du paramètre d'arrêt ε_r à spécifier dans les entrées de SIVIA est souvent bien arbitraire. De plus, dans de nombreuses situations, nous ne voulons pas un encadrement de l'ensemble solution, mais nous préférons un résultat plus concis formé d'un seul sous-pavage approximant l'ensemble solution. Dans ce cas, nous proposons de reformuler le problème d'inversion ensembliste comme suit.

Nouveau problème : Soient \mathbb{Y}^- et \mathbb{Y}^+ tels que $\mathbb{Y}^- \subset \mathbb{Y}^+$, trouver un sous-ensemble $\hat{\mathbb{X}}$ tel que $\mathbf{f}^{-1}(\mathbb{Y}^-) \subset \hat{\mathbb{X}} \subset \mathbf{f}^{-1}(\mathbb{Y}^+)$.

$\hat{\mathbb{X}}$, élément de l'intercompact $[\mathbb{X}^-, \mathbb{X}^+] = \mathbf{f}^{-1}([\mathbb{Y}^-, \mathbb{Y}^+])$, nous donne une caractérisation de l'incertitude rattachée à \mathbf{x} même si \mathbb{Y} n'est connu qu'approximativement. L'algorithme suivant est une adaptation de SIVIA à la résolution de ce problème.

- Pas 1 Si $\mathbf{f}(\mathbf{x}(k)) \subset \mathbb{Y}^+$, alors $\hat{\mathbb{X}} := \hat{\mathbb{X}} \cup \mathbf{x}(k)$. Aller au pas 4.
 Pas 2 Si $\mathbf{f}(\mathbf{x}(k)) \cap \mathbb{Y}^- = \emptyset$, alors aller au pas 4.
 Pas 3 Bisséquer $\mathbf{x}(k)$ et empiler les deux pavés résultants.
 Pas 4 Si la pile n'est pas vide, alors dépiler dans $\mathbf{x}(k+1)$, incrémenter k de un et aller au pas 1.
- FIN. (5.18)

Il est aisé de comprendre que, sous les conditions requises par le théorème 5.1, si $\partial\mathbb{Y}^- \cap \partial\mathbb{Y}^+ = \emptyset$, alors un ensemble $\hat{\mathbb{X}}$ peut être obtenu en un temps fini. Une telle approche est intéressante dans le cas de l'estimation à erreurs bornées, en posant $\mathbb{Y}^- = \mathbb{E}$ (ou \mathbb{Y}) et $\mathbb{X} = \mathbb{S}$. Nous avons alors $\mathbb{S} \subset \hat{\mathbb{S}}$ et obtenons donc un résultat garanti. L'ensemble \mathbb{Y}^+ joue alors le rôle du critère d'arrêt. Si l'on est pas trop exigeant, ce critère d'arrêt permet d'obtenir, en un temps beaucoup plus court qu'avec SIVIA, un résultat comparable. L'approche présente également de l'intérêt lorsque l'ensemble \mathbb{Y} à inverser est lui même

incertain. L'ensemble $\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$, obtenu souvent après un temps très long par SIVIA n'est alors pas d'une grande validité. On peut préférer enfermer l'ensemble incertain \mathbb{Y} dans l'intercompact $[\mathbb{Y}^-, \mathbb{Y}^+]$ pour obtenir un ensemble solution $\hat{\mathbb{X}}$ qui vérifie $\mathbf{f}(\hat{\mathbb{X}}) \in [\mathbb{Y}^-, \mathbb{Y}^+]$ et qui contienne (resp. ne contienne pas) tous les paramètres indiscutablement cohérents (resp. incohérents) avec les connaissances que nous avons sur le problème.

5.2.9. SIVIA pour l'analyse d'ensembles à test d'inclusion

Dans beaucoup de problèmes ensemblistes, qui ne sont pas des problèmes d'inversion ensembliste, nous disposons d'un test d'appartenance t , c'est-à-dire d'une procédure qui permet de vérifier qu'un vecteur appartient ou non à cet ensemble, que nous notons \mathbb{X} . Souvent, il est possible, à partir de ce test d'appartenance t , de trouver un test d'inclusion \mathbb{T} pour \mathbb{X} . En remplaçant alors dans SIVIA, le test du pas 1 par " $\mathbb{T}(\mathbf{x}) = \text{vrai}$ " et le test du pas 2 par " $\mathbb{T}(\mathbf{x}) = \text{faux}$ ", l'algorithme générera deux sous-pavages encadrant \mathbb{X} . De plus, si ce test d'inclusion est convergent, la précision de cet encadrement pourra être aussi grande que désirée.

Remarquons que si nous connaissons un test d'appartenance t pour l'ensemble \mathbb{X} , alors,

$$\mathbb{X} = t^{-1}(\{\text{vrai}\}). \quad (5.19)$$

Il s'agit donc là d'un problème d'inversion ensembliste un peu particulier où l'ensemble à inverser est le singleton $\{\text{vrai}\}$.

5.3. Application aux problèmes d'estimation à erreurs bornées

La table 1 donne une traduction dans le contexte de l'estimation à erreurs bornées des notations utilisées pour l'inversion ensembliste avec celles utilisées par la figure 5.1.

Fig. 5.1	Estimation à erreurs bornées
\mathbb{R}^n	espace des paramètres \mathbb{R}^{np}
\mathbb{R}^p	espaces des données (ou des erreurs) \mathbb{R}^{ny}
f	sorties modèles y_m (ou erreurs e)
\mathbb{Y}	ensemble \mathbb{Y} (resp. \mathbb{E}) des sorties modèles (resp. erreurs) acceptables
\mathbb{X}	ensemble \mathbb{S} des paramètres acceptables <i>a posteriori</i>
$f(\mathbb{X})$	ensemble de toutes les sorties modèles (ou erreurs) associées à \mathbb{S}

Table 5.1 Correspondance entre les notations utilisées en estimation et celles utilisées pour SIVIA.

5.3.1. Problème de Milanese et Vicino simplifié

Pour $\varepsilon_r = 0.005$ et $p(0) = [-0.1, 1] \times [-0.1, 1]$, après environ 17 secondes sur un Compaq 386/33, SIVIA génère le pavage représenté par la figure 5.6 en gardant à chaque instant en mémoire un nombre de pavés inférieur à 12 (le théorème 5.2 prédit un nombre inférieur ou égal à 16). Les pavés acceptables (resp. inacceptables et indéterminés) sont représentés en gris foncés (resp. gris clair et blanc). L'ensemble solution \mathbb{S} vérifie à coup sûr

$$[0.316, 0.754] \times [0.059, 0.336] \subset [\mathbb{S}] \subset [0.305, 0.770] \times [0.055, 0.344],$$

$$0.039 \leq \text{vol}(\mathbb{S}) \leq 0.045. \quad (5.20)$$

Pour essayer de donner une confirmation expérimentale des résultats sur la complexité donnés au paragraphe 5.2.4.2, calculons $\#\mathbb{K}$, où \mathbb{K} est le pavage obtenu avec une précision ε_r . Ici, la frontière $\partial\mathbb{S}$ a une dimension $m = 1$. D'après le théorème 4.1 nous devrions avoir $(\varepsilon_r)^{-1} \triangleleft \#\mathbb{K} \triangleleft (\varepsilon_r)^{-2}$ c'est-à-dire

$$1 \leq \lim_{\varepsilon_r \rightarrow 0} \alpha(\varepsilon_r) \leq 2, \text{ où } \alpha(\varepsilon_r) = \frac{\ln(\#\mathbb{K})}{-\ln(\varepsilon_r)}. \quad (5.21)$$

Pour $\varepsilon_r = 0.005$, $\#\mathbb{K} = 1154$, et donc $\alpha(\varepsilon_r) = 1.33$, pour $\varepsilon_r = 0.0005$, $\#\mathbb{K} = 19270$ et donc $\alpha(\varepsilon_r) = 1.3$. La borne supérieure de complexité donnée par le théorème 4.1 semble donc assez pessimiste.

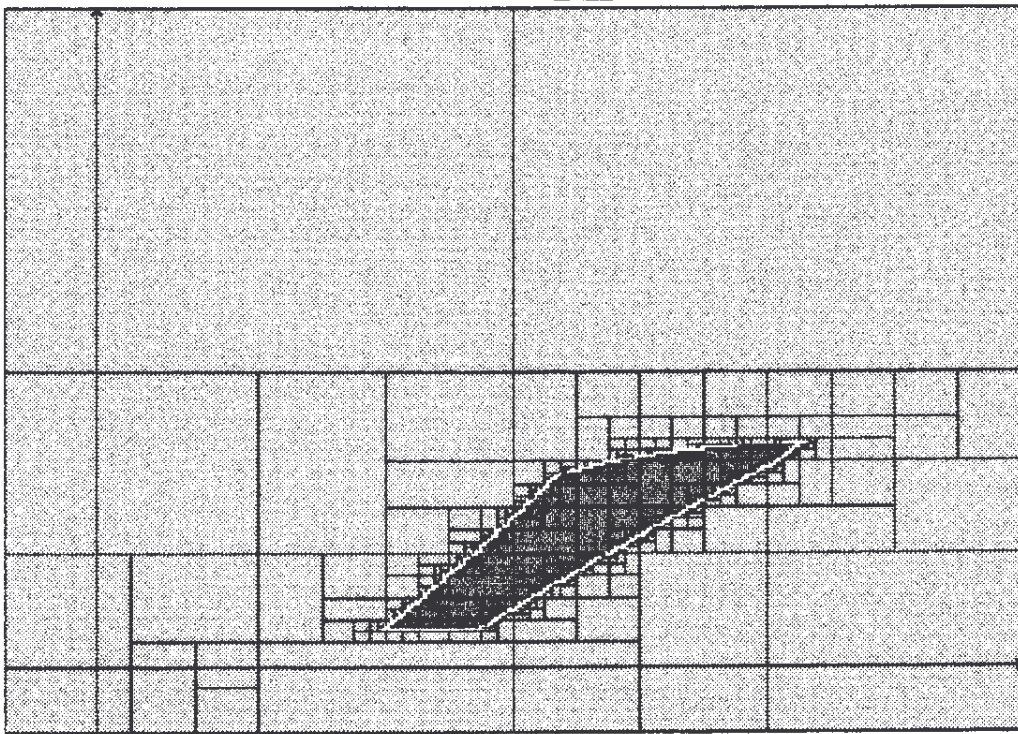


Fig. 5.6 Pavage généré dans l'espace (p_1, p_2) par SIVIA lors de la résolution du problème de Milanese et Vicino simplifié. Le cadre correspond au pavé de recherche $p(0) = [-0.1, 1]^2$.

5.3.2. Problème des exponentielles approximantes

Pour $\varepsilon_r = 0.02$, $p(0) = [0, 5] \times [0, 5]$ et après 16 secondes de calcul sur un Compaq 386/33, SIVIA génère le pavage représenté par la figure 5.7. Le nombre de pavés gardés en mémoire ne dépasse pas 10. Le théorème 5.2 prédit un nombre de pavés inférieur à 16. Les pavés acceptables, inacceptables et indéterminés sont respectivement représentés en gris foncés, gris clair et blanc. L'ensemble \mathcal{S} des vecteurs de paramètres acceptables satisfait à coup sûr

$$[0.342, 1.992] \times [0.420, 2.646] \subset [\mathcal{S}] \subset [0.303, 2.002] \times [0.400, 2.813],$$

$$0.72 \leq \text{vol}(\mathcal{S}) \leq 0.88. \quad (5.22)$$

Le centre du pavé enveloppe (3.19) $[\mathcal{S}]$ est un estimateur ponctuel classique (estimateur du min-max) utilisé dans le contexte des erreurs bornées. Dans le problème des exponentielles approximantes, cet estimateur donne une estimée qui n'est pas acceptable

puisque le centre de $[\mathcal{S}]$ n'est pas dans \mathcal{S} , comme cela est mis en évidence par la figure 5.7. La figure 5.8 représente la superposition de toutes les exponentielles associées aux vecteurs de paramètres acceptables et appartenant donc à \mathcal{S} .

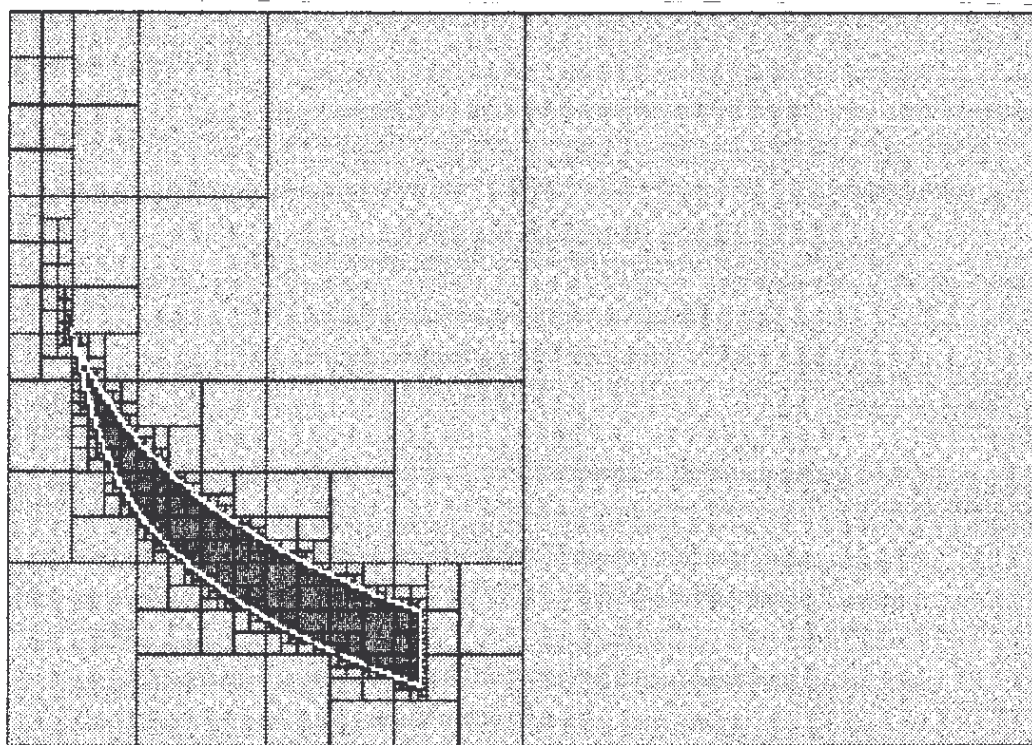


Fig. 5.7 Pavage obtenu par SIVIA dans l'espace (p_1, p_2) lors de la résolution du problème des exponentielles approximantes. Le cadre correspond au pavé de recherche $p(0) = [0, 5]^2$.

5.3.3. Problème de Milanese et Vicino

Redéfinissons les échelles pour les paramètres de telle façon que le pavé de recherche $p(0) = [2, 60] \times [0, 1] \times [-30, -1] \times [0, 0.5]$ devienne un cube de coté 1. Cela change uniquement la stratégie de bisection utilisée pendant le pas 3 de l'algorithme SIVIA. La table 5.2 indique les performances de SIVIA pour différentes précisions ϵ_r . Une importante information fournie par SIVIA, qui ne peut pas être présentée ici pour des raisons évidentes, est une description détaillée de tous les pavés des sous-pavages $\Delta\mathcal{S}$ et \mathcal{S}^- . Cette information est une description bien plus précise de \mathcal{S} que celle donnée par $[\mathcal{S}]$. En utilisant une approche signomiale, rappelée au chapitre 7, Milanese et Vicino (1991b) trouvent une très bonne estimée de $[\mathcal{S}] = [17.2, 26.9] \times [0.3, 0.49] \times [-16.1,$

$-5.4] \times [0.077, 0.136]$ après 10 minutes de calcul sur un VAX 8800. Le volume de l'ensemble d'incertitude sur la position des paramètres fourni par leur algorithme est $\text{vol}(\mathbb{S}) = 1.16$, alors que le volume du pavé de recherche initial est 841.

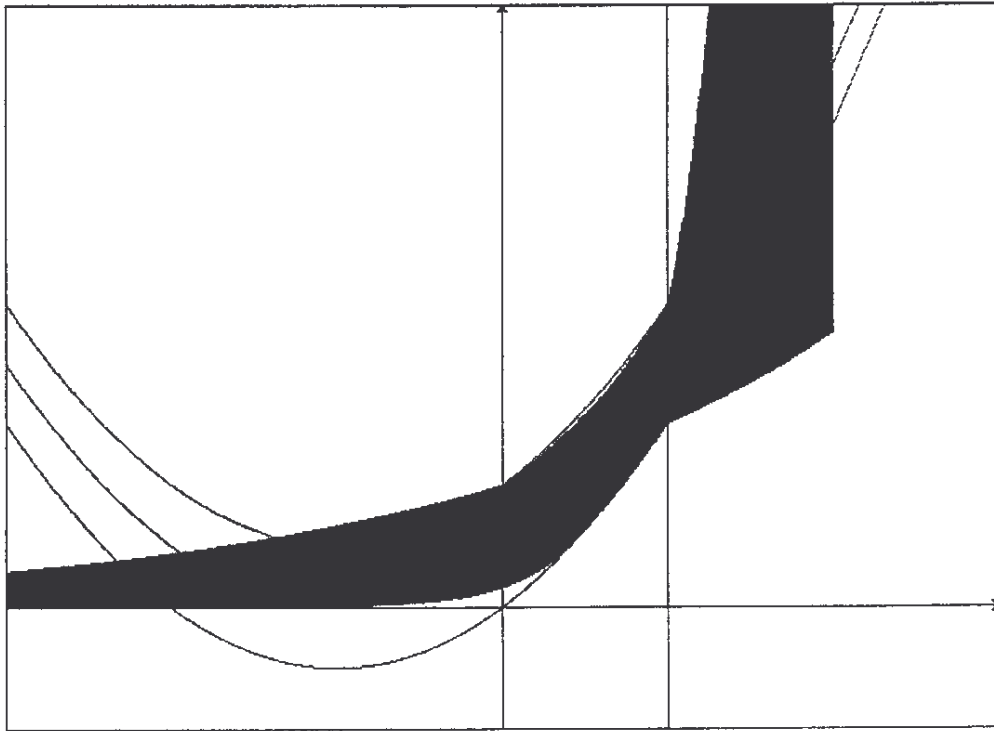


Fig. 5.8 Superposition de toutes les exponentielles acceptables apparaissant sur le cadre $[-3, 3] \times [-2, 10]$ de l'espace (t, y) .

Suivant le but recherché, l'une ou l'autre approche peut être plus adaptée. Si nous sommes intéressés à calculer avec précision l'intervalle d'incertitude associé à chacun des paramètres alors l'approche signomiale est plus efficace. En termes de volume du domaine dont on est sûr qu'il contient les paramètres, l'approche utilisée par SIVIA donne en 44 secondes de meilleurs résultats que ceux que donnent l'approche signomiale en 10 minutes avec un ordinateur de puissance équivalente. En deux minutes, SIVIA divise le volume garanti contenir les paramètres acceptables par un facteur de 10^4 . Le volume résultant $\text{vol}(\mathbb{S}^+)$ est alors 15 fois inférieur au volume de \mathbb{S} .

Remarque 5.2 : Dans de nombreux problèmes pratiques, l'ensemble des erreurs acceptables \mathbb{E} est connu approximativement. Il est donc important de s'intéresser au problème de la sensibilité des estimées obtenues vis-à-vis d'une variation ΔY de Y . D'après le lemme 1, SIVIA va générer un ensemble \mathbb{S}^+ tel que la proximité (4.3) de $y_m(\mathbb{S}^+)$ à Y et à $Y + \Delta Y$ soit petite. Cela veut dire que n'importe quel modèle $M(p)$,

avec \mathbf{p} appartenant à \mathbb{S}^+ , possède un comportement cohérent avec les données. D'un autre côté, \mathbf{p} peut être très loin de \mathbb{S} et pourtant tel que l'erreur $e(\mathbf{p})$ soit suffisamment proche de \mathbb{E} pour que \mathbf{p} appartienne à \mathbb{S}^+ . Notons qu'il peut exister certains paramètres \mathbf{p} dans $[\mathbb{S}]$ qui soient tels que $\mathbf{y}_m(\mathbf{p})$ soit complètement incohérent avec les mesures prélevées. \diamond

ε_r	durée	k_f	vol(\mathbb{S}^+)	vol(\mathbb{S}^-)	#pile	# $\Delta\mathbb{S}$	# \mathbb{S}^-
1	0.27 s	1	841	0	0	1	0
2-1	0.49 s	31	368	0	4	7	0
2-2	1 s	167	135	0	6	41	0
2-3	6 s	765	24	0	9	116	0
2-4	18 s	2 479	3.9	0	11	304	0
2-5	44 s	6 217	0.5	0	14	$3 \cdot 10^3$	0
2-6	2 mn	$14 \cdot 10^3$	$8 \cdot 10^{-2}$	0	16	$7 \cdot 10^3$	0
2-7	5 mn	$37 \cdot 10^3$	$16 \cdot 10^{-3}$	0	19	$5 \cdot 10^3$	0
2-8	15 mn	$12 \cdot 10^4$	$52 \cdot 10^{-4}$	0	23	$3 \cdot 10^4$	0
2-9	1h	$66 \cdot 10^4$	$26 \cdot 10^{-4}$	$26 \cdot 10^{-5}$	27	$19 \cdot 10^4$	$13 \cdot 10^3$
2-10	10 h	$46 \cdot 10^5$	$17 \cdot 10^{-4}$	$6 \cdot 10^{-4}$	30	$15 \cdot 10^5$	$22 \cdot 10^4$

Table 5.2 Performances de SIVIA.

La table 5.3 récapitule les propriétés des deux types d'approximation pour \mathbb{S} c'est-à-dire $[\mathbb{S}]$, que nous fournit par exemple l'approche signomiale, et \mathbb{S}^+ , produit par SIVIA.

$[\mathbb{S}]$	\mathbb{S}^+
1. Précision sur les intervalles d'incertitude de chacun des paramètres.	1. Intervalles d'incertitude pessimiste pour chacun des paramètres
2. Approximation très pessimiste de \mathbb{S} .	2. Description plus détaillée de \mathbb{S} .
3. Les modèles de $M([\mathbb{S}])$ peuvent avoir des comportements très différents.	3. Les modèles de $M(\mathbb{S}^+)$ ont des comportements proches.
4. Résultat concis.	4. Résultat adapté à une exploitation sur ordinateur.

Table 5.3 Comparaison entre les approches signomiale et inversion ensembliste.

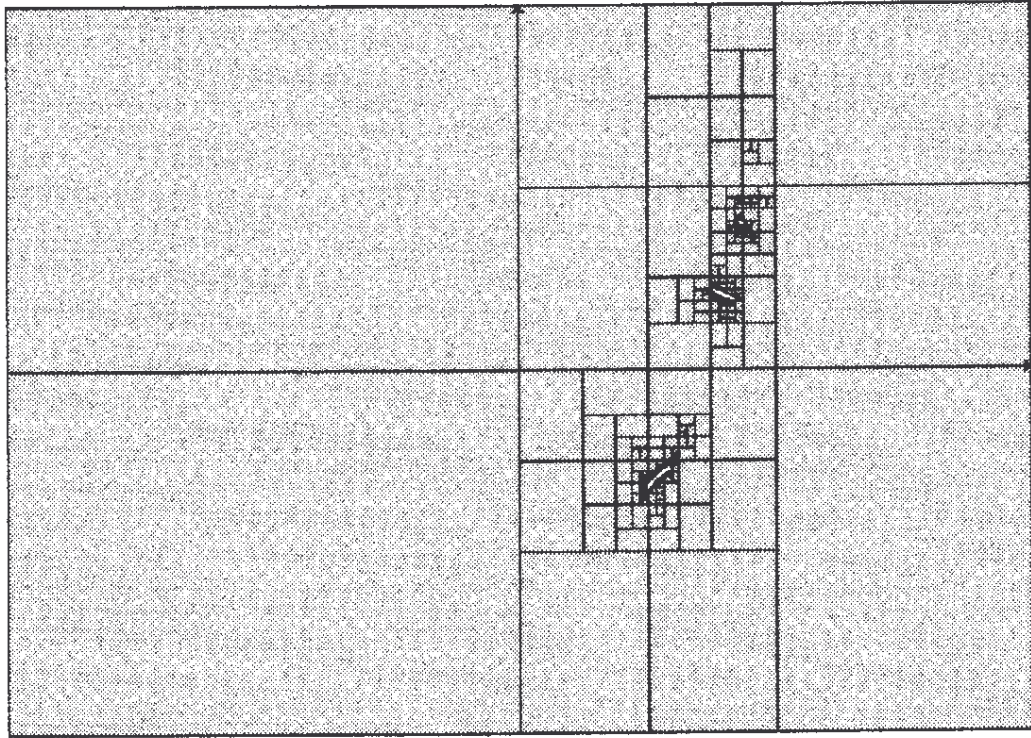


Fig. 5.9 Pavage obtenu par SIVIA pour le problème des conditions initiales. Le cadre correspond au pavé de recherche $\mathbb{p}(0) = [-5, 5] \times [-5, 5]$ de l'espace (p_1, p_2) .

5.3.4. Problème des conditions initiales

Pour $\epsilon_r = 0.01$ et $\mathbb{p}(0) = [-5, 5] \times [-5, 5]$ et en moins de 27 secondes, SIVIA génère le pavage représenté sur la figure 5.9. La pile ne contient jamais plus de 15 pavés. Les pavés inacceptables sont représentés en gris alors que les indéterminés apparaissent en blanc. Aucun pavé acceptable n'a été trouvé pour la précision choisie. Il semble intéressant de noter qu'une méthode de recherche aléatoire sur $\mathbb{p}(0)$ avec une distribution uniforme ne produit, en une heure de temps, aucun point acceptable. Pour comprendre la difficulté du problème, agrandissons la zone située autour de la vraie valeur de \mathbf{p} . Pour $\epsilon_r = 0.0001$, et $\mathbb{p}(0) = [1.98, 2.02] \times [0.98, 1.02]$, en moins de 10 minutes, le pavage représenté sur la figure 5.10 est obtenu. Les pavés acceptables, inacceptables et indéterminés sont représentés respectivement en gris foncé, gris clair et blanc. L'ensemble de vraisemblance \mathbb{S} est si étroit qu'il est pratiquement impossible à atteindre par une exploration aléatoire. Seule une technique déterministe et garantie est à même de fournir une réponse correcte

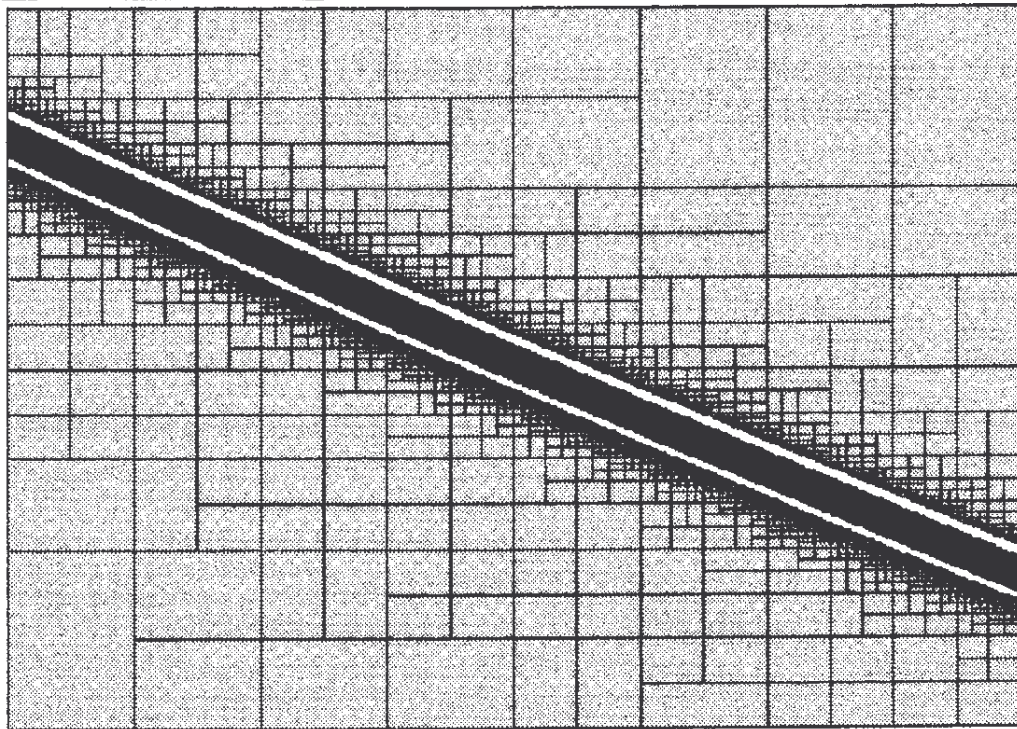


Fig. 5.10 Pavage obtenu par SIVIA pour le problème des conditions initiales après agrandissement de la zone située autour de p^ . Le cadre correspond au pavé de recherche $p(0) = [1.98, 2.02] \times [0.98, 1.02]$ de l'espace (p_1, p_2) .*

5.4. Application aux problèmes de stabilité paramétrique

5.4.1. Utilisation du théorème de Kharitonov

SIVIA s'applique directement pour caractériser le domaine de stabilité d'un système linéaire à paramètres incertains. Il est également possible de combiner la méthode de résolution du problème ensembliste utilisée par SIVIA avec l'utilisation de résultats du type Kharitonov (voir par exemple (Barmish, Hollot, Kraus et Tempo, 1992) pour des résultats récents). Le théorème de Kharitonov donne des conditions nécessaires et suffisantes simples de stabilité du polynôme $\sum_{k=0}^n a_k \cdot s^k$ pour tout \mathbf{a} appartenant au pavé \mathbf{a} . Ainsi, le test du pas 1 de SIVIA pourrait être remplacé par l'application du théorème de Kharitonov sur le pavé $\mathbf{a}(\mathbf{p})$, où \mathbf{a} représente la fonction d'inclusion pour $\mathbf{a}(\mathbf{p})$ que nous considérons. Le test résultant serait plus fort et pourrait accélérer l'exécution de l'algorithme. L'algorithme aurait alors la forme suivante

Entrées

Fonction d'inclusion de la fonction f des coefficients de première colonne de la table de Routh (2.52)	:	f .
Fonctions d'inclusion des coefficients du polynôme	:	a .
Pavé de recherche <i>a priori</i>	:	p .
Précision requise pour le pavage	:	ϵ_r .

Initialisation

pile := \emptyset , $X^- := \emptyset$, $\Delta X := \emptyset$.

Itération

- Pas 1 Si $a(p)$ est stable alors $X^- := X^- \cup p$. Aller au pas 4.
 - Pas 2 Si $\exists i / f_i(p) < 0$ alors aller au pas 4.
 - Pas 3 Si $w(p) \leq \epsilon_r$, alors $\Delta X := \Delta X \cup p$,
sinon bissecter p suivant le plan principal en deux pavés qu'on empile.
 - Pas 4 Si la pile n'est pas vide, alors dépiler dans p , aller au pas 1.
- FIN. (5.23)◇

Toutefois, l'amélioration de l'efficacité de l'algorithme ne serait réellement significative que si le test du pas 2 pouvait être remplacé par un test (similaire à celui de Kharitonov) qui fournirait des conditions nécessaires et suffisantes simples garantissant l'*instabilité* du polynôme $\sum_{k=0}^n a_k \cdot s^k$ pour tout a appartenant au pavé a . A notre connaissance, un tel test n'existe pas, ce qui devrait motiver des recherches ultérieures. ◇

5.4.2. Problème des palmes

Appliquons SIVIA dans le but de caractériser le domaine de stabilité \mathbb{S} du polynôme

$$P_p(s) = s^3 + \sin(p_1 p_2) s^2 + p_1^2 s + p_1 p_2. \tag{5.24}$$

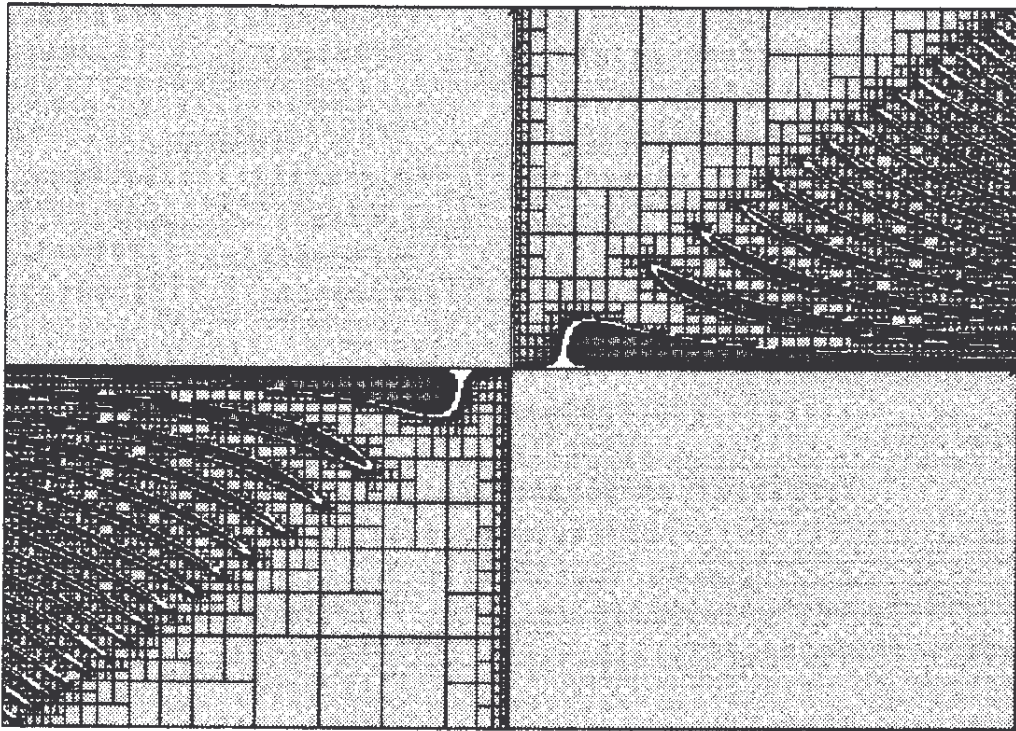


Fig 5.11 Pavage obtenu par SIVIA lors de la résolution du problème des palmes. Le cadre correspond au pavé de recherche $\mathfrak{p}(0) = [-10, 10]^2$ de l'espace (p_1, p_2) .

L'encadrement de \mathbb{S} par les sous-pavages \mathbb{S}^- et \mathbb{S}^+ obtenu par SIVIA pour $\epsilon_r = 0.04$ et $\mathfrak{p}(0) = [-10, 10] \times [-10, 10]$ est présenté sur la figure 5.11 dans le plan (p_1, p_2) . Les pavés gris clairs, blancs et gris foncés représentent respectivement les pavés inacceptables, indéterminés et acceptables. Cette figure a été dessinée en temps réel, sans mémoriser les sous-pavages. La taille de la pile et donc la mémoire requise pour l'algorithme ne dépasse pas 18 pavés. La durée de calcul sur un Compaq 386/33 est inférieure à 2 minutes. Notons que \mathbb{S}^+ contient les axes de l'espace des paramètres, contrairement à \mathbb{S}^- , bien que ceux-ci ne soient pas inclus dans \mathbb{S} . Lorsque ϵ_r tend vers zéro, \mathbb{S}^+ ne converge pas (dans le sens de Hausdorff) vers \mathbb{S} , en raison de la discontinuité de \mathbf{f}^{-1} en $\mathbb{Y} = (]0, \infty[)^3$. Si, par exemple, p_1 est nul, alors $\mathbf{p} \notin \mathbb{S}$, et il est facile de prouver que \mathbf{p} possède un voisinage qui n'intersecte pas \mathbb{S} bien que $\mathbf{f}(\mathbf{p})$ soit infiniment proche de \mathbb{Y} . Pourtant, pour $p_1 = 0$, $P_{\mathbf{p}}(s) = s^3$, qui est instable, possède un voisin stable infiniment proche dans l'espace des polynômes puisque $s^3 + \epsilon s^2 + 2\epsilon s + \epsilon^2$ est un polynôme stable quel que soit ϵ positif. La variété image de \mathbf{f} vient tangenter \mathbb{Y} sans le pénétrer localement.

En utilisant une méthode de D-partition (Leipholz, 1987), nous aurions eu un résultat graphique équivalent, mais SIVIA nous donne en plus un résultat directement exploitable par ordinateur, et ceci quelle que soit la dimension de l'espace des paramètres incertains.

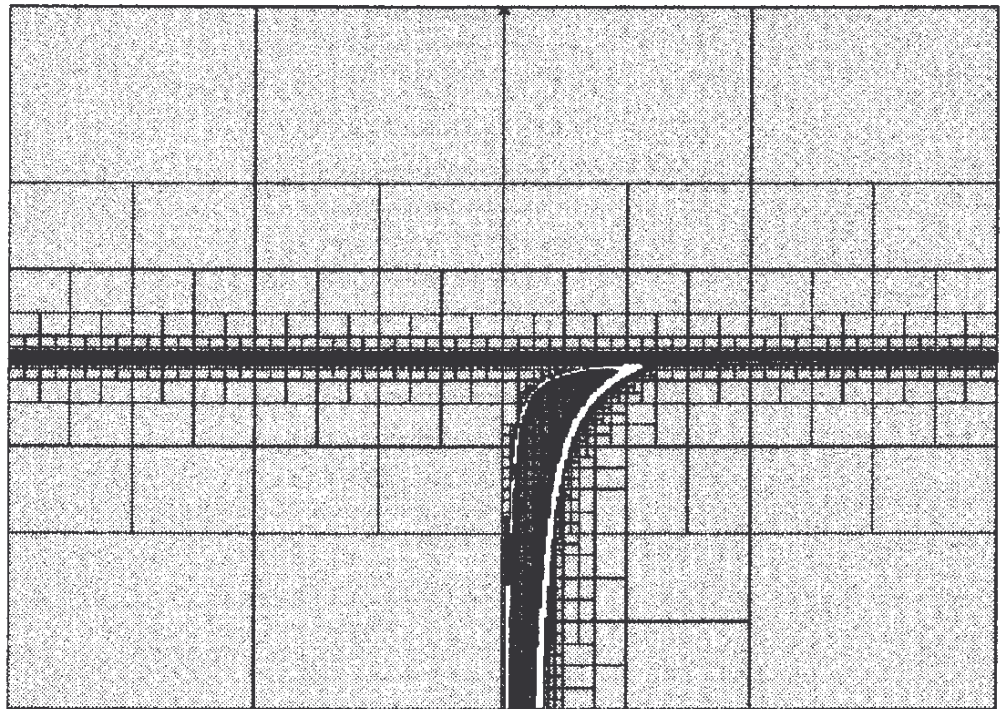


Fig. 5.12 Pavage obtenu par SIVIA lors de la résolution du problème du col de cygne. Le cadre correspond au pavé de recherche $p(0) = [-10, 10]^2$ de l'espace (p_1, p_2) .

5.4.3. Problème du col de cygne (suite)

Les sous-pavages S^- et S^+ représentés sur la figure 5.12 ont été obtenus en 35 secondes avec $\epsilon_T = 0.04$ et $p(0) = [-10, 10] \times [-10, 10]$. Les pavés gris foncés, blancs et gris clairs représentent respectivement les pavés acceptables, indéterminés et inacceptables. Comme pour l'exemple précédent, S^+ ne converge pas vers S , car S^+ contient l'axe d'équation $T = 0$, sur lequel f n'est pas défini. La traînée juste en dessous de l'axe d'équation $T = 0$ sur la droite de la figure 5.12 disparaît quand la précision augmente.

5.5. Conclusion

Quel que soit le nombre de paramètres, l'algorithme d'inversion ensembliste SIVIA est capable d'éliminer très rapidement une large portion de l'espace de recherche avant de se concentrer sur les régions indéterminées. Des résultats théoriques ont été donnés en ce qui concerne la complexité de l'algorithme, le temps de calcul, la taille mémoire nécessaire à son fonctionnement et ses conditions de convergence. La taille mémoire requise reste extrêmement limitée, même quand le nombre de paramètres devient très élevé. Du fait que le temps de calcul augmente exponentiellement avec le nombre de paramètres, SIVIA ne suffit pas lorsqu'il est utilisé avec une grande précision et un grand nombre de paramètres, à moins que le diamètre de l'ensemble solution soit de l'ordre de la précision ε_r demandée. Nous avons appliqué SIVIA à la résolution de problèmes d'estimation non linéaires à erreurs bornées et de caractérisation de domaine de stabilité. Il est clair que de multiples autres applications peuvent être imaginées.

L'estimation à erreurs bornées est un problème d'inversion ensembliste sur lequel SIVIA s'applique et génère un encadrement de l'ensemble de vraisemblance. D'autres méthodes de la littérature permettent aussi d'obtenir un résultat global et garanti dans le cas non-linéaire. C'est le cas de la méthode de programmation signomiale développée par Falk (1973) et appliquée par Milanese et Vicino (1991) dont nous reparlerons au chapitre 7. L'analyse signomiale, lorsqu'elle s'applique, nous donne une description précise du pavé enveloppe enfermant l'ensemble solution \mathbb{S} plus rapidement que SIVIA. D'un autre côté, SIVIA, s'applique à une classe plus large de problèmes et donne une caractérisation plus précise de \mathbb{S} (dans le sens où la perte d'information est moindre).

L'approche utilisée pour caractériser l'ensemble de stabilité, s'applique quelle que soit la dépendance des coefficients du polynôme caractéristique en les paramètres incertains contrairement aux méthodes fondées sur le théorème de Kharitonov. Une fois l'algorithme exécuté, le domaine de recherche a été partitionné en une région où la stabilité du modèle est garantie, une autre région sur laquelle son instabilité est garantie et une région indéterminée qui peut être réduite en augmentant la précision demandée au prix bien sûr de plus de calculs. L'utilisation systématique des fonctions d'inclusion permet de garantir la stabilité de toute une région, une des principales raisons du succès du théorème de Kharitonov chez les praticiens et les théoriciens de la commande des processus.

La principale différence entre SIVIA et l'algorithme proposé par Moore réside dans la quantité mémoire énorme utilisée par ce dernier qui utilise une file pour mémoriser les pavés intermédiaires alors que SIVIA utilise une pile.

CHAPITRE 6

ALGORITHMES POUR D'AUTRES PROBLÈMES ENSEMBLISTES

6.1. Introduction

Nous avons montré l'importance d'une formulation ensembliste des problèmes en estimation et en commande au chapitre 2. Par exemple, dans le cas où le problème ensembliste est associé à un problème ponctuel, une formulation ensembliste permet de caractériser complètement l'incertitude avec laquelle la solution du problème ponctuel est connue. Elle s'adapte particulièrement bien aux problèmes non-linéaires qui demandent à être traités globalement et avec un résultat garanti.

Nous avons vu au chapitre 5 que la structure simple des problèmes d'inversion ensembliste permettait, sous certaines hypothèses peu restrictives, d'enfermer l'ensemble solution \mathbb{X} dans un intercompact $[\mathbb{X}^-, \mathbb{X}^+]$. Pourtant, de nombreux problèmes ensemblistes, que l'on rencontre par exemple dans le domaine de l'estimation et de la commande, ne sont pas de ce type et c'est à ce type de problèmes non-résolus que nous allons nous intéresser maintenant. Chacun de ces problèmes demande une méthode adaptée à son cas, et il est impossible de trouver une méthode universelle qui s'adapte à tous comme c'était le cas pour les problèmes d'inversion ensembliste. Les méthodes de résolution seront cependant toutes fondées sur les concepts d'analyse par intervalles, de sous-pavages, d'intercompacts et d'encadrement d'ensembles.

Nous traiterons le problème d'estimation de régions de confiance dans un contexte probabiliste bayésien puis celui de la commande garantie dans le cas d'un système linéaire entrées-sorties, mais non linéaire paramètres-sorties. Ces deux problèmes ont été introduits dans le chapitre 2.

6.2. Calcul de régions de confiance Bayésiennes

6.2.1. Introduction

Nous avons montré au chapitre 2 que la région de confiance \mathbb{S}_α de probabilité α associée à la variable aléatoire vectorielle \mathbf{x} de loi de probabilité π vérifie le système d'équations ensemblistes

$$\begin{cases} \mathbb{S}_\alpha = \pi^{-1}([s_\alpha, +\infty[) \\ \Pi(\mathbb{S}_\alpha) = \alpha \end{cases} \quad (6.1)$$

où $\Pi(A)$ désigne la probabilité de l'ensemble A . La caractérisation de l'ensemble solution \mathbb{S}_α n'est pas un problème d'inversion ensembliste et ne peut donc pas être résolu directement par la méthode développée au chapitre 5. En gardant toutefois une approche similaire, il est possible d'adapter la méthode (Jaulin et Walter 1994b) afin de développer un algorithme spécifique qui générera deux sous-pavages \mathbb{S}^- et \mathbb{S}^+ tels que $\mathbb{S}^- \subset \mathbb{S}_\alpha \subset \mathbb{S}^+$, encadrant convenablement et en toute garantie l'ensemble solution \mathbb{S}_α .

6.2.2. Test de seuil

Pour encadrer l'ensemble \mathbb{S}_α , une première étape consiste à encadrer le seuil s_α . La proposition suivante fournit des conditions suffisantes pour que l'intervalle ouvert $]s^-, s^+[$ contienne s_α .

Proposition 6.1 : Soient π^- et π^+ deux fonctions encadrant π , c'est-à-dire telles que $\pi^- \leq \pi \leq \pi^+$. Alors,

- (a) $\Pi((\pi^-)^{-1}([s^-, \infty[)) > \alpha \Rightarrow s^- < s_\alpha$;
- (b) $\Pi((\pi^+)^{-1}(]s^+, \infty[)) < \alpha \Rightarrow s^+ > s_\alpha$;

où $(\pi^-)^{-1}$ et $(\pi^+)^{-1}$ sont les fonctions réciproques de π^- et π^+ . ◇

Démonstration : Remarquons tout d'abord que pour tout s , nous avons :

$$(\pi^-)^{-1}(]s, \infty[) \subset \pi^{-1}(]s, \infty[), \quad (6.2)$$

$$\pi^{-1}(]s, \infty[) \subset (\pi^+)^{-1}(]s, \infty[). \quad (6.3)$$

(a) $\Pi((\pi^-)^{-1}(]s^-, \infty[)) > \alpha$, ce qui implique, compte tenu de (6.2), $\Pi(\pi^{-1}(]s^-, \infty[)) > \alpha$. Or $\alpha \stackrel{\Delta}{=} \Pi(\pi^{-1}(]s_\alpha, \infty[))$ donc $s^- < s_\alpha$.

(b) $\Pi((\pi^+)^{-1}(]s^+, \infty[)) < \alpha$, ce qui implique, compte tenu de (6.3), $\Pi(\pi^{-1}(]s^+, \infty[)) < \alpha$. Or $\alpha \stackrel{\Delta}{=} \Pi(\pi^{-1}(]s_\alpha, \infty[)) \Rightarrow]s^+, \infty[\subset]s_\alpha, \infty[\Rightarrow s^+ > s_\alpha$. \diamond

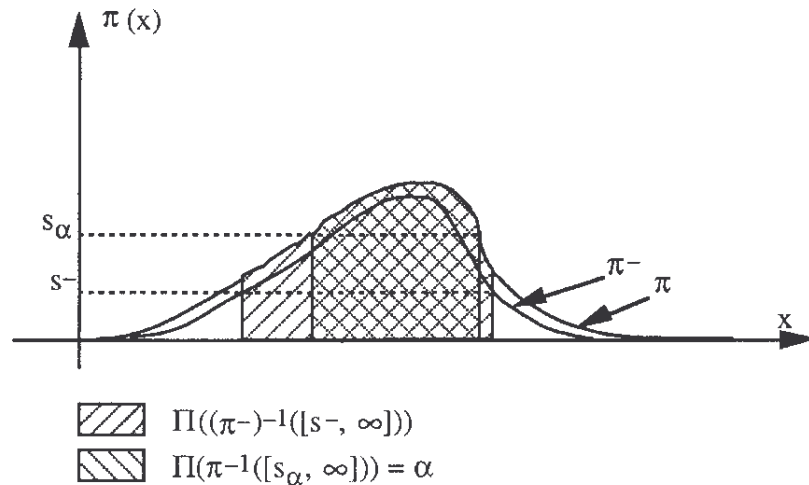


Fig. 6.1a Illustration de la proposition 6.1 (a).

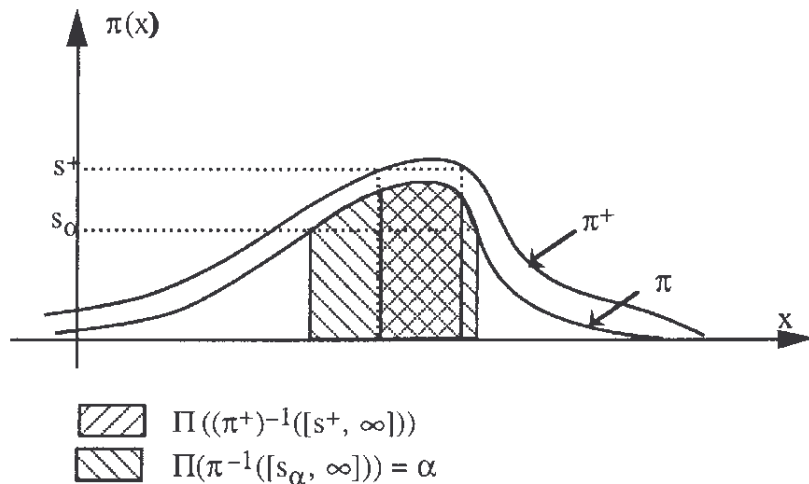


Fig. 6.1b Illustration de la proposition 6.1 (b).

L'ouverture du crochet gauche dans (b) permet d'avoir une implication plus forte. Notons de plus que la proposition 6.1 n'est plus valide si nous considérons des inégalités non strictes dans le cas où π est discontinue. Les figures 6.1a et 6.1b illustrent les deux implications de la proposition 6.1.

6.2.3. Encadrement de la région de confiance

Considérons un pavé $\mathbf{x}(0) \subset \mathbb{R}^n$ suffisamment grand pour que la fonction π puisse être considérée comme nulle en dehors de $\mathbf{x}(0)$. Considérons un pavage \mathbb{K} de $\mathbf{x}(0)$. Il est en général très facile (en utilisant par exemple l'analyse par intervalles) d'obtenir deux fonctions en escalier π^- et π^+ , constantes sur chacun des pavés de \mathbb{K} , qui encadrent la densité de probabilité π . Ainsi, pour tout s , les ensembles $(\pi^+)^{-1}(]s, \infty[)$ et $(\pi^-)^{-1}(]s, \infty[)$ utilisés dans les tests de seuil formeront des sous-pavages de \mathbb{K} . Ceci va être utilisé pour calculer un encadrement $]s^-, s^+[$ pour le seuil s_α à partir d'un pavage \mathbb{K} , d'un encadrement en escalier (π^-, π^+) pour π , et d'une procédure d'intégration calculant la probabilité $\Pi(\mathbf{x})$ d'un pavé \mathbf{x} . Nous en déduisons alors un encadrement de \mathbb{S}_α par deux sous-pavages \mathbb{S}^- et \mathbb{S}^+ formant ainsi un intercompact $[\mathbb{S}^-, \mathbb{S}^+]$ contenant \mathbb{S}_α .

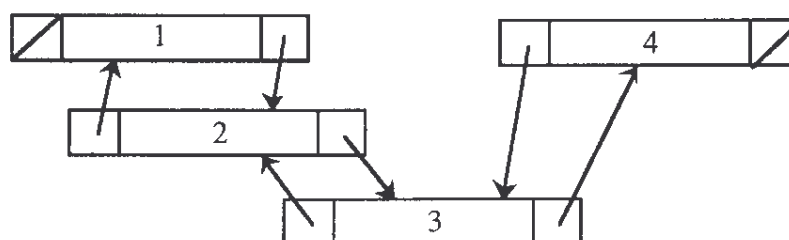


Fig. 6.2 Représentation en machine d'une liste.

Chacun des sous-pavages que nous considérerons lors de l'exécution de l'algorithme sera rangé dans une liste de pavés. Rappelons qu'une liste est une structure informatique contenant des éléments (ici des pavés). Chaque élément de la liste connaît l'emplacement mémoire de l'élément qui le suit et de celui qui le précède. Les enregistrements associés à chacun des éléments de la liste sont donc constitués de trois champs. Le premier contient l'emplacement mémoire de l'élément qui le précède dans la liste, le deuxième mémorise l'élément considéré et le troisième champ contient l'emplacement mémoire de l'élément suivant dans la liste. La figure 6.2 illustre la représentation machine d'une liste à quatre éléments. Sur cette figure, le premier élément porte le numéro 1 et le dernier porte le

numéro 4. Une liste est simplement repérée en machine par l'emplacement mémoire du premier élément.

6.2.3.1. Procédure pour calculer s^-

Trier les pavés de \mathbb{K} dans l'ordre décroissant des valeurs de π^- .

Soit \mathbf{x} le premier élément de la liste \mathbb{K} .

$\alpha^+ := \Pi(\mathbf{x})$

Tant que $\alpha^+ \leq \alpha$

$\mathbf{x} := \text{suivant}(\mathbf{x})$

$\alpha^+ := \alpha^+ + \Pi(\mathbf{x})$

$s^- := \pi^-(\mathbf{x})$.

α^+ représente la somme des probabilités des pavés qui ont été balayés. Nous avons $\Pi\{(\pi^-)^{-1}(\{s^-, \infty\})\} \geq \alpha^+ > \alpha$ et donc d'après la proposition 6.1 (a), $s^- < s_\alpha$.

6.2.3.2. Procédure pour calculer s^+

Trier les pavés de \mathbb{K} dans l'ordre croissant des valeurs de π^+ .

Soit \mathbf{x} le premier élément de \mathbb{K} .

$\alpha^- := 1 - \Pi(\mathbf{x})$

Tant que $\alpha^- \geq \alpha$

$\mathbf{x} := \text{suivant}(\mathbf{x})$

$\alpha^- := \alpha^- - \Pi(\mathbf{x})$

$s^+ := \pi^+(\mathbf{x})$.

Le réel α^- représente la somme des probabilités des pavés non balayés. Ainsi $\Pi\{(\pi^+)^{-1}(\{s^+, \infty\})\} \leq \alpha^- < \alpha$ et donc d'après la proposition 6.1 (b), $s^+ > s_\alpha$.

6.2.3.3. Encadrement de la région de confiance

Posons $\mathbb{S}^- = \{ \mathbf{x} \in \mathbb{K} \mid \pi^-(\mathbf{x}) \geq s^+ \}$ et $\overline{\mathbb{S}^+} = \{ \mathbf{x} \in \mathbb{K} \mid \pi^+(\mathbf{x}) \leq s^- \} \cup \{ \mathbf{x} \in \mathbb{K} \mid \Pi(\mathbf{x}) = 0 \}$. Posons $\mathbb{S}^+ = \mathbb{K} - \overline{\mathbb{S}^+}$, un encadrement de la région de confiance est donné par :

$$\mathbb{S}^- \subset \mathbb{S}_\alpha \subset \mathbb{S}^+. \quad (6.4)$$

Les pavés de probabilité nulle sont rangés dans $\overline{\mathbb{S}^+}$ car, lorsque π est discontinue, il peut arriver que sur un coin \mathbf{x} de \mathbb{K} , $\pi(\mathbf{x})$ soit non nul. Comme l'algorithme récursif qui sera utilisé dans la suite effectue ses bisections sur les pavés de probabilité maximum, \mathbf{x} ne serait jamais éliminé, ni découpé.

6.2.3.4. Procédure récursive d'encadrement de la région de confiance

A partir d'un pavage \mathbb{K} , nous savons maintenant trouver un encadrement de \mathbb{S}_α , qui sera d'autant plus fin que \mathbb{K} le sera. Proposons maintenant un algorithme récursif, qui affine \mathbb{K} afin d'obtenir un encadrement plus précis. Dans ce qui suit, $\mathbb{K} + \mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}$ signifie que l'on rajoute au pavage \mathbb{K} les pavés \mathbf{x}_1 et \mathbf{x}_2 , et que l'on lui enlève le pavé \mathbf{x} .

$\mathbb{K} := \{ \mathbf{x}(0) \}$

Répéter

Soit \mathbf{x} un pavé de \mathbb{K} qui maximise $\Pi(\mathbf{x})$.

Couper \mathbf{x} suivant son axe principal, générant ainsi deux pavés \mathbf{x}_1 et \mathbf{x}_2 .

$\mathbb{K} := \mathbb{K} + \mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}$.

Calculer s^- et s^+ .

Calculer l'encadrement de \mathbb{S}_α par \mathbb{S}^- et \mathbb{S}^+ .

Jusqu'à ce que le critère d'arrêt défini ci-dessous soit satisfait.

Critère d'arrêt : Il est convenable d'arrêter l'algorithme lorsque $\Pi(\mathbb{S}^+) - \alpha \leq \varepsilon$, où ε reste à choisir ($0 < \varepsilon \ll 1$). L'ensemble \mathbb{S}^+ contiendra alors \mathbf{x} avec une probabilité d'au moins α et sera de volume relativement modeste puisqu'il sera inclus dans $\mathbb{S}_{\alpha+\varepsilon}$, qui est le plus petit ensemble (au sens du volume) contenant \mathbf{x} avec une probabilité $\alpha + \varepsilon$. Notons enfin qu'un résultat satisfaisant peut être obtenu pour ε très grand devant la précision numérique. \diamond

Remarque 6.1 : L'algorithme ci-dessus demande un traitement global (différents tris de la liste et calcul de s^- et s^+) d'une liste de taille divergente. L'algorithme proposé dans le paragraphe suivant exploite une structure informatique adaptée qui permet à chaque itération de retrier la liste et recalculer s^- et s^+ en se servant des recherches effectuées aux itérations précédentes. \diamond

Remarque 6.2 : La convergence de S^- et S^+ vers S_α est assurée si la fonction d'inclusion qui à \mathbf{x} associe $[\pi^-(\mathbf{x}), \pi^+(\mathbf{x})]$ est convergente, c'est-à-dire telle que pour toute suite de pavé \mathbf{x}

$$w(\mathbf{x}) \rightarrow 0 \Rightarrow \pi^+(\mathbf{x}) - \pi^-(\mathbf{x}) \rightarrow 0, \quad (6.5)$$

Notons que l'analyse par intervalles permet de trouver de telles fonctions d'inclusion pour la plupart des loi de probabilité π pouvant être rencontrées dans les problèmes réels. L'intercompact $[S^-, S^+]$ définit dans l'ensemble des compacts un voisinage de l'ensemble solution S_α . \diamond

6.2.4. Algorithme efficace pour encadrer les régions de confiance

La méthode que nous venons de proposer pour l'encadrement de régions de confiance n'est pas directement programmable avec un langage structuré de type PASCAL ou C. Reste en effet à organiser la gestion en mémoire de l'ensemble des données à traiter à chaque itération. Nous allons maintenant décrire un algorithme récursif efficace et directement programmable avec un langage évolué pour encadrer par deux sous-pavages la région de confiance S_α . Le principe de l'algorithme est le même que ci-dessus, mais il utilise une gestion de la mémoire qui en améliore considérablement l'efficacité et permet une transcription immédiate en un langage structuré de type Pascal ou C. Le lecteur qui ne s'intéresse pas à cette partie plutôt technique, peut aller directement à l'exemple d'application du paragraphe 6.2.5, qui concerne la résolution du problème de la région de confiance triangulaire exposé au chapitre 2. Les résultats qui seront obtenus dans cet exemple d'application seront calculés à partir de cet algorithme récursif.

6.2.4.1. Structure et variables utilisées

Nous utiliserons une *multiliste* qui est une liste de pavés, triée à la fois par ordre décroissant de Π , décroissant de π^- et croissant de π^+ . Cette multiliste est représentée sous la forme d'une union d'enregistrements liés par des liens de succession représentés par des pointeurs (entité informatique contenant l'adresse de l'enregistrement concerné). Un enregistrement contient un pavé \mathbf{x} , sa probabilité $\Pi(\mathbf{x})$, un minorant $\pi^-(\mathbf{x})$ de π sur \mathbf{x} , un majorant $\pi^+(\mathbf{x})$ de π sur \mathbf{x} et six pointeurs, respectivement vers les π^- , π^+ et Π -suivants et les π^- , π^+ et Π -précédents de \mathbf{x} . Par l'expression π^- -suivant (resp π^+ , Π -suivant), il faut comprendre l'enregistrement suivant au sens de π^- (resp π^+ , Π). La figure 6.3 illustre un tel enregistrement. Sur cette figure, les flèches représentent des pointeurs vers d'autres enregistrements, c'est-à-dire que les cases d'où partent les flèches contiennent les adresses des enregistrements concernés.

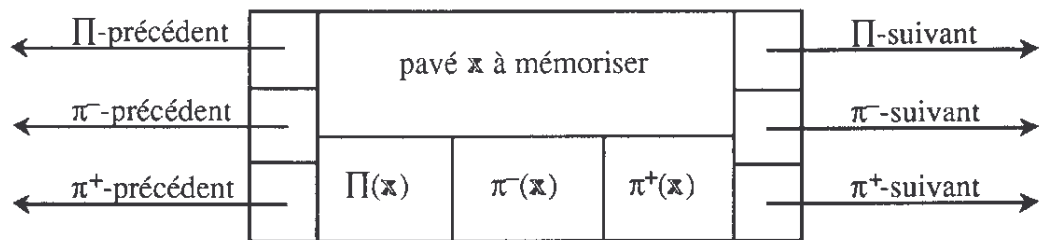


Fig. 6.3 Enregistrement représentant un pavé dans la multiliste.

En mémoire, les enregistrements sont désordonnés, mais cette structure dynamique de multiliste mémorise toutes les relations d'ordre utilisées, et rend possible un changement immédiat de la relation d'ordre considérée sans avoir à modifier l'implantation en mémoire de la multiliste.

La multiliste contient tous les pavés dont nous n'avons pas su déterminer s'ils étaient à l'intérieur ou à l'extérieur de \mathbb{S}_α . Ceux dont il a été démontré qu'ils étaient à l'intérieur (resp. à l'extérieur) de \mathbb{S}_α sont rangés dans \mathbb{S}^- (resp. dans $\overline{\mathbb{S}^+}$). Le pavé courant \mathbf{x} , assimilé désormais à l'enregistrement (pointé par le pointeur z) qui le contient, sera noté \hat{z} . Le pointeur z_Π (resp. z_{π^-} , z_{π^+}) pointe vers un pavé \hat{z}_Π (resp. \hat{z}_{π^-} , \hat{z}_{π^+}) qui maximise Π (resp. maximise π^- , minimise π^+). Ces pointeurs sont donc les premiers éléments de la multiliste suivant chacun des critères de tri que nous nous sommes proposés. Le pointeur $z_{\Pi\text{end}}$ pointe vers le dernier élément de la multiliste triée suivant Π , et sa probabilité est

donc minimale. La figure 6.4 représente cette multiliste lorsque celle-ci contient seulement trois pavés : a, b, c.

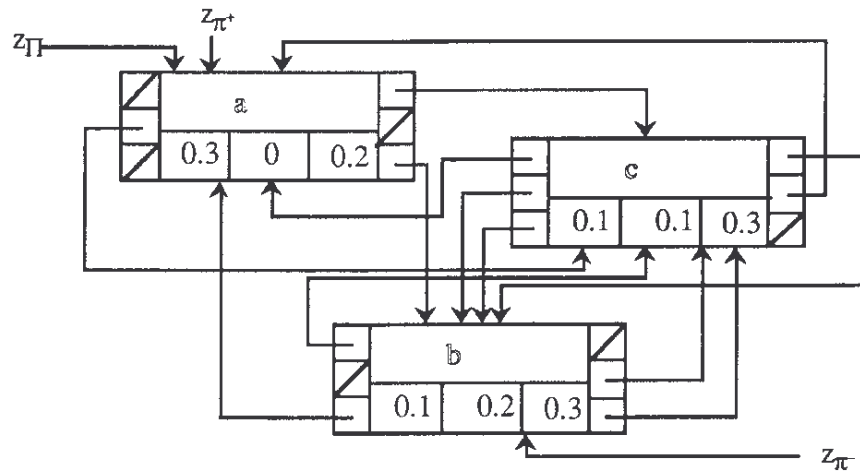


Fig. 6.4 Représentation en machine d'une multiliste à 3 éléments.

Sur cette figure, selon que la relation d'ordre retenue est Π (resp. π^-, π^+), la multiliste équivaut à la liste {a, c, b} (resp. {b, c, a}, {a, b, c}). Aux pointeurs auxiliaires z_{α^+} et z_{α^-} sont associés deux réels α^- et α^+ . Ces deux variables sont mises à jour récursivement à chaque modification de la multiliste. La variable α^- vaut la somme de la probabilité de S^- , avec celle de tous les pavés qui suivent strictement (au sens de π^+) le pavé pointé par z_{α^-} . La variable α^+ vaut la somme de la probabilité de S^- , avec celle de tous les pavés qui précèdent non strictement (au sens de π^-) le pavé pointé par z_{α^+} .

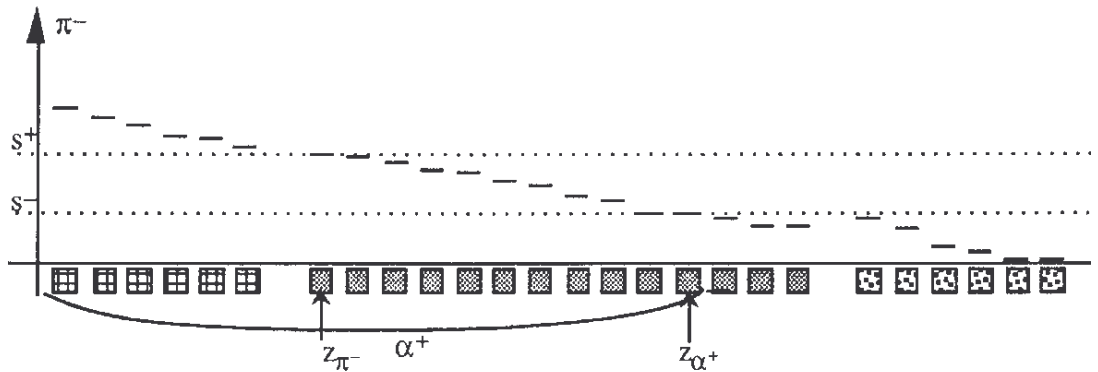


Fig. 6.5a Pavés de S^- , de la multiliste et de S^+ triés dans l'ordre décroissant des π^- .

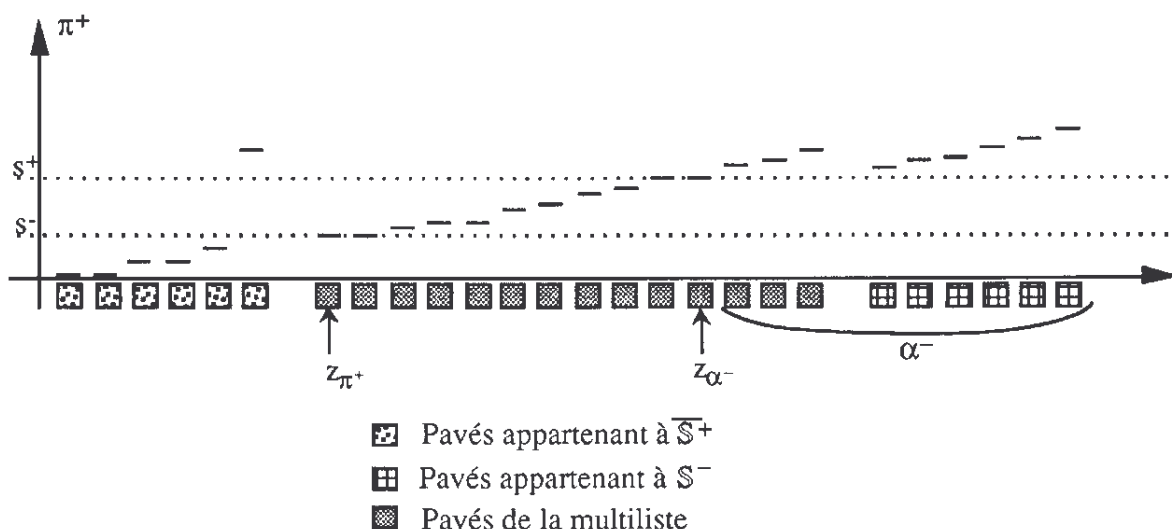


Fig 6.5b Pavés de \overline{S}^+ , de la multiliste et de S^- triés dans l'ordre croissant des π^+ .

Fig 6.5 Illustration de la procédure d'encadrement de seuil dans l'algorithme.

Les figures 6.5 présentent S^- , \overline{S}^+ et les pavés de la multiliste, qui forment un pavage complet de $\mathbf{x}(0)$, trié suivant π^- , puis suivant π^+ . Remarquons qu'un pavé de \overline{S}^+ peut avoir un π^+ supérieur à s^- . Ceci vient du fait que la probabilité de ce pavé est nulle et que son intersection avec S_α est donc de mesure nulle.

6.2.4.2. Algorithme

L'algorithme récursif qui suit, utilise la méthode développée au paragraphe 6.2.3 avec la structure de la multiliste dont nous venons de parler. Cette organisation en mémoire des pavés permet d'enregistrer à chaque itération toutes les informations nécessaires pour passer à l'itération suivante.

Entrées : $\mathbf{x}(0)$, ε , α .

Initialisation

$$\begin{aligned} z_{\Pi\text{end}} &:= z_{\Pi} := z_{\pi^+} := z_{\pi^-} := z_{\alpha^+} := z_{\alpha^-} := \text{adresse}(\mathbf{x}(0)), \\ S^- &:= \overline{S}^+ := \emptyset, \end{aligned}$$

$$s^- := \pi^-(\mathbf{x}(0)), s^+ := \pi^+(\mathbf{x}(0)),$$

$$\alpha^+ := 1, \alpha^- := 0.$$

Itération k :

- Pas 1 Couper \hat{z}_Π , le pavé pointé par z_Π , en deux suivant l'axe principal.
- Pas 2 Insérer les deux pavés résultant de la bisection dans la multiliste et retrier cette dernière suivant les ordres décroissant de Π , décroissant de π^- et croissant de π^+ . Pointer z_Π sur le pavé de probabilité la plus élevée z_{π^-} sur celui qui maximise π^- et z_{π^+} , sur celui qui minimise π^+ . Retirer \hat{z}_Π de la multiliste.
- Pas 3 Pointer z_{α^+} sur le pavé tel que $\alpha^+ > \alpha$, α^+ soit minimum, et tel que $\pi^-(\pi^- \text{-suivant}(\hat{z}_{\alpha^+})) \neq \pi^-(\hat{z}_{\alpha^+})$. Pointer z_{α^-} sur le pavé tel que $\alpha^- < \alpha$, α^- soit maximum, et tel que $\pi^+(\pi^+ \text{-suivant}(\hat{z}_{\alpha^-})) \neq \pi^+(\hat{z}_{\alpha^-})$.
- Pas 4 $s^- := \pi^-(\hat{z}_{\alpha^+})$.
- Pas 5 $s^+ := \pi^+(\hat{z}_{\alpha^-})$.
- Pas 6 Si $\pi^-(\hat{z}_{\pi^+}) \geq s^+$, alors $\mathbb{S}^- := \mathbb{S}^- + \hat{z}_{\pi^+}$, enlever \hat{z}_{π^+} de la multiliste.
- Pas 7 Si $\pi^+(\hat{z}_{\pi^-}) \leq s^-$, alors $\overline{\mathbb{S}^+} := \overline{\mathbb{S}^+} + \hat{z}_{\pi^-}$, enlever \hat{z}_{π^-} de la multiliste.
- Pas 8 Si probabilité $\Pi(\hat{z}_{\Pi \text{end}}) = 0$, alors $\overline{\mathbb{S}^+} := \overline{\mathbb{S}^+} + \hat{z}_\Pi$, enlever \hat{z}_Π de la multiliste.
- Pas 9 Remettre la multiliste à jour en actualisant les pointeurs.
- Pas 10 Si $1 - \Pi(\overline{\mathbb{S}^+}) - \alpha \geq \varepsilon$ alors aller au pas 1.
- Pas 11 FIN.

6.2.5. Application au problème de la région de confiance triangulaire

Supposons, comme dans le chapitre 2, qu'après application de la règle de Bayes l'expression analytique obtenue pour la densité *a posteriori* pour le vecteur des paramètres \mathbf{p} soit donnée par

$$\pi(\mathbf{p}) = \begin{cases} \exp(-p_1) * \exp(-p_2) & \text{si } \mathbf{p} \geq \mathbf{0} \\ 0 & \text{sinon} \end{cases} \quad (6.6)$$

Caractérisons l'ensemble $\mathbb{S}_{0.95}$ de volume minimal tel que $\Pi(\mathbb{S}_{0.95}) = 0.95$, avec la

méthode développée ci-dessus. Si \mathfrak{p} est un pavé, alors un encadrement de π sur \mathfrak{p} est donné par :

$$\pi^-(\mathfrak{p}) = \pi(p_1^+, p_2^+) \text{ si } p_1^- \text{ ou } p_2^- \geq 0 \text{ et } 0 \text{ sinon ,} \quad (6.7)$$

$$\pi^+(\mathfrak{p}) = \pi(\text{sup}(0, p_1^-), \text{sup}(0, p_2^-)) \text{ si } p_1^+ \text{ ou } p_2^+ \geq 0 \text{ et } 0 \text{ sinon} \quad (6.8)$$

La probabilité de \mathfrak{p} s'obtient après un simple calcul d'intégrale double sous la forme :

$$\begin{aligned} \Pi(\mathfrak{p}) = & (\exp(-\text{sup}(0, p_1^-)) - \exp(-\text{sup}(0, p_1^+))) \\ & * (\exp(-\text{sup}(0, p_2^-)) - \exp(-\text{sup}(0, p_2^+))) \end{aligned} \quad (6.9)$$

Choisissons $\mathfrak{p}(0) = [-1000, 1000] \times [-1000, 1000]$, $\alpha = 0.95$, $\varepsilon = 0.01$. En moins de 12 secondes, l'algorithme décrit génère la figure 6.6. Notons que \mathbb{S}_α a bien une allure triangulaire et correspond tout à fait au résultat calculé analytiquement au chapitre 2.

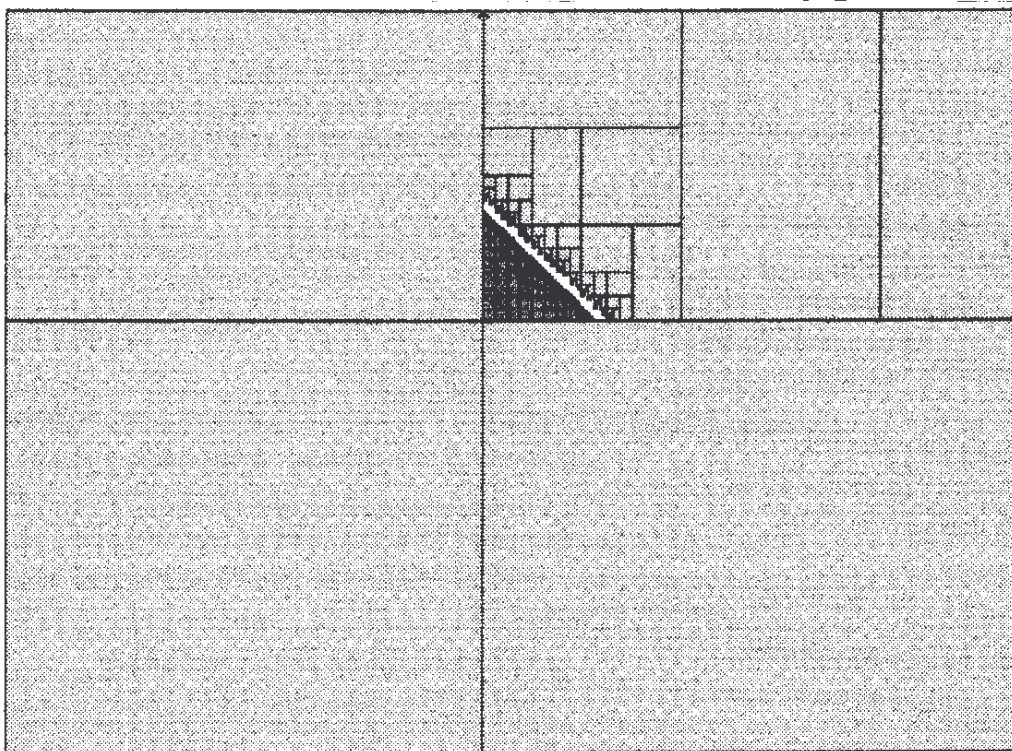


Fig. 6.6 Région de confiance Bayésienne à 95 %.

6.2.6. Conclusion

L'algorithme proposé est à notre connaissance le premier qui permette d'approcher de façon garantie une région de confiance de volume minimal pour des densités de probabilité non gaussienne, en l'enfermant dans un intercompact dont les bornes sont des sous-pavages. Les conditions de convergence se comprennent aisément, mais il reste toutefois à donner des conditions théoriques suffisantes sur π qui assurent cette convergence. L'intérêt de l'approche pour l'approximation d'une région de confiance \mathbb{S}_α est que lorsque α est proche de 1, \mathbb{S}_α , et donc l'union de pavés obtenue par notre algorithme, reste de volume aussi modeste que possible et contient le paramètre à estimer avec une forte probabilité.

6.3. Caractérisation de l'ensemble stabilisant

6.3.1. Introduction

Nous nous intéressons ici à trouver les caractéristiques de l'ensemble des paramètres \mathbb{S}_c d'une structure de commande en boucle fermée qui stabilisent tous les modèles $M(\mathbf{p})$ tels que $\mathbf{p} \in \mathbb{S}_p$, où \mathbb{S}_p a été déterminé, par exemple, par une technique d'estimation ensembliste. Nous avons vu au chapitre 2 que l'ensemble \mathbb{S}_c pouvait être mis sous la forme

$$\mathbb{S}_c = \{\mathbf{c} \mid \forall \mathbf{p} \in \mathbb{S}_p, \mathbf{f}(\mathbf{c}, \mathbf{p}) > \mathbf{0}\} \quad (6.10)$$

où \mathbf{f} est une fonction vectorielle obtenue à partir de la table de Routh. Nous nous plaçons bien sûr dans le cas de structures de modèles et de régulateurs invariants dans le temps et linéaires entrées-sorties. En utilisant le même type d'approche qu'au chapitre 5 pour encadrer l'ensemble solution d'un problème d'inversion ensembliste, nous allons construire des sous-pavages \mathbb{S}^- et \mathbb{S}^+ qui encadrent l'ensemble solution \mathbb{S}_c dans le sens où

$$\mathbb{S}^- \subset \mathbb{S}_c \subset \mathbb{S}^+. \quad (6.11)$$

Les sous-pavages \mathbb{S}^- et \mathbb{S}^+ forment ainsi un intercompact $[\mathbb{S}^-, \mathbb{S}^+]$ contenant \mathbb{S}_c à coup sûr. L'algorithme de résolution est composé de deux algorithmes A1 et A2. A1 met en œuvre un test d'inclusion pour décider si un pavé \mathbf{c} se trouve à l'intérieur ou à l'extérieur

de \mathbb{S}_c . Il correspond à un sous programme de A2 qui effectue l'encadrement ensembliste.

6.3.2. Algorithme A1

Quand il est possible de calculer une fonction d'inclusion $f(c,p)$ de la fonction f , deux conditions suffisantes nous permettent d'étudier la stabilité de l'ensemble des polynômes associés au pavé $c \times p$:

$$f(c,p) > 0 \Rightarrow P_{c \times p}(s) \text{ est stable,} \quad (6.12)$$

$$\exists i \mid f_i(c,p) < 0 \Rightarrow P_{c \times p}(s) \text{ est instable,} \quad (6.13)$$

où $P_{c \times p}(s)$ représente l'ensemble des polynômes $P_{c,p}(s)$ tels que $c \in c$ et $p \in p$. Dire que $P_{c \times p}(s)$ est stable, resp. instable, est équivalent à dire que tous les éléments de $P_{c \times p}(s)$ sont stables, resp. instables. A1 exploite ces deux conditions suffisantes pour démontrer l'une ou l'autre des deux propositions

$$c \subset \mathbb{S}_c \text{ c'est-à-dire que } c \text{ est stabilisateur,} \quad (6.14)$$

$$c \cap \mathbb{S}_c = \emptyset \text{ c'est-à-dire que } c \text{ n'est pas stabilisateur.} \quad (6.15)$$

Notons \mathbb{K}_p un sous-pavage de l'ensemble des paramètres qui contient \mathbb{S}_p de telle façon que $h_\infty(\mathbb{S}_p, \mathbb{K}_p)$ soit négligeable. L'algorithme A1 utilise une pile S_p qui mémorise les pavés courants p de l'espace des paramètres du système à commander. La variable binaire b prend la valeur *vrai* si A1 peut encore conclure que $c \subset \mathbb{S}_c$ et b prend la valeur *faux* dans le cas contraire.

Paramètres d'entrées transmis par le programme principal A2 :

Sous-pavage de l'espace des paramètres vraisemblables	:	\mathbb{K}_p
Pavé des paramètres du système de commande à étudier	:	c

Initialisation

Mettre tous les pavés de \mathbb{K}_p en mémoire dans la pile S_p .

$b := \text{vrai}$.

Itération k

- Pas 1 Dépiler S_p dans p .
- Pas 2 Si $\exists i \mid f_i(c, \text{centre}(p)) < 0$ alors retourner $c \cap S_c = \emptyset$. FIN.
- Pas 3 Si $f(c, p) > 0$ alors aller au pas 6.
- Pas 4 Si $w(p) < w(c)$, $b := \text{faux}$ aller au pas 6.
- Pas 5 Bisséquer p suivant son axe principal et empiler des deux pavés résultants.
- Pas 6 Si S_p n'est pas vide alors aller au pas 1.
- Pas 7 Si $b = \text{vrai}$ alors retourner $c \subset S_c$. FIN.
- Pas 8 Conclure à l'échec. FIN.

Si A1 conclut au pas 2 : Il existe un vecteur de paramètres $p \in S_p$ tel qu'aucun des éléments de c ne stabilise le modèle $M(p)$ et donc $c \cap S_c = \emptyset$.

Si A1 conclut au pas 7 : $b = \text{vrai}$ et S_p est vide. Ce qui veut dire que tous les pavés descendant de K_p vérifient les inégalités de stabilité. Ainsi, $c \subset S_c$.

Si A1 conclut au pas 8 : $b = \text{faux}$ et donc l'algorithme n'a pas pu démontrer que tous les éléments de c stabilisaient K_p . De plus il n'a trouvé aucun p qu'aucun des éléments de c ne stabilisait. Nous ne pouvons donc rien conclure. L'algorithme A2 devra alors chercher à lever l'ambiguïté en bissant c , à moins que la taille de celui-ci soit inférieure à la précision demandée. Remarquons que le test du pas 4, qui a conduit à donner à b la valeur *faux*, permet d'éviter de découper trop finement le pavé p quand le pavé c reste grossier.

Remarque 6.3 : A1 forme un test d'inclusion pour l'ensemble S_c . Dans la plupart des situations, ce test semble être convergent. Ainsi, SIVIA peut s'appliquer pour encadrer S_c par deux sous-pavages. L'algorithme A2, proposé au prochain paragraphe, est donc une réécriture, avec les notations adaptées, de SIVIA. ◇

6.3.3. Algorithme A2

A2 encadre l'ensemble solution \mathbb{S}_c en appelant le sous-programme A1.

Entrées du programme fournies par l'utilisateur

Pavé de recherche dans l'espace des paramètres de contrôle : $c(0)$.
Précision demandée pour le pavage : ϵ_r .

Initialisation

$\mathbb{S}_c := \emptyset, \mathbb{S}^- := \emptyset, \mathbb{S}^+ := \emptyset.$
 $c := c(0).$

Itération k

Pas 1 Si A1 prouve que $c \subset \mathbb{S}_c$, alors $\mathbb{S}^- := \mathbb{S}^- \cup c, \mathbb{S}^+ := \mathbb{S}^+ \cup c$, aller au pas 5.
Pas 2 Si A1 prouve que $c \cap \mathbb{S}_c = \emptyset$ alors aller au pas 5.
Pas 3 Si $w(c) \leq \epsilon_r$ alors $\mathbb{S}^+ := \mathbb{S}^+ \cup c$, aller au pas 5.
Pas 4 Bisséquer c suivant son plan principal et empiler les pavés résultants dans \mathbb{S}_c .
Pas 5 Si $\mathbb{S}_c \neq \emptyset$ alors dépiler \mathbb{S}_c dans c , aller au pas 1.
Pas 6 FIN.

6.3.4. Application au problème du lobe stabilisateur

Rappelons que dans le problème du lobe stabilisateur, nous cherchons à caractériser l'ensemble \mathbb{S}_c défini par

$$\mathbb{S}_c = \{c \mid \forall p \in \mathbb{S}_p, f(c, p) > 0\} \quad (6.16)$$

avec

$$p = (z, T, \omega_0, k)^T, \quad (6.17)$$

$$\mathbf{c} = (c_1, c_2)^T, \quad (6.18)$$

$$\mathbb{S}_p = \mathbb{K}_p = [0.95, 1.05] \times [-1.05, -0.95] \times [0.95, 1.05] \times [0.95, 1.05], \quad (6.19)$$

$$\mathbf{f}(\mathbf{c}, \mathbf{p}) = \mathbf{g} \circ \mathbf{a}(\mathbf{c}, \mathbf{p}), \quad (6.20)$$

$$\mathbf{a}(\mathbf{p}, \mathbf{c}) = \begin{pmatrix} a_0(\mathbf{p}, \mathbf{c}) \\ a_1(\mathbf{p}, \mathbf{c}) \\ a_2(\mathbf{p}, \mathbf{c}) \\ a_3(\mathbf{p}, \mathbf{c}) \end{pmatrix} = \begin{pmatrix} \omega_0^2 k c_1 T^{-1} \\ \omega_0^2 (1 + c_2 k) T^{-1} \\ 2 z \omega_0 T^{-1} + \omega_0^2 (1 - 6 k T^{-1}) \\ 2 z \omega_0 + T^{-1} \end{pmatrix}, \quad (6.21)$$

$$\mathbf{g}(\mathbf{a}) = \begin{pmatrix} a_0 \\ a_3 \\ a_2 a_3 - a_1 \\ a_1 a_2 a_3 - a_1^2 - a_0 a_3^2 \end{pmatrix}. \quad (6.22)$$

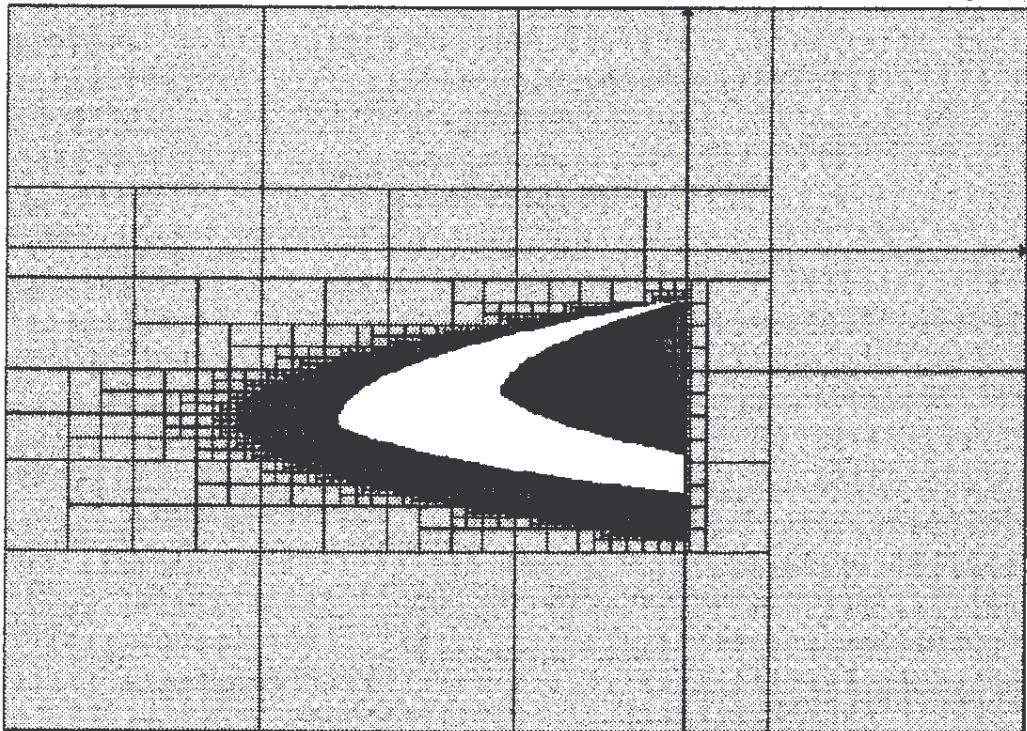


Fig. 6.7 Pavage obtenu par l'algorithme A1 pour la résolution du problème du lobe stabilisateur. Le cadre correspond au pavé de recherche $\mathbf{c}(0) = [-10, 5] \times [-10, 5]$ dans l'espace (c_1, c_2) .

Avec $\varepsilon_r = 0.01$ et $\mathfrak{c}(0) = [-10, 5] \times [-10, 5]$, A2 encadre l'ensemble solution \mathbb{S}_c entre deux sous-pavages \mathbb{S}^- et \mathbb{S}^+ dessinés sur la figure 6.7. La couche d'incertitude, représentée en blanc est très épaisse et délimite le sous pavage \mathbb{S}^- dont tous les pavés sont d'une taille relativement petite. Le temps de calcul est énorme (sur un Compac 386/33, de l'ordre de 15 heures).

6.3.5. Conclusion sur la caractérisation de l'ensemble stabilisant

Nous venons de proposer un algorithme qui caractérise l'ensemble des régulateurs, de structure paramétrique donnée, qui stabilise un ensemble de modèles paramétriques linéaires entrées-sorties et non-linéaires paramètres-sorties. L'ensemble solution est décrit par un encadrement entre deux sous-pavages, correspondant à un intercompact contenant de façon garantie l'ensemble solution. Le calcul de cet intercompact est très coûteux en temps de calcul. Notons que, dans de nombreux problèmes, la connaissance d'un seul point de cet ensemble est amplement suffisante car elle fournit une commande stabilisant à coup sûr tout l'ensemble \mathbb{K}_p et *a fortiori* \mathbb{S}_p . Pour cela, il existe des algorithmes efficaces, comme par exemple l'algorithme 3 de (Ratschek et Rokne 1988), qui se limitent à trouver un point admissible d'un ensemble dont on connaît un test d'inclusion, sans chercher à le caractériser complètement.

6.4. Conclusion

Dans ce chapitre, nous avons proposé deux méthodes pour résoudre deux problèmes ensemblistes qui ne pouvaient pas être formulés en terme d'inversion ensembliste. Le premier problème considéré est celui de la caractérisation de régions de confiance de probabilité α d'une loi de densité connue. Le second problème est celui de la caractérisation de l'ensemble des régulateurs, d'une structure paramétrique donnée, qui stabilise un ensemble de modèles paramétriques linéaires entrées-sorties où le vecteur des paramètres est inconnu mais appartient à coup sûr à un compact connu. Ces deux types de problèmes ensemblistes sont très importants en théorie des signaux et systèmes et n'avaient été résolus que dans certains cas particuliers (par exemple lorsque la loi de probabilité de la région de confiance est gaussienne ou lorsque la dépendance entre les paramètres et la sortie est linéaire). Nous avons pu, à l'aide des notions introduites dans les chapitres 3 et 4, trouver des algorithmes simples pouvant résoudre de façon garantie et

pour des classes de problèmes très générales (loi de probabilité non gaussienne et modèles non-linéaires paramètres-sorties) ces problèmes ensemblistes.

A travers ces types de problème, nous avons voulu montrer que les approches utilisées étaient relativement semblables et que l'on pouvait utiliser les notions d'analyse par intervalles et d'encadrement ensembliste pour résoudre un grand nombre de problèmes ensemblistes suggérés par les mathématiques appliquées (dans les sciences de l'ingénieur, mais aussi en physique, chimie, biologie ...).

CHAPITRE 7

OPTIMISATION GLOBALE

SOUS CONTRAINTES

7.1. Introduction

Dans les chapitres 5 et 6, nous avons donné des algorithmes permettant, pour une grande classe de problèmes ensemblistes, d'enfermer dans un intercompact l'ensemble $\mathbb{X} \subset \mathbb{R}^n$ des solutions. Le temps de calcul de ces algorithmes croît très rapidement avec le nombre n des paramètres. Pour le réduire, nous sommes souvent prêts à nous contenter d'une caractérisation partielle de \mathbb{X} . Or, le calcul d'une caractéristique de \mathbb{X} équivaut souvent à la résolution d'un ou de plusieurs problèmes d'optimisation d'un critère (linéaire ou non) sur \mathbb{X} . Par exemple, le calcul du pavé enveloppe $[\mathbb{X}]$ de \mathbb{X} revient à faire $2.n$ optimisations linéaires sur \mathbb{X} , où la fonction critère à minimiser $f_0(\mathbf{x})$ prend successivement les valeurs $-x_i, x_i$ ($i = 1, \dots, n$). Ou encore, si nous cherchons le vecteur de plus grande norme sur \mathbb{X} , il faudra maximiser le critère $x_1^2 + x_2^2 + \dots + x_n^2$.

Bien sûr, la caractéristique que l'on se propose de calculer contient beaucoup moins d'information que l'ensemble solution lui-même, mais peut suffire dans de nombreuses applications. Il faut que l'approche utilisée soit globale car les contraintes (inégalités ou test) définissant \mathbb{X} sont souvent non convexes, et un résultat fourni par un algorithme d'optimisation locale n'aurait en général aucun sens. De plus, il faut que le résultat soit garanti et que l'incertitude avec laquelle il est obtenu soit connue. En effet, dans le cas du calcul du pavé enveloppe, nous ne nous contentons pas d'avoir pour résultat un pavé qui approxime $[\mathbb{X}]$, nous voulons qu'il le contienne à coup sûr afin de préserver l'aspect garanti de l'approche ensembliste.

Intéressons nous, tout d'abord, à la minimisation globale et garantie d'un critère linéaire ou non linéaire sur l'ensemble solution \mathbb{X} , où \mathbb{X} est défini par un ensemble d'inégalités non linéaires. La résolution de problèmes d'optimisation non linéaire globale et garantie n'en est encore qu'à ses débuts (Horst et Tuy, 1990 ; Hansen, 1992). Après avoir fixé les notations utilisées, nous expliquerons l'approche développée par Falk (1973) pour

résoudre ce type de problèmes lorsque les fonctions intervenant dans le problème appartiennent à la classe des *fonctions signomiales*. Nous proposerons ensuite un algorithme du nom de GOVIA (*Global Optimizer Via Interval Analysis*) permettant de résoudre les problèmes d'optimisation linéaire sur un ensemble pour lequel nous connaissons un test d'inclusion convergent. Nous utiliserons cet algorithme pour résoudre le problème de Milanese et Vicino, où l'on cherche le pavé enveloppe d'un ensemble de vraisemblance, et comparerons les résultats obtenus avec ceux de l'approche signomiale utilisée par Milanese et Vicino (1991).

7.2. Les problèmes d'optimisation sous contraintes

Plaçons nous tout d'abord dans le cas où l'ensemble \mathbb{X} peut être défini à l'aide d'inégalités. Un problème d'optimisation sur \mathbb{X} peut être formulé comme celui de la minimisation d'un critère scalaire f_0 , où le vecteur argument \mathbf{x} de dimension n est soumis à m contraintes inégalités scalaire $\mathbf{f}(\mathbf{x}) \leq \mathbf{0}$. Le problème P peut s'écrire de façon plus concise par

$$P(f_0, \mathbf{f}) \Leftrightarrow \begin{cases} \text{minimiser } f_0(\mathbf{x}) \\ \text{sous les contraintes } \mathbf{f}(\mathbf{x}) \leq \mathbf{0} \\ \mathbf{x} \in \mathbb{R}^n. \end{cases} \quad (7.1)$$

Notons que les fonctions f_i , $i = 0, \dots, m$ peuvent être linéaires ou non en \mathbf{x} . L'ensemble \mathbb{X} des \mathbf{x} vérifiant les contraintes $\mathbf{f}(\mathbf{x}) \leq \mathbf{0}$ sera fréquemment noté $\mathbb{X}_{\mathbf{f}}$ pour rappeler la fonction qui le définit. Dans la suite, \mathbf{x} sera supposé appartenir à un ensemble borné $\mathbb{X}_0 \subset \mathbb{R}^n$, qui servira de domaine initial de recherche. Dans le but de simplifier les notations, nous supposerons, sans perte de généralité, que $\mathbb{X}_{\mathbf{f}} \subset \mathbb{X}_0$. Si $\mathbb{A} \subset \mathbb{X}_0$, le problème restreint à \mathbb{A} sera noté $P(f_0, \mathbf{f})|_{\mathbb{A}}$. Dans les problèmes d'optimisation, il est possible de distinguer les problèmes d'optimisation *centrés sur le critère* qui s'intéressent à trouver le minimum global f_0^* du critère f_0 sur l'ensemble des contraintes $\mathbb{X}_{\mathbf{f}}$, et les problèmes d'optimisation *centrés sur l'argument* qui cherchent l'ensemble \mathbb{X}^* des minimiseurs (arguments de tous les minimums) globaux de f_0 sur $\mathbb{X}_{\mathbf{f}}$.

Parfois, l'ensemble des contraintes \mathbb{X} ne peut être exprimé à l'aide d'inégalités, c'est le cas du problème des exponentielles approximantes, de celui du lobe stabilisant et de celui des conditions initiales proposés au chapitre 2. Il nous faudra alors pour résoudre le problème connaître un test d'inclusion pour \mathbb{X} . Le problème s'écrira sous la forme

$$P(f_0, \mathbf{f}) \Leftrightarrow \begin{cases} \text{minimiser } f_0(\mathbf{x}) \\ \text{sous les contraintes } \mathbf{x} \in \mathbb{X} \\ \mathbf{x} \in \mathbb{R}^n. \end{cases} \quad (7.2)$$

7.3. Approche signomiale

La programmation signomiale s'applique à une classe limitée de problèmes d'optimisation non linéaires. Falk (1973) a présenté un algorithme permettant de résoudre d'une façon garantie tous les problèmes de ce type, que Milanese et Vicino (1991) ont appliqué à l'estimation à erreurs bornées. Avant l'introduction de l'analyse par intervalles dans le domaine de l'automatique et du traitement du signal (Moore, 1992 ; Jaulin et Walter, 1993a), il s'agissait de l'unique approche numérique globale et garantie applicable à une classe de problèmes assez large (non linéaires, non gaussiens ...).

7.3.1. Définitions et notations

Définition 7.1 : $P(f_0, \mathbf{f})$ est *convexe* si les fonctions f_k ($k = 0, \dots, m$) sont convexes. \diamond

Si le problème est convexe, il est très facile de trouver le minimum global en utilisant une méthode locale. D'où l'idée, dans le cas non convexe, de partitionner le domaine de recherche de façon à approximer le problème, sur chaque sous ensemble de la partition, par ensemble de problèmes convexes. Notons

$$f^-(\mathbb{A}) = \inf_{\mathbf{a} \in \mathbb{A}} f(\mathbf{a}), \quad (7.3)$$

où f est une fonction réelle et A un ensemble quelconque. Le minimum pour le problème $P(f_0, f)$ est alors $f_0^-(X_f)$. Avec cette notation, nous avons

$$A \subset B \Rightarrow f^-(A) \geq f^-(B), \quad (7.4)$$

$$f^-(A \cup B) = \min\{f^-(A), f^-(B)\}. \quad (7.5)$$

La plupart des problèmes non-convexes rencontrés peuvent être approximés par une partition en problèmes convexes en relaxant certaines contraintes ainsi que le critère. Une approche possible pour choisir une stratégie de relaxation utilise la définition suivante.

Définition 7.2 : $P(g_0, g)|_A$ est plus optimiste que $P(f_0, f)|_A$ si $\forall k \in \{0, \dots, m\}, g_k \leq f_k$ sur A . ◇

Comme $g \leq f$ sur A , $(X_f \cap A) \subset (X_g \cap A) \stackrel{(7.4)}{\Rightarrow} f_0^-(X_f \cap A) \geq f_0^-(X_g \cap A) \geq g_0^-(X_g \cap A)$. La solution de $P(g_0, g)|_A$ sera donc nécessairement meilleur en terme de critère que celle du problème $P(f_0, f)|_A$ dont $P(g_0, g)|_A$ donne une vision plus optimiste.

7.3.2. Programmation signomiale

Un problème d'optimisation est *signomiale* s'il peut se mettre sous la forme

$$P(f'_0, f')|_{\mathbb{R}^{n+}} \Leftrightarrow \begin{cases} \text{minimiser } f'_0 = h'_0(\mathbf{x}') - g'_0(\mathbf{x}'), \\ \text{sous les contraintes } f'(\mathbf{x}') = \mathbf{h}'(\mathbf{x}') - \mathbf{g}'(\mathbf{x}') \leq \mathbf{0}, \\ \mathbf{x}' \in \mathbb{R}^{n+} = \{\mathbf{x}' \in \mathbb{R}^n \mid \mathbf{x}' > \mathbf{0}\}, \end{cases} \quad (7.6)$$

où les fonctions $h'_k(\mathbf{x}')$ et $g'_k(\mathbf{x}')$ ($k = 0, \dots, m$) sont des *posynômes*, c'est-à-dire des polynômes généralisés avec des coefficients réels positifs ou nuls et des exposants réels. Par exemple $x^{1.3} + 7x^{-0.2} + xy^{0.5}$, est un posynôme à deux indéterminées x et y . La structure relativement simple des problèmes signomiaux a incité les chercheurs à exploiter leurs nombreuses propriétés pour trouver des algorithmes efficaces de résolution. Falk (1973) a ainsi proposé une méthode de résolution générale. Cette méthode, qui sera

exposée au prochain paragraphe, n'utilise en fait que très peu les nombreuses propriétés des problèmes signomiaux.

Un posynôme f d'indéterminé \mathbf{x}' n'est pas nécessairement convexe, mais il peut être transformé en fonction convexe par le changement de variable

$$\mathbf{x} = \log_e \mathbf{x}', \quad (7.7)$$

qui doit bien sûr être compris composante par composante. Par exemple si $f(\mathbf{x}') = (x_1')^{-1} + (x_2')^{0.5} * (x_1')^3$, alors, $f(\mathbf{x}) = e^{-x_1} + e^{0.5x_2} * e^{3x_1}$ obtenue par le changement de variable (7.7) est bien convexe. Posons $f_i(\mathbf{x}) = f_i'(\exp(\mathbf{x}))$, $i = 0, \dots, m$, comme $\log_e(\mathbb{R}^{n+}) = \mathbb{R}^n$, le problème transformé est donné par

$$P(f_0, \mathbf{f}) \Leftrightarrow \begin{cases} \text{minimiser } f_0(\mathbf{x}) = h_0(\mathbf{x}) - g_0(\mathbf{x}) \\ \text{sous les contraintes } \mathbf{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x}) - \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{cases} \quad (7.8)$$

Tous les problèmes d'optimisation qui peuvent être générés à partir d'un problème d'optimisation signomiale par le changement de variable (7.7) seront qualifiés de *log-signomiaux* et peuvent être résolus par la méthode de Falk, que nous allons maintenant exposer.

7.3.3. Méthode de Falk pour résoudre les problèmes log-signomiaux

Bien que le problème log-signomial $P(f_0, \mathbf{f})$ ne soit, en général, pas convexe, l'algorithme itératif proposé par Falk (1973) permet de le résoudre grâce à une technique de type *branch-and-bound* (voir par exemple Horst et Thuy 1990). Cette technique converge à coup sûr vers la solution globale. Le principe général est de considérer successivement des sous-ensembles partitionnant le domaine de recherche initial \mathbb{X}_0 . A chaque itération, le sous-ensemble \mathbb{A} de \mathbb{X}_0 considéré est coupé en deux parties \mathbb{A}_1 et \mathbb{A}_2 (*branching*) sur lesquelles un encadrement de la valeur de la fonction objectif peut être calculée (*bounding*). Une borne inférieure pour l'optimum global f_0^* est ainsi déterminée. Après rejet de tous les sous-ensembles de \mathbb{X}_0 qui ne peuvent pas contenir f_0^* , la recherche

continue uniquement sur les sous-ensembles restant. L'algorithme de Falk repose sur deux principes.

- Le *principe de bisection* : Si (A_1, A_2) forment une partition de A , en prenant (7.5) en compte, nous avons

$$\begin{aligned} \bar{f}_0(X_f \cap A) &= \bar{f}_0((X_f \cap A_1) \cup (X_f \cap A_2)) \\ &= \min(\bar{f}_0(X_f \cap A_1), \bar{f}_0(X_f \cap A_2)). \end{aligned} \quad (7.9)$$

- Le *principe d'approximation convexe* : Soit $P(h_0 - l_0, \mathbf{h} - \mathbf{l})|_A$ le problème obtenu en remplaçant chacun des g_i , $i = 0, \dots, m$ dans $P(h_0 - g_0, \mathbf{h} - \mathbf{g})|_A$ par la plus petite fonction affine l_i telle que $l_i \geq g_i$ sur A . $P(h_0 - l_0, \mathbf{h} - \mathbf{l})|_A$ est convexe et plus optimiste que $P(h_0 - g_0, \mathbf{h} - \mathbf{g})|_A$. De plus,

$$(h_0 - g_0)^-(X_{h-\mathbf{l}} \cap A) \xrightarrow{\leq} \bar{f}_0(X_{h-\mathbf{g}} \cap A) \quad (7.10)$$

lorsque le diamètre de A tend vers zéro.

A partir du problème $P(h_0 - g_0, \mathbf{h} - \mathbf{g})|_{X_0}$, l'algorithme de Falk génère un arbre à partir du principe de bisection. Grâce au principe d'approximation convexe optimiste, chaque nœud de l'arbre est associé à un problème convexe. Plus le diamètre du sous-ensemble A considéré sera petit, plus précise sera la solution du problème transformé. La solution du problème initial est alors obtenue en revenant à la paramétrisation originale en \mathbf{x}' .

Milanese et Vicino (1991) ont utilisé l'algorithme de Falk pour trouver le pavé enveloppe de l'ensemble solution de leur problème d'estimation à erreurs bornées. Puisque le modèle dépend de quatre paramètres, il leur a fallu effectuer 8 minimisations linéaires sur l'ensemble solution. Les 8 problèmes d'optimisation correspondant ne sont pas signomiaux mais log-signomiaux. L'algorithme de Falk s'applique donc directement sans changement de variables préalable. Milanese et Vicino ont ainsi trouvé le pavé enveloppe de l'ensemble solution de leur problème, avec une précision de 3 chiffres significatifs :

$$\mathbb{S} \subset [17.2, 26.9] \times [0.3, 0.49] \times [-16.1, -5.4] \times [0.077, 0.136]. \quad (7.11)$$

Beaucoup de problèmes d'optimisation non linéaire rencontrés dans la pratique, comme par exemple celui de l'optimisation d'un critère linéaire sur l'ensemble solution d'un problème des palmes, ne sont pas ni signomiaux, ni log-signomiaux. Dans le problème des palmes, intervient, en effet, dans la définition de l'ensemble solution, un sinus qui n'est pas une fonction signomiale. L'algorithme de Falk ne peut donc pas être appliqué. L'algorithme à base d'analyse par intervalles que nous allons maintenant proposer s'applique à une classe beaucoup plus large que celle des problèmes signomiaux, celle des problèmes d'optimisation faisant intervenir des fonctions pour lesquelles nous connaissons un test d'inclusion convergent pour \mathbb{X} .

7.4. GOVIA

Cette section décrit un nouvel algorithme basé sur l'analyse par intervalles pour trouver la valeur de l'optimum global associé au problème $P(f_0, \mathbb{X})$, où f_0 est un critère (linéaire ou non) et \mathbb{X} est un ensemble pour lequel nous connaissons un test d'inclusion. Cet algorithme sera testé sur le problème d'estimation à erreurs bornées de Milanese et Vicino afin de confirmer sa validité.

Dans la suite, la recherche sera limitée au pavé $\mathbb{X}_0 = \mathbf{x}(0)$. Supposons que nous disposions d'un test d'inclusion (3.38) convergent pour \mathbb{X} et d'une méthode permettant de calculer rapidement le minimum $f_0^-(\mathbf{x})$ de f_0 sur tout pavé \mathbf{x} dans \mathbb{X}_0 , comme c'est le cas par exemple lorsque le critère est linéaire, quadratique, convexe, concave ou croissant en \mathbf{x} . D'après (7.5), le minimum de f_0 sur n'importe quel sous-pavé \mathbb{K} , est immédiatement donné par

$$f_0^-(\mathbb{K}) = \inf_{\mathbf{x} \in \mathbb{K}} f_0^-(\mathbf{x}), \quad (7.1)$$

Rappelons que la notation \mathbb{K} dénote indifféremment un pavé, comme dans le second membre de cette égalité, ou le sous-ensemble $\{\mathbb{K}\}$ de \mathbb{R}^n correspondant, comme dans le premier membre.

7.4.1. Principe

Pour faciliter la présentation de GOVIA, nous allons nous intéresser à la résolution d'un problème centré sur le critère. La méthode ainsi développée s'étend aux problèmes centrés sur l'argument. GOVIA présente de fortes analogies avec un de ses prédécesseurs (Ratschek et Rokne, 1988) qui sera lui aussi décrit. Nous allons tout d'abord énoncer les principes fondamentaux sur lesquels s'appuient ces deux algorithmes. Ils diffèrent essentiellement sur la représentation et la gestion des données, ce qui influence considérablement sur le temps de calcul.

L'idée de base consiste à générer une suite $\mathbb{K}(k)$ de sous-pavages qui s'accumule sur l'ensemble de tous les minimiseurs globaux. Cette suite est initialisée par $\mathbb{K}(0) = \mathbf{x}(0) = \mathbb{X}_0$. Supposons (hypothèse de récurrence) qu'à l'itération k , le sous-pavage $\mathbb{K}(k)$ de $\mathbf{x}(0)$ obtenu vérifie

$$\mathbb{X} \subset \mathbb{K}(k). \quad (7.13)$$

Or $\mathbb{X} \subset \mathbb{X}_0$, et donc $\mathbb{K}(0)$ satisfait l'hypothèse de récurrence. Le sous-pavage $\mathbb{K}(k+1)$ sera alors construit comme suit. Notons $\tilde{\mathbb{K}}(k)$ l'ensemble de tous les pavés \mathbf{x} de $\mathbb{K}(k)$ tels que

$f_0(\mathbf{x}) = f_0(\mathbb{K}(k))$. Prenons un pavé quelconque \mathbf{x} de $\tilde{\mathbb{K}}(k)$. D'après (7.13) et (7.4) $f_0(\mathbf{x}) = f_0(\mathbb{K}(k)) \leq f_0(\mathbb{X}_f)$. En notant \mathfrak{t} le test d'inclusion (3.38) pour l'ensemble des contraintes, trois cas doivent être considérés.

- (i) Si $\mathfrak{t}(\mathbf{x}) = \text{vrai}$, alors \mathbf{x} est acceptable $f_0(\mathbf{x}) = f_0(\mathbb{K}(k)) = f_0(\mathbb{X}_f)$. Le problème est donc résolu.
- (ii) Si $\mathfrak{t}(\mathbf{x}) = \text{faux}$ alors \mathbf{x} est inacceptable et peut être enlevé. Nous poserons $\mathbb{K}(k+1) := \mathbb{K}(k) - \mathbf{x}$.
- (iii) Si $\mathfrak{t}(\mathbf{x})$ est indéterminé, nous couperons le pavé \mathbf{x} en deux pavés plus petits de façon à permettre une analyse plus fine. Ceci sera fait en posant $\mathbb{K}(k+1) := \mathbb{K}(k) - \mathbf{x} + \mathbf{x}_1 + \mathbf{x}_2$, où \mathbf{x}_1 et \mathbf{x}_2 résultent de la bisection de \mathbf{x} suivant un plan principal.

Si la condition (i) est satisfaite pour un certain k , une solution exacte est trouvée en un temps fini, mais cette situation n'apparaît presque jamais. Dans le cas contraire, nous disposons d'un sous-pavage $\mathbb{K}(k+1)$ qui vérifie les hypothèses de récurrence. La figure 7.1 illustre une situation à deux dimensions, avec deux minimums globaux et un critère linéaire.

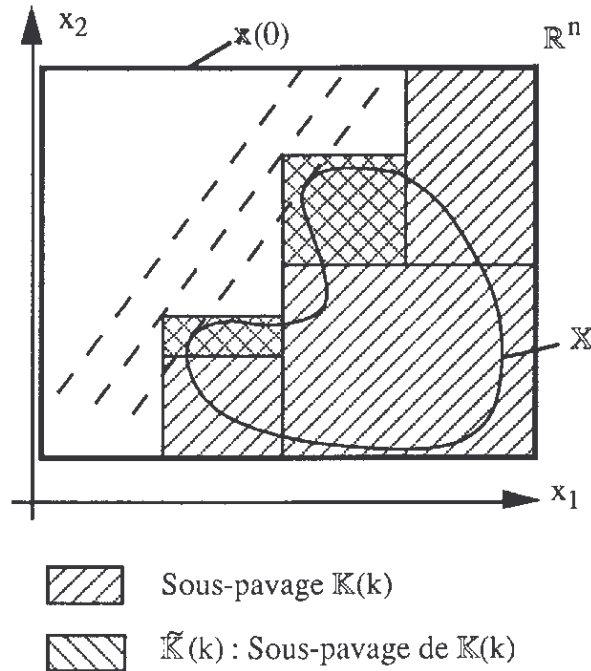


Fig. 7.1 Minimisation d'un critère linéaire sur X_f .
Les isocritères sont repérés par les lignes en pointillés.

Lorsque $X = X_f$ est défini des inégalités, les tests $t(\mathbf{x}) = \text{vrai}$ et $t(\mathbf{x}) = \text{faux}$ sont respectivement remplacé par les tests $f(\mathbf{x}) \subset (\mathbb{R}^-)^m$ et $f(\mathbf{x}) \cap (\mathbb{R}^-)^m = \emptyset$, où \mathbb{R}^- désigne l'ensemble des réels strictement négatifs.

Le théorème suivant, qui repose sur les condition de continuité de la fonction réelle $f_0 \circ f^{-1}$, donne des conditions nécessaires de convergence pour GOVIA. A tout sous-ensemble A de \mathbb{R}^m , la fonction réelle $f_0 \circ f^{-1}$ sera (h_∞, L_∞) -continue autour de A si et seulement si $L_\infty(f_0 \circ f^{-1}(A), f_0 \circ f^{-1}(B)) \rightarrow 0$ lorsque $h_\infty(A, B) \rightarrow 0$. Dire que $f_0 \circ f^{-1}$ est (h_∞, L_∞) -continue autour de $(\mathbb{R}^-)^m$ signifie que si dans (7.1) les contraintes $f(\mathbf{x}) \leq \mathbf{0}$ sont remplacées par $f(\mathbf{x}) \leq \varepsilon \cdot \mathbf{v}$, avec ε infiniment proche de zéro, et \mathbf{v} étant un vecteur arbitrairement choisi, alors la modification de $f_0(X_f)$ sera infiniment petite. Dans ce cas et dans la situation générique où la convergence n'est pas finie, le théorème s'applique :

Théorème 7.1: Si $f_0 \circ f^{-1}$ est (h_∞, L_∞) -continue autour de l'ensemble $(\mathbb{R}^-)^m$, GOVIA convergera vers le minimum global du problème

$$f_0(\mathbb{K}(k)) \xrightarrow{\leq} f_0^* . \quad \diamond$$

Démonstration : Quelle que soit l'itération k , un pavé est soit éliminé du sous-pavage $\mathbb{K}(k)$, soit coupé en deux suivant son axe principal. Ainsi $\mathbb{K}(\infty)$ a nécessairement un ensemble d'accumulation non-vidé (sauf dans le cas atypique où la convergence est finie). Comme toutes les bisections sont faites dans $\mathbb{K}(k)$, cet ensemble d'accumulation est le même que celui de $\mathbb{K}(\infty)$. Ainsi, pour tout ε , qui peut être choisi aussi petit que voulu, il existe une itération k_ε telle que $\mathbb{K}(k_\varepsilon)$ possède un pavé \mathbf{x} indéterminé avec une taille inférieure à ε . Faisons maintenant tendre ε vers zéro. Comme f est une fonction d'inclusion convergente et \mathbf{x} est indéterminé, les deux conditions suivantes sont satisfaites.

$$\begin{aligned} w(f(\mathbf{x})) &\rightarrow 0, \\ f(\mathbf{x}) \cap (\mathbb{R}^-)^m &\neq \emptyset. \end{aligned} \quad (7.14)$$

D'après (4.10), $h_\infty^0(f(\mathbf{x}), (\mathbb{R}^-)^m) = h_\infty(f(\mathbf{x} \cup (\mathbb{R}^-)^m), (\mathbb{R}^-)^m) \rightarrow 0$. Comme $f_0 \circ f^{-1}$ est (h_∞, L_∞) -continue autour de $(\mathbb{R}^-)^m$, $L_\infty(f_0 \circ f^{-1}\{f(\mathbf{x}) \cup (\mathbb{R}^-)^m\}, f_0 \circ f^{-1}((\mathbb{R}^-)^m)) = |\min\{f_0 \circ f^{-1} \circ f(\mathbf{x}), f_0 \circ f^{-1}((\mathbb{R}^-)^m)\} - f_0 \circ f^{-1}((\mathbb{R}^-)^m)| \rightarrow 0$. Donc

$$|\min\{f_0 \circ f^{-1} \circ f(\mathbf{x}), f_0(\mathbb{X})\} - f_0(\mathbb{X})| \rightarrow 0. \quad (7.15)$$

Or $\mathbf{x} \subset f^{-1} \circ f(\mathbf{x}) \subset f^{-1} \circ f(\mathbf{x})$. Nous avons donc d'après (7.4), $f_0 \circ f^{-1} \circ f(\mathbf{x}) \leq f_0(\mathbf{x})$. De plus $f_0(\mathbf{x}) = f_0(\mathbb{K}(k_\varepsilon)) \leq f_0(\mathbb{X})$. Donc, $f_0 \circ f^{-1} \circ f(\mathbf{x}) \leq f_0(\mathbf{x}) = f_0(\mathbb{K}(k_\varepsilon)) \leq f_0(\mathbb{X})$. D'après (7.15) nous obtenons $|f_0(\mathbb{K}(k_\varepsilon)) - f_0(\mathbb{X})| \rightarrow 0$. \diamond

Une fois l'algorithme exécuté, nous avons en mémoire le sous-pavage $\mathbb{K}(k_f(\varepsilon_r))$ qui s'accumule sur \mathbb{X}^* lorsque ε_r tend vers zéro sous certaines conditions de continuité. Il semblerait que si $f_0 \circ f^{-1}$ est (h_∞, L_∞) -continue autour de $(\mathbb{R}^-)^m$, et si f_0^{-1} est (L_∞, h_∞) -continue autour de f_0^* le sous-pavage $\mathbb{K}(k)$ s'accumule sur \mathbb{X}^* lorsque ε_r tend vers zéro. Mais cette proposition, qui permettrait de résoudre le problème d'optimisation centré sur l'argument, n'est pour le moment qu'une conjecture. Enfin, la méthode qui vient d'être

exposée, nécessite la gestion d'un très grand nombre de pavés. Il convient donc de développer un algorithme qui prenne en compte ce problème et utilise la représentation en mémoire adéquate. Une mauvaise organisation des données en mémoire peut, en effet, avoir des conséquences dramatiques sur les performances de l'algorithme. Nous allons proposer maintenant deux algorithmes utilisant la méthode que nous venons d'exposer. Le premier (Ratschek et Rokne, 1988) utilise une structure de liste pour représenter le sous-pavage en mémoire alors que le deuxième utilise une structure d'arbre binaire.

7.4.2. Représentation des sous-pavages par des listes

La méthode présentée ci-dessus apparaît dans la littérature sous une forme différente dans le cas plus général où $f_0(\mathbf{x})$ n'est pas calculable directement, mais où une fonction d'inclusion est connue pour f_0 (Ratschek et Rokne, 1988). La représentation choisie pour le sous-pavage est une liste composée de pavés qui doivent être triés par ordre croissant en fonction de $f_0(\mathbf{x})$. L'algorithme généré est très gourmand en temps de calcul car chaque fois qu'une bisection est effectuée deux pavés sont rajoutés dans la liste qui doit alors être retriée. Il est bien connu qu'une représentation par arbres peut apporter une gestion plus efficace d'un ensemble ordonné.

7.4.3. Représentation des sous-pavages par des arbres

Tous les algorithmes d'optimisation classique, fondés sur l'analyse par intervalles, utilisent une représentation linéaire (pile, file, ...) du sous-pavage courant (Ratschek et Rokne, 1988). Il s'ensuit une perte des connaissances topologiques du sous-pavage parce que deux pavés qui sont proches dans l'espace des paramètres ne le sont pas nécessairement dans la représentation (et *vice versa*), d'où une influence considérable sur le temps de calcul. La recherche d'une solution globale du problème (7.1) requiert à la fois une connaissance globale de l'état courant de la recherche et des connaissances locales. Une représentation par arbre binaire semble adaptée à cette gestion de connaissances multi-échelles. Rappelons qu'un *arbre binaire* est un ensemble de *nœuds* (ici, chacun de ces nœuds sera associé à un sous-pavage) liés par des relations familiales.

Chaque nœud a un père, sauf l'un d'entre eux appelé *racine*. Chaque nœud possède soit deux fils soit zéro. Un nœud sans fils est appelé *feuille* (pour plus d'informations, voir par exemple (Knuth, 1977, pp. 305-346)).

Tout sous-pavage \mathbb{K} qui n'est pas un pavé peut être partitionné en deux sous-pavages \mathbb{K}_1 et \mathbb{K}_2 . Cette représentation récursive d'un sous-pavage génère un arbre binaire où \mathbb{K}_1 et \mathbb{K}_2 , fils de \mathbb{K} , peuvent admettre d'autres fils. Dans l'algorithme, que nous allons développer, toute l'information sera mémorisée par un seul arbre binaire. La racine de l'arbre sera associée au sous-pavage $\mathbb{K}(k)$. Chaque feuille correspondra à un pavé de $\mathbb{K}(k)$ et pourra générer une paire de fils par bisection. Cette opération sur l'arbre est appelée *éclosion*.

Dans la mémoire de l'ordinateur, tout sous-pavage \mathbb{K} est représenté par un pointeur vers un nœud qui contient les informations sur \mathbb{K} qui doivent être mémorisées, c'est-à-dire

- les sous-pavages parents (père, frère, fils), représentés par des pointeurs,
- la valeur minimale du critère f_0 sur \mathbb{K} , rangé dans le champ $\tilde{f}_0(\mathbb{K})$.

Lorsque le sous-pavage \mathbb{K} est réduit à un pavé (ce qui signifie que \mathbb{K} est représenté par une feuille), l'enregistrement contient un pointeur vers une zone mémoire contenant les coordonnées de ce pavé

- $\mathbf{x} = [x_1^-, x_1^+] \times [x_2^-, x_2^+] \times \dots \times [x_n^-, x_n^+]$.

Exemple 7.1: Si $f_0(x_1, x_2) = x_1 + 2x_2$, et $\mathbb{K}(k) = \{[1, 2] \times [1, 2], [2, 3] \times [0, 2], [1, 4] \times [3, 4]\}$ (figure 7.2a), alors $\mathbb{K}(k)$ peut être représenté en mémoire comme sur la figure 7.2b. Le nombre indiqué dans chaque nœud est la valeur minimum du critère sur le sous-pavage correspondant. Dans cet exemple, nous avons bien évidemment $\hat{\mathbb{K}} = \{[2, 3] \times [0, 2]\}$.

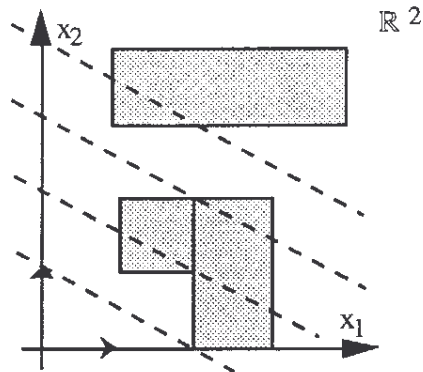


Fig. 7.2a Sous-pavage $\mathbb{K}(k)$ de l'exemple 7.1. Les isocritères sont indiquées par les hachures en pointillés.

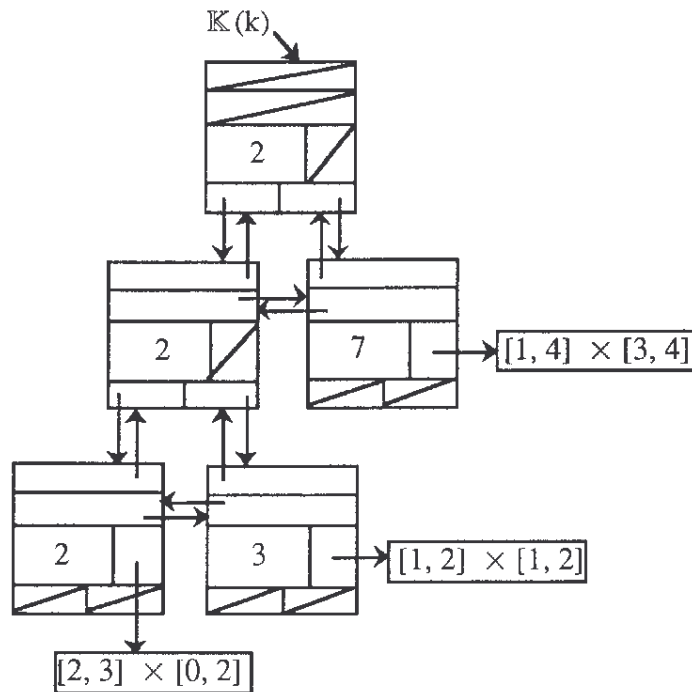


Fig. 7.2b Représentation en mémoire du sous-pavage $\mathbb{K}(k)$ de l'exemple 7.1.

Supposons maintenant f telle que le pavé $[2, 3] \times [0, 2]$ soit classé comme inacceptable. Alors le nouvel arbre $\mathbb{K}(k+1)$ obtenu est représenté sur la figure 7.2c. Bien que $\tilde{f}_0(\mathbb{K}(k+1))$ ait changé, aucune évaluation de \tilde{f}_0 n'a dû être effectuée. Le calcul de \tilde{f}_0 pour un pavé ne sera nécessaire que lorsqu'un nouveau pavé sera créé à la suite d'une bisection.

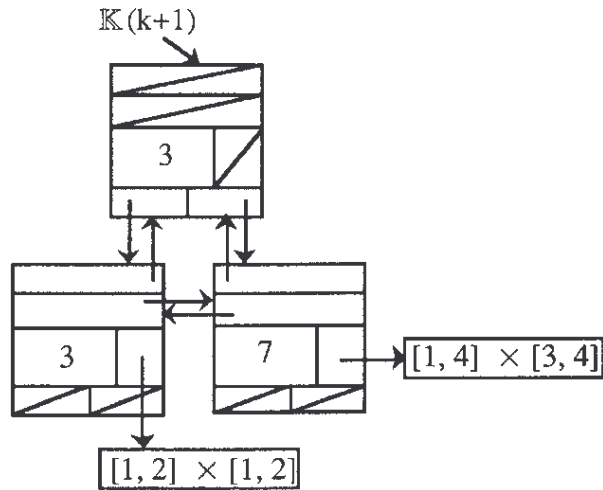


Fig. 7.2c Représentation en mémoire du sous-pavage $\mathbb{K}(k+1)$ obtenu après rejet du pavé inacceptable.

7.4.4. GOVIA

Nous allons maintenant utiliser la représentation des sous-pavages par arbres binaires pour générer un algorithme implémentable sur machine à l'aide d'un langage permettant l'allocation dynamique de mémoire tel que Pascal ou C. Dans ce qui suit, ζ est un pointeur représentant un sous-pavage \mathbb{K}^ζ de $\mathbb{K}(k)$. Le nombre \hat{f}_0^- est une borne inférieure déjà calculée par l'algorithme pour la solution du problème d'optimisation $f_0(\mathbb{X})$, qui va converger vers $f_0(\mathbb{X})$, d'après le théorème 1.

Entrées du programme

Test d'inclusion pour \mathbb{X}	:	t.
Pavé initial de recherche	:	$\mathbf{x}(0)$.
Précision demandée	:	ϵ_r .
Critère à minimiser	:	f_0 .

Initialisation

$\mathbb{K}^\zeta := \mathbf{x}(0)$. (ce sous-pavage est constitué d'un seul pavé. Il s'agit de la racine.)
 $\hat{f}_0^- = \tilde{f}_0^-(\mathbf{x}(0)) := f_0^-(\mathbf{x}(0))$.

Itération (à ce niveau, \mathbb{K}^ζ est une feuille c'est-à-dire un pavé)

Pas 1 Si $t(\mathbf{x}) = \text{vrai}$ alors aller au pas 4. (le pavé \mathbb{K}^ζ est acceptable)

Pas 2 Si $t(\mathbf{x}) = \text{faux}$ alors (le pavé \mathbb{K}^ζ est inacceptable)

Si le pavé \mathbb{K}^ζ n'a pas de père (pas de point acceptable). FIN.

Échanger le frère de \mathbb{K}^ζ avec son père, pointer ζ sur le nouveau père maintenant appelé \mathbb{K}^ζ . Éliminer les deux fils de \mathbb{K}^ζ .

Répéter

pointer ζ sur le père de \mathbb{K}^ζ .

$\tilde{f}_0^-(\mathbb{K}^\zeta) := \min(\tilde{f}_0^-(\text{fils}_1(\mathbb{K}^\zeta)), \tilde{f}_0^-(\text{fils}_2(\mathbb{K}^\zeta)))$.

Si ζ pointe vers la racine, alors $\hat{f}_0^- := \tilde{f}_0^-(\mathbb{K}^\zeta)$.

jusqu'à ce que $\hat{f}_0^- = \tilde{f}_0^-(\mathbb{K}^\zeta)$.

Pointer ζ sur la feuille descendante de \mathbb{K}^ζ qui minimise \tilde{f}_0^- .

Aller au pas 1.

Pas 3 Sinon, (le pavé \mathbb{K}^ζ est indéterminé)

Si $w(\mathbb{K}^\zeta) \leq \epsilon_r$ alors aller au pas 4.

Bisséquer le pavé \mathbb{K}^ζ suivant son axe principal.

Attacher les deux pavés résultants comme fils de \mathbb{K}^ζ .

$\tilde{f}_0^-(\text{fils}_1(\mathbb{K}^\zeta)) := f_0^-(\text{fils}_1(\mathbb{K}^\zeta))$.

$\tilde{f}_0^-(\text{fils}_2(\mathbb{K}^\zeta)) := f_0^-(\text{fils}_2(\mathbb{K}^\zeta))$.

Pointer ζ sur le fils qui minimise \tilde{f}_0^- .

Aller au pas 1.

Pas 4 Une borne inférieure pour $f_0^-(\mathbb{X})$ est \hat{f}_0^- .

Remarque 7.1 : Quand le test t revient à tester m inégalités $f(\mathbf{x}) \leq \mathbf{0}$, tester si $f(\mathbf{x}) \subset (\mathbb{R}^-)^m$ revient à tester m inclusions d'intervalles. Définissons le vecteur masque associé à \mathbb{K}^ζ comme un vecteur booléen de dimension m avec sa $i^{\text{ème}}$ composante vraie si et seulement si la $i^{\text{ème}}$ inclusion est satisfaite. Si un pavé \mathbf{x} satisfait certaines de ces inclusions, tous

ses descendants générés par bisections successives satisferont aussi à ces inclusions. Le test peut donc être limité à l'espace non masqué qui est le sous-espace de \mathbb{R}^m engendré par les vecteurs de la base canonique associés aux composantes fausses du vecteur masque. Cela réduit considérablement le volume des calculs à effectuer lorsque la taille des pavés devient petite. \diamond

Remarque 7.2 : A l'optimum global, les conditions de stationnarité ou de Kuhn-Tucker s'écrivent

$$\begin{aligned} df_0(\mathbf{x}) + \sum_{i=1}^m u_i df_i(\mathbf{x}) &= \mathbf{0}, \\ u_i f_i(\mathbf{x}) &= 0, \\ u_i &\geq 0. \end{aligned} \tag{7.16}$$

Une méthode de Newton par intervalles peut donc s'insérer dans GOVIA afin de réduire, à l'aide de (7.16), la taille du pavé \mathbf{x} considéré, ce qui accélèrerait considérablement la convergence de l'algorithme. \diamond

Remarque 7.3 : GOVIA génère une borne inférieure \hat{f}_0 pour $f_0(\mathbb{X})$. Si jamais le centre $\bar{\mathbf{x}}$ du pavé courant \mathbf{x} appartient à \mathbb{X} , une borne supérieure $f_0(\bar{\mathbf{x}})$ pour $f_0(\mathbb{X})$ peut alors être donnée. Il est ainsi très facile de générer une suite de bornes supérieures pour $f_0(\mathbb{X})$ qui tend vers $f_0(\mathbb{X})$ lorsque ε_r tend vers zéro. La connaissance de bornes inférieure et supérieure pour la solution nous donne une estimée garantie de l'erreur commise lors de l'estimation de $f_0(\mathbb{X})$, et peut être utilisée comme critère d'arrêt. En pratique, du fait que ε_r ne tend pas vers zéro, il arrive que le volume de \mathbb{X} soit tellement petit qu'aucun de tous les pavés parcourus n'ait son centre dans \mathbb{X} , et donc qu'aucune borne supérieure pour $f_0(\mathbb{X})$ ne puisse être donnée. Cela ne signifie pas nécessairement que le pavé enveloppe $[\mathbb{X}]$ soit petit, mais peut correspondre à des situations où \mathbb{X} est à la limite d'avoir une dimension inférieure à n . \diamond

7.5. GOVIA pour l'optimisation sur l'ensemble solution

Si l'ensemble solution \mathbb{X} est défini par $\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$, où \mathbb{Y} est défini par un ensemble d'inégalités $\mathbf{g}(\mathbf{y}) \leq \mathbf{0}$, alors, comme nous l'avons fait remarquer dans le chapitre 2, \mathbb{X} peut aussi être redéfini par les inégalités $\mathbf{g} \circ \mathbf{f}(\mathbf{x}) \leq \mathbf{0}$. Comme mentionné dans l'introduction de ce chapitre, le calcul du pavé enveloppe $[\mathbb{X}]$ peut être fait en résolvant $2.n$ problèmes d'optimisation d'un critère linéaire sous contraintes inégalité non linéaires. Chacun de ces problèmes d'optimisation correspond à l'optimisation d'une composante x_i de \mathbf{x} sujette à toutes les contraintes définissant l'ensemble solution \mathbb{X} . La fonction critère à minimiser prendra respectivement les valeurs $-x_i, x_i$ ($i = 1, \dots, n$). L'algorithme GOVIA s'applique donc immédiatement, si nous connaissons une fonction d'inclusion pour $\mathbf{g} \circ \mathbf{f}$, mais il s'applique aussi lorsque seul un test d'inclusion convergent est connu pour l'ensemble \mathbb{X} .

Problème de Milanese et Vicino

Dans le problème de Milanese et Vicino (1991), le nombre de paramètres (égal à 4) et la surface de la frontière $\partial\mathbb{S}$ de l'ensemble solution \mathbb{S} sont trop grand pour avoir, en utilisant SIVIA, une description fine de \mathbb{S} en un temps raisonnable. Cherchons maintenant à utiliser GOVIA pour le calcul du pavé enveloppe $[\mathbb{S}]$ de l'ensemble solution \mathbb{S} . La table 7.1 illustre les correspondances entre les notations utilisées dans GOVIA, décrit ci-dessus, et celles utilisées pour le problème d'estimation à erreurs bornées de Milanese et Vicino.

\mathbf{x}	\mathbf{p}
$f_0(\mathbf{x})$	p_j si $j \leq n_p, -p_j$ si $j > n_p$
$\mathbf{f}(\mathbf{x}) \leq \mathbf{0}$	$\mathbf{e}(\mathbf{p}) \in \mathbb{E}$
\mathbb{X}_f	\mathbb{S}
\mathbb{X}_0	\mathbb{P}_0

Table 7.1 Correspondances entre les notations de GOVIA et celles utilisées par l'estimation à erreurs bornées

Une approximation pour $[\mathcal{S}]$ est obtenue après avoir appliqué GOVIA 8 fois avec pour le critère $f_0(\mathbf{p})$ les valeurs respectives $p_1, -p_1, p_2, -p_2, p_3, -p_3, p_4, -p_4$. Nous avons choisi les coefficients de pondération pour les paramètres de telle façon que le pavé des paramètres acceptables *a priori* \mathbb{P}_0 devienne un cube de côté 1. Cela change uniquement la stratégie de bisection utilisée lors du pas 3 de l'algorithme. Les résultats suivants, obtenus pour différentes précisions sur un Compaq 386/33, donnent une idée de l'efficacité de l'algorithme. Dans le vecteur \mathbf{k}_f et \mathbf{n}_{\max} sont rangés respectivement le nombre d'itérations et le nombre maximum de pavés dans l'arbre pour chacune des huit optimisations faites par GOVIA.

Pour $\varepsilon_r = 0.5$,

$$\begin{aligned}\mathbf{k}_f &= (116, 39, 166, 80, 147, 213, 24, 438), \\ \mathbf{n}_{\max} &= (45, 27, 143, 37, 77, 73, 27, 131), \\ \mathcal{S} &\subset [9.3, 60] \times [0.016, 1] \times [-30, -3.7] \times [0, 0.17], \\ &\text{temps de calcul : inférieur à 33 secondes.}\end{aligned}$$

Pour $\varepsilon_r = 0.05$,

$$\begin{aligned}\mathbf{k}_f &= (3\ 221, 18\ 654, 21\ 281, 3\ 282, 17\ 442, 1\ 642, 3\ 454, 15\ 453), \\ \mathbf{n}_{\max} &= (781, 647, 1\ 055, 705, 385, 209, 755, 1\ 193), \\ \mathcal{S} &\subset [17, 28] \times [0.29, 0.50] \times [-17, -5.3] \times [0.075, 0.14], \\ &\text{temps de calcul : inférieur à 37 minutes.}\end{aligned}$$

Pour $\varepsilon_r = 0.01$,

$$\begin{aligned}\mathbf{k}_f &= (5\ 350, 29\ 577, 32\ 850, 4\ 160, 26\ 711, 3\ 226, 4\ 888, 24\ 444), \\ \mathbf{n}_{\max} &= (1\ 043, 1\ 045, 1\ 513, 749, 437, 481, 1\ 163, 2\ 001), \\ \mathcal{S} &\subset [17.13, 27.12] \times [0.298, 0.495] \times [-16.35, -5.34] \times [0.076, 0.137], \\ &\text{temps de calcul : inférieur à 87 minutes.}\end{aligned}$$

A chaque itération, GOVIA effectue un déplacement moyen dans l'arbre de 8 nœuds. Ce déplacement moyen semble ne pas croître avec la taille de l'arbre. La gestion de l'arbre reste donc extrêmement simple même lorsque l'arbre devient gigantesque. Rappelons enfin que Milanese et Vicino (1991) obtiennent un meilleur résultat en seulement 8 minutes sur un ordinateur équivalent, puisqu'ils obtiennent la certitude que

$$\mathcal{S} \subset [17.2, 26.9] \times [0.3, 0.49] \times [-16.1, -5.4] \times [0.077, 0.136]. \quad (7.17)$$

Incontestablement, l'algorithme de Falk, conçu pour résoudre les problèmes signomiaux, est plus adapté pour résoudre le problème posé (bien qu'il ne s'applique qu'à une classe de modèles plus restreinte). L'efficacité de GOVIA pourrait être considérablement améliorée par insertion de procédures locales garanties comme la méthode de Newton par intervalles.

Problème des exponentielles approximantes

Le calcul de $[\mathcal{S}]$ requiert 4 optimisations effectuées dans le pavé de recherche $\mathfrak{p}(0) = [0, 5] \times [0, 5]$. Pour une précision de $\varepsilon_r = 0.01$, en 4 secondes sur un Compaq 386/33, GOVIA démontre numériquement que \mathcal{S} satisfait

$$\mathcal{S} \subset [0.302, 2.002] \times [0.399, 2.813]. \quad (7.18)$$

Lors des quatre minimisations effectuées, GOVIA génère quatre pavages dont la superposition est représentée sur la figure 7.3.

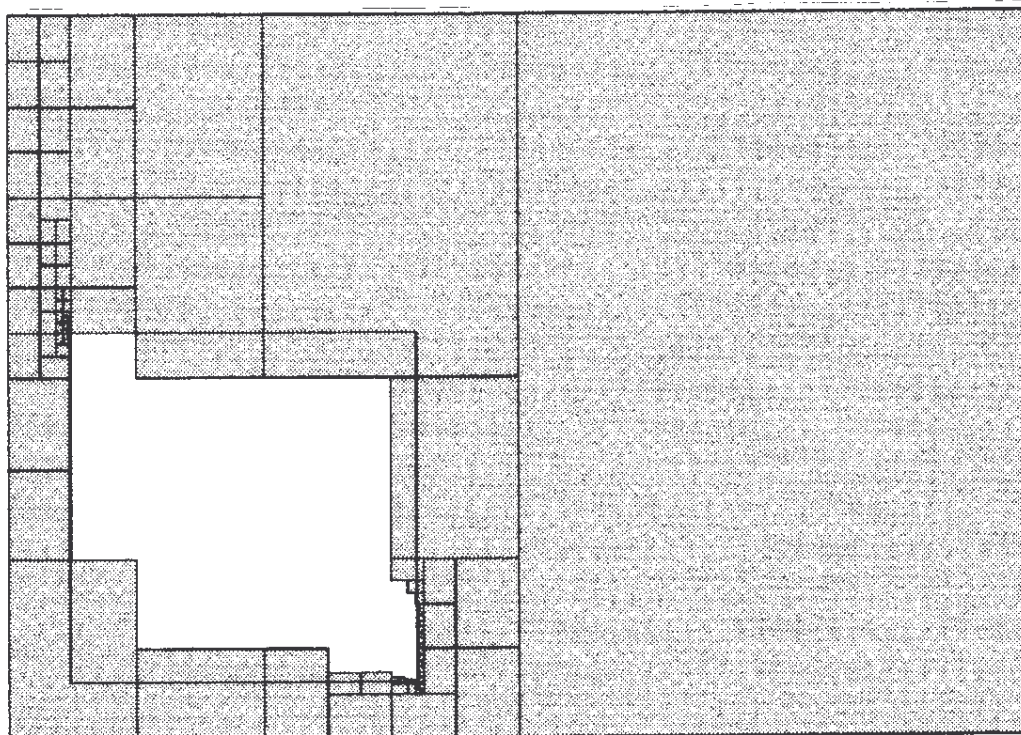


Fig. 7.3 Superposition des quatre pavages générés par GOVIA pour le calcul du pavé enveloppe de l'ensemble des exponentielles approximantes.

Les pavés en gris clairs sont prouvés inacceptables et ont été éliminés de l'arbre. La zone blanche contient \mathbb{S} . Elle est formée de tous les pavés qui restent dans l'arbre après exécution de l'algorithme. Pour une caractérisation complète de \mathbb{S} , SIVIA demandait, pour une précision et un domaine de recherche identiques, un temps de calcul de 33 secondes.

Comme l'ensemble solution \mathbb{S} ne peut pas être exprimé par des inégalités, l'algorithme de Hansen (1992) ainsi que celui de Falk (1973) ne peuvent traiter un tel problème.

7.6. Conclusion

Dans ce chapitre, nous avons proposé un algorithme déterministe GOVIA permettant de résoudre les problèmes d'optimisation linéaire ou non linéaire de façon garantie sur un ensemble pour lequel nous connaissons un test d'inclusion convergeant. GOVIA utilise l'analyse par intervalles et se sert d'une structure d'arbre binaire pour une gestion aisée de l'ensemble des pavés du sous-pavage courant. Pour valider la méthode, nous avons traité le problème signomial de Milanese et Vicino (1991) et le problème du calcul du pavé enveloppe de l'ensemble des exponentielles approximantes sur lequel nous connaissons un test d'inclusion. Pour leur problème, Milanese et Vicino ont obtenu des résultats plus précis avec l'algorithme de Falk, exposée en début de chapitre, spécialement conçu pour la résolution des problèmes signomiaux de façon garantie. Bien que notre méthode semble moins efficace, elle permet de traiter une classe beaucoup plus large de problèmes d'optimisation non linéaire. Pour une utilisation dans le cas particulier des problèmes d'optimisation à contraintes inégalité, il conviendrait de fusionner GOVIA avec l'algorithme de Hansen (1992) de façon à profiter de procédures efficaces de bisection, d'élimination, de procédures locales garanties comme la méthode de Newton par intervalles ... tout en conservant la gestion efficace des pavés en mémoire qu'autorise la structure arborescente de GOVIA.

CHAPITRE 8

OPTIMISATION GLOBALE SANS CONTRAINTE PAR DÉCOMPOSITION FINIE

8.1. Introduction

Dans ce chapitre, nous allons proposer une nouvelle méthode de minimisation globale et garantie d'un critère f sans contraintes. Cette méthode combine la vision globale qu'autorise l'analyse par intervalles avec les techniques puissantes de résolution des problèmes d'optimisation convexe. L'algorithme qui sera proposé partitionnera l'espace de recherche en un sous-pavage dont chacun des pavé \mathbf{x} satisfera une des deux conditions suivantes.

- Le critère est *convexe* sur \mathbf{x} . Il ne possède donc pas plus d'un minimiseur local à l'intérieur de \mathbf{x} . La combinaison du critère de Routh, largement utilisé pour l'étude de la stabilité des systèmes linéaires (voir par exemple DiStefano, Stubberud et William (1990) ; Faedo (1953) ; Leipholz (1987) ; Walter et Jaulin (1994)), et de l'analyse par intervalles nous donnera un test de convexité. Ce test permettra de démontrer numériquement que le critère est convexe sur \mathbf{x} afin d'appliquer en toute garantie une technique d'optimisation convexe.
- Le pavé \mathbf{x} est *inacceptable*, c'est-à-dire qu'il ne peut pas contenir un minimiseur global. C'est le cas par exemple lorsque le gradient ne peut s'annuler sur \mathbf{x} , lorsque que l'on connaît un vecteur dont l'image par f minore l'ensemble $f(\mathbf{x})$, ou lorsque que le critère n'est nulle part convexe sur \mathbf{x} .

A notre connaissance, aucun test d'inclusion pour la convexité n'existait jusqu'à présent dans la littérature. Un tel test constitue pourtant le lien fondamental entre les algorithmes

d'optimisation globale d'analyse par intervalles et les techniques convexes. Dans le but de faciliter la compréhension, nous allons implanter ce test d'inclusion dans un algorithme élémentaire, mais il conviendrait, pour plus d'efficacité, de l'utiliser dans un algorithme du type Hansen (1992).

8.2. Tests d'inclusion pour la convexité et l'inacceptabilité

Dans ce paragraphe, nous allons chercher des méthodes numériques capables de démontrer que le critère à minimiser est convexe sur un pavé donné \mathbf{x} ou que \mathbf{x} est inacceptable. Pour cela, supposons que le critère à minimiser $f : \mathbb{R}^n \rightarrow \mathbb{R}, \mathbf{x} \rightarrow f(\mathbf{x})$ soit deux fois différentiable et qu'il admette un nombre fini de minimiseurs. Si \mathbf{x}^* est un minimiseur, nous avons nécessairement

$$\mathbf{g}(\mathbf{x}^*) = \mathbf{0} \text{ et } \mathbf{H}(\mathbf{x}^*) \geq \mathbf{0} \quad (8.1)$$

où \mathbf{g} et \mathbf{H} sont respectivement le gradient et le Hessien de f et où $\mathbf{H}(\mathbf{x}) \geq \mathbf{0}$ signifie que la matrice est semie définie positive, c'est-à-dire que toutes ses valeurs propres sont positives ou nulles. De même, $\mathbf{H}(\mathbf{x}) > \mathbf{0}$ signifie que $\mathbf{H}(\mathbf{x})$ a toutes ses valeurs propres strictement positives et $\mathbf{H}(\mathbf{x}) \not\geq \mathbf{0}$, que l'une d'entre elle est strictement négative. Les tests d'inclusion qui suivent peuvent être utilisés pour analyser la convexité ou l'inacceptabilité de \mathbf{x} .

$$\begin{aligned} \mathbb{T}_1(\mathbf{x}) &\Leftrightarrow \mathbf{0} \notin \mathbf{g}(\mathbf{x}), \\ \mathbb{T}_2(\mathbf{x}) &\Leftrightarrow \mathbf{H}(\mathbf{x}) \not\geq \mathbf{0}, \\ \mathbb{T}_3(\mathbf{x}) &\Leftrightarrow \mathbf{H}(\mathbf{x}) > \mathbf{0}. \end{aligned} \quad (8.2)$$

Lorsque \mathbb{T}_1 et \mathbb{T}_2 prennent la valeur "vrai", f ne peut admettre un minimum local ou global sur \mathbf{x} . Si \mathbb{T}_3 est vrai, f est convexe sur \mathbf{x} et admet au plus un minimiseur sur \mathbf{x} et une technique de minimisation convexe fournit donc un résultat garanti. Notons que les tests \mathbb{T}_i ($i = 1, 2, 3$), qui ne sont jamais indéterminés, sont très difficiles à évaluer. C'est pourquoi nous allons les remplacer par d'autres tests d'inclusion convergeants \mathfrak{t}_i faciles à évaluer mais plus faibles dans le sens où

$$\begin{aligned} \mathfrak{t}_i(\mathbf{x}) = \text{vrai} &\Rightarrow \mathbb{T}_i(\mathbf{x}) = \text{vrai}, \\ \mathfrak{t}_i(\mathbf{x}) = \text{faux} &\Rightarrow \mathbb{T}_i(\mathbf{x}) = \text{faux}. \end{aligned} \quad (8.3)$$

Cette fois, $t_1(\mathbf{x})$ peut être indéterminé. Soit g , une fonction d'inclusion pour g . Un test t_1 plus faible que T_1 est donné par

$$t_1(\mathbf{x}) \Leftrightarrow \mathbf{0} \notin g(\mathbf{x}). \quad (8.4)$$

Soit $P_x(s)$ le polynôme caractéristique associé à H . Il satisfait

$$H(\mathbf{x}) > \mathbf{0} \Leftrightarrow P_x(-s) \text{ est Hurwitz.} \quad (8.5)$$

Rappelons qu'un polynôme est *Hurwitz* si toutes ses racines sont à parties réelles strictement négatives. L'application du critère de Routh à $P_x(-s)$, génère une fonction vectorielle $r(\mathbf{x})$, dont les fonctions coordonnées sont associées aux éléments de la première colonne de la table. Nous appellerons r la *fonction de Routh* du polynôme $P_x(-s)$. Elle satisfait

$$P_x(-s) \text{ Hurwitz} \Leftrightarrow r(\mathbf{x}) > \mathbf{0}. \quad (8.6)$$

Le calcul par intervalles permet de calculer une fonction d'inclusion r de r à partir de laquelle nous obtenons les tests suivants

$$t_2(\mathbf{x}) \Leftrightarrow r(\mathbf{x}) \not\geq \mathbf{0} \Leftrightarrow \exists i, r_i^+ < 0, \quad (8.7)$$

$$t_3(\mathbf{x}) \Leftrightarrow r(\mathbf{x}) > \mathbf{0} \Leftrightarrow \forall i, r_i^- > 0, \quad (8.8)$$

qui sont respectivement plus faibles que T_2 et T_3 . Pour un pavé donné \mathbf{x} , t_1 , t_2 et t_3 sont faciles à évaluer et seront utilisés dans l'algorithme de décomposition finie que nous allons maintenant proposer.

8.3. Algorithme

En s'inspirant des algorithmes ensemblistes développés dans les chapitres 5 et 6, l'algorithme de minimisation sans contraintes par décomposition finie vient naturellement.

Entrées : Pavé de recherche initial : \mathbf{x} .
Fonctions d'inclusion convergentes : f, r, g .

Initialisation : $f^* := \infty$, pile := \emptyset .

Itération k

- Pas 1 Si $(f(\mathbf{x}))^- \geq f_{\min}$ ou $\mathbf{0} \notin g(\mathbf{x})$ ou $r(\mathbf{x}) \neq \mathbf{0}$ alors aller au pas 4.
 Pas 2 Si $r(\mathbf{x}) > \mathbf{0}$ alors $f_{\min} := \min(f_{\min}, \min_{\mathbf{x} \in \mathbf{x}} f(\mathbf{x}))$. Aller au pas 4.
 Pas 3 Bisséquer \mathbf{x} suivant un plan principal et empiler les deux pavés résultants.
 Pas 4 Si la pile n'est pas vide, alors dépiler dans \mathbf{x} et aller au pas 1.
 FIN. ◇

Remarque 8.1 : Lorsque l'algorithme se termine, la variable f_{\min} contient le minimum global f^* de la fonction f sur le domaine de recherche. Au pas 2, la quantité

$$\min_{\mathbf{x} \in \mathbf{x}} f(\mathbf{x}) \tag{8.9}$$

est calculée à l'aide d'un algorithme d'optimisation convexe classique. Comme f est convexe sur \mathbf{x} , cette optimisation se fait en toute garantie. L'union de tous les pavés \mathbf{x} intervenant dans l'algorithme forme un pavage fini du pavé de recherche initial. Pour chaque \mathbf{x} du pavage, soit le critère est convexe sur \mathbf{x} , soit \mathbf{x} est inacceptable. ◇

Remarque 8.2 : Il est facile d'obtenir tous les minimiseurs globaux de f en rangeant tous les minimiseurs locaux obtenus au pas 2, et en rejetant ceux qui ne correspondent pas au minimum f^* obtenu. Cette technique peut aussi être implantée récursivement.

Remarque 8.3 : Au pas 2, nous devons minimiser un critère convexe sur un pavé \mathbf{x} . Il y a donc une probabilité non négligeable que le minimiseur se trouve sur l'un des sommets de \mathbf{x} . De ce fait, il peut être judicieux d'initialiser la procédure de minimisation convexe sur le sommet pointé par le gradient de f pris au milieu de \mathbf{x} . Rappelons que ce coin est immédiatement repéré par le vecteur signe de ce gradient. ◇

Remarque 8.4 : L'algorithme nécessite un traitement formel préparatoire assez lourd (calcul du gradient et du Hessien, écriture de la table de Routh ...). Il peut donc s'avérer indispensable d'automatiser cette tâche à l'aide d'un logiciel de calcul formel (voir par exemple Davenport, Siret et Tournier E. (1987)). ◇

Remarque 8.5 : Soient $d_g(\epsilon)$ la durée moyenne de la minimisation globale ci-dessus et $d_c(\epsilon)$ la durée moyenne de la minimisation convexe utilisée au pas 2, pour une précision ϵ demandé sur la localisation du minimiseur. Puisque la décomposition finie du pavé de

recherche ne dépend pas de ϵ , si n_c est le nombre total de minimisations convexes requises, nous avons

$$\lim_{\epsilon \rightarrow 0} \frac{d_g(\epsilon)}{d_c(\epsilon)} = n_c. \quad (8.10)$$

Les complexités des algorithmes d'optimisation convexe et globale — considérées comme fonctions de ϵ — sont donc équivalentes. \diamond

8.4. Exemple

L'exemple académique auquel nous allons nous intéresser se limitera à illustrer la méthode développée. Il a été choisi bidimensionnel afin de faciliter l'illustration graphique du mécanisme. Il est toutefois évident que la méthode pourrait résoudre des problèmes d'optimisation bien plus complexes.

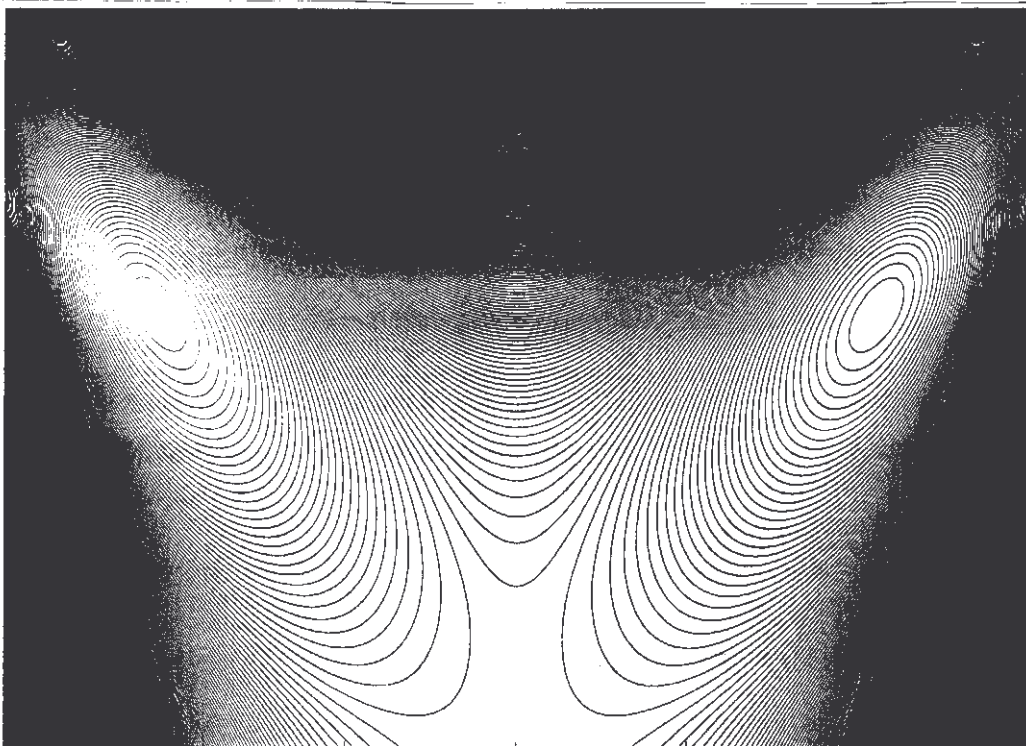


Fig. 8.1 Isocritères mettant en évidence l'existence de deux minimiseurs. Le cadre correspond au pavé $[-3, 3] \times [0, 3]$ de l'espace (x_1, x_2) .

Le critère à minimiser est défini par

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}, \mathbf{x} \rightarrow x_1^4 - 5 x_1^2 x_2 + x_2^4. \quad (8.11)$$

Ce critère possède deux minimiseurs globaux. La figure 8.1 représente les isocritères. La fonction d'extension naturelle de f

$$f(\mathbf{x}) = x_1^4 - 5 x_1^2 x_2 + x_2^4 \quad (8.12)$$

est non minimale. Le gradient du critère, obtenu par simple dérivation de f ,

$$\mathbf{g}(\mathbf{x}) = (4 x_1^3 - 10 x_1 x_2 \quad -5 x_1^2 + 4 x_2^3)^T, \quad (8.13)$$

admet pour fonction d'extension naturelle

$$\mathbf{g}(\mathbf{x}) = (4 x_1^3 - 10 x_1 x_2 \quad -5 x_1^2 + 4 x_2^3)^T. \quad (8.14)$$

Le Hessien du critère

$$\mathbf{H}(\mathbf{x}) = \begin{pmatrix} 12 x_1^2 - 10 x_2 & -10 x_1 \\ -10 x_1 & 12 x_2^2 \end{pmatrix}, \quad (8.15)$$

admet pour polynôme caractéristique

$$P_{\mathbf{x}}(s) = s^2 + (-12 x_2^2 - 12 x_1^2 + 10 x_2) s + 144 x_1^2 x_2^2 - 100 x_1^2 - 120 x_2^3. \quad (8.16)$$

Ce polynôme a été obtenu par simple calcul du déterminant de la matrice $s\mathbf{I} - \mathbf{H}(\mathbf{x})$, où \mathbf{I} représente la matrice identité. La matrice $\mathbf{H}(\mathbf{x})$ sera définie positive si et seulement si $P_{\mathbf{x}}(-s)$ est Hurwitz. Le tableau de Routh associée au polynôme de degré deux $P(s) = s^2 + a_1 s + a_0$, est donnée par

1	a_0
a_1	0
a_0	0

(8.17)

Il permet d'établir l'équivalence

$$s^2 + a_1 s + a_0 \text{ Hurwitz} \Leftrightarrow \begin{cases} a_0 > 0 \\ a_1 > 0 \end{cases}. \quad (8.18)$$

Ainsi

$$\mathbf{H}(\mathbf{x}) > \mathbf{0}$$

$$\begin{aligned} \Leftrightarrow P_{\mathbf{x}}(-s) = s^2 + (12 x_2^2 + 12 x_1^2 - 10 x_2)s + 144 x_1^2 x_2^2 - 100 x_1^2 - 120 x_2^3 \text{ Hurwitz} \\ \Leftrightarrow \begin{cases} 144 x_1^2 x_2^2 - 100 x_1^2 - 120 x_2^3 > 0, \\ 12 x_2^2 + 12 x_1^2 - 10 x_2 > 0. \end{cases} \end{aligned} \quad (8.19)$$

Une fonction de Routh est donc

$$\mathbf{r}(\mathbf{x}) = \begin{pmatrix} 144 x_1^2 x_2^2 - 100 x_1^2 - 120 x_2^3 \\ 12 x_2^2 + 12 x_1^2 - 10 x_2 \end{pmatrix}, \quad (8.20)$$

qui admet pour fonction d'extension naturelle

$$\mathbf{r}(\mathbf{x}) = \begin{pmatrix} 144 x_1^2 x_2^2 - 100 x_1^2 - 120 x_2^3 \\ 12 x_2^2 + 12 x_1^2 - 10 x_2 \end{pmatrix}. \quad (8.21)$$

Bien que des fonctions d'inclusion beaucoup plus fines puissent être obtenues pour \mathbf{f} , \mathbf{g} et \mathbf{H} (Hansen (1992)), nous avons choisi de garder la fonction d'extension naturelle pour faciliter la compréhension. Toutes les fonctions d'inclusion requises par l'algorithme sont maintenant à notre disposition. En prenant comme pavé de recherche initial

$$\mathbf{x}(0) = [-5, 15] \times [-5, 15], \quad (8.22)$$

l'algorithme génère le pavage représenté sur la figure 8.2 composé de 89 pavés, en 1.64 secondes sur le Compaq 386/33 après 14 minimisations convexes. Un minimum global $f^* = -9.765625$ est trouvé. Il correspond aux minimiseurs

$$\hat{\mathbf{x}}_1 = \begin{pmatrix} 2.10224 \\ 1.76777 \end{pmatrix} \quad \text{and} \quad \hat{\mathbf{x}}_2 = \begin{pmatrix} -2.10224 \\ 1.76777 \end{pmatrix} \quad (8.23)$$

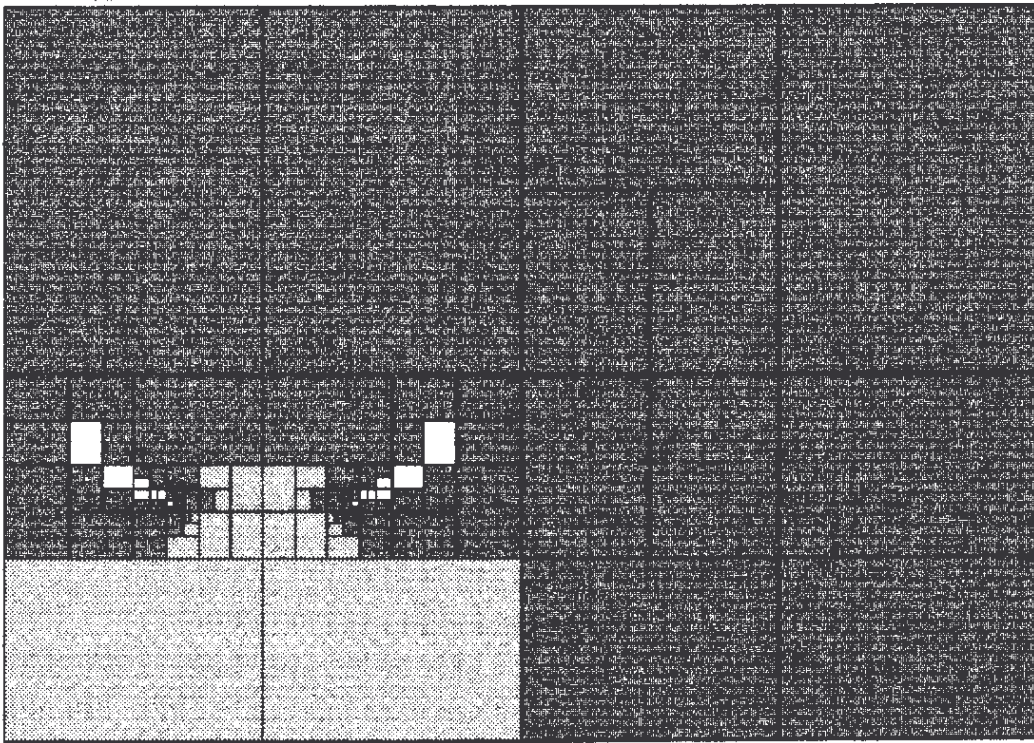


Fig. 8.2 Pavage généré par l'algorithme dans l'espace (x_1, x_2) .
Le cadre correspond au pavé de recherche $\mathbf{x}(0) = [-5, 15] \times [-5, 15]$.

Les pavés en blanc satisfont $r(\mathbf{x}) > \mathbf{0}$, ceux en gris clair vérifient $\mathbf{0} \notin g(\mathbf{x})$ et ceux en gris foncés sont tels que $(f(\mathbf{x}))^- \geq f_{\min}$. Aucun pavé ne satisfait $r(\mathbf{x}) \geq \mathbf{0}$, mais ceci est dû aux caractéristiques spécifiques du critère. En prenant maintenant comme pavé de recherche

$$\mathbf{x}(0) = [-100, 100] \times [-100, 100], \quad (8.24)$$

L'algorithme produit un résultat identique en 1.75 secondes, toujours après 14 minimisations convexes. Le pavage généré est composé de 94 pavés.

8.5. Conclusion

L'algorithme par décomposition finie que nous venons de proposer combine la vision globale et garantie des algorithmes d'optimisation par analyse par intervalles avec la rapidité des algorithmes d'optimisation convexe. Les algorithmes classiques pour l'optimisation globale par analyse par intervalles, tels que ceux d'Hansen ou de Moore, opèrent par élimination de pavés qui ont été prouvés inacceptables. Ici, de telles

éliminations sont aussi effectuées, mais une nouvelle approche combinant le critère de Routh et l'analyse par intervalles est utilisée pour détecter des pavés pour lesquels des approches convexes vont donner des résultats garantis. L'algorithme proposé partitionne le pavé de recherche en un nombre fini de pavés. Sur chacun de ces pavés, soit le pavé est inacceptable (et donc ne peut contenir un minimiseur global), soit le critère est convexe (une approche convexe adaptée peut alors être utilisée en toute garantie pour localiser, s'il existe, l'unique minimiseur local contenu dans ce pavé).

A notre connaissance, aucun test d'inclusion pour la convexité n'existait auparavant dans la littérature. Nous en avons proposé un qui, grâce au critère de Routh, permet de transformer un problème de convexité en un ensemble d'inégalités à satisfaire. Pour faciliter la compréhension, nous avons montré, sur un algorithme élémentaire d'optimisation globale par analyse par intervalles, comment utiliser ce test pour pouvoir lancer en toute garantie une procédure de minimisation convexe. Afin de donner toute sa dimension au test de convexité proposé, il conviendrait de l'insérer dans un algorithme plus efficace tel l'algorithme d'Hansen.

CHAPITRE 9

CONCLUSION

Nous avons tenté de montrer ce que peut apporter une formulation ensembliste de problèmes de la physique par rapport aux formulations ponctuelles classiques. Rappelons les points essentiels. Une telle approche permet de prendre en compte les incertitudes sur les données du problème et les erreurs numériques engendrées par le calculateur lors de résolution et d'obtenir un ensemble solution caractérisant de façon garantie l'incertitude sur le résultat obtenu.

En général, les problèmes ensemblistes (c'est-à-dire ceux dont la solution est un ensemble) rencontrés dans les sciences pour l'ingénieur font intervenir des relations non linéaires. Pour proposer des méthodes capables de résoudre de tels problèmes et qui caractérisent la solution de façon garantie, nous avons dû utiliser trois concepts. Le premier est celui de *calcul par intervalles* qui semble être le support naturel du calcul ensembliste. Il définit les opérations élémentaires sur les ensembles les plus simples que l'on puisse trouver, les intervalles de \mathbb{R} . Il devient alors à la fois un outil pour calculer sur les ensembles et un langage pour manipuler les inégalités. Un rappel des notions essentielles à sa manipulation a été donné au chapitre 3. Le second concept correspond à la notion de *sous-pavage fini* utilisée pour approximer les ensembles manipulés. Tout sous-pavage fini peut être représenté en machine. De plus, pour la distance de Hausdorff (distance entre compacts), l'ensemble des sous-pavages est dense dans l'espace des compacts, ce qui justifie son utilisation pour l'approximation des ensembles de \mathbb{R}^n . Le troisième concept utilisé est la notion d'*intercompact*. Un intercompact est un couple de sous-pavages, l'un qui contient, l'autre qui est contenu dans l'ensemble qu'il caractérise. Ce couple définit un ensemble de compacts formant un voisinage dans l'espace des compacts de l'ensemble à caractériser. Un intercompact caractérise de façon garantie cet ensemble alors que le sous-pavage n'en donnait qu'une approximation. Les sous-pavages et les intercompacts ont été introduits au chapitre 4.

Il nous a alors été possible de développer des algorithmes permettant de résoudre différents types de problèmes ensemblistes, ainsi que quelques problèmes ponctuels qui ont été abordés eux aussi par une approche ensembliste. Une première classe importante de problèmes correspond à la notion d'*inversion ensembliste*. L'ensemble solution peut

alors s'exprimer comme l'inverse (ou image réciproque) d'un ensemble par une fonction vectorielle. Les problèmes d'inversion ensembliste interviennent souvent dans les sciences pour l'ingénieur comme par exemple pour l'estimation à erreur bornée et la caractérisation de domaines de stabilité. Au chapitre 5, nous avons proposé un algorithme SIVIA capable de les résoudre dans un cadre général. La complexité de SIVIA, les conditions de continuité qui garantissent la convergence ainsi que quelques exemples d'application résolus ont été donnés.

Le même type d'approche a pu être adopté pour développer des algorithmes résolvant *d'autres types de problèmes ensemblistes*, tels que celui de la détermination de régions de confiance de volume minimum associées à une loi de probabilité, ou celui de la détermination d'ensembles stabilisants (ensemble des paramètres d'une structure de commande qui stabilisent à coup sûr un modèle dont les paramètres inconnus appartiennent à un ensemble connu). Ces algorithmes ensemblistes, illustrés de quelques exemples d'application, sont donnés au chapitre 6.

Outre ces problèmes de caractérisation d'ensembles, deux problèmes d'optimisation ont été considérés. Le premier correspond à une *optimisation linéaire sur l'ensemble solution*. Il s'agit là d'un problème ponctuel puisque la solution est généralement constitué d'un ensemble fini de vecteurs. Une approche ensembliste, développée au chapitre 7, nous a permis d'obtenir un algorithme efficace convergeant à coup sûr vers la solution à condition de disposer d'un test d'inclusion convergent pour l'ensemble solution, et de satisfaire des conditions de continuité. Cet algorithme a alors été appliqué dans ce même chapitre au calcul de pavés enveloppe d'ensembles de vraisemblance dans un contexte à erreur bornée.

Le second correspond à un problème d'*optimisation sans contrainte par décomposition finie*. Pour minimiser un critère non convexe sans contraintes sur les paramètres, nous avons joint, par le critère de Routh, l'approche locale ponctuelle classique (rapide et garantie si le critère est convexe) avec la vision globale et garantie que permet l'analyse par intervalles. L'algorithme proposé au chapitre 8 partitionne le domaine de recherche en un nombre fini de pavés. Sur chacun de ces pavés, soit le pavé est inacceptable (et ne peut donc pas contenir un minimiseur global), soit le critère est convexe (une approche locale adaptée peut alors être utilisée en toute garantie).

Les algorithmes proposés et appliqués à la résolution des problèmes introduits ont été programmés en Turbo Pascal. Il conviendrait, afin de palier les erreurs numériques et d'avoir ainsi une stricte garantie sur le résultat obtenu, d'effectuer à chaque opération sur

r les intervalles une approximation extérieure sur l'intervalle résultat et donc d'utiliser pour
1 cela un langage adapté (voir par exemple Kulisch (1987)). Il serait souhaitable qu'un
1 langage combinant à la fois traitement formel et calcul par intervalles soit développé afin
3 d'éviter de reprogrammer systématiquement les résultats formels associés à chacun des
2 problèmes traités. Notons que la technique stochastique CESTAC (voir par exemple
3 Vignes (1993)) pourrait elle-aussi être utilisée pour évaluer l'incertitude avec laquelle les
extrémités des intervalles sont calculées.

3 L'analyse par intervalles semble avoir un champ d'application immense dans les
3 domaines de l'automatique (par exemple pour la commande et l'étude de stabilité des
2 systèmes non linéaires) et du traitement du signal (par exemple pour la conception de
t filtres). Elle forme un outil numérique qui se manipule aisément, qui s'adapte tout à fait
3 au traitement informatique, qui permet de traiter de façon globale et garantie des
t problèmes non linéaires de dimension modeste et qui s'applique aussi bien pour la
2 résolution des problèmes ensemblistes que pour des problèmes ponctuels. Elle devrait, de
3 ce fait, se banaliser dans les sciences pour l'ingénieur. A titre d'exemples de domaines
2 auxquels on pourrait penser, mentionnons la recherche d'un filtre (temporel ou image) de
t structure donnée passant par un gabarit donné ou la démonstration numérique du fait
2 qu'une fonction est de Lyapunov pour un système non linéaire ou encore le calcul d'une
commande stabilisant de façon garantie un système non linéaire.

1 Une disquette contenant les programmes (exécutables sur tout compatible IBM avec écran
EGA-VGA) associés aux exemples d'application se trouve jointe à ce document.

BIBLIOGRAPHIE

Bányász Cs. et Keviczky L. (Eds.) (1991). *Prep. 9th IFAC/IFORS Symposium on Identification and System Parameter Estimation, 1 & 2*, IFAC, Budapest.

Barmish B.R., Hollot C.V., Kraus F.J. et Tempo R. (1992). "Extreme point results for robust stabilization of interval plants with first order compensators," *IEEE Trans. on Automat. Contr.*, **AC-37**, 707-714.

Belforte G., Bona B. et Cerone V. (1990). Parameter estimation algorithms for a set membership description of uncertainty. *Automatica*, **26**, 887-898.

Belforte G. et Milanese M. (1981). Uncertainty interval evaluation in presence of unknow-but-bounded errors: nonlinear families of models. *Proc. 1st IASTED Internat. Symp. Modeling, Identification, Control*, Davos, 75-79.

Berger M. (1979). *Espace euclidien, triangles, cercles et sphères; Géométrie 2*, Cedic/Fernand Nathan, Paris, 127-129.

Broman V. et Shensa M.J. (1990). A compact algorithm for the intersection and approximation of n-dimensional polytopes. *Mathematics and Computers in Simulation*, **32**, 469-480.

Burgmeier P., Jahn K.U. et Plochov A.G. (1990). An interval computational method for approximating controllability sets. *Computing*, **44**, 35-46.

Cherbit G. (1987). *Fractals* Masson, Paris, 96-97.

Clément T. et Gentil S. (1990). Recursive membership set estimation for output-error models. *Mathematics and Computers in Simulation*, **32**, 505-513.

Davenport J., Siret Y. et Tournier E. (1987), *Calcul formel*, Masson, Paris.

DiStefano J.J., Stubberud A.R. et Williams I.J. (1990), *Feedback and Control Systems* (2nd Edition), Schaum's Outline Series, McGraw-Hill, NewYork.

Dixon L.C.W. et Szego G.P. (Eds.) (1975). *Towards Global Optimization*, North-Holland, Amsterdam.

Dixon L.C.W. et Szego G.P. (Eds.) (1978). *Towards Global Optimization 2*, North-Holland, Amsterdam.

Faedo S. (1953). Un nuovo problema di stabilità per le equazioni algebriche a coefficienti reali, *Ann. Sc. Norm. Sup. Pisa, Sci. Fiz. Mat.*, **1-2**, pp. 53-65.

Falk J.E. (1973). Global solutions for signomial programs. *Tech. Rep. T-274*, George Washington Univ., Washington, DC.

Fogel E. et Huang Y.F. (1982). On the value of information in system identification - bounded noise case. *Automatica*, **18**, 229-238.

Hansen E. (1992). *Global optimization using interval analysis*, Marcel Dekker, New York.

Horst R. et Tuy H. (1990). *Global Optimization, Deterministic Approaches*, Springer-Verlag, Berlin.

IBM (1986). *High-accuracy arithmetic subroutine library*, (ACRITH). Program description and user's guide, SC 33-6164-02, 3rd edition.

Jahn K.U. (1992). Approximation and manipulation of imprecise data by enclosing sets, *Modelling uncertain data*, Editeur : H.Bandemer, Akademie-Verlag Berlin.

Jaulin L. et Walter E. (1993a). Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis. *Mathematics & Computers in Simulation*, **35**, 123-127.

Jaulin L. et Walter E. (1993b). Set inversion via interval analysis for nonlinear bounded-error estimation, *Automatica*, **29**(4), 1053-1064.

Jaulin L. et Walter E. (1993c). Guaranteed nonlinear estimation and robust stability analysis via set inversion. *Proc. 2nd European Control Conference*, 28 juin - 1er juillet 1993, 818-821.

Jaulin L. et Walter E. (1993d). Guaranteed nonlinear parameter estimation via interval computations. Paru dans dans *Interval Computations* avec une sélection de papiers présentés à la *Conférence on Numerical Analysis with Automatic Result Verification*, Lafayette (25 Février-1er Mars 1993).

Jaulin L. et Walter E. (1994a). Set inversion with application to guaranteed nonlinear estimation and robust control. *Proc. of the Workshop on Modelling Techniques for Uncertain Systems*, Sopron, 6-10 Juillet 1992. A paraître chez Birkhäuser, Boston.

Jaulin L. et Walter E. (1994b). Guaranteed nonlinear set estimation via interval analysis. Contribution invitée à un ouvrage collectif, à paraître chez Plenum.

Jaulin L. et Walter E. (1994c). Routh-Based Convexity Check, Application to Guaranteed Optimization. Soumis à *Journal of Global Optimisation*, North-Holland.

Kearfott B. (1987). Abstract generalized bisection and cost bound, *Math. of Comput.* **49**, 187-202.

Kharitonov V.L. (1978). Asymptotic stability of an equilibrium position of a family of systems of linear differential equations, *Differentsial'nye Uravneniya*, vol. 14, pp. 2086-2088.

Kienld H. et Michalske A. (1992). Robustness analysis of linear control systems with uncertain parameters by the method of convex decomposition, Robustness of dynamic Systems with parameter uncertainties, Editeur : M.Mansour, S.Balemi, W.Truöl, Birkhäuser Verlag, Basel, 189-198.

Klate R., Kulisch U.W., Neaga M., Ratz D. and Ullrich C. (1992), *PASCAL-XSC, Language Reference with Example*, Springer-Verlag, Heidelberg.

Knuth D.E. (1977). *Fundamental Algorithms*, 2nd edition, Addison-Wesley, Stanford, 305-346.

Kolev L. (1993), An interval first-order method for robustness analysis, Proc. 1993 IEEE Int. Symp. on Circuits and Systems, Chicago, to appear.

Kulisch U. (Ed.) (1987). *PASCAL-SC: A PASCAL Extension for Scientific Computation, Information Manual and Floppy Disks*, Teubner, Stuttgart.

Kurzanski A.B. et Vályi I. (1991). Guaranteed state estimation for dynamic systems: beyond the overviews. *Prep. 9th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, Budapest, 1 033-1 037.

Leipholz H. (1987), *Stability Theory*, (2nd edition), Teubner, Stuttgart.

Messaoud H., Favier G. et Santos Mendes R. (1992). Adaptive robust pole placement by connecting identification and control, *Proc. 4th IFFAC Int. Symp. on adaptive systems in control and signal processing*, Grenoble, 41-46.

Milanese M. et Belforte G. (1982). Estimation theory and uncertainty intervals evaluation in presence of unknown-but-bounded errors. Linear families of models and estimators. *IEEE Trans. Aut. Control*, **AC-27**, 408-414.

Milanese M. et Vicino A. (1991a). Estimation theory for dynamic systems with unknown but bounded uncertainty: an overview. *Prep. 9th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, Budapest, 859-867.

Milanese M. et Vicino A. (1991b). Estimation theory for nonlinear models and set membership uncertainty. *Automatica*, **27**, 403-408.

Mo S.H. et Norton J.P. (1990). Fast and robust algorithm to compute exact polytope parameter bounds. *Mathematics and Computers in Simulation*, **32**, 481-493.

Mockus J. (1989). *Bayesian Approach to Global Optimization*, Kluwer, Dordrecht.

Moore R.E. (1979). *Methods and Applications of Interval Analysis*, SIAM, Philadelphia.

Moore R.E. (1992). Parameter sets for bounded-error data. *Mathematics and Computers in Simulation*, **34**, 113-119 (1992).

Neumaier A. (1990). *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge.

Norton J.P. (1986). An introduction to identification. *Academic Press*, London.

Norton J.P. (1987a). Identification and application of bounded-parameter models. *Automatica*, **23**, 497-507.

Norton J.P. (1987b). Identification of parameter bounds for ARMAX models from records with bounded noise. *Internat. J. Control*, **45**, 375-390.

Piet-Lahanier H. et Walter E. (1990). Characterization of non-connected parameter uncertainty regions. *Mathematics and Computer in Simulation*, **32**, 553-560.

Pronzato L. et Walter E. (1990). Experiment design for bounded-error models, *Mathematics and Computers in Simulation*, **32(5&6)**, 571-584 (1990).

Pronzato L., Walter E. et Piet-Lahanier H. (1989). Mathematical equivalence of two ellipsoidal algorithms for bounded-error estimation. *Proc. 28th IEEE Conf. on Decision and Control*, Tampa, 1 952-1 955.

Rall L.B. (1981). *Automatic Differentiation*, Springer-Verlag, Berlin.

Ratschek A. et Rokne J. (1988). *New Computer Methods for Global Optimization*, Ellis Horwood Limited, John Wiley & Sons, New York.

Richalet J. (1991). *Pratique de l'identification*, Hermès, Paris.

Routh E.J. (1877). *A Treatise on the stability of a given state of motion*, MacMillan & Co, London.

Siljak D.D. (1989). Parameter space methods for robust control design: a guided tour, *IEEE Trans. on Automat. Contr.*, **AC-34**, 674-688.

Vicino A. et Zappa G. (1992). Sequential approximation of uncertainty sets via parallelotopes, *Proc. Workshop on the modeling of uncertainty in control systems*, Springer-Verlag, Heidelberg.

Vignes J. (1993). A stochastic arithmetic for reliable scientific computation, *Mathematics & Computers in Simulation*, **35**, 233-261.

Walter E. (Ed.) (1990). Special issue on parameter identifications with error bound. *Mathematics & Computers in Simulation*, **32**, 447-607.

Walter E. et Piet-Lahanier H. (1989). Exact recursive polyhedral description of the feasible parameter set for bounded error models. *IEEE Trans. Aut. Control*, **AC-34**, 911-915.

Walter E. et Piet-Lahanier H. (1990). Estimation of parameter bounds from bounded-error data: a survey. *Mathematics and Computers in Simulation*, **32**, 449-468.

Walter E. et Jaulin L. (1994). Guaranteed characterization of stability domains via set inversion, accepté pour publication dans *IEEE Trans. Autom. Control*.

Walter E. et Pronzato L. (1994). *Identification de modèles paramétriques à partir de données expérimentales*, Masson, Paris.

Zhigljavsky A.A. (1991). *Theory of Global Random Search*, Kluwer, Dordrecht.

TABLE DES MATIÈRES

REMERCIEMENTS	(ii)
TABLE DES NOTATIONS.....	(iii)
CHAPITRE 1 INTRODUCTION	1
1.1 Problèmes ponctuels et ensemblistes	1
1.1.1 Les problèmes ponctuels.....	1
1.1.2 Les problèmes ensemblistes.....	2
1.2 Modélisation, estimation et commande	5
1.3 Plan de la thèse	7
CHAPITRE 2 LES PROBLÈMES ENSEMBLISTES	11
EN ESTIMATION ET EN COMMANDE	
2.1 Introduction	11
2.2 Notation vectorielle	11
2.3 Définition d'un problème d'inversion ensembliste	13
2.4 Estimation ponctuelle	15
2.5 Estimation ensembliste à erreurs bornées	16
2.5.1 Introduction à l'estimation à erreurs bornées.....	16
2.5.2 Problème des exponentielles approximantes.....	19
2.5.3 Problème de Milanese et Vicino simplifié	20
2.5.4 Problème de Milanese et Vicino	22
2.5.5 Problème des conditions initiales.....	23
2.6 Régions de confiance Bayésiennes	25
2.6.1 Introduction.....	25
2.6.2 Problème de la région de confiance triangulaire.....	26
2.7 Ensemble de stabilité	28
2.7.1 Introduction.....	28
2.7.2 Problème des palmes.....	31
2.7.3 Problème du col de cygne.....	35
2.8 Ensemble stabilisant	35
2.8.1 Introduction.....	35
2.8.2 Problème du lobe stabilisant.....	36
2.9 Conclusion	38
CHAPITRE 3 CALCUL ET ANALYSE PAR INTERVALLES	39
3.1 Introduction	39
3.2 Intervalles réels et arithmétique sur les intervalles	40
3.2.1 Intervalles	40
3.2.2 Arithmétique sur les intervalles	41
3.2.3 Arithmétique sur les intervalles représentables en machine.....	42
3.2.4 Propriétés algébriques de l'arithmétique sur les intervalles.....	43

3.3	Fonctions d'inclusion	43
3.3.1	Définition, inclusion monotone, propriété de convergence	43
3.3.2	Fonction d'inclusion minimale.....	44
3.3.3	Fonction d'extension naturelle.....	45
3.3.4	Fonction d'inclusion par développement de Taylor.....	46
3.4	Cas multidimensionnel	47
3.4.1	Pavés	47
3.4.2	Fonctions d'inclusion vectorielles	49
3.4.3.1	<i>Définitions</i>	49
3.4.3.2	<i>Fonction d'inclusion minimale</i>	50
3.4.3.3	<i>Fonctions d'extension naturelles</i>	52
3.4.3.4	<i>Test d'inclusion</i>	55
3.5	Analyse par intervalles	55
3.5.1	Résolution des équations linéaires perturbées par Gauss-Seidel.....	56
3.5.2	Méthode de Newton par intervalles.....	57
3.5.3	Algorithme d'optimisation globale de Moore-Skelboe.....	59
3.6	Conclusion	60
CHAPITRE 4 ENCADREMENT ENSEMBLISTE ET SOUS-PAVAGES		63
4.1	Introduction	63
4.2	Distance sur l'ensemble des compacts	64
4.2.1	Distance de Hausdorff	64
4.2.2	Distance de Hausdorff complémentaire	66
4.2.3	Distance de Hausdorff généralisée.....	67
4.3	Sous-pavages	67
4.3.1	Définitions.....	68
4.3.2	Propriétés du cardinal d'un sous-pavage	69
4.4	Encadrement d'un compact entre deux sous-pavages	70
4.5	Les intercompacts	73
4.6	Topologie du problème d'inversion ensembliste	75
4.7	Conclusion	78
CHAPITRE 5 ALGORITHME D'INVERSION ENSEMBLISTE		79
5.1	Introduction	79
5.2	Algorithme d'inversion ensembliste	80
5.2.1	Tests d'acceptabilité et d'inacceptabilité d'un pavé.....	80
5.2.2	Principe de l'algorithme d'inversion ensembliste	82
5.2.3	SIVIA	83
5.2.4	Propriétés de SIVIA	85
5.2.4.1	<i>Convergence</i>	85
5.2.4.2	<i>Complexité</i>	87
5.2.4.3	<i>Mémoire utilisée</i>	88
5.2.4.4	<i>Parallélisation de SIVIA</i>	89
5.2.4.5	<i>Stratégie de bisection</i>	90

5.2.5	Algorithme d'inversion ensembliste de Moore	91
5.2.6	Discontinuité de la fonction réciproque.....	92
5.2.7	Inversion intercompacte	93
5.2.8	Reformulation du problème d'inversion.....	94
5.2.9	SIVIA pour l'analyse d'ensembles à test d'inclusion.....	95
5.3	Application aux problèmes d'estimation à erreurs bornées.....	95
5.3.1	Problème de Milanese et Vicino simplifié	96
5.3.2	Problème des exponentielles approximantes.....	97
5.3.3	Problème de Milanese et Vicino	98
5.3.4	Problème des conditions initiales.....	101
5.4	Application aux problèmes de stabilité paramétrique.....	102
5.4.1	Utilisation du théorème de Kharitonov.....	102
5.4.2	Problème des palmes.....	103
5.4.3	Problème du col de cygne.....	105
5.5	Conclusion.....	106
 CHAPITRE 6 ALGORITHMES POUR D'AUTRES PROBLÈMES ENSEMBLISTES... 109		
6.1	Introduction.....	109
6.2	Calcul de régions de confiance Bayésiennes.....	110
6.2.1	Introduction.....	110
6.2.2	Test de seuil	110
6.2.3	Encadrement de la région de confiance.....	112
6.2.3.1	<i>Procédure pour calculer s^-</i>	113
6.2.3.2	<i>Procédure pour calculer s^+</i>	113
6.2.3.3	<i>Encadrement de la région de confiance</i>	114
6.2.3.4	<i>Procédure récursive d'encadrement de la région de confiance</i>	115
6.2.4	Algorithme efficace pour encadrer les régions de confiance	115
6.2.4.1	<i>Structure et variables utilisées</i>	116
6.2.4.2	<i>Algorithme</i>	118
6.2.5	Application au problème de la région de confiance triangulaire.....	119
6.2.6	Conclusion.....	121
6.3	Caractérisation de l'ensemble stabilisant.....	121
6.3.1	Introduction.....	121
6.3.2	Algorithme A1.....	122
6.3.3	Algorithme A2.....	124
6.3.4	Application au problème du lobe stabilisateur	124
6.3.5	Conclusion sur la caractérisation de l'ensemble stabilisant	126
6.4	Conclusion.....	126
 CHAPITRE 7 OPTIMISATION GLOBALE SOUS CONTRAINTES..... 129		
7.1	Introduction.....	129
7.2	Les problèmes d'optimisation sous contraintes.....	130
7.3	Approche signomiale.....	131
7.3.1	Définitions et notations	131

7.3.2	Programmation signomiale.....	132
7.3.3	Méthode de Falk pour résoudre les problèmes log-signomiaux.....	133
7.4	GOVIA	135
7.4.1	Principe.....	136
7.4.2	Représentation des sous-pavages par des listes.....	139
7.4.3	Représentation des sous-pavages par des arbres	139
7.4.4	GOVIA.....	142
7.5	GOVIA pour l'optimisation sur l'ensemble solution	145
7.6	Conclusion	148
 CHAPITRE 8 OPTIMISATION GLOBALE SANS CONTRAINTE.....		149
PAR DÉCOMPOSITION FINIE		
8.1	Introduction.....	149
8.2	Tests d'inclusion pour la convexité et l'inacceptabilité.....	150
8.3	Algorithme.....	151
8.4	Exemple	153
8.5	Conclusion.....	156
 CHAPITRE 9 CONCLUSION.....		159
 BIBLIOGRAPHIE.....		163
TABLE DES MATIÈRES.....		169

Résumé : Une formulation ensembliste de nombreux problèmes mathématiques apparaissant dans les sciences pour l'ingénieur et usuellement traités de façon ponctuelle permet de prendre en compte l'incertitude sur les données de ces problèmes, les erreurs numériques introduites par le calculateur lors de leur résolution et de caractériser l'incertitude avec laquelle les résultats sont obtenus.

L'algorithme SIVIA (*Set Inverter Via Interval Analysis*) qui a été développé utilise l'analyse par intervalles (outil mathématique permettant de calculer sur les ensembles) pour résoudre de façon globale et garantie les problèmes d'*inversion ensembliste* (c'est à dire ceux où l'ensemble solution peut être défini comme l'inverse d'un ensemble par une fonction). Les propriétés de SIVIA (convergence et complexité) sont établies après une étude approfondie de la structure topologique de l'ensemble des compacts. SIVIA permet de traiter une classe importante des problèmes comme l'estimation non linéaire ou la caractérisation de domaines de stabilité.

Des méthodes pour caractériser des régions de confiance Bayésiennes pour des paramètres et pour calculer des commandes robustes qui stabilisent à coup sûr des modèles incertains sont également proposées, ainsi qu'un algorithme d'optimisation globale et garantie sans contrainte.

Ce travail établit un pont entre l'analyse par intervalles (très peu connue dans le milieu des signaux et systèmes) et l'automatique non linéaire (largement ignorée par les spécialistes de l'analyse par intervalles)

Mots clefs : Analyse par intervalles, Commande robuste, Erreur bornée, Estimation non linéaire, Inversion, Optimisation globale et garantie, Problème ensembliste, Stabilité.

Abstract : A set-theoretic formulation of many mathematical problems of engineering usually dealt with in a pointwise fashion makes it possible to account for the uncertainty on the data of the problems, the numerical errors induced by the computer during their solutions and to characterize the uncertainty with which the results are obtained.

The algorithm SIVIA (*Set Inverter Via Interval Analysis*) that has been developed uses interval analysis (a tool allowing computation with sets) in order to solve set-inversion problems (i.e. those where the solution set can be defined as the inverse of a set by a function) in a global and guaranteed way. The convergence and complexity properties of SIVIA are established after a detailed study of the topological structure of the set of compacts. SIVIA makes it possible to consider a large class of problems such as nonlinear estimation or the characterization of stability domain.

Methods to characterize Bayesian confidence regions for parameters and to compute robust controls guaranteed to stabilize uncertain models are also proposed, as well as an algorithm for guaranteed global unconstrained optimization.

A bridge is established between interval analysis (unfamiliar to system and signal-processing theoreticians) and nonlinear system theory (largely ignored by specialists of interval analysis).

Key words : Interval analysis, Robust control, Bounded error, Nonlinear estimation, Inversion, Guaranteed and global optimization, Set problem, Stability.